

RTM

ESSENTIAL 1.0





CONTENIDO

Perl Basico I parte 3
vendett@
vendetta@hackertm.org

Integracion de DHCP-WINS-DNS 8
OpTix
optix@hackertm.org

Proyecto Toro 16
Zorro
matiasvara@yahoo.com

KPGP & GPG 21
Ksaver
ksaver@hackertm.org

Creacion de Subredes 30
OpTix
optix@hackertm.org

Despedida 36
RTM Hacker Staff
hackertm.org

Edición y Programación Gráfica by **e x a k e a w**

Road Technology Minds ofrece su primer proyecto e-zine basico para sus usuarios, es grato poder desarrollar los temas influyentes, discutidos y hasta desapercibidos que en cierto momento son clave para el entendimiento de aplicaciones avanzadas. Es la primera version de Essential, por ello iniciamos con basics, es decir, llevar al usuario por la guia paso a paso de la evolucion hacia los item mas complejos y completos para estudiar. Hoy en dia aun seguimos buscando el fin de suministrar la clara definicion del termino hack, aun con la desinformacion de los medios no solo nosotros sino varias comunidades nos hemos puesto mas firmes en su concepto.

Por esta razon inducimos una vision profesional, didactica y llena de sentido de exploracion.

RTM inicialmente busca desarrollar un soporte basico y fuerte a la vez, pero en si su mayor ambicion es conformar una comunidad amplia, global, no solo nacional, en donde se establezca la informacion completa, llena de proyectos y soluciones, soporte en los principales sistemas operativos sin importar un windows o un linux o lo que sea, solo estaremos ahi en mejora de los sitemas.

RTM Hacker Staff

www.hackertm.org
staff@hackertm.org
[#hackertm.org](http://irc.red-latina.org)

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





Perl

by vendett@

En esta ocasión me he dedicado a uno de los lenguajes preferidos de los usuarios de software libre, éste es Perl, el cual considero el mejor lenguaje cuando uno es novato en la programación.

Debido a lo amplio que es llegar a tocar todos los aspectos de Perl, al menos de una forma básica iré dividiendo en varias colaboraciones todo sobre Perl.

En ésta ocasión tocare una breve introducción a lo que es Perl y el por qué elegirlo, además relataré lo mas relevante en la historia de Perl, veremos cómo hacernos de Perl o saber que lo tenemos y por último haremos nuestro primer script en Perl. Espero les guste y estén al pendiente de las siguientes partes que conforman esta serie de colaboraciones ;)

Introducción

Perl significa *“Pathologically Eclectic Rubbish Lister”*, aunque su definición de uso es *“Practical Extraction & Report Language”*, que traducido al español es mas o menos Lenguaje Practico de Extracción e Informes.

Es un lenguaje muy usado para simplificar tareas de administración de sistemas Unix/Linux, para administrar sitios web, como es el pedir información de registros, monitorear sitios de internet, tratamiento y generación de archivos de texto y un gran etc.

Muchas veces Perl es usado por su facilidad para la programación rápida y sucia, es decir un programa rápido, no planeado, pero que funcione; esto ha llevado a que se use para mostrar prototipos rápidos de algún algoritmo y ver su funcionamiento antes de ser codificado en lenguajes como C o C++.

A últimas fechas se ha encontrado su aplicación en la escritura de CGI (Common Gateway Interface), o scripts ejecutados desde sitios web del lado del servidor.

Mediante módulos adicionales como el DBD o el ODBC, Perl puede llegar servir para acceder a bases de datos, desde bases de datos como MySQL hasta el Microsoft SQL server usando ODBC. Además esto se puede combinar con un CGI para hacer aplicaciones tales como un carrito de compras para un sitio web, aunque normalmente estos se le dejan a PHP.

Perl es un lenguaje que hereda ciertas estructuras de los intérpretes de comandos de UNIX, en especial de csh (C shell), y otras utilidades como awk y sed.

Perl está diseñado para hacer todo lo que csh, awk y sed podrían hacer, pero de varias formas, mejores, mas comprensibles y mas fáciles de depurar. Perl guarda una estrecha relación con el shell scripting.

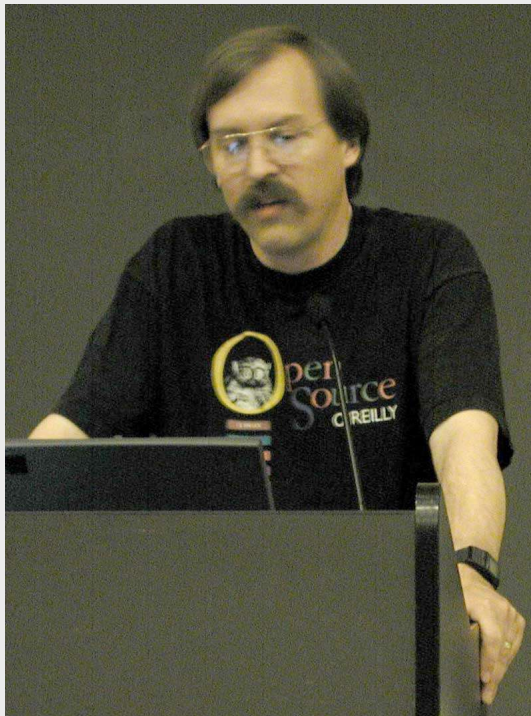
Perl es un lenguaje interpretado, aunque como todos los interpretes modernos, compila los



programas antes de ejecutarlos, por ello es común que nos refiramos a ellos como *scripts de Perl* debido a que no son compilados al lenguaje de máquina.

Aunque en los sistemas tipo UNIX las extensiones no son de gran importancia debido a que en estos sistemas todo se maneja como archivos, la extensión que se le asigna a los archivos de Perl normalmente es *.pl.

Historia de PERL



Perl fue creado por Larry Wall, su objetivo era el de simplificar las tareas de administración de un sistema UNIX, aunque en la actualidad se ha convertido en mas que eso, ahora Perl es usado como un lenguaje de propósito general, siendo hoy en día una de las herramientas base de cualquier webmaster o administrador de redes.

Cuando Perl empezó a ser un gran conocido fue en su versión 4, esta versión fue dada a conocer junto con el libro "*Programming Perl*" de Larry Wall o mejor conocido como "*El libro el camello*".

La versión 4 se fue desarrollando desde 1991 hasta 1993, durante este tiempo se incrementó una gran popularidad de Perl como lenguaje de programación para servidores de Internet, a pesar de que originalmente había sido diseñado como lenguaje para administración de sistemas.

En 1994 apareció la versión 5, en esta versión se introdujeron muchas características que han echo de Perl un lenguaje con el que la programación se simplifica y que lo hace un excelente candidato para los novatos en la programación, incluyó módulos, facilidades de programación orientada a objetos, referencias y una mejor documentación.

Durante este año y debido a la popularidad que va tomando Perl, aparecen muchos otros libros.

En la versión 5.6 Perl trajo muchas mejoras, como soporte pleno de caracteres internacionales, hebras y un mejor compilador. Aparece un sistema de *Patch Pumpkin*(encargado de cada nueva versión), el cual decide lo que será agregado y lo que no, sustituyendo a Larry Wall.

ActiveState, una empresa que participaba activamente en el desarrollo de Perl, comienza a controlar más de cerca el proyecto, lo que a la vez trajo herramientas mas potentes y comerciales :(, para el desarrollo de Perl.



En el año 2000 se empieza a discutir sobre la versión 6, donde se plantea un avance en rapidez y flexibilidad.

Mientras escribo estas líneas la versión usada de Perl es la 5.8.5.

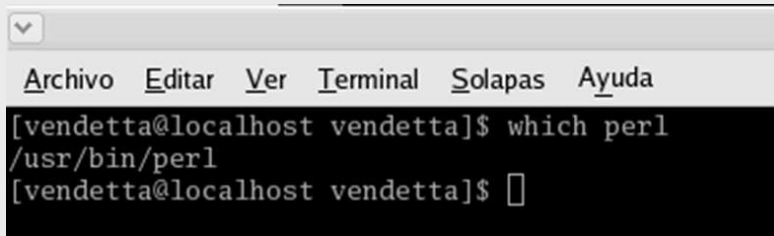
Empecemos con PERL

Para empezar con Perl, primero hay que tener Perl en nuestra computadora, Perl está disponible para todos los sistemas operativos más populares. Sólo hay que descargar la versión para Perl que nos será útil.

En el caso de Linux, todas las distros le incluyen, sólo deben de asegurarse de instalarlo, en el caso de otros S.O. como Windows hay herramientas proporcionadas por ActiveState, pero nos centraremos más en el uso de Perl sobre un S.O. libre, en este caso Linux.

Como les mencioné, Perl acompaña a las distros Linux desde hace mucho tiempo, ahora lo único que necesitan es saber en donde se encuentra, esto lo debemos de saber pues necesitaremos la ruta del interprete para empezar a programar. Saber en donde esté Perl es sencillo, desde la línea de comandos tecleemos:

`$which perl`



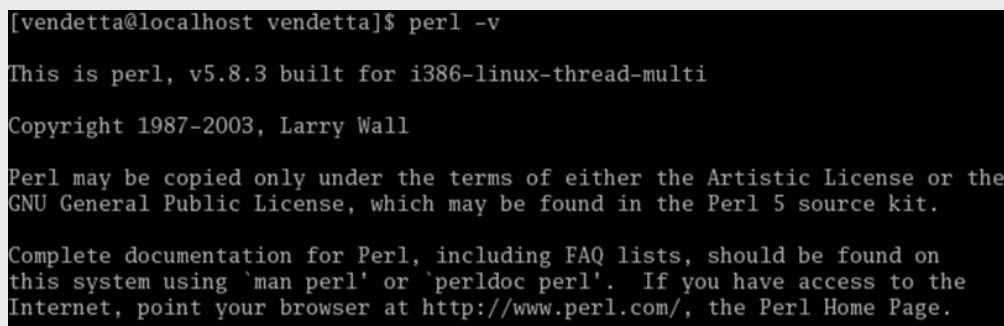
```

Archivo  Editar  Ver  Terminal  Solapas  Ayuda
[vendetta@localhost vendetta]$ which perl
/usr/bin/perl
[vendetta@localhost vendetta]$ █
    
```

Este comando nos devolverá algo parecido a lo que se muestra en la imagen en donde se muestra la ruta en donde se encuentra Perl, tomen nota pues ésta es importante para cuando programemos.

También podemos averiguar otras cosas de interés sobre la versión de Perl que tenemos instalada mediante el comando:

`$ perl -v`



```

[vendetta@localhost vendetta]$ perl -v
This is perl, v5.8.3 built for i386-linux-thread-multi

Copyright 1987-2003, Larry Wall

Perl may be copied only under the terms of either the Artistic License or the
GNU General Public License, which may be found in the Perl 5 source kit.

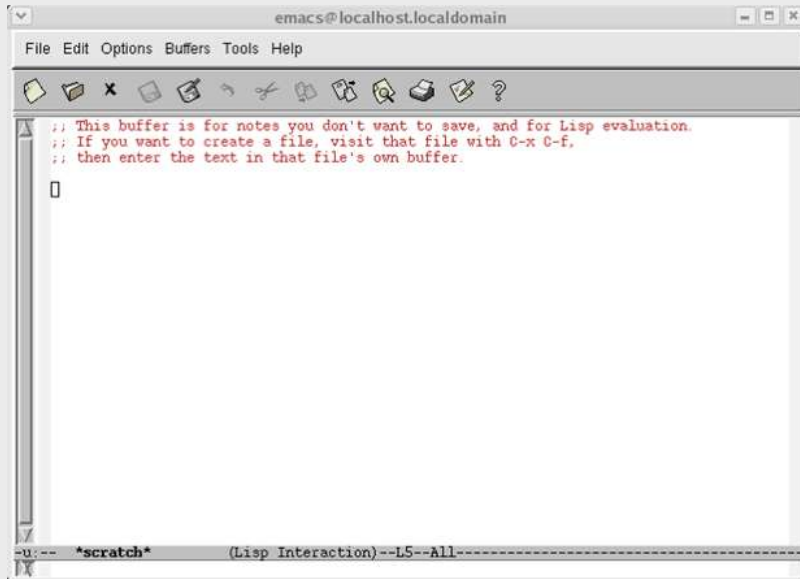
Complete documentation for Perl, including FAQ lists, should be found on
this system using `man perl' or `perldoc perl'.  If you have access to the
Internet, point your browser at http://www.perl.com/, the Perl Home Page.
    
```

Así nos mostró la versión y otras cosas más.



Nuestro Primer Script en PERL

Ok, hemos llegado a la parte emocionante y ésta es el primer script, en esta ocasión será algo muy sencillo, pero prefiero que vayamos lento para que no se confundan y todo lo aprendamos bien.



Para escribir un script en Perl hay muchas formas, pueden hacerlo desde un editor de texto plano como puede ser gedit, pueden hacerlo en editores como vi o vim, o en emacs, todo es según su gusto y como estén acostumbrados a trabajar. Al igual que Perl estos editores siempre acompañan a las distros Linux, tal vez no todos pero aunque sea uno estará allí. En este caso usare emacs, no por que sea mejor, solo por que es mi preferido.

Bien, ahora si a programar ;)

Como mencioné en la introducción, los programas en Perl no se compilan a lenguaje de máquina, por lo que siempre hay que escribir la ruta donde está el intérprete. Esto lo haremos siempre en la primera línea del script de la siguiente forma:

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

Como se habrán dado cuenta esa fue la ruta que obtuvimos con el comando `which perl` si la suya es distinta deberán colocar la que les corresponda.

Ahora escribiremos nuestra primera palabra reservada **print** de ahora en adelante las palabras reservadas del lenguaje las marcamos con rojo para que sea más fácil identificarlas.

Para este primer programa escribiremos algo así:

```
print "Los integrantes del RTM HT son los mas guapos del mundo :P \n";
```

Las comillas que escribí quieren decir que el contenido entre ellas será mostrado en pantalla, y el `\n` es un retorno de carro [ENTER] que se debe de escribir para que nos devuelva el prompt

Ahora lo guardaremos con el nombre que queramos, yo lo haré con el nombre "prog1_perl.pl" ya había mencionado que la extensión no importa, pero es buena costumbre ponerla para tener orden en nuestros archivos; es importante que asignemos permisos de ejecución al



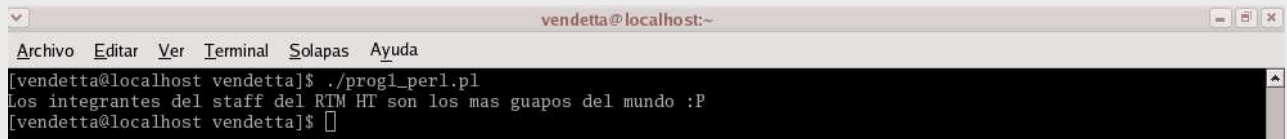
script pues normalmente no los tiene al momento de crearlo, bastará con que usemos el comando:

```
$ chmod 777 prog1_perl.pl
```

Y ahora si, ya todo esta listo para ejecutarlo...

```
$ ./prog1_perl.pl
```

El resultado de nuestro primer programa en Perl sera....



```
vendetta@localhost:~  
Archivo Editar Ver Terminal Solapas Ayuda  
[vendetta@localhost vendetta]$ ./prog1_perl.pl  
Los integrantes del staff del RTM HT son los mas guapos del mundo :P  
[vendetta@localhost vendetta]$
```

XDDD,,, bueno, esto es el final de la primer entrega de esta serie de Perl, espero les sea de utilidad, aun estamos empezando, mientras vayamos avanzando iremos haciendo cosas mas entretenidas. Pero hasta el siguiente numero....

vendett@
vendetta@hackertm.org
www.hackertm.org

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

END 

Integracion de DHCP-WINS-DNS

by OpTix

Bueno lo que vamos a ver es tratar de lograr una convivencia pacifica entre una subred basada en Windows NT y otros sistemas de esta manera complementaremos el sistema de nombres de dominio (DNS) con un sistemas de nombres de Windows (WINS) Y la forma en que un sistema ajeno al protocolo Lan manager puede aprovechar el protocolo (DHCP) para obtener direcciones dinámicas. Bueno si todo esto nos parece medio confuso vamos a ir viéndolo por pasos para que este mas claro, y espero te sirva de utilidad. Antes de poner las manos en el teclado. Vamos a recordar un poco estos servicios que se prestan no tanto como teoría si no un complemento para entender mejor

DNS

Y el principio se llamo DNS o Sistema de Nombres de Dominio. Este sistema responde a la necesidad casi innata de dar nombre a los equipos que pueblan el Internet es casi igual de antigüita porque nació por la necesidad de comunicarse los equipos, bueno todos estamos acostumbrados a teclear en nuestro navegador yahoo.com pero en realidad es bastante mas complejo que eso porque diremos que la red no entiende de nombres si no de números Exactamente de direcciones IP de ahí que ningún router es capaz de llegar a la maquina yahoo.com si no que habrá que decirle cual es su dirección IP para ubicarla y más si son números de 32 bits nosotros como que no nos llevamos bien con los números, imaginemos esto se resuelve en números binarios o sea cada IP tiene su identificación binaria por lo tanto surge la necesidad de darle nombre a los equipos. En los tiempos de arpanet digamos que era sencillo ya que no habían muchas maquinas entonces cada maquina tenia un archivo de host en donde se guardaban los nombres de las maquinas y sus ips el DNS es casi igual pero ahora es mas complejo, con mas funcionalidades que necesitaríamos hablar un capitulo solo de eso pero bueno esto es a breve nada mas, el DNS tiene dos características importantes las cuales son:

1) permite consultas Remotas...

De esta manera puedo establecer un maquina como servidor DNS y mantener un archivo que será usado por los otros nodos que quieren coactarse a nuestra red si añadimos o modificamos un IP solo se cambiara el archivo de direcciones del DNS

2) Jerárquico:

El DNS sigue una estructura piramidal en árbol ejemplo: si quiero buscar la maquina "galileo.fie.us.mx" el cliente DNS preguntara o se comunicara con el DNS root o Raíz cuyas direcciones son conocidas de antemano para averiguar que maquina asigna el dominio "mx" una vez averiguado dicha maquina preguntara quien sirve el dominio "us" y así sucesivamente...eventualmente llegara nuestra solicitud a la subred donde esta la maquina esta nos dará su dirección IP del nodo galileo.fie.us.mx" y entonces empieza a remontar todo el árbol de nuevo pasando por el DNS root o raíz y a nuestra maquina.

Bueno esto es algo simplificado a lo que en verdad se mueve esto xD porqué no queremos profundizar la belleza que diría yo del DNS radica en su recursividad una petición de nuestro



cliente puede remontar miles de kilómetros antes de respondernos; como nota curiosa digo que funciona bastante bien. Bueno con todo esto lindo no esta exenta de problemas como los de seguridad; por ejemplo un hacker consiguiera dominar el DNS raíz y modificar la base de datos del DNS después se haría pasar por un host en el que la victima confié, cosas similares ha esto han sucedido pero creo que el servicio DNS estará aun por mucho tiempo.

WINS y DHCP

Bueno por si fuera poco apareció Microsoft y decidió que no estaba bien la forma en que DNS controlaba la red. Mientras DNS se creo par el Internet WINS estaba dirigido a lo que era las redes de área local privadas, con ordenadores donde mantener una IP fija no era muy necesario con todo WINS también da la posibilidad de tener IP fijas pero eso lo vemos después.

Por definición DNS es estático, esto es que no es muy fácil así por así cambiar una IP en el caso de redes que no están conectadas a Internet no es algo necesario de tener ip fijas por ejemplo hoy puedo ser A.b.c.d y mañana ser A.b.c.x la parte de asignacion dinámica de IPs corresponde al servicio de DHCP o Protocolo de Configuración Dinámica de Host este servicio es el que responde a asignar a los clientes dirección IPs dentro de un determinado rango esta se hace cada vez que el cliente se conecte o como nosotros lo hayamos establecido imaginemos un entorno de trabajo donde disponemos de 10 direcciones IP "reales" esto es directamente conectadas a Internet y un número superior de ordenadores. bueno diremos que un máximo de 10 personas podrán acceder a Internet simultáneamente por ejemplo los de la primera planta de la oficina para que visiten la paginita de playboy :) y si mañana queremos que los demás también ingresen entonces tendríamos que cambiar las configuraciones de cada maquina si los demás quisieran conectarse seria un problemas como administradores de la red, esto es solo un ejemplo para que entendamos la función de los servicios que estamos hablando.

Que es donde entra la integración de WINS/DHCP. Puede existir un servidor WINS que atienda las peticiones de los clientes que por lo usual son dos:

Registros:

son mensajes del tipo Mm... quiero que sepas que mi dirección IP es X.X.X.x y que mi nombre es ABCD que vendría hacer la identidad del cliente.

Consultas:

esta es una fusión de WINS

supongamos que una maquina pregunte al servidor por la identidad de otra... Quiero saber la dirección IP de tal maquina x.x.x... si la maquina se ha registrado previamente entonces WINS le responderá por ella veamos un pequeño resumen de lo que es esto.

El sistema operativo se inicia, y se inicializa la interfaz de red. El cliente carece ahora de cualquier identidad IP entones lo que hace es emitir una Broadcast a la red local buscando desesperadamente un servidor DHCP par que le asigne una IP es lo más parecido a una llamado de auxilio de un cliente :)



El servidor DHCP lo que hace es buscar en su configuración o rango una IP disponible para asignarle al cliente si tiene una lo que expide el servidor sería como un ACK y le manda los datos de la identidad, con su nueva y flamante IP el chico nuevo del barrio se alista para anunciarse en la red de su identidad ahí le toca el turno a WINS que como ya sabemos lo que hace es registrarlo al cliente identificado tras el registro la base de datos del servidor WINS ya contiene los datos hago una aclaración, lo que cambia cada vez el la dirección IP no el nombre del cliente, ese si es fijo por ejemplo "FAERIE" que sea el nombre después que te identificas ahora le toca al jefe contactarte y hacer una Chat contigo para hablarte de las horas extras que tienes que hacer si crees que DHCP te salvara no creas porque WINS ya te tiene registrado y solo le bastara saber tu nombre que sería FAERIE y pues DHCP le da a WINS tu IP y listo.

Haber habíamos quedado en que lo que cambia con el DHCP es al IP pero que el nombre es fijo en el cliente, así es cuestión es mucho mas sencillo.

todo esto suena bien en una LAN privada con pocos clientes que usen Lan manager (WfW, Windows /98/NT) pero empieza a dar problemas si las cosas las ponemos más complejas para empezar recordemos el que el protocolo netBeui es no enrutable o podríamos decir que tienen capacidades pequeñas de enrutado con algunas limitaciones

lo que quiere decir que un servidor WINS no recibirá los registros o consultas de cliente que residan en otra subred si es que están separadas por un router ya sea hardware o software. podríamos decir que DHCP se salvaría por los agentes de transito que permiten el trafico de peticiones a través de routers pero aun así seguimos con los problemas de WINS, por no decir que solo las plataformas Lan manager mejor dicho Windows son capaces de hacer peticiones al Server WINS, ya que otra plataforma usaría DNS en su lugar. la buena noticia es que se puede hacer a estas instancias y formar una red mixta con distintas plataformas windows /UNIX / OS/2 / Linux /Mac power etc.

Pues ha llegado la hora de poner las manos en el teclado

haber supongamos que tenemos un Server NT4 (PDC) varias estaciones de trabajo NT, un Linux que hará las veces de Router/gateway/firewall a Internet y más clientes basados en distintas plataformas, bueno vamos a la configuración WINS.

Instalando WINS

Esta configuración e un paso sencillo mejor digo que es difícil de equivocarnos se instala como un servicio mas de Red y una ves reiniciada la maquina empieza a captar los registros de clientes y respondiendo a las peticiones de resolución de IPS, Su aspecto debería de parecerse a la imagen de la figura 1.

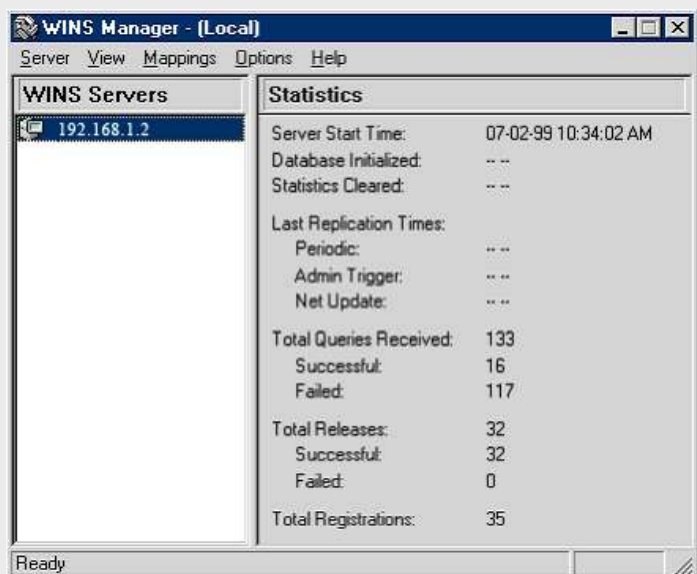


fig 1.



En el administrador de WINS se encuentra lo siguiente:

Total de registros:

Aquí residen el numero de maquinas que han tratado de registrarse al Server un cliente puede ser un nombre de estación como un ID de usuario o un dominio esto permite a un servidor WINS resolver también peticiones de localización de usuarios (por ejemplo para ordenes de tipo " net send <username> esto es para que las maquinas no tengan que recurrir a la multidifusion

Total de liberaciones:

esto representa el numero de clientes que se han dado de baja en el Server Wins esto es cuando cierran su sesión pueden haber liberaciones erróneas pero esto es cuando alguien que no se ha registrado en el servidor se da de baja todos estos datos son necesarios para conocimiento del Adm. de la red

Total de peticiones Recibidas:

esto es el numero de consultas a la base de datos WINS recibidas desde cualquier cliente o sea alguien que pregunte por una maquinas que este en la red supongamos que quiero saber la ip de X maquinas entonces esta consulta para por el Server y queda registrada.

Las duplicaciones:

digamos que no es tan necesario pero eso depende del ancho de la red esto seria por ejemplo si el Server se llegase a saturar ya sea por los registros consultas lo que hace es tener un Server esclavo y configuras en el Server principal quien te ayudara en eso lo único que haces es actualizar tu base de datos en el Server que te servirá de esclavo WINS entonces ya el cliente ira al WINS que este mas libre podría decirse haber ¿como agregamos el "esclavo"? bien en el servidor principal solo debemos irnos al apartado que dice " Agregar servidor WINS" proporcionamos la dirección IP de nuestra maquina y listo, podríamos monitorear todo el trafico desde es el principal seria evitarnos un dolor de cabeza :) es sencillo pues estaríamos acabando con un montón de tráfico de red estúpido en multidifusiones que podrían cargar la red. Bueno nos quedaría por instalar DHCP y los DNS

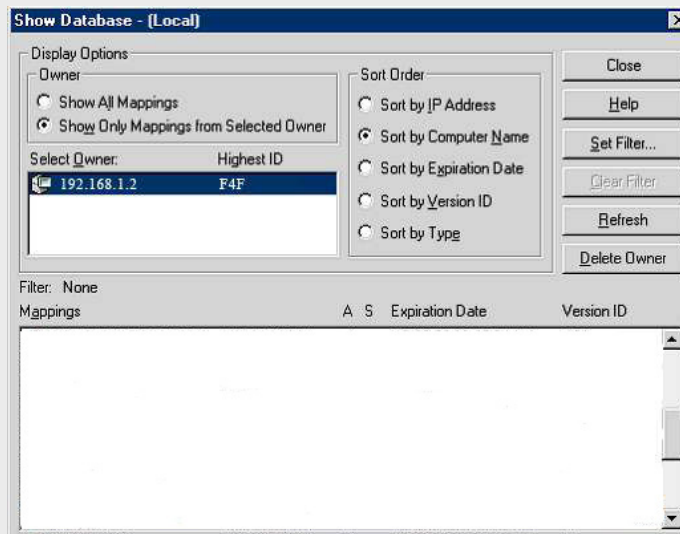


fig 2 - Aspecto de una Base de Datos WINS.



Instalando DHCP...

Después de tener a alguien que escuche nuestras peticiones de registros ha llegado la hora de convertir nuestra Workstation en clientes dinámicos para ello tenemos a nuestra disposición el protocolo de configuración dinámica de host, que básicamente funciona según un conjunto de rangos de direcciones ips disponibles para los clientes estos son los llamaditos ámbitos. Un ámbito como mínimo debe de contener una IP de entrada o inferior y una superior y estas serán asignadas dinámicamente haber pongamos un ejemplo de ámbito activo 192.168.1.0 [sala]

La palabra sala es el identificador del ámbito es como una especie de alias que podemos asignar a los ámbitos para reconocerlos mas fácil; sabemos que esta ip no es real pero es para identificar, después de agregar a nuestro DHCP la IP en el menú de "servidores" a igual que hicimos en el caso de servidor WINS bueno es todo fácil creo que es solo pinchar y listo seleccionamos la opción " Crear" en el menú "Ámbito" aparecerá la ventana de creación de ámbito aquí podemos agregar muchas cosas podemos especificar algunas de las propiedades mas importantes como son la mascara de subred que utilizaremos ya sea que la hayamos configurado o la que viene por defecto en las redes de tipo C también dirección IP inicial, final, exclusiones etc.

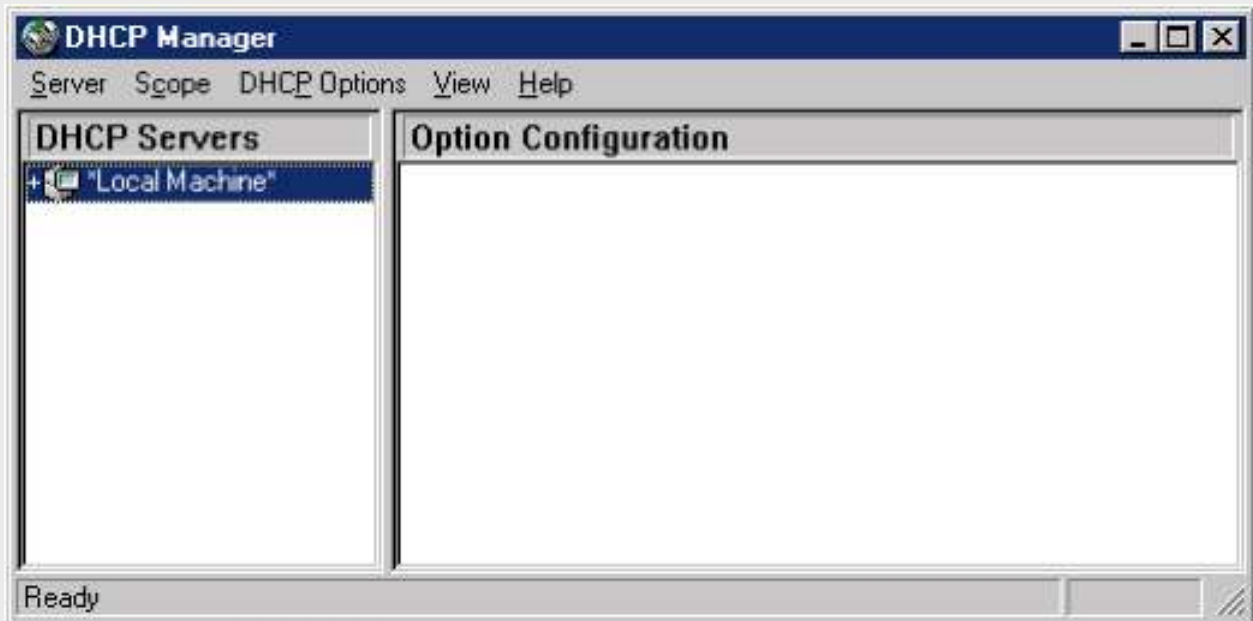


Fig. 3: Ejemplo de ámbito en un servidor DHCP

recordemos que un servidor DHCP puede proporcionar mucha información aun cliente conectado esta información viene en opción de ámbitos en verdad son como 59 opciones pero veremos las más básicas o mejor las necesarias para que todo se integre:



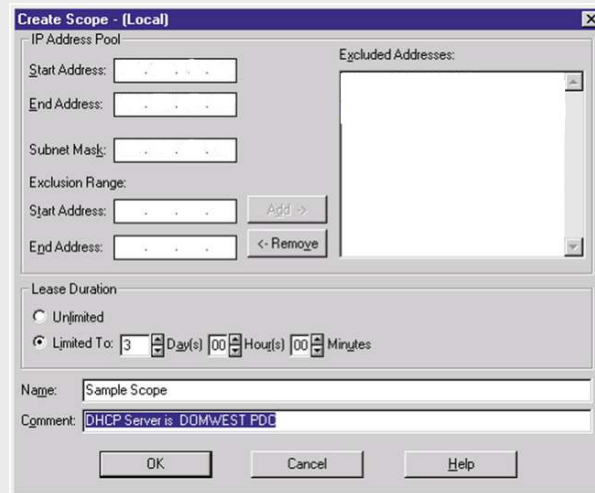


Fig.4: Ventana de ámbito

1) mascara de subred que por cierto ya lo mencione seria 255.255.255.0 por defecto yo uso 255.255.255.224

3) ROUTERS
aquí proporcionamos una lista de enrutadores para la red

6) Servidores DNS....
aquí los servidores disponibles para la red del ámbito

15) NOMBRES DE DOMINIO
esta parte se combina con el nombre de la maquina por ejemplo si ponemos como rsh.es y una maquina del ámbito se llama legolas su nombre completo seria legolas.rsh.es

44) SERVIDORES WINS/NBNS
ya saben especificamos los servidores Wins pero para esto necesitamos configurar la siguiente opción.

46) tipo de WINS/ NBT
bueno para esto tendríamos que ver como funciona el protocolo Lan manager utiliza multidifusiones para anunciarse en la red esto es Broadcast a grito pelado para no alargarnos miren hay tres formas de configurar el Server wins que vamos a usar por ejemplo esta el b-nodo que seria para las multidifusiones como lo anterior dicho, también esta el p-nodo pero es mejor que lo dejemos h-nodo o sea híbrido de los dos modos ¿por que así? bueno porque si por ejemplo un Pc con Linux se quiere registrar, como no se registra con el Servidor de WINS le dará la opción de hacerlo con el DN, por eso usamos el híbrido esta opción es 0x80. que es el código del h-nodo; por ultimo podemos comprobar que todo funcione del siguiente modo: Ipconfig/renew; esto es casi como si hubiéramos presentado nuestra maquina por que empieza el proceso de petición DHCP, y después, Ipconfig/all; ahí te debe de salir una pantalla en donde veremos todo lo que hemos configurado en el Servidor DHCP, esto es casi todo prácticamente por ultimo nos quedaría.



Instalar DNS

...Que bueno llegamos a la última pieza de nuestro esquema, esto de DNS es medio complejo pero solo veremos la instalación de uno pequeño para esto en nuestra Intranet crearemos una zona primaria llamada "rsh.es" como una subzona "sala".

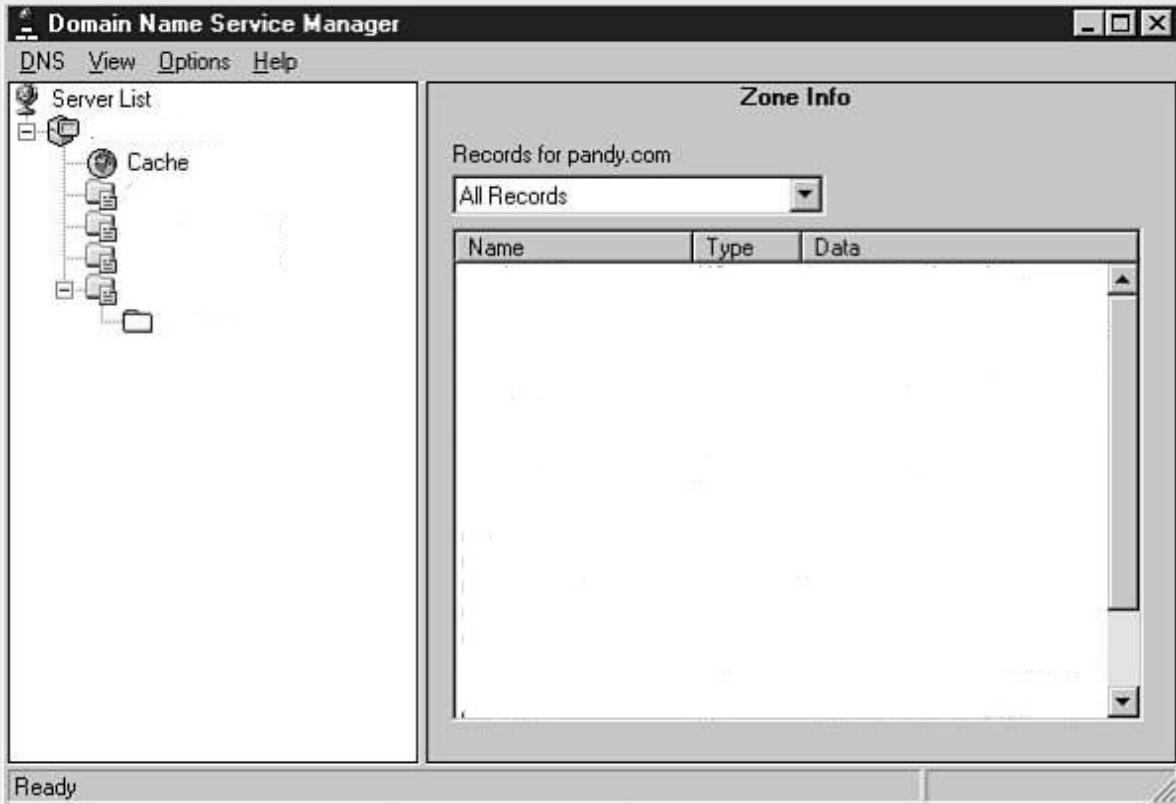


Fig. 5: Zona d

Seguidamente crearemos la zona inversa que se obtiene invirtiendo la dirección de la subred y añadiendo in-addr.arpa

¿porque esto así? Bueno todo esto tiene una explicación lógica pero esto no lo veremos simplemente así se establecerá ya que no pretendo entrar en profundidad del tema siguiendo con lo nuestro:

tenemos la zona inversa como 1.168.192.in-addr.arpa

las otras zonas de cache, 0.in-addr.arpa 255.in-addr.arpa y 127.in-addr.arpa estas ultimas se crean automáticamente por el servidor al realizar las primera ahora es un buen momento para introducir algunos Host, nuestro gateway, que seria mmmm "caronte" este ira justo debajo de la zona rsh.es

por lo que el nombre final de la estación quedara caronte.rsh.es

en la subzona sala Irán todas las maquinas NT de esta forma sus nombres DNS quedaran como <nombre>.sala.rsh.es



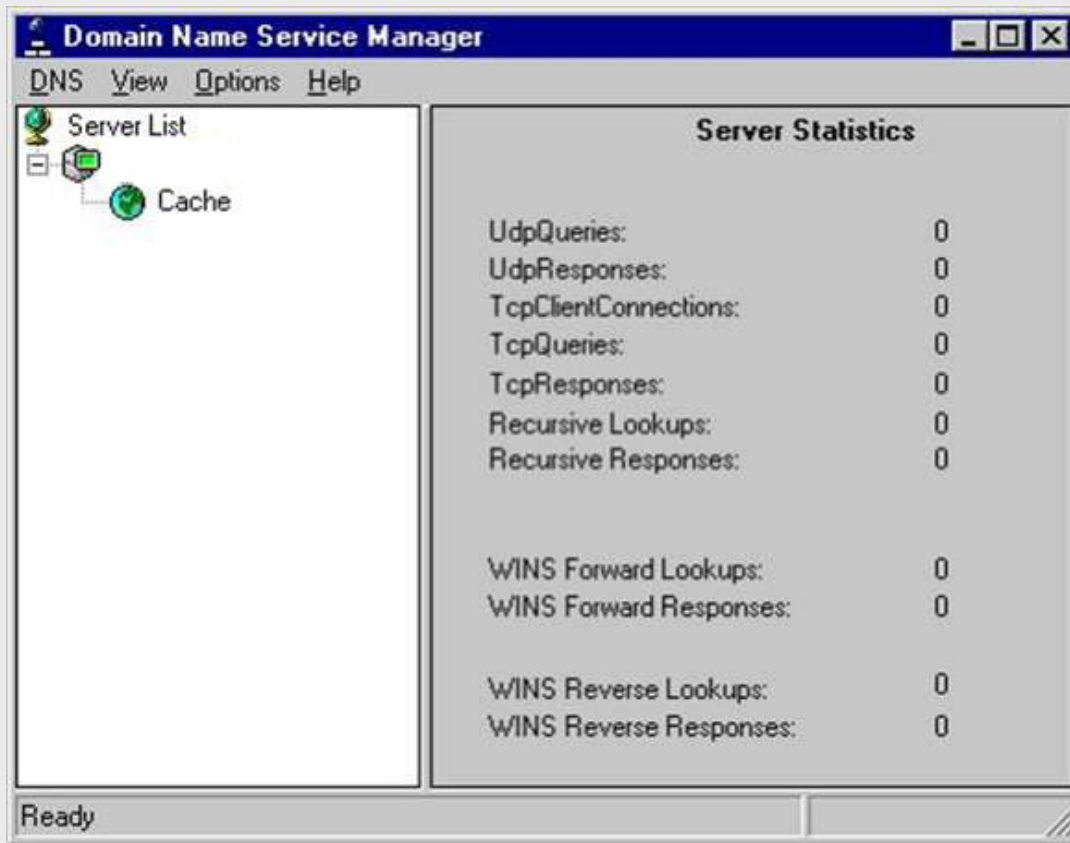


Fig. 6: Estadísticas del Servidor DNS

Lo que mas nos interesa es el registro de tipo WINS este registro indica que el servidor DNS redireccionará al servidor WINS

por ejemplo si un cliente hace una petición al DNS sobre una maquina Windows que utilice WINS el cliente podrá hallar su dirección IP dado que si no esta registrada el DNS estará en el WINS; bueno para que fusiones todo esto aclaro debemos especificar un opcional configurar el DNS. Decíamos que de la pestaña Wins Look Up esto es español viene a ser como "enganche de WINS" activamos la casilla y ponemos "usar resolución WINS" y agregamos la dirección del mismo...con esto DNS preguntara por una maquina cuando no encuentre algún nombre en su base de datos, esto es el flujo de datos DNS-->WINS. Vamos a rematar la faena

esta es una forma de centralizar la administración de la resolución de nombres bueno desde el DNS podremos también ver un vistazo de todo si es que queremos con esto hemos integrado el tráfico de red con todas las plataformas y los protocolos.

OpTix
optix@hackertm.org
www.hackertm.org

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

END 

Proyecto TORO

by Zorro

El argentino Matias Vara, creador de Toro, quien aun sigue en su proyecto nos muestra una leve pero interesante nota acerca de su funcionamiento y desarrollo.

El proyecto surge a comienzos del año 2003 , cuando se penso en el foro de prehackers.com diseñar un S.O básico , yo que era y soy lector de ese foro , leí el post y me surgió la curiosidad . Comencé buscando por la web información sobre Sistemas Operativos , de la cual encontré un montón . Pero aun no me sentía totalmente informado como para encarar un proyecto de esta magnitud , es por eso que leí los manuales de programación de Intel sobre el x86 , también el libro de Tanenbaum sobre su S.O. Minix , entre otros . En un primer momento me uní al proyecto de prehackers , pero viendo que este no se continuaba , decidí abrirme y desarrollarlo yo mismo . Al momento de decidir en que lenguaje lo desarrollaría , pense en c y pascal , pero no tenia un gran manejo c , a aparte , quería escapar de la rutina y poder lograr un S.o. escrito íntegramente en Pascal.

Navegando por la web, me encontré con el Proyecto Freepascal, que es un compilador Pascal de 32 bits, comencé a probar el lenguaje y me llamo mucha la atención la potencia y la posibilidades que da su sintaxis. Al principio comencé realizando bocetos a mano de cómo seria un manejador de interrupciones, un scheduler, etc, y cree un ambiente de trabajo que me permitiera compilar el núcleo rápidamente:

- * Configure el boch para que boote desde un disket.
- * Desarrolle un programa para copiar el sector de booteo y el kernel a la imagen de boch
- * Cree un simple booteador , que copiaba el kernel a memoria , pasaba a modo protegido y salta al inicio del kernel.

Tuve que configurar el compilador FPC para que genere un binario plano y que no utilizase llamadas al sistema de Windows . Esto fue fácil puesto que FreePascal genere el archivo en assembler , luego utilizando grep quitaba los procedimientos de propios del compilador , como el procedimiento de salida y otros mas . Luego que generaba el archivo assembler .s , es generado el archivo objeto utilizando As . Todos los archivos objeto son linkeados con ld , en un único binario plano llamado Toro.bin .

La escritura del núcleo comenzó aproximadamente en Agosto del 2003 apartir del modulo de la memoria . En Mayo del 2004 termine un precario FS a través de inodos , para esa epoca me aceptaron el proyecto en sourceforge , y subi la version 1.0.0 . En la actualidad , voy por la version 1.0.3 , en la cual he logrado un sistema bastante estable , no puedo decir que a un 100% pero voy en camino a eso .



Estructura de Toro

El booteo de Toro se realiza por ahora desde un disket 3 • , porque es la forma mas rapida de debuguearlo en diferentes maquinas . Corre sobre Modo protegido por ahora solo sobre arquitecturas x86 y es multitasking .. Es un sistema del tipo monolitico .

```

Bochs for Windows [F12 enables mouse]
Total Memoria: 16777216
Memoria Libre: 14680064
Cantidad de Paginas : 3581
Primer Pagina : 515
Paginas en Mem_Map : 3
Iniciando Malloc ... Ok
Iniciando las Irq ... Ok
Cpu : Intel Family: 5 Model : 1 Types : 0
Iniciando Buffer ... 407 Bufferes
Iniciando PCI Bus ... Ok
Pci Bus :Presente
Pci Bus: Encontrados 0
Iniciando FDC Driver ... Ok
fd0 : 3.5
fd1 : No encontrada
Iniciando Ide Driver ... Ok
hda0 : No Encontrada
hda1 : No Encontrada
hdb0 : No encontrada
hdb1 : No encontrada
Iniciando TTY Driver ... Ok
Iniciando Driver de Keyboard ... Ok
Iniciando Scheduler ...Ok
Unidad Root : fd
    
```

Booteo de Toro utilizando el emulador Bochs

Memoria :

Aplica un modelo paginado – segmentado de memoria .

La memoria Física es utilizada a partir de *Mem_Ini* , esta es igual a el 2 MB de memoria Física , puesto que el segmento desde 1 MB – 2MB es utilizado para alojar al binario del kernel .

Existen 4 descriptores globales , el nulo (obvio) , el de datos y código del kernel con una extensión de 4 Gb , y los datos y código de usuario , también con 4gb de extensión . El espacio virtual del kernel es todo el primer 1 gb de memoria y del 2 al 4 es la memoria de usuario .

Esto hace que a través del PDT de usuario el kernel no pueda acceder a paginas mas haya del 1 gb , y si quiero que pueda deberían realizarse muchas modificaciones del registro cr3 que haria mas lento el sistema , Por lo tanto las paginas son agrupadas en dos pilas , aquellas que se encuentran antes del 1 Gb y las que se encuentra mas haya del 1 gb ,



llamado `High_Memory` , estas son las asignadas al usuario . Como no se aplica `swap` , si la memoria fuese de menos de 1gb , los procesos de usuario comenzaran a tomar paginas de la zona baja . En el caso de que la pila del kernel se agote , el sistema colapsa , puesto que el kernel no puede tomar memoria de la zona alta .

El kernel pide memoria a través de la llamada `kmalloc()` , que maneja bloque de 2 , 4 , 16 , 32 , 64 , 128 , 256 , 512 , 2048 y 4096 bytes . Cada conjunto de objetos posee un limite de asignación de descriptores , este es de $1024 * \text{Max_Malloc_Page_Desc}$ descriptores .

Los procesos de usuario poseen la memoria organizada en estructuras del tipo `vmm_area_struct` . Estas descriptas en las estructuras `tarea_struct` de cada proceso . Cada una posee características propias como permisos de acceso , longitud , comienzo , etc . Cada proceso de usuario posee por ahora dos áreas , la de código + datos , que comienza a partir de `High_Memory` , y la de pila de usuario , que es a partir del 3 gb .

Procesos :

Los procesos son creados con la llamada `Proceso_Crear()` , esta crea un proceso vacío . Los procesos son identificados por su `Pid` y su `ppid` . El `pid` 1 corresponde a `Init` y el 2 a el thread nulo . Los procesos se encuentran agrupados en una tabla `Hash` , para su fácil acceso . Se tiene soporte de `Signal` y `timers` . El `scheduler` , por ahora , soporta solo un algoritmo , este es una simple cola `RR` , en el cual se le asigna a cada proceso un `quantum` de tiempo .

Son implementadas la mayoría de las llamadas al sistema de Unix , como `Exit()` , `Fork()` , `Waitpid()` , `Getpid()` , `getppid()` , `Signal()` , `Kill()` , , entre otras , cuyos resultados son iguales que en las de Unix .

También se cuenta con la posibilidad de crear `Threads` de kernel que corren en el espacio del kernel , un ejemplo es el thread nulo , que corre como si fuese una tarea cualquiera , en la cola `RR` .

Entrada – Salida :

La entrada - salida esta muy vinculada al `Fs` , ya que este a través de la llamada `Register_Chrdev()` y `Register_blkdev()` , permite el acceso a los dispositivos como si fueran archivos , este tratamiento es muy similar al implementado en Linux con su `VFS` .

Cada dispositivo esta identificado por un numero Mayor y uno Menor . Con el Mayor se identifica que tipo de dispositivos es (`fdc,ide,tty,keyb`) y con el menor se identifica a un dispositivo de esa clase en particular . Por cada nuevo numero mayor , hay una array de manejadores que puntean a las funciones `Open` , `Write` , `Seek` , `close` , `ioctl` y `Read` . Hasta la actualidad se cuentan con `drivers` para `tty` , `fdc` , `ide` y `keyb` .

Cada dispositivo posee una cola `wait` , en las que duermen los procesos que quieren acceder al `hard` y este se encuentra ocupado .

Son identificados dos tipos de `irq` de `hard` , las `irq` cortas y las largas . Las cortas son aquellas que se producen de forma sucesivamente rápidas y para captarlas se utiliza la llamada `Wait_Short_Irq()` , en las que el kernel deja que `soft` realice todo el tratamiento de la `irq` .



la irq . Las largas , por el contrario , son aquellas que pueden esperar a ser atendidas como son el caso de las irq de disco , estas son captadas a través de la llamada *Wait_Long_Irq()* , y que , a diferencia de las cortas , el proceso solicitante es dormido y despertado cuando se produce la irq , aquí es el kernel quien realiza todo el tratamiento , enviado un EOI .

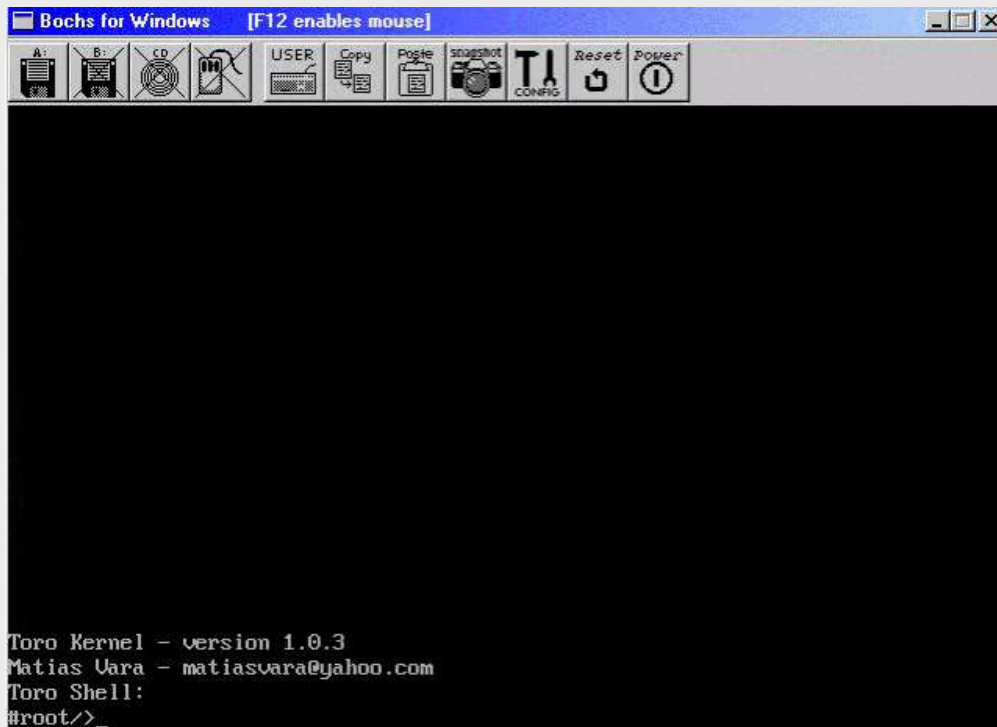
FileSystem :

El Filesystem es atrevas de inodos y superbloques . Esta basado en el Fs de Minix , salvo algunas excepciones . En próximas versiones se implementara un Fs virtual que de soporte a cualquier tipo de Sistema de Archivo . Esta parte del S.o. es la que mas me costo , puesto que después de un año de estar trabajando solo en ella puedo decir que es lo suficientemente estable .

La capa mas baja del Fs , es el buffer cache , que mantiene en memoria , a los bloques utilizados y los que no , a traves de dos Colas , la cola Hash ordenada por numero mayor y la cola Lru que se encuentra ordenada por los bloques mas utilizados .

También esta implementada la llamada al sistema *Exec()* que , a diferencia de Unix , no sobrescribe al proceso ejecutante , sino que crea a un nuevo proceso , el cual tiene como ppid al proceso ejecutante . Esta soporta argumentos .

También son implementadas la mayoría de las llamadas al sistema de Unix , como *chmod()* , *mkdir()* , *mknod()* , *open()* , *write()* , *mount()* , *unmount()* , entre otras .



Toro Shell cargada



Kernel :

La ejecución del S.O. se realiza a partir del archivo kernel.pas . Este crea la tarea Init y el Thread nulo , luego se entra a en un bucle .

Las llamadas al sistema se realizan a través de la interrupción 50 , y los parámetros son pasados por los registros .

La tarea Init , se encarga de montar la unidad root y de ejecutar la shell , que es el archivo *sh* . La unidad root puede ser especificada en cualquier dispositivo físico , aunque por ahora se implemente un disket . Luego de esto permanece en la cola Waitpid aguardando por procesos huérfanos .

```

Bochs for Windows [F12 enables mouse]
-----
Toro Kernel - version 1.0.3
Matias Vara - matiasvara@yahoo.com
Toro Shell:
#root/>>/ls /
ewrr-----      sh                11332
ewrr-----      ls                10804
ewrr-----      mk_dir           9632
ewrr-----      mknod            9504
ewrr-----      mount            9560
-wr-c-----      tty0              0
-wr-c-----      keyb0             0
-wr-b-----      fd0                0
ewr-d-----      /dev              0
#root/>>_
    
```

Programa clásico de Linux el ls listando root

Bueno este es un breve resumen del funcionamiento de Toro , hay una descripción mas profunda en toro.sourceforge.net/estructura.htm .

También pueden encontrar una guía de cómo compilar Toro aquí, <http://toro.sourceforge.net/compilando.html> . Esta se refiere a la versión que se encuentra en cvs y que es un release de la version 1.0.3 , que corrige gran cantidad de bugs .

Zorro
matiasvara@yahoo.com
<http://toro.sourceforge.net>

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

END

KPGP & GPG

por Jimmy O'Regan

Traducción Español by Ksaver

Una de las virtudes de Linux es que su herencia de Unix le ha dado una poderosa línea de comandos. Usar la línea de comandos puede ser difícil sin embargo, y mucha gente prefiere usar GNOME o KDE para hacer su trabajo. Afortunadamente, muchos desarrolladores eligen acoger la virtud de la "holgazanería" de Larry Wall, y en lugar de elegir reinventar la rueda, crean interfaces gráficas de uso fácil para aplicaciones CLI comunes.

Éste es el primer artículo de una serie ocasional que intenta mirar a algunas de éstas interfaces gráficas, mostrar al nuevo usuario cómo usarlas, y, más importantemente, proveer los comandos correspondientes para el modo de consola para futura referencia - nunca sabes cuándo puedes necesitar hacer algo sobre una conexión SSH, después de todo! KGPG y GPG

KGPG es una interfaz gráfica para GPG, el "GNU Privacy Guard" (Guardián de la Privacidad GNU). GPG fué creado para sustituir a PGP, el popular programa de encriptado. GPG es un componente común en sistemas Linux - casi todos los paquetes lo usan para verificación, por ejemplo.

GPG es una implementación de OpenPGP (RFC 2440), un estándar creado a partir del desarrollo de PGP, para proveer seguridad para, entre otras cosas, correo electrónico. PGP es mejor conocido como una implementación de criptografía de clave pública - cada usuario tiene dos claves, una pública, y una privada. Si yo quiero enviarle un correo electrónico a Mark, lo encripto usando mi clave privada y su clave pública; Mark es capaz entonces de desencriptarlo usando su clave privada y mi clave pública.

El uso más común de PGP, sin embargo, es como una forma de firmar algo digitalmente - normalmente correo electrónico, o como se mencionó antes, paquetes de software - entonces el receptor puede verificar que el artículo viene de la persona que dice haberlo enviado.

Iniciando KGPG



Cuando inicias KGPG por primera vez, presenta un Wizard que te ayuda a configurar GPG. Está bien dejar los defaults. El paso final, y en el que estamos interesados, es el diálogo de creación de claves.





El diálogo de generación de claves de KPGP contiene los valores comunes por defecto, todo lo que necesitas hacer es ingresar tu nombre y correo electrónico en las áreas apropiadas, y hacer click en OK. Observa que el "Modo Experto" te provee con una consola, e inicia GPG por tí.



KPGP entonces te presenta un diálogo que contiene la firma de tu clave ID y huella digital. También te ofrece crear un Certificado de Revocación. Puedes usar esta opción - la mejor opción es escoger "Guardar como", mover el fichero a un lugar seguro, y borrarlo de tu computadora. Ésta es una opción de la que desearía haber tomado ventaja - hay dos viejas claves que me pertenecen pululando por ahí de las que desearía haberme librado!



Si quieres generar un Certificado de Revocación en algún momento posterior, simplemente haz un click derecho en la llave en la ventana de Administración de Claves, y selecciona "Revocar Clave".

Iniciando con GPG

Crear un par de claves en GPG es muy fácil: usando el comando `gpg --keygen` se te ofrecen los mismos valores por defecto que en KPGP, simplemente presiona la tecla 'Enter' para aceptarlos y escribe 'y' seguido de 'Enter' cuando pregunte "Es esto correcto". Cuando lo pida, ingresa tu nombre y dirección de correo electrónico.

Aquí está una muestra de una sesión de creación de claves:

```
[foo@dhcpc0 foo]$ gpg --gen-key
gpg (GnuPG) 1.2.4; Copyright (C) 2003 Free Software Foundation, Inc.
Este programa no tiene ABSOLUTAMENTE NINGUNA GARANTÍA.
Es software libre, y eres bienvenido a redistribuirlo
bajo ciertas condiciones. Ver el archivo COPYING para detalles.
```

Por favor Selecciona que tipo de clave quieres:

- (1) DSA and ElGamal (default)
- (2) DSA (sólo firma)
- (4) RSA (sólo firma)

Tu selección?

Par de claves DSA tendrá 1024 bits.

Para generar un nuevo par de claves ELG-E.

tamaño mínimo de clave es 768 bits

tamaño por defecto de clave es 1024 bits

tamaño máximo aconsejado es 2048 bits

Que tamaño de clave quieres? (1024)

Tamaño solicitado es 1024 bits

Por favor especifica por cuánto tiempo será válida la clave.

0 = la clave no expira

= la clave expira en n días

w = la clave expira en n semanas

m = la clave expira en n meses

y = la clave expira en n años

La clave es válida por? (0)

La clave no expira en absoluto

Es correcto (y/n)? y

Necesitas un ID de usuario para identificar tu clave; el programa construye la id de usuario a partir del Nombre Real, Comentario y Dirección de Correo Electrónico en ésta forma:

"Heinrich Heine (Der Dichter) "

Nombre real: Foo McBar

E-mail: foo@bar.com

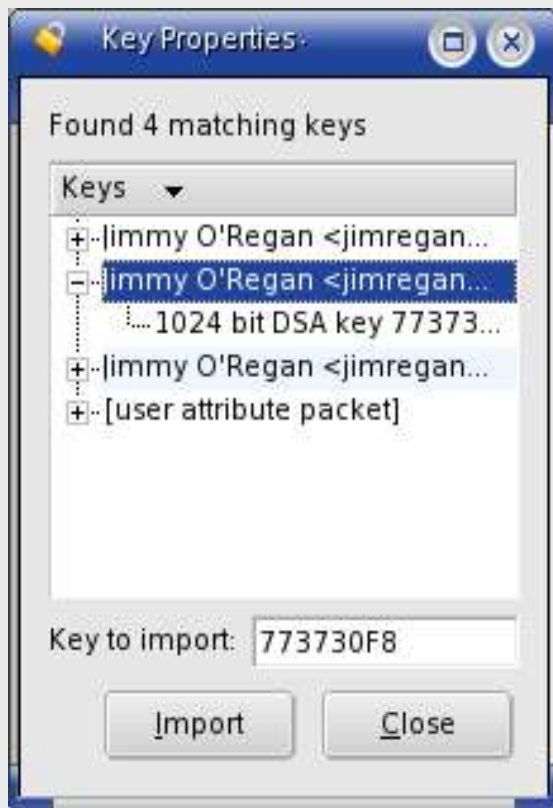


Por favor muevelo a un medio que puedas esconder lejos; si alguien malintencionado tiene acceso a este certificado puede usarlo para hacer tu clave inservible. Es inteligente imprimir este certificado y guardarlo lejos, sólo en caso de que tu medio se vuelva ilegible. Pero ten algo de cuidado: el sistema de impresión de tu máquina puede guardar los datos y hacerlos disponibles para otros!

Importando claves



Para encontrar una clave con KGP, escoge "Archivo->Diálogo del servidor de claves". En el diálogo, ingresa el nombre, dirección de correo electrónico o ID de la clave que deseas importar. Si hay múltiples coincidencias, se te presenta un diálogo desde el que puedes escoger la clave correcta.



Para encontrar una clave con KGP, escoge "Archivo->Diálogo del servidor de claves". En el diálogo, ingresa el nombre, dirección de correo electrónico o ID de la clave que deseas importar. Si hay múltiples coincidencias, se te presenta un diálogo desde el que puedes escoger la clave correcta.



```
[foo@dhcpc0 foo]$ gpg --search-keys "Jimmy O'Regan"
gpg: buscando "Jimmy O'Regan" en servidor HKP subkeys.gpg.net
Claves 1-3 de 3 para "Jimmy O'Regan"
(1) Jimmy O'Regan <jimregan@o2.ie>
    1024 bit clave DSA 773730F8, creada 2004-06-19
(2) Jimmy O'Regan <jimregan@o2.ie>
    1024 bit clave DSA DA974449, creada 2004-06-05
(3) Jimmy O'Regan <jimregan@lit.compsoc.com>
    1024 bit clave DSA FF5D8291, creada 2000-08-22
Ingresa número(s), N)ext, o Q)uit > 1
gpg: key 773730F8: "Jimmy O'Regan <jimregan@o2.ie>" importada
gpg: Número total procesado: 1
gpg:          importado: 1
```

Observa que si no tienes un servidor de claves configurado en ~/.gnupg/gpg.conf puedes agregar la opción --keyserver [servidor de claves]. Por ejemplo:

```
gpg --keyserver subkeys.gpg.net --search-keys "Jimmy O'Regan"
```

Si ya tienes una clave pública en un fichero, puedes importarla a KGPG usando "Claves->Importar clave", donde puedes también ingresar el nombre de archivo, o navegar hasta él. Este diálogo también da la opción de importar desde el portapapeles - He encontrado esto útil cuando necesito contactar al administrador de sistemas de Gaceta Linux, Kayos, quien mantiene su clave en su página de contactos.

En GPG, para importar desde un fichero, usa gpg --import nombre-de-archivo. Nota que esto acepta tuberías estándar, entonces hay maneras de emular la entrada del portapapeles de KGPG. El modo más usado para trabajar donde sea es usar un "documento de este directorio":

```
[foo@dhcpc0 jimmy]$ gpg --import <<EOF
> (pegar clave pública)
>EOF
```

Cómo obtener el contenido del portapapeles en las "X" llegó recientemente en The Answer Gang, donde Ben apuntó a su tip de xclip del Número 78. Para usar esto con GPG, puedes intentar:

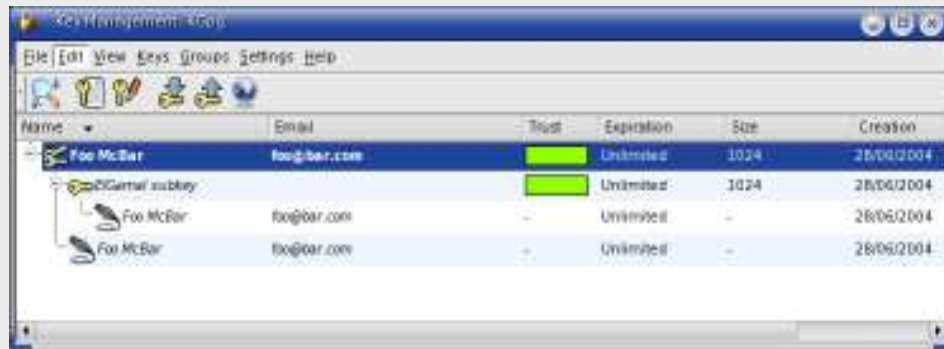
```
xclip -o|gpg --import
```

Alternativamente, si usas KDE, pero no usas KGPG, puedes intentar:

```
dcop klipper klipper getClipboardContents|gpg --import
```



Viendo claves



No hay pasos especiales a seguir en KGPG para ver clave o firmas - la ventana de Administración de Claves muestra todas las claves en tu llavero. Puedes hacer click en el signo '+' junto a cada nombre de clave para ver alternadas las UIDs, firmas, IDs de foto etc. Si deseas, puedes tener IDs de foto apareciendo en la ventana principal seleccionando una de las opciones in "Ver->IDs de foto".

En GPG, puedes usar una de varias opciones, dependiendo de que quieres ver. Si deseas ver claves públicas, usa `--list-keys`; si deseas ver claves secretas, usa `--list-secret-keys`; y si deseas ver claves secretas y sus firmas, usa `--list-sigs`. Con todas estas opciones si especificas un nombre, GPG mostrará sólo los detalles para ése nombre.

Exportando Claves

Para exportar tu clave pública en KGPG, escoge "Claves->Exportar Claves Públicas", o presiona Ctrl-C. Esto te ofrece exportar la clave en un correo electrónico, al portapapeles, al servidos de claves por defecto, o a un archivo - `$HOME/$USER.asc` por defecto.

En GPG, para exportar una clave, usa `gpg --export -a [id de clave]`. La opción `-a` es importante si quieres usar la clave en correo electrónico, ésto codifica la clave en "armadura" ASCII. si no deseas especificar la ID de clave, el comportamiento por defecto es exportar todas las claves, que probablemente no es lo que tú quieres.

Alternar IDs

Si tienes múltiples direcciones de correo electrónico, y deseas usar la misma clave para cada una de ellas, puedes desear agregar una ID de usuario extra. En KGPG, simplemente haz un click derecho en el archivo y selecciona "Agregar ID de Usuario". Esto te muestra un diálogo que te pide un nombre, dirección de correo electrónico y un comentario opcional.

En GPG, usa `gpg --edit-key [id de clave]`. Esto te da un prompt diciendo `Command>`. En este prompt, escribe `adduid`, e ingresa el nombre, correo electrónico, y un comentario opcional según te lo pida.

Para agregar una ID de foto en KGPG, haz un click derecho en el nombre al que deseas agregar la foto, escoge "Agregar foto", y navega hasta la localización del archivo. En GPG, debes usar otra vez `gpg --edit-key`. Esta vez, desde el prompt `Command>` escribe `addphoto`. Cuando te lo pida, escribe la ruta hacia la foto.



Es importante señalar que KGPG, o al menos KGPG1.1, como viene con Mandrake 10.0, define una nueva UID como la UID primaria, en lugar de una alternativa, como lo hace GPG. Para cambiar la UID primaria, debes usar el modo de edición de GPG. Selecciona el número de la UID, sigue los pasos de arriba, y usa el comando primary Borrando

En KGPG, ésto es muy sencillo. Selecciona la clave o firma que deseas borrar, y presiona la tecla 'Borrar'. No es posible borrar todas las firmas desde KGPG sin embargo - las firmas en una UID alternativa o ID de foto deben ser borradas desde la línea de comando. Puedes tener acceso rápidamente a GPG desde "Claves-> Editar Clave en Terminal", o presionando Alt-Enter.

En GPG, para borrar una clave pública, usa `gpg --delete-keys [id de clave]`; para borrar una clave secreta, usa `gpg --delete-secret-keys [id de clave]`. Para borrar una firma, debes usar otra vez el modo de edición. GPG enumera los UIDs de la clave por tí. Para trabajar en una UID, escribe su número. GPG entonces listará cada UID otra vez, con un asterisco junto a cada clave seleccionada. Usa el comando `delsig` para borrar una firma. GPG te preguntará con cada firma en turno; presiona 'y', 'n' o 'q' seguido de 'Enter' para borrar, saltar una firma, o dejar el modo de borrado de firmas, respectivamente. Simplemente presionando 'Enter' escoge la opción por defecto, lo cual es saltarse la firma.

Firmado de claves

Para firmar una clave, primero impórtala. En KGPG, escoge "Claves->Firmar Clave(s)"; en GPG, usa `gpg --sign-key [id de clave]`. Ambos te solicitarán confirmar ésto, y ambos preguntarán que tan bien has verificado la identidad del dueño de la clave. En GPG ésto aparece como:

- (0) No contesto. (por defecto).
- (1) No lo he comprobado en absoluto.
- (2) He hecho una comprobación informal.
- (3) Lo he comprobado meticulosamente.

Si no puedes decir que has hecho una comprobación meticulosa, es recomendable que no firmes la clave en absoluto. Una vez que hayas firmado la clave, puedes exportarla, entonces el dueño puede importar tu firma. Para mayor información sobre firmado de claves, ver el CÓMO de firmado de claves GPG o La Guía de firmado de claves de Debian.

Encriptando, desencriptando y firmando

Con KGPG instalado, puedes encriptar cualquier archivo que desees desde konqueror haciendo un click derecho, y seleccionando "Acciones->Encriptar archivo"; o soltándolo sobre el ícono DE KGPG en la Barra de Tareas. Selecciona a la persona que desees que reciba el fichero, y KGPG creará una versión codificada en ASCII del archivo encriptado. Para desencriptarlo, de la misma forma arrastra el fichero sobre la barra de tareas.

En GPG, usa `gpg -e [nombre de archivo]`, e ingresa la ID de la clave cuando te lo pida. Puedes usar entonces `gpg --enarmor [nombre de archivo]` para codificarlo en ASCII. Usa `gpg --decrypt [nombre de archivo]` para desencriptarlo.

Para firmar ficheros en KGPG, primero debes cambiar las preferencias para permitirlo. Ve



a "Preferencias->Configurar KGpg"; en el panel "Interfaz de usuario", cambia "Acción al soltar un archivo sin cifrar" a "Preguntar" (o "Firmar" si no tienes intenciones de usar el encriptado). Esto te permite soltar ficheros sobre la barra de tareas para firmado.

En GPG, para firmar un fichero usa `gpg --sign [nombre de archivo]`. Puedes codificar ésto en formato ASCII, como para encriptado, si deseas enviar por correo electrónico el fichero.
Conclusión

Espero que alguien ahí afuera encuentre ésto útil - si aún un usuario de KPGP encuentra un comando GPG correspondiente usando ésto, estaré feliz. Si alguien lo encuentra útil como una introducción a cualquiera, KPGP o GPG, siéntase libre de enviarme un correo electrónico encriptado - mi clave pública está disponible aquí. Hasta la próxima, cuidense!
P.D.

Me gustaría tomar un momento para corregir un error en el artículo sobre las Extensiones Mozilla del último mes, y agradecer a quienes hicieron observaciones.

Mencionando PopUpALT, dije "Esta característica, que estaba presente en Netscape 4, fué removida desde Mozilla por alguna razón desconocida." Lo que Quise decir fué "por alguna razón que no he descubierto".

Marcin Gil fué el primero en escribir, diciendo "PopUpALT fué probablemente removido de mozilla porque (si mi memoria no me falla) el atributo ALT es para lectores de pantalla, navegadores de texto, etc. Tal uso es recomendado por W3 en las especificaciones de XHTML (creo que Jeffrey Zeldman lo escribe en su libro). El atributo TITLE es para presentar textos en pop-ups.."

Crystle aclaró ésto: "De hecho, el W3 dice que el alt tag debe ser usado cuidadosamente, y he visto algunos alt tags realmente largos - esto causa problemas si tu navegador traduce a voces o braille, lo cual es el propósito del alt tag.

" Ver: <http://www.w3.org/TR/html401/struct/objects.html#alternate-text>

A lo cual Josh Ockert agregó: "El comportamiento correcto es desplegar el valor *entero* de un atributo alt si la imagen no es cargada".

Ksaver
ksaver@hackertm.org
www.hackertm.org

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

END 

Creación de SUBREDES

by OpTix

Es sabido que una red proporciona los medios para la comunicación de Pcs y otros dispositivos; lo que hace la subred es subdividir una red en secciones distintas. Recuerda que la dirección IP está dividida en la parte IP de red y en la parte que identifica a el equipo.

*Como por ejemplo 192.168.1.1 En este ejemplo notamos que los tres primeros octetos identifican a la red y el último sería al equipo por lo que es de clase C. Algunas veces es necesario dividir la parte de la red de modo que pueda representar múltiples redes, esta división de la identificación (ID) de red se conoce como creación de subredes y se la lleva a cabo por medio de una máscara de Subred personalizada "Subred".

¿Por que queremos dividir la parte de red de la dirección IP?

*miremos un ejemplo: supongamos que donde trabajamos haya una red con menos de 100 usuarios y nuestro ISP nos ha asignado una dirección de red de clase C, 192.168.1.0 supongamos que queremos separar la red en dos segmentos físicos; primero podríamos poner encaminadores y segmentar la red, pero para segmentarla tendríamos que separar la red en dos partes y luego se podría ir al ISP y pagar por otra IP de clase C. Pero esto no es lo mas inteligente, lo recomendable es dividir la red en dos segmentos esto puede lograrse aplicando una máscara de subred personalizada.

Al escoger la máscara de subred apropiada se consigue que la red tenga un número pequeño de ID de red con una mayor número de ID de equipos o que tenga un mayor número de subredes más pequeñas, bueno, existen las máscaras de subred por defecto *Las que maneja nuestra IP dependiendo su clase.

¿Que es lo que llamamos máscara de subred y que hace?

Una máscara de subred es una secuencia de números de 4 octetos muy similar a una dirección IP. Su trabajo es identificar que parte de una dirección IP es el ID de la red y que parte es el ID del equipo; vamos a mirar una tablita que hice de las máscaras de subred por defecto.

Clases de direcciones IP	Máscara de subred
Clase A:	255.0.0.0
Clase B:	255.255.0.0
Clase C:	255.255.255.0

*ok estas máscaras de subred vienen por defecto

Cuando se establece TCP/IP en un sistema ya sea Win o Unix debe proporcionárseles una máscara de subred, esto permite que TCP/IP sepa como dividir la dirección en sus partes



de red y de equipo. Las mascararas de subred por defecto permiten a las diferentes clases de dirección dividir sus direcciones IP en distintas relaciones. Haber como vimos la mascara de subred de clase A reserva el primero octeto para la parte del ID de red y los tres últimos Octetos para los equipos y como funciona las mascara de Subred.

Las mascararas de Subred son como un tipo de direcciones IP se usan para dividir las direcciones IP en el ID de la Red y en el ID de equipo también digo que podemos escribir las direcciones IP en binarios.

Por ejemplo la dirección IP 192.168.16.1 se escribirá en binario como:

```
11000000 10101000 00010000 00000001
```

así seria, podemos hacer lo mismo con la mascara de subred puesto que la dirección es ejemplo: 192.168.16.1

La mascara seria 255.255.255.0

Que en binarios seria 11111111.11111111. 11111111. 00000000

Ósea la mascara de Subred por defecto en binario.

Si colocamos ahora la mascara de Subred binaria debajo de la dirección IP binaria podemos utilizarlas para saber las partes del equipo y de la red de la dirección IP.

Los bits de la dirección IP que corresponden a 1 en la mascara de subred corresponden a los bits de la dirección IP que conforma la parte del equipo de la dirección que corresponde a 1, haber veámoslo para mejor entendimiento.

```
11000000 10101000 00010000 00000001 dirección IP
11111111. 11111111. 11111111. 00000000 Mascara de subred
11000000 10101000 00010000 00000000 ID de red
```

ok, tenemos la dirección IP arriba en binarios, de ahí vemos la mascara de subred por defecto; entonces si buscamos los unos en la mascara de subred estos hacen parte de lo que identifica a la red verdad?... y ponemos su valor entonces por defecto abajo, la final solo vemos que no hay nada porque eso ya seria del equipo. Al traducirlo de nuevo quedaría así 192.168.16.0

Seria de la red OK ¿ahora vamos a ver la parte que identificaría al equipo es igual solo que ahora buscamos 0 en ves de unos en la mascara de subred; veámoslo como quedaría la obtención de la ID de equipo buscando ceros

```
11000000 10101000 00010000 00000001 dirección IP
11111111. 11111111. 11111111. 00000000 mascara de subred
```

```
00000000. 00000000. 00000000 .00000000
```

Ahí seria, al traducir el ID de equipo a decimales, tenemos 0.0.0.1 lo que nos da un ID de equipo que corresponde exactamente al ultimo octeto de la dirección.



OK en este ejemplo que vimos hay 24 bits que hacen referencia a la red de la dirección y quedarían 8 bits en la parte distinta de cero que están disponibles para los equipos. Ah por cierto no podemos usar solo unos o ceros para identificar a un equipo en la red, entonces dado esto solo podríamos usar $2^8 - 2$ o 254 equipos en esta red. Este proceso que hemos visto trabaja en cualquier tipo de dirección o clase de red.

Veamos otro ejemplo que ocurre si tomamos la dirección 172.21.192.7 y aplicamos la máscara de subred por defecto, puesto que el primero octeto se encuentra en 128 y 191 deducimos de que es una red de tipo B por lo tanto la máscara de subred por defecto sería 255.255.0.0, traducida a binarios sería 10101100 00010101 11000000 00000111.

Bueno esa sería la IP traducida
Ahora la máscara
11111111 .11111111. 00000000. 00000000

Poniendo la dirección Ip binaria encima de las máscara de subred

```
10101100 00010101 11000000 00000111   Dirección IP
11111111 11111111 00000000 00000000   máscara de Subred
00000000 00000000 11000000 00000111   ID de equipo
```

Recordemos los procedimientos para sacar el ID de equipo el resultado en decimal corresponde a una ID de red 172.21.0.0.

Haciendo la misma operación para ver el ID de equipo tenemos:
10101100 00010101 11000000 00000111 Dirección IP
11111111. 11111111. 00000000. 00000000 máscara de Subred
00000000. 00000000. 11000000. 00000111 ID de equipo

Bueno ahora vemos los ejemplos para sacar primero los ID que son necesarios conocerlos para después personalizar una máscara de subred, por ejemplo en una red de clase B tenemos 65.534 posible dirección de equipo... porque?

Se debe a que la máscara de subred por defecto de clase B divide la dirección justo por la mitad. La parte de ID de red constaría de 16 bits y la parte de ID de equipo sería de 16 bits también por lo que vemos más equipos :)

Ahora entremos de lleno a la personalización de Subredes hemos visto la mecánica de como funciona esto de y notamos que la parte de la red es fija, bueno cuando usaríamos una máscara de subred personalizada?. Esto sería cuando deseamos dividir nuestra red en diferentes segmentos, por ejemplo podrían haber algunos segmentos remotos que se necesitan conectar o reducir las difusiones o el tráfico local etc.

Bueno teniendo el único ID de Red empezaríamos antes de dividir la máscara de subred hay que tener en cuenta ciertos aspectos.

- 1) Cuantas subredes se quieren hacer o dividir la red
- 2) tener en cuenta cuantos equipos se necesitan por Subred , etc.



Vamos a ejemplificarlo todo:

Supongamos que tenemos una compañía de desarrollo de software con un espacio de direcciones de clase C 192.168.183.0

Se quiere dividir la red en cinco segmentos... Mmmmm... haber, podrían ser desarrollo ventas marketing control de calidad y administración de la compañía, se sabe que en un buen tiempo no abra más de 100 empleados en la empresa y que no se necesitan más de 30 equipos por subred; creo que esto sería razonable: OK para crear las cinco subredes de clase C se toman algunos bits de la parte del ID de equipo de la red i se asignan a la ID de red. Acordémonos que solo podemos mover lo que es del los equipos, es decir, el último octeto, o lo primero que se necesita es saber cuantos bits se necesitan para crear la mascara de subred personalizada, es decir, cuantos se van a tomar y así pasarlos a la parte del ID de red. Igualmente sabemos que las matemáticas par esto son en binarios :) también recordemos que podemos usar solo ceros o unos para los equipos.

Las subredes se asignaran en base dos usando tres bits que le pertenecen al ID de equipo nos daría 6 subredes ya que esto se hace en base dos, osea que nos sobraría una, la dejaríamos para después. Pero es lo mas lógico que se haría al pasar tres bits de la parte de ID de quipo a la parte de ID de red quedan solo cinco ID para los ID de equipos.

Esto da 2^5-2 o 30 ID de equipos validos por Subred, puesto que solo se necesitan 20 ID de equipo por subred parece que tomar tres bits es lo mejor :) las mascara de subred por defecto para una red de clase C es 255.255.255.0

OK, ahora lo que en binarios hemos visto seria,

11111111.11111111.11111111.00000000

Ahora añadiendo los tres bits de la parte de equipo al lado derecho de la parte de red quedaría así en binarios:

11111111.11111111.11111111.11100000
traducido seria 255.255.255.224

Nueva mascara de subred

Por fin esta es la nueva mascara de subred.

Antes de conectar la nueva mascara de subred y crear nuevas subredes tenemos que saber cuales van a ser los nuevos ID de equipos y de Red. Esto puede resultar algo complicado, pero lo veremos despacio.

La dirección original de clase C era 192.168.183.0
Que corresponde a 11000000.10101000. 101110111.00000000

OK, como estamos añadiendo tres bits al id de red la dirección de red se expandiría igual que la mascara de subred . Para obtener los nuevos id de red añadimos todas las posibles permutaciones del valor de tres bits excluyendo 000 y 111 ya sabemos porque, acordémonos que esto esta reservado para las difusiones.



Ahora veamos las permutaciones posibles.
Después de haber tomado tres bits.

TABLA 7.3 seis nuevas subredes de la 192.168.183.0

Numero de Subred	Subred en binario	Subred en decimal
subred1	11000000 10101000 10110111 00100000	192.168.183.32
subred2	11000000 10101000 10110111 01000000	192.168.183.64
subred3	11000000 10101000 10110111 01100000	192.168.183.96
subred4	11000000 10101000 10110111 10000000	192.168.183.128
subred5	11000000 10101000 10110111 10100000	192.168.183.160
subred6	11000000 10101000 10110111 11000000	192.168.183.192

Pero bueno, miremos el ultimo octeto y veamos los últimos tres bits del último octeto, vemos que están las permutaciones; es decir,

001
010
011
010
etc...

Haber mejor veamos la primera subred como quedaría
Subred 1 11000000 10101000 10110111 00100000 192.168.183.32
...así sería.

Estas nuevas subredes pueden parecer un poco confusas porque no termina en 0, solo hay que recordar que existen 5 bits par identificar a los equipos.

¿Como se asignan los ID de equipos?

No hay mas que empezar a rellenar los ID de equipos por el final y por supuesto hay 30 dirección es posibles para cada subred, por eso es que la primera es 192.168.183.32 Las direcciones de equipo simplemente se cuenta hacia adelante desde le numero de la subred hasta uno menos que el de la subred siguiente, así en la subred .32 la primera dirección de equipo es .33 y la ultima .63 ya que el numero de la siguiente subred es .64.

Ahora veamos el como quedarían los ID de equipo solo en la primera subred

Numero de dirección De Equipo	Dirección de equipo en binario	de equipo en decimal
Equipo 1	11000000 10101000 10110111 00100001	192.168.183.33
Equipo 2	11000000 10101000 10110111 00100010	192.168.183.34



Equipo 3	11000000 10101000 10110111 00100011	192.168.183.35
Equipo 4	11000000 10101000 10110111 00100100	192.168.183.36
.....
Equipo 28	11000000 10101000 10110111 00111101	192.168.183.61
Equipo 29	11000000 10101000 10110111 00111110	192.168.183.62
Equipo 30	11000000 10101000 10110111 00111111	192.168.183.63

Vemos las del equipo 1 hasta el 4 y desde el 28 hasta el equipo 30 que sería el último equipo de la primera subred. Por último miremos como quedaría par una subred de clase C y B y los bits que se necesitarían.

Tabla 7.7 Información desubredes para clase C

Numero de Subredes	Equipos por subred	Bits necesarios	Mascara de Subred
2	62	2	255.255.255.192
6	30	3	255.255.255.224
14	14	4	255.255.255.240
30	6	5	255.255.255.248
62	2	6	255.255.255.252
Invalido	invalido	7	255.255.255.254
invalido	invalido	8	255.255.255.255

Tabla 7.8 Información de subredes para clase B

Numero de Subredes	Equipos por subred	Bits necesarios	Mascara de Subred
2	16.382	2	255.255.192.0
6	8.190	3	255.255.224.0
14	4.094	4	255.255.240.0
30	2.046	5	255.255.248.0
62	1.022	6	255.255.252.0
126	510	7	255.255.254.0
254	254	8	255.255.255.0

es todo.

OpTix
optix@hackertm.org
www.hackertm.org

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

END 

He aqui nuestra primera publicacion hemos arrancado una etapa como team encontrando afinidades las cuales se han plasmado en esta entrega, esperemos continuar aportando de una u otra manera a la comunidad Under latina y ser cada dia mejores.

agradecimientos:
a todos nuestros lectores y compañeros de RTM por el fruto de sus esfuerzos.

pronto publicaremos nuestro siguiente numero, por tal razon invitamos a los que deseen que publiquemos sus textos,enviarlos al staff .Esperamos sus comentarios inquietudes,sugerencias etc, seran bien recibidas.

Hubieramos querido sacar algunos articulos mas pero estaran para nuestra proxima publicacion.



staff@hackertm.org
www.hackertm.org