Tuxinfo

/tuxinfo
@tuxinfo

una revista libre, para un mundo libre.

nro.60

Iniciando con **OpenOffice Writer**

Virtualizando en Fedora
Parte III

La indefinición de la pantalla
Opinión

Redes para las masas
Parte V

Auditando una Red WiFi
Ataque Chop-Chop a WEP

ROOT o no ROOT, Esa es la cuestión...
Opinión

¿Y como elegir una distribución?
Nuestros lectores escriben

Martín Escboyez | LibreGraphics/Crafts

#60
julio 2013

Dirección
Ariel M. Corgatelli

Corrección
Luis Luque
Oscar Reckziegel

Diseño de tapa
Martín Eschoyez

Diseño
Ariel M. Corgatelli

www
http://www.tuxinfo.com.ar

facebook
http://www.facebook.com/tuxinfo

email
info@tuxinfo.com.ar

twitter
@tuxinfo

EDITORIAL

Como todos los meses les entregamos un nuevo número de nuestra querida revista Tuxinfo. En esta oportunidad tenemos una colección muy interesante de artículos dedicados, desde tutoriales hasta artículos avanzados.

Como tema destacado para esta editorial podemos hablar del programa secreto que está llevando adelante el gobierno de Estados Unidos y la NSA llamado Prims.
El mismo tiene como finalidad poder controlar a todos los gobiernos del mundo y su actividad cibernética ya sea oficial como no oficial.
La divulgación de la misma fue un escándalo a lo largo de todo el mundo, ya que si bien se tenía en cuenta que ese gobierno tenía un plan de este tipo, jamás se pensaba hasta dónde el mismo podía tener acción.

Dicha filtración fue llevada adelante por un ex agente de la CIA, quien además de exiliarse en Rusia, fue quien divulgó los detalles más escalofriantes de intrusión como así también dio a conocer la lista de países afectados. Dentro de ellos por supuesto esta nuestro país y todos los afectados al MERCOSUR.

Y esto no no es todo, ya que además se dio a conocer información en donde la misma empresa Microsoft habría brindado información y acceso al gobierno de USA sobre sus servicios más importantes, como ser Outlook, Skype, incluso Office.

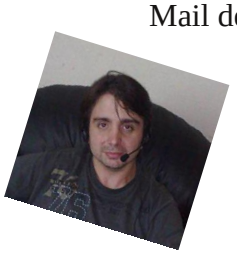
Por supuesto las declaraciones desde Redmond fueron que no habían brindado información a la NSA. Y además reafirmaron que Microsoft no brinda información a terceros o gobiernos ya que tienen como preponderancia la privacidad de sus usuarios.
Muchas de las empresas más importantes salieron al cruce con declaraciones en donde desconocían que la NSA y el gobierno de USA los estaba espiando.
Por nuestra parte nos guardamos las opiniones al respecto sobre las acciones de la NSA y de Microsoft como empresa colaboradora.

Otro tema importante fue el lanzamiento de Ubuntu EDGE, el cual intenta por medio del crowdfunding, llevar adelante un proyecto muy ambicioso sobre un smartphone corriendo Ubuntu Linux, y que además pueda ser conectado a una TV para tal fin. (para mayor información del mismo pueden acceder al siguiente url http://www.indiegogo.com/projects/ubuntu-edge).
Y como para cerrar les comparto la lista de los artículos más destacados de este número.

Iniciando con OpenOffice Writter; Virtualizando en Fedora - Parte III; La indefinición de la pantalla - Opinión; Redes para las masas – Parte V; Auditando una Red WiFi - Ataque Chop-Chop a WEP; ROOT o no ROOT - Esa es la cuestión...

Y como todos los meses, repetimos la misma convocatoria en donde podamos tener más sugerencias de ustedes y así adaptar los contenidos de las notas a vuestras necesidades y preferencias, las mismas las podrán realizar a nuestros medios de contacto.

Fan page: https://www.facebook.com/tuxinfo
User Twitter: @tuxinfo
Mail de contacto: info@tuxinfo.com.ar



Ariel M. Corgatelli
@arielmcorg

índice

Nexus 7 2013 vs Nexus 2012

Lanzamiento del mes.

Virtualizando en Fedora Parte III

POR **RINO RONDAN**

Nos encontramos nuevamente en otra entrega de KVM, vamos a comenzar directamente con la línea de comandos, así podemos mostrarles cómo crear una vm utilizando lvm, para luego instalarle un gentoo base/Fedora ya instalado.

Creación del sistema de archivos:

En esta parte vamos a crear un sistema de archivos dentro de nuestro lvm para lo cual suponemos que tenemos creados un vg que se llame kvm.

Creamos Volumen

#lvcreate -L8G -n gentoo-base kvm

Creamos FS

#mkfs.ext4 /dev/mapper/kvm-gentoo--base

Importar S.O.:

En este paso lo que podríamos hacer es un tar de todo nuestro sistema operativo y luego ponerlo en el fs que tenemos.

Montamos el FS:

#mount /dev/mapper/kvm-gentoo—base /mnt

Copiamos nuestro sistema:

#cd /mnt

#tar xvjf backup-fedora19.tar.tbz

o

#tar xjf stage3-*-*.tar.bz2

En estos pasos ya tenemos el sistema en nuestro fs, basta con entrar en los directorios correspondientes y setear ip, hostname, etc...

Crear XML máquina virtual:

Vamos a ver cómo exportar un xml ya creado y luego editar lo que haga falta, o también crear desde un xml el comando de creación. Los xml se encuentran en /etc/libvirt/qemu/*.xml

Obtener comando a partir de xml.

#domxml-to-native qemu-argv /etc/libvirt/qemu/vm.xml

Esa línea les va a dar el comando que tienen que ejecutar en conjunto a las variables a incluir en la ejecución.

Crear un xml a partir de una máquina

#virsh dumpxml nombre_vm > /tmp/nueva_vm.xml

En el paso anterior le pasamos una vm existente y obtuvimos el mismo xml.

Definir la vm en base a un xml.

Teniendo ya el xml anterior, siendo el mismo un producto de otro xml, hay que eliminarle ciertos tags.

-> Eliminar toda la línea que contengan los siguientes tags: uuid, mac address, address.

Tener en cuenta cambiar el nombre de la vm y otros parámetros como memoria, disco, etc.

Parámetros a cambiar en el xml

Domain, es el tag principal y tiene el id de la vm

Name, el nombre de nuestra máquina

uuid, mejor que lo genere sólo al levantar la vm

memory, la memoria que va a contener

currentMemory, por ahora usemos sólo la misma cantidad que memory

vcpu, procesadores

os, acá van cosas importantes, se desprenden otros tags (type, kernel, cmdline, boot)

Ejemplo:

```
<os>
<type arch='x86_64' machine='pc-0.14'>hvm</type>
<kernel>/var/kvm/bzImage-3.8.8</kernel>
<cmdline> console=ttyS0 ignore_lost_ticks root=/dev/vda </cmdline>
<boot dev='hd'/>
</os>
```

Estos parámetros nos van a servir para que arranque de otro kernel propio y no usamos ningún bootloader.

Emulator, por omisión usamos el que viene que es qemu-kvm

disk, acá vamos a tener varios tags, (driver, source, alias, address)

```
<disk type='block' device='disk'>
<driver name='qemu' type='raw'/>
<source dev='/dev/fedora/gentoo-base'/>
<target dev='vda' bus='virtio'/>
<alias name='virtio-disk0'/>
<address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</disk>
```

Podríamos tener otro tipo de disco también:

```
<disk type='file' device='disk'>
 <driver name='qemu' type='qcow2' cache='writeback'/>
 <source file='/home/crond1/ibm/discos/pc01-a.qcow2'/>
 <target dev='sdb' bus='virtio'/>
</disk>
```

Interfaces, para definir nuestra red, tenemos más tags adentro (mac address, source network, target dev, model type, alias name, address)

```
<interface type='network'>
<mac address='52:54:00:a9:8f:c2'/>
<source network='virbr0'/>
<target dev='vnet0'/>
<model type='rtl8139'/>
<alias name='net0'/>
<address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```

Podríamos utilizar macvtap en lugar de bridge y nos olvidamos de tantas vueltas:

```
<interface type='direct'>
<mac address='52:54:00:0a:2f:b2'/>
<source dev='enp16s0f0' mode='bridge'/>
<target dev='macvtap1'/>
<model type='virtio'/>
<alias name='net0'/>
<address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```

Luego tenemos otros tags como usb, serial, console, y muchos más, depende de las necesidades.

Features, tenemos varios tags (acpi, apic, pae, hap), siempre pueden ser muchos más los que tengamos, esto es un ejemplo.

```
<features>
<acpi/>
<apic/>
<pae/>
<hap/>
</features>
```

Algunos otros importantes:

```
<clock offset='utc'>
<timer name='rtc' tickpolicy='catchup' track='guest'>
```

```
                    <catchup  threshold='123'
slew='120' limit='10000'/>
                  </timer>
                 </clock>
          <on_poweroff>destroy</on_poweroff>
          <on_reboot>restart</on_reboot>
          <on_crash>destroy</on_crash>
         <watchdog model='i6300esb' action='reset'>

                 <alias name='watchdog0'/>
                 <address type='pci'
domain='0x0000' bus='0x00' slot='0x09'
function='0x0'/>
        </watchdog>
```

En estos casos el reloj y determinadas acciones que puede tomar en ciertas instancias, no olvidarse de watchdog que va a estar ahí presente dando batalla :)

```
          <rng model='virtio'>
             <backend
model='random'>/dev/random</backend>
          </rng>
```

Para poder generar la entropía propia.

Una vez que tenemos todo ya podemos importar nuestro xml.

Importar xml

Ahora sí vamos a crear nuestra vm.

#virsh define /tmp/nueva_vm.xml

Ahora sí nuestra vm está creada, basta listarla (virsh list –all)

Tener en cuenta que la vm va a arrancar utilizando el kernel que le pusimos en los tags “os” así que van a tener que bajarse el kernel y compilarlo o utilizar alguno que ya sabemos que anda bien, también hay que tener en cuenta todos los seteos correspondientes.

Encender la vm.

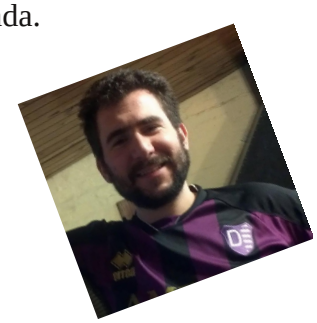
Ahora para encender la vm, no se olviden de desmontar el /mnt.

#virsh start nueva_vm –console

Acá pueden pasar muchas cosas, entre ellas muchos kernels panic, eso sí uno por vez :)

Manos a la obra e ir mirando qué hace falta para que arranque su vm.

En la próxima entrega vamos a ver más comandos para interactuar con la vm ya creada.



Rino Rondan

Fan de Villa Dalmine
LPIC-2 – RHCE – RHCVA

Bibliografía: http://libvirt.org/formatdomain.html

ROOT o no ROOT
Esa es la cuestión...

POR **JUAN MANUEL DANSA**

Millones de dispositivos Android se encuentran distribuidos por todo el planeta, estos a su vez poseen un kernel GNU/Linux, por ende una de las preguntas que se nos vendría a la mente enseguida podría ser: ¿tenemos el acceso ROOT (/) en nuestro dispositivo?, la respuesta tendría que ser SI, pero es un rotundo NO. Esto trae infinidad de conjeturas y porqués, seguido de diferentes políticas de fabricación de las compañías, sumado a las garantías de productos, que lo único que provoca es inseguridad, miedo y desconcierto al usuario final. La pregunta que un "linuxero" se tendría que hacer es, ¿qué diferencia hay con un equipo portátil o PC con Windows u otro Sistema privativo preinstalado?, yo creo que ninguna, por ende si para estos dispositivos no tenemos ningún tipo de problema de instalar una distribución GNU/Linux, por qué no hacerlo en un dispositivo con Android y así poseer, si se lo desea, una ROM (Firmware) diferente, junto con un acceso total al sistema como es lo normal; obviamente no es necesario cambiar el sistema para tener acceso total a este, pero esto del ROOT ¿qué ventajas nos trae?, para los usuarios de sistemas GNU/Linux – BSD es muy fácil responderse, para el resto es más difícil ya que lo más probable es que vengan y/o utilicen sistemas privativos como lo son Windows y MAC-OSX.

Lo que vemos algunos como ventajas del "rooteo" de un dispositivo, es tratar de acercase lo más posible a las ya conocidas libertades del Software Libre, en especial a la siguiente: *"La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera"*. Para llegar a cumplir esta libertad en un dispositivo de venta masiva obligatoriamente se necesita tener el acceso no sólo al código sino al kernel, librerías y procesos como mínimo; se podría decir que Google libera el código de Android, pero este no es el mismo en dispositivo de venta, ni siquiera en la línea NEXUS ya que integra programas privativos propios de la compañía. Y la nueva pregunta que se me ocurre es, ¿cómo borro aplicaciones que pueden vulnerar la seguridad del sistema o simplemente por su estado de privativos, y que dudamos de ellas y son instaladas por la prestadora o las compañías?...la respuesta es sólo con ROOT. Existen distribuciones Android como CyanogenMod (http://www.cyanogenmod.org/), que no llegan a tener un 100% de código libre, por un problema de drivers privativos del hardware, pero que se acerca mucho a este porcentaje, y no sólo nos da el acceso total al sistema sino que también trae mejoras sustanciales que hacen que nuestro dispositivo funcione en casi todos los casos mucho mejor que con su firmware de fábrica y o modificado por una prestadora.

Pero si descartamos el cambio de ROM, aceptando la que viene instalada, ¿qué otras ventajas tendría si tuviera acceso ROOT?, si lo miramos desde la óptica de la utilización habitual de un sistema operativo de ordenador, por ejemplo, ¡poder realizar un simple BACKUP de configuraciones, datos y aplicaciones!, parece una función obvia pero no lo es, sólo conozco una aplicación que incluye el fabricante ASUS, la cual si realiza esta acción desde la ROM de fábrica y sin ROOT pudiendo guardar este backup hasta en la memoria microSD, pero no es lo normal. Para estos

menesteres existen dos excelentes aplicaciones, como son Titanium Backup (https://play.google.com/store/apps/details?id=com.keramidas.TitaniumBackup&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5rZXJhbWlkYXMuVGl0YW5pdW1CYWNrdXAiXQ..) y Helium - App Sync and Backup (https://play.google.com/store/apps/details?id=com.koushikdutta.backup&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5rb3VzaGlrZHV0dGEuYmFja3VwIl0.), esta última posee como diferencia que puede realizar el deseado Backup sin permisos de superusuario pero solamente con un equipo con sistema Windows, MAC Osx o Linux (http://www.clockworkmod.com/carbon) y no en una memoria y o en el mismo dispositivo.

Otra de las ventajas del acceso completo a nuestro sistema es el de devolver a los móviles algunas funciones deshabilitadas por las prestadoras, como ser en algunos casos la del tethering. También en muchos dispositivos con poca capacidad interna, se nos abre la posibilidad de mover aplicaciones a la memoria microSD, estén preparadas estas para esa función o no, no importando la versión de Android. Podemos optimizar el consumo de batería con aplicaciones muy efectivas pero que necesitan el acceso ROOT, como ser Battery Saver (https://play.google.com/store/apps/details?id=com.antutu.powersaver&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5hbnR1dHUucG93ZXJzYXZlciJd), de los desarrolladores del famoso AnTuTu Benchmark.
Como nos encontramos con un kernel GNU/Linux, como es lo normal lo podemos cambiar por alguno con más funciones como la de poder overclockear el equipo ganando más performance y más consumo de batería :-); y no nos olvidemos de las molestas publicidades en aplicaciones como en la navegación, que con programas como el famoso Adblock (http://adblockplus.org/en/android-install), expulsado no hace mucho del Google Play por obvias razones, que siendo ROOT podremos bloquear la publicidad en todo el terminal y no sólo en el navegador. Pero esto no sería todo, por ejemplo encontramos dando vuelta muchos terminales con menos de 1GB RAM, los cuales no suelen ir muy fluidos y una de las opciones para tratar de mantener estabilidad es crear una especie de memoria de intercambio SWAP en una tarjeta MicroSD o en el mismo dispositivo, un ejemplo podría ser ROEHSOFT RAMEXPANDER (SWAP) (https://play.google.com/store/apps/details?id=com.swapit.expander.de&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5zd2FwaXQuZXhwYW5kZXIuZGUiXQ.), pero para estos menesteres se requiere el tan citado acceso ROOT.

Que se “cocina” hoy...

Como ya hemos dicho existen varias ROMs (Firmware) custom o como se le dice en la jerga “cocinadas” y estas aparte de darnos el acceso ROOT, en dispositivos olvidados les alarga la vida útil en el tiempo, y un gran ejemplo de esto es Galaxy S (I9000) de la compañía Samsung, que a pesar de que esta quitara el soporte en la versión 2.3.6 de Android (*Gingerbread*), hoy en día se pueden encontrar ROMs de la versión 4.2.x (*Jelly Bean*) adaptadas a este terminal.

Esta necesidad de cambiar los firmware antiguos por más modernos se ha acrecentado en los últimos tiempos, ya sea por culpa de las prestadoras como del fabricante, donde muchos de estos sólo al año de ponerlo a la venta quitan su soporte dejando al usuario con un trago más que amargo. Pero hoy en día...¿Encontramos alternativas Open Source?, y la respuesta es “SI”, son las ROM's **AOSP** (Android Open Source Project); un buen ejemplo son los Firmware **AOKP** (Android Open Kang Project – http://aokp.co/), desarrollo muy interesante, que da a dispositivos de potencia gran variedad de funciones.

La cantidad de ROM's desarrolladas es grande, pero la cantidad de equipos dando vuelta es mayor, por ende no todos los equipos gozan de estas increíbles posibilidades, cayendo en un rápido olvido y abandono ya sea de forma oficial como en el desarrollo paralelo; ante esto siempre al recomendar o adquirir un dispositivo móvil con Android, investigo si se ha desarrollado Firmware paralelo, cuántos de estos, quién los desarrolla, si existe la posibilidad de desbloquear el Bootloader y por supuesto si existe la posibilidad de adquirir el tan deseado acceso ROOT!.
Para ir terminando no olvidarse que obtener el acceso ROOT no significa cambiar la ROM del terminal...¡GOD SAVE THE ROOT!

Juan Manuel Dansa (Amonal)
amonal88@gmail.com
twitter: @Amonal_

Redes para las masas – Parte V

POR **HERNAN SALTIEL**

Ya contamos con los conocimientos teóricos para comprender el funcionamiento de las redes de datos que encontremos delante nuestro. En las entregas anteriores pudimos interiorizarnos en cuestiones de base, fundamentales y no siempre conocidas por los administradores de sistemas que configuran entre otras cosas las redes que usamos para hacer funcionar nuestras aplicaciones.

En esta entrega pondremos las manos en la grasa, y configuraremos los elementos necesarios para probar nuestros conocimientos. A no asustarnos, comienza aquí una nueva etapa, excitante y movida, que si seguimos al pie de la letra nos aportará fuertes conceptos, ahora tanto teóricos como prácticos. Y como debe ser, no perderemos oportunidad para abordar nuevos conocimientos ahora ya más avanzados. Súbanse al tren, no paramos.

Preparando la escena

Las prácticas que veremos estarán basadas en determinados parámetros de nuestra red. Sabemos que, como pasa con las familias, no hay dos redes iguales. Pero tomaremos como punto de partida algunos parámetros que podrán ajustarse a cada caso, según corresponda. Los parámetros iniciales serán los siguientes:

- Conexión a Internet: Consideramos tener una conexión a Internet medianamente decente, nada ambiciosa, pero sí que esté funcionando. La usaremos para bajar imágenes de sistemas operativos, si aún no los tenemos. En mi caso, me conecto a Internet a través de un pequeño router WiFi conectado a un cablemodem configurado con la dirección IP interna 10.100.100.1/24. Eso quiere decir que siempre que configure un router predeterminado y esté utilizando una conexión que carezca de la configuración “NAT” (paciencia, ya revisaremos este concepto) lo haré con esa dirección. Este punto se podrá ajustar a los parámetros de la red que tengamos. Si no tuviéramos un router, y sí una conexión directa a Internet, podremos ver los parámetros de ruteo ejecutando el comando “netstat -nr” desde una terminal GNU/Linux, y revisando la columna “Gateway” en la línea que tenga el valor “UG” en la columna “Flags”. Para las prácticas iniciales, utilizaremos facilidades que nos provee el mismo virtualizador, y que no desmerecen ninguno de los puntos que estudiaremos. Todo lo contrario, cuanto más complicado, mejor. Más adelante, jugaremos con estos parámetros.

- Direcciones IP disponibles: Si contamos en nuestra red con un servidor DHCP (más adelante veremos qué es eso) utilizaremos las direcciones IP que él nos entregue. En mi caso tengo uno que entrega direcciones IP desde la 10.100.100.100 a la 10.100.100.150, siempre con máscara 24. Si no usamos esa tecnología, no hay

problema, veremos opciones para estos casos. E inclusive, para nuestra práctica jugaremos con el servidor DHCP que tiene el software hipervisor.

- Espacio en disco: Si bien no es concluyente, ya que no lo utilizaremos en forma completa, el contar con unos 30~60 GB de espacio en disco será conveniente para la generación de las máquinas virtuales que utilizaremos en nuestras prácticas. Para la totalidad de este artículo, 30 GB son más que suficientes, y hasta me animaría a decir que demasiado.

- Memoria RAM: Convendrá tener por lo menos 1,5 GB de memoria RAM disponible. Si tenemos muchas aplicaciones abiertas mientras ejecutamos las prácticas, tendremos que cerrar algunas hasta que podamos contar con máquinas virtuales funcionales. No es tan terrible, en mi caso estoy ejecutando todo en una pequeña notebook.

- Sistema operativo: Cualquier sistema operativo que nos permita la ejecución de un programa de virtualización como lo es VirtualBox nos servirá. Eso incluye GNU/Linux en casi todos sus sabores, OpenIndiana, *BSD, y "los que ya no son primeros en ventas", también.

- Software de virtualización: Como software de virtualización utilizaremos VirtualBox. Dependiendo de la versión de GNU/Linux de que dispongamos podremos utilizar el set de comandos "apt-get", "yum", "pacman", o bajarlo desde el sitio http://www.virtualbox.org para nuestro sistema operativo y arquitectura de procesador. Recordemos también bajar la "VM VirtualBox Extension Pack" para usar funciones avanzadas de este software.

- Imágenes de sistemas operativos: En mi caso particular deposito siempre los archivos "*.iso" correspondiente a los CD/DVD de sistemas operativos en un único subdirectorio, como para tener todo organizado. Las que utilizaremos serán las siguientes:
  - Debian 7 x86: Para simplificar, bajaremos la imagen de Debian GNU/Linux desde http://cdimage.debian.org/debian-cd/7.0.0/i386/iso-cd/debian-7.0.0-i386-netinst.iso . Sí, ejecutaremos la instalación por red.
  - CentOS 6.x x86: Lo bajaremos desde el mismo sitio http://www.centos.org. En mi caso particular, bajo la imagen desde uno de los sitios de descarga listados, específicamente de http://mirrors.usc.edu/pub/linux/distributions/centos/6.4/isos/i386/CentOS-6.4-i386-bin-DVD1.iso . Si queremos aprovechar el tiempo, bajemos primero la imagen de Debian, y cuando estemos haciendo alguna parte de las prácticas, dejemos bajando la de CentOS.

El motivo por el cual estamos bajando dos versiones diferentes de GNU/Linux es el poder experimentar por una lado con una basada en ".deb", como lo es Debian, Ubuntu, Linux Mint, etc., y por el otro con una basada en ".rpm", como lo son Red Hat, CentOS, Scientific Linux, Fedora, etc.

No dejaré de lado la configuración de otras distribuciones, como lo son las *BSD, como FreeBSD, NetBSD o PC-BSD; y las basadas en IllumOS, como OpenIndiana, por ejemplo. Ellas quedarán, por no ser las más populares (no saben lo que se pierden) para el final de esta sección.

Debian Virtual

Lo primero que haremos es implementar una máquina virtual basada en Debian GNU/Linux 7, y para ello haremos uso de la primer imagen que hemos bajado. Para realizar esta instalación, abriremos el programa VirtualBox y en la ventana principal seleccionaremos el botón superior “New” o “Nueva”.

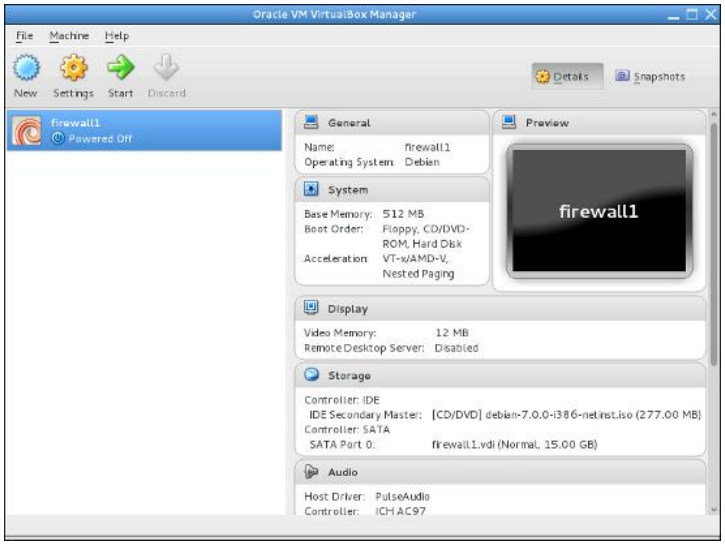
En la primer ventana, donde se nos pide “Name and Operating System”, colocaremos, como nombre (“Name”), “firewall1”, como tipo de sistema operativo (“Type”), “Linux”, y como versión (“Version”), “Debian”.

En la siguiente pantalla, donde se define el tamaño de la memoria, asignaremos no más de 512 MB. En la siguiente, donde se debe configurar el disco virtual a generar, seleccionaremos “Crear un disco virtual ahora” (“Create a virtual hard drive now”), y presionaremos el botón “Crear” (“Create”). Debemos seleccionar el tipo de archivo a utilizar para el disco virtual, que en nuestro caso será “VDI”.

El almacenamiento de este disco será “Dinámicamente alojado” (“Dynamically allocated”), y su tamaño será no menor a los 15 GB. Con eso prácticamente quedará creado el entorno que necesitamos para hacer nuestra práctica. Sólo nos queda cambiar algo de la configuración.

Para ello, presionaremos el botón derecho sobre el nombre de la máquina virtual, y en el menú que se desplegará seleccionamos “Configuración” (“Settings”). El primer cambio que introduciremos será dejar el tipo de red como “NAT” (veremos en breve qué es eso). Luego, seleccionaremos en el menú de la sección izquierda de la pantalla la parte de “Almacenamiento” (“Storage”), y en el disco óptico simulado, seleccionaremos la imagen que bajamos, “debian-7.0.0-i386-netinst.iso”, o la que hayamos bajado.

Ahora, con los cambios introducidos, podremos presionar el botón “OK”. La máquina virtual quedará configurada como lo vemos en la figura, y estaremos listos para encenderla presionando el botón “Iniciar” (“Start”).



Como el orden de inicio lo determina, comenzaremos a ver el booteo desde el CD-ROM, que en nuestro caso, es la imagen del CD de sistema operativo Debian 7.

En la primer pantalla, seleccionaremos como idioma de instalación “English”. En “Select your location”, elegiremos “other”, luego “South America”, luego “Argentina”.

En “Select your locales”, elegiremos “United States”. En “Configure the Keyboard”, a menos que tengamos un teclado algo distinto, “Latin American” será adecuado. Así llegaremos a la sección “Configure the network”.

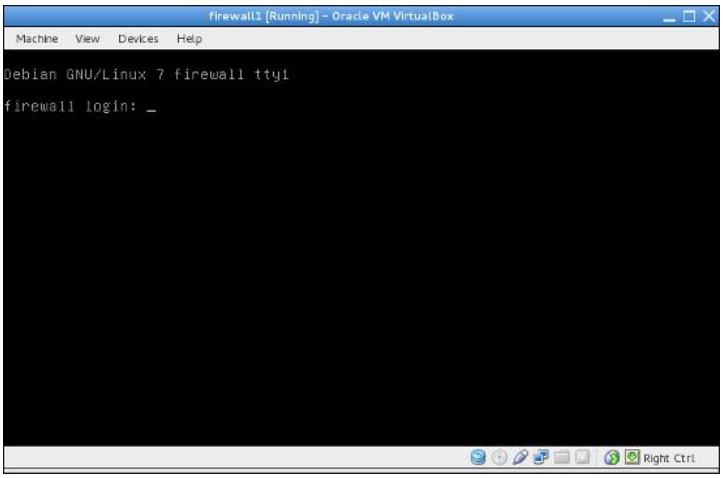
Como hostname escribiremos “firewall”. Domain name lo dejaremos en blanco. Colocaremos la clave de root dos veces. Crearemos un nuevo usuario colocando su nombre, username, e ingresando la contraseña dos veces.

En la sección de particionamiento de discos seleccionaremos la opción “Manual”, luego elegiremos el disco virtual que hemos configurado antes, y como “Partitioning scheme”, para simplificar este paso, elegiremos “All files in one partition (recommended for new users”. Seleccionaremos “Finish partitioning and write changes to disk”, y lo confirmaremos.

Luego de unos segundos se nos solicitará que configuremos el gestor de paquetes (“Package Manager”), para lo cual, en mi caso selecciono “Argentina”, y luego “ftp.ccc.uba.ar”, pero que podrá ser cualquier otro valor según la zona donde estén viviendo.

Si no uso un proxy para salir a internet, dejaré este campo en blanco. Selecciono “No” a participar del concurso de popularidad de paquetes (“popularity contest”). En la sección “Software selection” sólo dejo seleccionado “SSH Server” y “Standard System Utilities”.

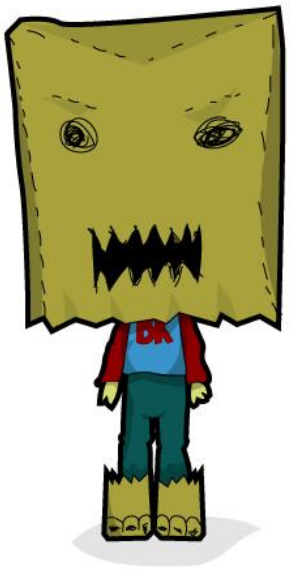
En la sección de instalación de GRUB, digo “Yes” a instalarlo en el MBR (“Master Boot Record”). Cuando termine la instalación podré ingresar a mi nuevo sistema operativo, ya que me encontraré ante una ventana típica de login en modo caracter.



NAT King Cole y el enmascarado misterioso

Bueno, venimos hablando de NAT hace un par de párrafos, llegó el momento de definir qué es eso.

NAT es una sigla que significa “Network Address Translation”, o “Traducción de Direcciones de Red”. Éste es un sistema de cambio de direcciones IP utilizado para interconectar dos o más redes que tienen rangos de direcciones incompatibles. Imaginemos una empresa donde haya un telefonista que dedica sus días a recibir llamadas desde dentro de esa entidad, escuchar a alguien decir “comuníqueme con el teléfono xxx”. Esa persona lo que hará será utilizar alguna de sus líneas disponibles, y comunicará al empleado INTERNO, a través de un número de teléfono que es público, a otro número de teléfono que también es público. Si del otro lado de la comunicación tuviéramos también un conmutador telefónico, estaríamos entrando a esa red de teléfonos a través de un número público, y luego nuestra llamada sería enrutada hacia el número interno que corresponda.



Entonces, las redes de datos tendrán la forma de hacer algo similar. Como ya sabemos, las direcciones IP rápidamente se están agotando, y si tuviérmos que colocar una dirección IP pública (es decir, alguna que pueda ser directamente contactada en Internet) a cada dispositivo que en este momento utilizamos, estaríamos en serios problemas. En una empresa de cinco mil empleados, utilizaríamos esa cantidad de direcciones IP públicas. Por lo tanto, el proceso de NAT, en este caso, permitirá tener un ruteador en el borde de la red (esto es, el punto donde se interconecta la red privada y la pública, internet), que haría una traducción de las direcciones internas para que todas contacten sus destinos a través de una única, o únicas direcciones IP públicas.

Supongamos que tenemos una red que internamente tiene el rango de direcciones IP 10.100.100.0/24 (sí, máscara 255.255.255.0),

es decir, que soporta unos doscientos cincuenta y tres direcciones internas. En lugar de contratar con nuestro proveedor ese número de direcciones IP públicas, contratamos una sola, supongamos la 200.12.13.14, y colocamos en el punto de interconexión el ya tan famoso ruteador.

Cada vez que algún usuario interno quiera contactar un sitio de internet, podrá hacerlo, y ese sitio leerá que la dirección que lo contactó fue la 200.12.13.14, no la dirección interna. Y cuando ese servidor envíe información de vuelta hacia la red que lo contactó, esa información le llegará al puesto de trabajo interno que originalmente la solicitó.

La primer pregunta que nos haríamos es "¿cómo sabe el servidor a qué dirección interna enviar la información, si todos los pedidos salieron por la misma dirección IP pública, que es la 200.12.13.14?". El servidor enviará un paquete de red de respuesta, que de seguro tendrá "algo" para saber que debe llegar a ese puesto de trabajo. Caso contrario, sería muy cómico que yo pida una página de internet, y que el resultado le llegue a cualquier persona, sea mi jefe, o mi empleado.

¿Qué es ese "algo"? Es un mecanismo. Por un lado, TCP/IP en tiempo real edita las cabeceras de los paquetes de red, y agrega información destinada a esta traducción. Si la conexión fuera del tipo "uno a uno" ("one to one"), estaríamos ante lo que se denomina "basic NAT". Pero cuando tenemos una red entera escondida detrás de una única dirección IP pública, aparte de llamarlo "uno a muchos" ("one to many") lo que se hace es armar en el ruteador una tabla de puertos de salida para que, cuando el servidor contactado devuelve su respuesta, se verifique por qué puerto está llegando, y entonces dicha respuesta sea enviada al sistema que la solicitó. Sencillo, ¿no? A veces, este tipo de NAT es llamado NAPT, por la sigla de "Network Address and Port Translation", ya que se están traduciendo tanto las direcciones IP como los puertos.

Ahora bien, supongamos que en nuestra red interna tengo un servidor web en una máquina que tiene la dirección IP 10.100.100.10, luego un servidor FTP en otra máquina que tiene la dirección IP 10.100.100.20, y un servidor SSH en una tercera máquina, con la dirección IP 10.100.100.30. ¿Cómo puedo hacer para que desde afuera de mi red un cliente externo pueda, por un lado, ver páginas alojadas en mi sitio interno, por el otro transferir archivos utilizando el protocolo FTP, y por el otro conectarse con un servidor mediante SSH?

Para eso tenemos una de las variantes del NAT, llamado DNAT por su significado, "Destination Network Address Translation", o "Traducción de direcciones de red de destino". Esto también se encontrará en el ruteador que está conectado a la dirección IP pública, y tendrá una tabla con reglas de este estilo: "si el paquete de red tiene como destino el puerto 80, debe ir al servidor web, que está en la dirección IP 10.100.100.10; si el destino es el puerto 21 para comenzar una transferencia FTP, se debe redirigir a la dirección 10.100.100.20; finalmente, si el destino es el puerto 22, se redirigirá a la dirección IP interna 10.100.100.30".

De esta forma, tendremos inclusive la oportunidad de tener en nuestra red interna más de un servidor web. Supongamos que el sistema que muestra fotos está en la dirección IP 10.100.100.50; luego el que muestra la interfaz gráfica de un webmail está en la dirección IP 10.100.100.60; y finalmente el portal web está en la dirección 10.100.100.70. Podré definir, en el ruteador, que todo lo que ingrese por el puerto 8080 sea redirigido al puerto 80 del primer servidor, luego lo que ingrese con un destino de puerto 8090 sea redirigido al puerto 80 del segundo servidor, y lo que ingrese con puerto 8100 como destino sea redirigido al tercer servidor. Sencillo, pero se complica un poco, ¿no? No tanto, sería peor si el DNAT no existiera. DNAT en algunos casos es denominado también "port forwarding", dado que se toma un puerto determinado, y se reenvía a otro puerto, de otra dirección IP.

Si en cambio, necesito que una serie de direcciones internas de una red sean "NATeadas" para cambiar su dirección IP por la

pública, estaré ejecutando “SNAT”, o “Source Network Address Translation”, que es similar al caso que vimos al principio.

Ahora bien, ya sabiendo todo esto, ¿qué es “IP Masquerade”, o “Enmascaramiento de IP”? Es casi lo mismo que NAT, tal como lo vimos en un primer momento, pero que puede funcionar inclusive con interfaces de red que no tengan una dirección IP estática asignada. Eso quiere decir que funciona inclusive con las conexiones de internet que tenemos en nuestra casa, así sea un cablemodem, una conexión ADSL, por módem 3G, etc.

Ah, ¿no saben qué es una dirección IP estática, y una dinámica? No hay problema, ya vamos, ya vamos.

Dinámicas y estáticas

Habrán notado que en la instalación de nuestra máquina virtual como único parámetro de red hemos configurado su hostname. No le hemos agregado, aún, ninguna dirección IP, máscara de subred, ruteador predeterminado, ni nada que se le parezca. ¿Qué valores obtendrá cuando este sistema virtualizado inicie su procesamiento?

Ejecutemos un “ifconfig eth0” y veamos su salida:

```
root@firewall:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr
08:00:27:31:01:50
          inet addr:10.0.2.15
Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr:
fe80::a00:27ff:fe31:150/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST
MTU:1500  Metric:1
          RX packets:125 errors:0
dropped:0 overruns:0 frame:0
          TX packets:123 errors:0
dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15005 (14.6 KiB)  TX
bytes:12720 (12.4 KiB)
```

Esto quiere decir que nuestra máquina virtual obtuvo de “algún lado” la dirección IP 10.0.2.15. Esa dirección es claramente interna, ya que al menos en mi caso no se corresponde con ninguna de las redes que tengo disponible, y no la hemos configurado nosotros, entonces ¿quién la configuró? Sencillamente la configuró nuestro hipervisor, VirtualBox, a través de un servidor de direcciones dinámicas que posee embebido.

Ahora bien, como teníamos configurada nuestra tarjeta de red con el protocolo NAT, de seguro vamos a poder llegar a cualquier dirección IP pública a través de ésta, privada, y haremos la prueba con la 8.8.8.8, perteneciente a Google:

```
root@firewall:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of
data.
64 bytes from 8.8.8.8: icmp_req=1 ttl=63
time=43.9 ms
64 bytes from 8.8.8.8: icmp_req=2 ttl=63
time=42.2 ms
64 bytes from 8.8.8.8: icmp_req=3 ttl=63
time=51.2 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0%
packet loss, time 2006ms
rtt min/avg/max/mdev =
42.260/45.805/51.240/3.902 ms
```

Qué bueno, tenemos salida a internet, y no hemos sacrificado ninguna dirección IP pública más. Ésta es una muestra de cómo NAT puede ayudarnos a solucionar parte de la problemática de consumo de direcciones IP públicas.

Pero volviendo al tema de este apartado, decimos que VirtualBox nos ha entregado una dirección IP “dinámica”, dado que la misma cambiará cada vez que la máquina virtual se encienda. Si fuera estática, la misma se mantendría constante y estaría asignada en forma manual a nuestra máquina.

Pero ¿es el único parámetro que nuestro servidor de direcciones IP entregó? Claro que no.

Ejecutemos este comando para entender cuál es el juego de servidores DNS que hemos recibido:

```
root@firewall:~# more /etc/resolv.conf
domain fibertel.com.ar
search fibertel.com.ar
nameserver 200.42.4.207
nameserver 200.49.130.44
```

Estas direcciones IP son las correspondientes a los servidores que transformarán nombres del tipo “DNS”, como lo es http://www.google.com a direcciones IP, ya que en Internet, si bien uno escribe en su navegador esa dirección, denominada URL (por “Uniform Resource Locator”, o “Localizador de recursos uniforme”), cuando se accede a un sitio web, se accede a una dirección IP provista por esos servidores de DNS, o “Domain Name Service”, “Servicios de nombres de dominios”.

Por lo tanto, un servidor de direcciones dinámicas podrá entregarnos una serie de parámetros para que nuestra máquina posea una dirección IP, parámetros de servidores de nombres, ruteador predeterminado, y un sinfín de otros. El protocolo que permite este tipo de configuración dinámica se denomina “DHCP”, o “Dynamic Host Configuration Protocol”, “Protoclo de configuración de hosts dinámico”.

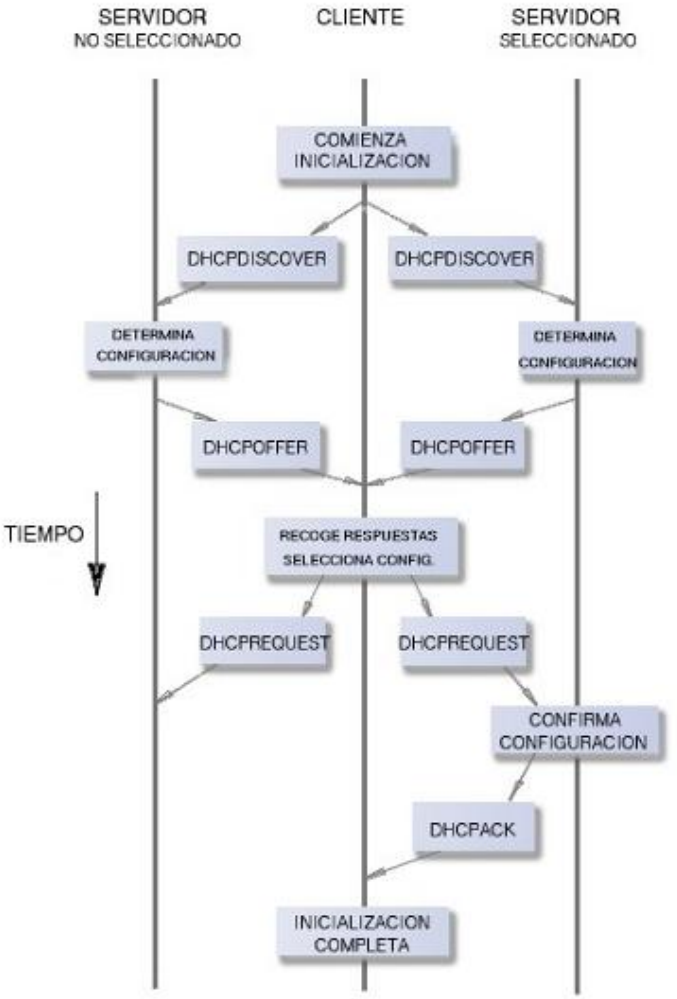
¿Pero qué pasaría si necesitara que sí o sí un sistema cuente con una dirección IP que no cambie cada vez que se reinicie? En ese caso lo que debiera hacer es configurar una dirección IP estática para esta máquina, y por ende también configurar otros parámetros, como ser sus servidores DNS, y su ruteador predeterminado, por lo menos, para permitir que nuestra máquina virtual tenga conexión con el mundo exterior.

Esto también se puede hacer utilizando el protocolo DHCP, ya que podría, como se hace en muchos casos, fijar del lado del servidor DHCP una determinada dirección IP a una dirección MAC, y aprovechar todos los demás servicios que este protocolo supone.

Una de las primeras conclusiones ya está saliendo a la luz, y es que claramente DHCP trabaja en un nivel, en la capa OSI, más abajo que “IP”, ya que cuando una máquina envía paquetes para buscar un servidor DHCP, lógicamente aún no posee esa dirección IP, y espera que el servidor se la entregue.

Profundamente dinámico

En esta sección veremos el protocolo DHCP en profundidad, para comprender bien cómo es que funciona, ya que el que asigne direcciones IP bajo pedido puede ser algo confuso de comprender.



Cuando un cliente configurado con DHCP se conecta a una red, envía paquetes de broadcast (para todos lados, para ser simplista) pidiendo por un servidor DHCP. El DHCP server, que tiene bajo su gestión un conjunto de direcciones IP (“IP address pool”), casi como si fuera un almacén, y esas direcciones se trataran de su stock, asigna a esa computadora una dirección, enrutador predeterminado, servidores DNS, servidor de zona horaria predeterminado, tiempo de alquiler (“lease

time", o el tiempo que dichos parámetros serán válidos en el cliente antes que el mismo deba enviar un nuevo pedido a la red, y renovar o conservar dichos valores) y otros tantos parámetros posibles. Cuando el cliente DHCP se desconecta de la red, por ejemplo al apagar la máquina, la dirección IP es devuelta al servidor DHCP para que luego pueda ser utilizada por otro cliente que así lo necesite.

DHCP utiliza dos puertos para su funcionamiento, que son lógicamente UDP (¿podrían ser de otro tipo, considerando que aún no hay direcciones IP en el cliente?), y que se corresponden con los puertos 67 y 68. DHCP, entonces, hace uso de cuatro fases para poder entregar las direcciones a sus clientes, a saber:

- DHCP discovery: El cliente envía mensajes en la subred física de forma tal de descubrir un servidor DHCP que pueda atenderlo. Si la subred estuviera separada de la que posee el servidor DHCP, se podría jugar con los ruteadores para que reenvíen los pedidos de los clientes, y les lleguen. Y claro está, si el cliente antes estuvo configurado con DHCP, puede solicitar un DHCP server que le entregue la misma dirección IP.
- DHCP offer: Cuando el servidor DHCP recibe un pedido de un cliente, reserva una dirección IP de su pool, y le extiende un "contrato de alquiler" de ese tipo de direcciones, o un "lease offer". Esto no es más que un mensaje que contiene la MAC address del cliente, la dirección IP que se ofrece, su máscara de red, etc.
- DHCP request: Como respuesta a la oferta de alquiler enviada por el servidor DHCP, el cliente envía un pedido de parámetros DHCP para recibir formalmente esos parámetros ofrecidos, y definir que la dirección IP en cuestión ya será asignada a esa máquina, quitándose del pool del servidor.
- DHCP aknowledgement: En esta fase final, el servidor DHCP le envía un paquete al cliente con el tiempo de alquiler, y otros tantos parámetros de red.

Dependiendo de la implementación que se efectúe, un servidor DHCP puede tener tres posibles métodos de asignación de direcciones IP, a saber:

- Alojamiento dinámico ("dynamic allocation"): Las direcciones IP disponibles se encuentran en un conjunto previamente configurado por el administrador del servidor DHCP. Las mismas son entregadas a los clientes que así las soliciten, y cuando el tiempo de alquiler termina, el servidor las reclama, y las vuelve a colocar en el conjunto o pool para ser entregadas a otro cliente. El esquema típico de los ISP ("Internet Service Providers") es éste, ya que necesitan poder entregar direcciones IP a más clientes que direcciones IP tengan asignadas en sus rangos.
- Alojamiento automático ("automatic allocation"): El servidor DHCP entrega direcciones IP en forma dinámica, pero guarda una tabla de correspondencias MAC – IP, de forma tal de entregar, de ser posible, la misma IP a la misma MAC, si estuviera aún disponible. Los ISP más serios tienen este esquema, y generalmente también los servidores DHCP de las empresas, también serias.
- Alojamiento estático ("static allocation"): Las direcciones IP están fijadas y asignadas a las MAC de cada equipo. Es útil para cambiar direcciones IP de clientes sin que éstos siquiera tengan que tocar un parámetro, o para asegurar que no habrán en nuestra red máquinas que no hayan sido expresamente declaradas.

Archívese

Vimos que los puestos de trabajo cliente podrán tener dos tipos de configuraciones de direcciones IP, estática o dinámica.

Ha llegado el momento de ver cuáles son los parámetros que se deben tocar en los diferentes puestos de trabajo *NIX para responder a una u otra configuración.

Estos parámetros se configuran, dependiendo del sabor de GNU/Linux, en diferentes archivos. Para el caso de Debian y sus derivados (Ubuntu, Mint, etc.), se debe configurar el archivo "/etc/network/interfaces", colocando, por ejemplo, las siguientes entradas si quisiéramos

que reciba direcciones IP dinámicas en una tarjeta de red “eth0”:

```
root@firewall:~# more
/etc/network/interfaces
# This file describes the network
interfaces available on your system
# and how to activate them. For more
information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp
```

Nótese que en este archivo se ha resaltado la entrada que indica que la interfaz de red eth0 utilizará “DHCP” para funcionar. Cuando el sistema operativo esté arrancando, y configure esta tarjeta de red, sabrá que debe salir a buscar un servidor DHCP para recibir sus parámetros de red.

Si quisiéramos asignarle, por ejemplo, la dirección IP 10.0.2.25/24, conservando su ruteador predeterminado (10.0.2.2), tendríamos que modificar este archivo para que tenga las siguientes entradas:

```
root@firewall:~# more
/etc/network/interfaces
# This file describes the network
interfaces available on your system
# and how to activate them. For more
information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
      address 10.0.2.25
      netmask 255.255.255.0
      gateway 10.0.2.2
```

Nótese la palabra “static”, resaltada en la sección que configura la tarjeta eth0, así como también la dirección IP asignada, su máscara de subred, y el ruteador predeterminado.

Si tuviéramos un programa como el “resolvconf” instalado en nuestro sistema, podríamos también agregar un par de entradas correspondientes a los DNS servers a utilizar, del estilo:

```
dns-nameservers 200.42.4.207,
200.49.130.44
```

Claro está, esto se agrega en la sección de la tarjeta eth0...eso quiere decir entonces que ¿se podrían tener parámetros de DNS diferentes para tarjetas diferentes?

Ahora bien, ¿qué pasa cuando lo que tenemos que implementar es esto mismo, pero en un sistema que posee Fedora GNU/Linux, o sus derivados (Red Hat, CentOS, etc.)? En este caso el archivo a modificar no será uno solo, ya que existirá uno por cada tarjeta de red a configurar, y estará ligado en forma biunívoca a esa interfaz, sea la misma tanto física como virtual; única como conjunto de otras interfaces:

Por ejemplo, si se quiere configurar la tarjeta de red eth0, nos encontraremos con el archivo /etc/sysconfig/network-scripts/ifcfg-eth0, que para el caso dinámico tendrá un contenido como el siguiente:

```
[root@server1 ~]# more
/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
ONBOOT="yes"
HWADDR=52:54:00:37:4B:A5
TYPE=Ethernet
BOOTPROTO=dhcp
```

Nuevamente, nótese la palabra “dhcp”, resaltada.

Y para el caso estático, tendrá un contenido como el siguiente, considerando que la dirección IP que deseo es la 10.0.2.35/24, con el mismo ruteador predeterminado:

```
[root@server1 ~]# more
/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
ONBOOT="yes"
HWADDR=52:54:00:37:4B:A5
TYPE=Ethernet
BOOTPROTO=static
IPADDR=10.0.2.35
PREFIX=24
```

```
GATEWAY=10.0.2.2
DEFROUTE=yes
```

Virtualmente, jugando un poco

Ahora bien, supongamos que decidimos verificar cuál es la dirección de la que salen paquetes de red correspondientes a un ping efectuado desde nuestra máquina virtual, basada en una dirección IP entregada por un DHCP server, y que tiene configurado un NAT. Para ello, utilizaremos, desde nuestra máquina física (no la virtual) el comando “tcpdump”. Supongamos que estamos conectados a Internet a través de la interfaz de red “eth0”.

Entonces ejecutaremos el comando “tcpdump -i eth0”, en nuestra máquina física, y mientras tanto, “ping 8.8.8.8” en nuestra máquina virtual. La salida del comando en cuestión será la siguiente (algo recortado, ya que los mensajes son larguísimos):

```
[root@zentraedy-l ~]# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v
or -vv for full protocol decode
listening on eth0, link-type EN10MB
(Ethernet), capture size 65535 bytes
00:48:07.407951 IP 10.100.100.105.59955 >
scl03s06-in-f22.1e100.net.https: Flags
[FP.], seq

(...)

00:48:27.238728 IP 10.100.100.105 >
google-public-dns-a.google.com: ICMP echo
request, id 2555, seq 6, length 64

(...)

00:48:35.309868 IP google-public-dns-
a.google.com > 10.100.100.105: ICMP echo
reply, id 2555, seq 14, length 64
^C
73 packets captured
73 packets received by filter
0 packets dropped by kernel
[root@zentraedy-l ~]#
```

Esto quiere decir que, para el sistema que contacte, la dirección que lo estará contactando será la 10.100.100.105, que en mi caso es la dirección IP que tengo asignada en la interfaz física de mi máquina, valga la redundancia, física, y diferente de la 10.0.2.15, configurada en mi máquina virtual.

Hemos tenido, entonces, la oportunidad de verificar cómo funciona NAT en carne propia, y hasta hemos contactado máquinas en Internet desde una máquina virtual cuya interfaz de red eth0 ha recibido una dirección IP gracias a un servidor DHCP.
Ahora, subiremos la apuesta, generando una máquina virtual que se conecte a Internet a través de esta máquina con la que hemos practicado hasta ahora. Bajemos nuestra máquina virtual, para poder modificar un poco su configuración. Ejecutemos con mucho cuidado el comando “poweroff”.

Primero, seleccionemos la configuración de nuestra máquina virtual, específicamente la sección de “Red” (“Network”), y seleccionaremos la solapa que se refiere al segundo adaptador (“Adapter 2”). En ella clickearemos el cuadro que habilita dicha tarjeta de red. En la sección “Adjunta a” (“Attached to”) seleccionaremos “Internal Network”, y como nombre de red pondremos “interna1”, por ejemplo. Luego de esto, presionaremos “OK” para aceptar todos los cambios.

¿Qué hemos hecho? Hemos agregado una tarjeta de red a nuestra máquina virtual, y la hemos apuntado a una red interna que hemos generado, llamada “interna1”, a la cual luego conectaremos otra máquina virtual.

Ahora, crearemos otra máquina virtual, pero basada en RPM, es decir, una VM CentOS, para la cual utilizaremos la imagen CentOS-6.0-i386-bin-DVD.iso.

Seguiremos los mismos pasos que antes seguimos, pero como nombre de máquina seleccionaremos “centos1”, luego como tipo seleccionaremos “Linux” y como versión “Red Hat”. También asignaremos 512 MB de memoria RAM, y crearemos un nuevo disco del tipo de archivo “VDI”, dinámicamente alojado, de 15 GB de tamaño total.

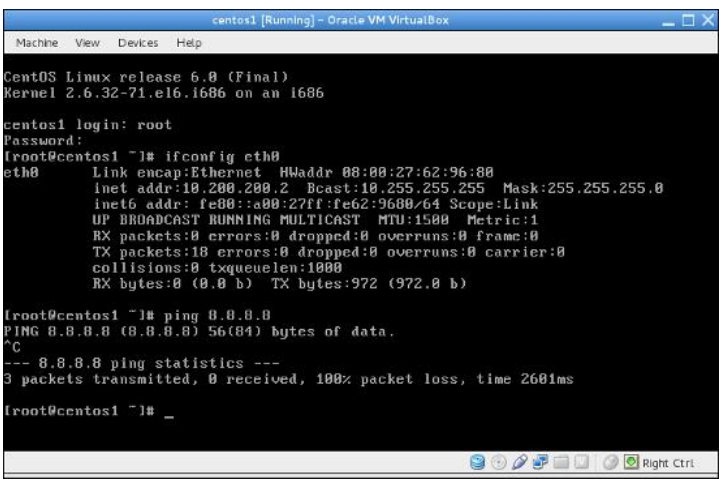
Seleccionaremos como lo hemos hecho antes la opción “Configuración” con el botón derecho

del mouse sobre la máquina virtual, asignando al CD la imagen CentOS-6.0-i386-bin-DVD.iso, y a la red la configuración “Internal Network”, y como nombre de la red interna “interna1”, la misma que agregamos a la segunda tarjeta de red de nuestra máquina virtual Debian.

Seleccionaremos “OK” para confirmar los cambios, e iniciaremos la instalación de la máquina virtual. Seleccionaremos, luego del booteo, “Install or upgrade an existing system”, saltarmos la comprobación del disco de instalación (“Disk Found” -> “Skip”), con lo que el instalador gráfico dará inicio.
Seleccionaremos las siguientes opciones, a saber:

- En la primera pantalla, sólo seleccionaremos “Next”.
- Como idioma de la instalación, seleccionaremos “English”, y presionaremos el botón “Next”.
- Como teclado seleccionaremos “Latin American”, y presionaremos “Next”.
- Como tipo de almacenamiento, seleccionaremos “Basic Storage Device”, y presionaremos “Next”.
- Si apareciera un error de inicialización de disco, presionar el botón “Re-initialize all”, siempre verificando antes que se haga referencia al “VBOX DISK”.
- Como hostname coloquémosle “centos1”, y presionemos el botón “Configure Network”.
- En la ventana de “Network Connections” seleccionemos la solapa “Wired”, y dentro de ella la interfaz “System eth0”.
- Presionemos el botón “Edit”.
- En la nueva ventana, “Editing system eth0”, seleccionemos el checkbox “Connect automatically”.
- Seleccionemos la solapa “IPv4 Settings”.
- En la lista que aparece en “Method” cambiemos de “Automatic (DHCP)” a “Manual”.
- En la sección de “Addresses” presionemos el botón “Add”.
- Configuremos la nueva dirección con “10.200.200.2” como “Address”, “255.255.255.0” como “Netmask”, y “10.200.200.1” como “Gateway”.
- En la sección de DNS Servers coloquemos “10.200.200.1”, y presionemos el botón “Apply”.
- En la ventana de “Network Connections” presionemos el botón “Close”.
- Presionemos el botón “Next”.
- Configurar la zona horaria según el lugar donde estemos. En mi caso, es “America/Argentina/Buenos_Aires”, y presiono el botón “Next”.
- Configuremos la clave de root dos veces, y presionemos “Next”.
- En la sección de particionamiento, seleccionemos “Use all space”, y seleccionemos “Next”.
- Aceptemos el particionamiento propuesto por el instalador presionando “Next”.
- Si se nos pregunta por la opción de formatear, presionar el botón “Format”.
- Confirmar con “Write changes to disk”.
- Dejar la opción “Install boot loader on /dev/sda”, y presionar “Next”.
- En la selección de software, seleccionemos “Minimal”, y presionemos “Next”.
- Cuando termine la instalación, presionar el botón “Reboot”.
- Al final esta instalación, relativamente chica, tendremos un sistema CentOS 6 implementado, con el cual jugaremos para ver cómo se comporta detrás de una máquina Debian, y conectándose a internet a través de ella.

Cuando la máquina virtual reinicie, levantemos también firewall1, para ver qué ocurre al intentar ejecutar un ping como antes lo hicimos. ¿Sorpresa? Los paquetes de red de la máquina centos1 no salen a la red. ¿Motivo? Sencillo, su ruteador predeterminado, 10.200.200.1, no existe aún. Esa será la dirección IP que configuraremos en la segunda tarjeta de red que hemos definido para la máquina firewall1.



Para eso, ingresaremos a la máquina virtual “firewall1”, que tiene hostname “firewall”, y configuraremos su segunda tarjeta de red. Primero veremos cuáles son ahora sus tarjetas de red disponibles:

```
root@firewall:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr
08:00:27:31:01:50
          inet addr:10.0.2.15
Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr:
fe80::a00:27ff:fe31:150/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST
MTU:1500  Metric:1
          RX packets:75 errors:0 dropped:0
overruns:0 frame:0
          TX packets:68 errors:0 dropped:0
overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9344 (9.1 KiB)  TX
bytes:8579 (8.3 KiB)

eth1      Link encap:Ethernet  HWaddr
08:00:27:20:21:ba
          BROADCAST MULTICAST  MTU:1500
Metric:1
          RX packets:0 errors:0 dropped:0
overruns:0 frame:0
          TX packets:0 errors:0 dropped:0
overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0
(0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1
Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436
Metric:1
          RX packets:0 errors:0 dropped:0
overruns:0 frame:0
          TX packets:0 errors:0 dropped:0
overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0
(0.0 B)
```

Editemos el archivo /etc/network/interfaces, para que quede parecido a lo siguiente (nótense los agregados de las líneas que comienzan con “auto”):

```
root@firewall:~# vi
/etc/network/interfaces

auto eth0
allow-hotplug eth0
iface eth0 inet dhcp

auto eth1
allow-hotplug eth1
iface eth1 inet static
        address 10.200.200.1
        netmask 255.255.255.0
```

Ahora relancemos los servicios de red en el sistema Debian, y veamos lo que ocurre al verificar nuevamente las direcciones IP existentes en él:

```
root@firewall:~# service networking
restart
[....] Running /etc/init.d/networking
restart is deprecated because it may not
r[warnble some interfaces ... (warning).
[....] Reconfiguring network
interfaces...Internet Systems Consortium
DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems
Consortium.
All rights reserved.
For info, please visit
https://www.isc.org/software/dhcp/

Listening on LPF/eth0/08:00:27:31:01:50
Sending on   LPF/eth0/08:00:27:31:01:50
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255
port 67 interval 8
DHCPREQUEST on eth0 to 255.255.255.255
port 67
DHCPOFFER from 10.0.2.2
DHCPACK from 10.0.2.2
RTNETLINK answers: File exists
bound to 10.0.2.15 -- renewal in 38093
seconds.
ifup: interface eth1 already configured
done.
```

En este último comando tenemos varios elementos interesantes. Veamos algunos de los marcados en negrita. DHCPDISCOVER, DHCPREQUEST, DHCPOFFER, y DHCPACK nos deben sonar bastante conocidos, para lo que estuvimos viendo en apartados anteriores, ¿verdad?

Por otro lado, vemos un mensaje que anuncia una renovación de dirección en una cantidad de tiempo determinado. Claro, ¿verdad?

Verificaremos con ifconfig nuevamente:

```
root@firewall:~# ifconfig -a
```

```
eth0      Link encap:Ethernet  HWaddr
08:00:27:31:01:50
          inet addr:10.0.2.15
Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr:
fe80::a00:27ff:fe31:150/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST
MTU:1500  Metric:1
          RX packets:1052 errors:0
dropped:0 overruns:0 frame:0
          TX packets:690 errors:0
dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:89314 (87.2 KiB)  TX
bytes:83407 (81.4 KiB)

eth1      Link encap:Ethernet  HWaddr
08:00:27:20:21:ba
          inet addr:10.200.200.1
Bcast:10.200.200.255  Mask:255.255.255.0
          inet6 addr:
fe80::a00:27ff:fe20:21ba/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST
MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0
overruns:0 frame:0
          TX packets:30 errors:0 dropped:0
overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX
bytes:2340 (2.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1
Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436
Metric:1
          RX packets:0 errors:0 dropped:0
overruns:0 frame:0
          TX packets:0 errors:0 dropped:0
overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0
(0.0 B)
```

Ahora, intentaremos llegar desde “firewall” hasta “centos”, con el siguiente comando:

```
root@firewall:~# ping 10.200.200.2
PING 10.200.200.2 (10.200.200.2) 56(84)
bytes of data.
64 bytes from 10.200.200.2: icmp_req=1
ttl=64 time=33.1 ms
^C
--- 10.200.200.2 ping statistics ---
1 packets transmitted, 1 received, 0%
packet loss, time 0ms
rtt min/avg/max/mdev =
33.162/33.162/33.162/0.000 ms
```

Efectivamente, ya la máquina “centos” forma parte de nuestra red, y podemos alcanzarla. Ese será el puntapié para las prácticas que haremos en próximos artículos. A no borrar las máquinas virtuales.

Conclusión

En este artículo hemos aprendido algo sobre NAT, DNAT, SNAT, IP Masquerade, DHCP, configuración de direcciones estáticas y dinámicas, y armamos una red interna que utilizaremos para las prácticas que vendrán. En el artículo próximo jugaremos con protocolos como DHCP, DNS, comprenderemos qué es y armaremos nuestro firewall, permitiremos y denegaremos puertos, colocaremos servidores internos y los accederemos (o no), traeremos vía TCP/IP a Snowden, seremos testigos de su casamiento en Venezuela con la hermosa espía rusa que se lo propuso, y mucho más.

¡Hasta la próxima!

Hernán “HeCSa” Saltiel
AOSUG Leader
CaFeLUG Member
Twitter: @hcsaltiel
hsaltiel@gmail.com
http://www.facebook.com/hcsaltiel
http://www.aosug.com.ar

Opinión - La indefinición de la pantalla

La pantalla ha sido, en los últimos años, el periférico que definía a la PC. Ya no es la Pantalla monocromática Hércules ni esos pobres intentos de llevar el Color con la CGA y EGA, sobre todo si las comparamos con la calidad y rapidez de Apple Lisa a pesar de su falta de color. Fuera de las Home Computers, los primeros años de la computación utilizada por la gente común, era algo deprimente hasta la llegada del VGA. La cosa se estabilizó hasta la llegada del manejo de las texturas en donde la imágenes empezaron a ser más realistas, y más exigentes de datos.

Para 1999 una guerra entre 3Dfx y nVidia se veía más interesante que la batalla entre Intel, AMD y el agonizante Cirux, era más encarnizada, más cruda. Finalmente nVidia ganó y hasta terminó comprando a 3Dfx; las otras empresas debieron correr Overclockeando sus períodos de desarrollo. Incluso Apple estuvo en peligro en esta batalla y en otros temas hasta que Jobs regresó.

La pantalla fue, obviamente, la parte más notoria de la computación, pues impulsó y requirió más de los equipos: discos de mayor capacidad, mayores velocidades de transferencia, más potencia de cálculo del procesador, más cache, más RAM, más potencia de la fuente. Pero entre todo la pantalla varió en mayor definición 800, 1024, 1280, 1366, 1600, HD y 4K; más cantidad de colores, mayores frecuencias de sincronismo. El monitor se hizo más profundo y más ancho y alto. Siempre creció.

Incluso en los asistentes personales, agendas, relojes y celulares creció y se aumentó en resolución. Desde las 4 líneas de la Casio 4600 a las Palm. Los relojes inteligentes no habían prosperado hasta ahora, el timex que tomaba los datos del monitor, el Microsoft que tomaba datos de una sub portadora de radio frecuencia FM. Ni hablar del Casio reloj cámara, (con una resolución horrible), con excepción del reloj calculadora de Casio, todos los demás no duraron. Los que tuvieron más éxito hasta ahora después del calculadora son los que incluían sensores biométricos para deportistas, sensores meteorológicos y GPS. ¿Ahora que vendrá? ¿El Tricoder en la muñeca?

Los celulares también han crecido en su pantalla. Hasta el modelo que más se resistía a agrandar su pantalla se agrandó. Y aquí llegó el punto de inflexión. Y todo se mezcló.

Ahora hay teléfonos desde las 3" a las 7", handset inteligentes con el tamaño de un reproductor MP3 y relojes. Sony es la que se está esforzando. Un teléfono enorme y un reloj con un accesorio del mismo con una resolución apenas menor que algunas Palm. ¿No era mejor una tableta grande y un celular pequeño?

Y lo último es lo más revolucionario. Los lentes de Google. La gente se obsesiona con el problema de una cámara apuntándoles constantemente. Pero allí está, pueden sacarle la cámara si quieren, pero en una pequeña pantalla que se ve enorme en tu ojo. Tiene un terrible potencial a futuro. Mayor que la PC, la tableta y el celular, o todo junto.

Es obvio que la era de la PC está llegando a su fin, lo que parece que se viene podría ser calificado como "La era de las pantallas". Pero, ¿Necesitamos una pantalla grande o chica? ¿Más o menos resolución? ¿HD, 3D, 4K, 8K? ¿Qué relación geométrica, 4/3, 16/9, 22/10, otras? ¿Cuántas de ellas? ¿En el escritorio, la pared, el bolsillo, el bolso, la muñeca, el lente, o en todos lados? Esa es la contradicción de hoy día: la indefinición de la pantalla, y si de algo estoy seguro, es que va a ser muy variado.

Claudio De Brasi.
@Doldraug.

PD: si de ésto sale un fracaso o un nuevo paradigma es cuestión de tiempo. Pero si fracasa ahora, no quiere decir que no sea un paradigma en el futuro. Pues éste es tan indefinido como la pantalla.

Consecuencias de la nota anterior:
Google reconsideró y dejó sin efecto su eCorralito. No por mi nota sino por la queja de muchos desarrolladores. Que haya primado la lógica es lo que importa.

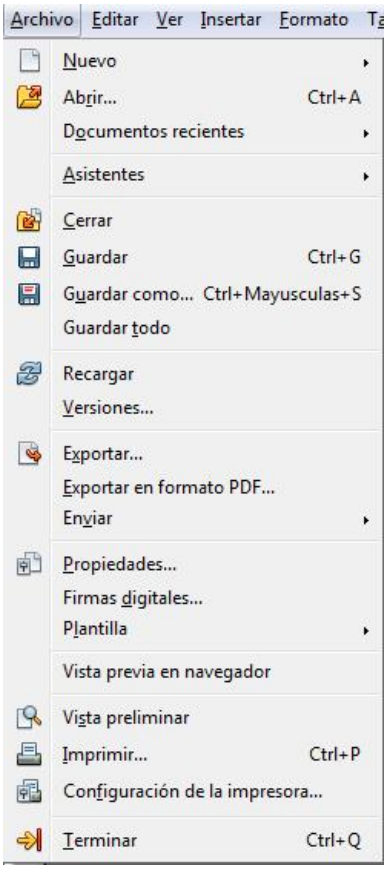
Fe de Erratas:
Un error en la nota 59, se me olvidó una denominación monetaria. Lo que vuelve a la economía nacional un poco más... deprimente.
Los ceros son 13 en total. Ley 2, Argentinos 4, Austral 3, Pesos convertibles 4. 1$ = 10.000.000.000.000 $M/n

Iniciando con OpenOffice Writter

POR **RAFAEL MURILLO**

Como había ya comentado en las entregas anteriores de esta guía de OpenOffice, si estás acostumbrado a manejar procesadores de textos, como es el caso del Word de Microsoft, te resultará muy familiar OpenOffice Writer. Sin embargo, que hagas uso de la "Ayuda" que el programa posee y que aparece en el menú principal, es una muy buena práctica por si tienes dudas.

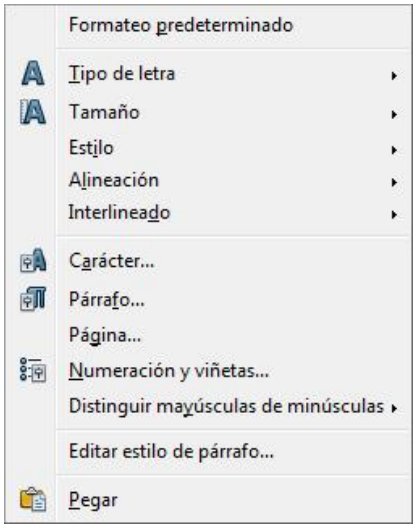
Vamos a ver ahora que las funciones y funcionalidades de OpenOffice son básicamente idénticas a las de cualquier otro procesador de textos, ya sea que utilicemos el mouse y el teclado para trabajar, o que hagamos una cosa distinta con cada uno (porque hay quien ya se tiene aprendidos los atajos del teclado para realizar ciertas cosas, como por ejemplo dar formato al texto).



Las acciones con el mouse

Bien, lo que se puede hacer con el mouse en OpenOffice es exactamente lo mismo que podemos hacer en cualquier otro programa. Clic izquierdo siempre hará la función principal, es decir, ejecutar el comando al que demos clic. Por ejemplo, si damos clic izquierdo en cualquiera de los menús, podremos ver su contenido, y si volvemos a repetir dicha acción en cualquiera de sus opciones, se ejecutará la opción seleccionada. Lo mismo pasa si damos clic izquierdo en cualquiera de los botones de las barras de herramientas: vamos a ejecutar dicho comando.

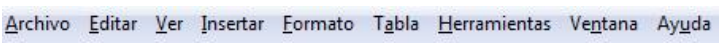
Y clic derecho mostrará un menú contextual con diversas opciones que podremos ejecutar al darle clic izquierdo en cualquiera de ellas.



Las acciones con el teclado

Tal como en el Word de Microsoft, si somos observadores, podremos notar que cada uno de los menús del Writer de OpenOffice tiene una de las letras siempre subrayada.

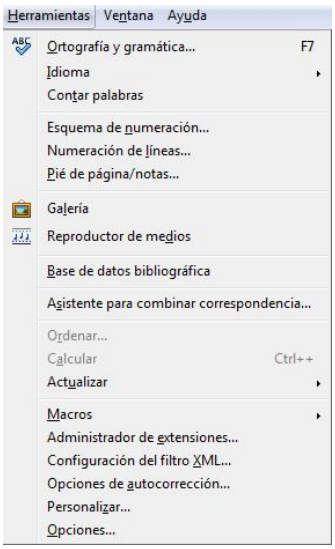
Ahora bien, puedes abrir el menú que quieras, presionando la tecla de la letra correspondiente (la que esté subrayada) mientras presionas **al mismo**



tiempo la tecla Alt del lado izquierdo de tu teclado.

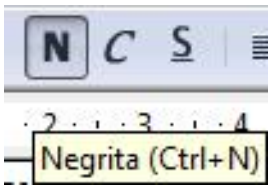
Observa igualmente que en los menús desplegables también aparecen letras subrayadas. Su finalidad es exactamente la misma.

Y por último, algunas opciones de menú también son accesibles a través de ciertas combinaciones de teclas. Son las llamadas **teclas de método abreviado** (o atajos de teclado), que se pueden utilizar para casi todo en Writer (en realidad muchas de ellas son aplicables a toda la suite de OpenOffice). Si queremos conocer una tecla de método abreviado para algún botón que esté en las barras de herramientas, simplemente vamos a posicionar el puntero del mouse en dicho botón, y nos aparecerá un recuadro con el nombre del botón y su método abreviado.

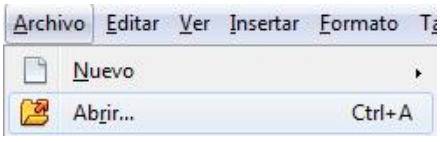


En resumen, y aplicando los conocimientos anteriores, podemos por ejemplo, abrir un documento existente de distintas maneras:

- Clic izquierdo en el menú “Archivo” y luego clic izquierdo en la opción “Abrir”.



- Utilizar el método abreviado, pulsando la tecla “Control” y al mismo tiempo la tecla “A”.
- Utilizar el teclado únicamente, oprimiendo primero la tecla “Alt” y al mismo tiempo la tecla “A” para abrir el menú Archivo, posteriormente, pulsando la tecla “Alt” nuevamente y al mismo tiempo la tecla “R”.



Incluso se pueden utilizar combinaciones entre estos tres métodos...

¡Ya no sirve mi mouse, no puedo trabajar!

Esto quizás podría pasarte en el Word de Microsoft, sin embargo, en Writer de OpenOffice existe la opción de trabajar completamente sin el mouse. Para eso simplemente debemos pulsar la tecla “**F6”.** Observa cómo se selecciona el primer menú. Si vuelves a pulsar la tecla “F6” se seleccionará el primer icono de la barra estándar, una nueva pulsación sobre “F6” seleccionará el primer icono de la barra de formato. Pulsa “F6” de nuevo y se mostrará el cursor del ratón en el área de texto para que puedas comenzar a escribir. La combinación de teclas Mayús (Shift) + F6 hará que la selección vuelva hacia atrás (en el orden antes mencionado).

Para moverte por las diferentes entradas de cada menú o por los iconos de las barras de herramientas sólo tendrás que pulsar la tecla de tabulación o incluso las teclas de dirección. Presiona la tecla “Intro” para activarlos.

¡Necesito una pantalla más grande!

Finalmente, para aquellos a los que “les estorban” las barras de herramientas, menús, etc... Writer permite trabajar utilizando la totalidad de la pantalla, tal vez te resulte útil. Para utilizar esta opción, presiona la combinación de teclas **Ctrl + Mayús (Shift) + “J”** tanto para ver el documento en pantalla completa como para regresar al modo normal de vista, también puede regresar con tecla “Esc”.

Como podrás darte cuenta, y claro, como ya lo mencionamos anteriormente, no es el objetivo de esta guía decirte para qué sirve cada uno de los botones (para eso está la sección de ayuda del mismo programa). El objetivo radica en poder ayudarte a conocer algunas funciones extras que es probable puedan hacerte una persona más productiva y eficiente al momento de utilizar las herramientas libres.

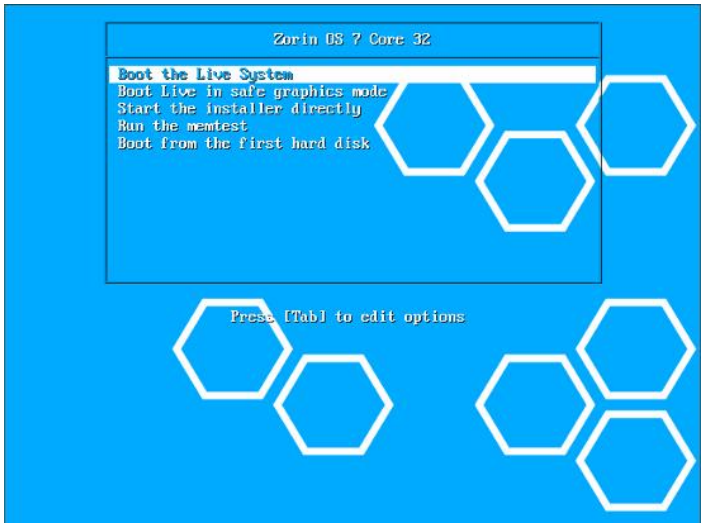
Rafael Murillo
@linxack
linxack@gmail.com

Tutorial de Instalación: ZorinOS 7

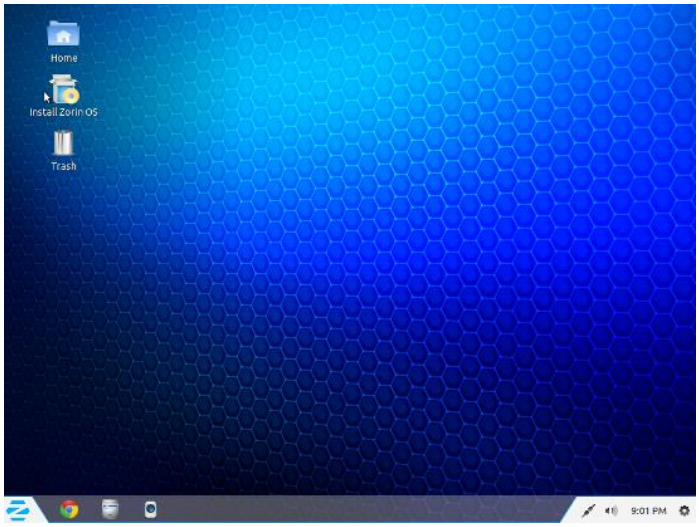
POR **NATAEL ANDRES GARRIDO**

Hola a todos los lectores de esta gran revista. En este número les traigo un tutorial de instalación y una rápida muestra de una distro digna de los novatos.
Es, nada más y nada menos, que ZorinOS 7, una distribución dedicada a aquellas personas que quieren cambiarse al mundo GNU/Linux sin perder el entorno al que está acostumbrado la persona que usa cualquier versión de Windows.
Esto es una característica fuerte de ZorinOS 7 ya que ese es su objetivo, ayudar a los novatos a sentirse cómodos en GNU/Linux. Esta distro está basada en Ubuntu 13.04 por lo que la instalación es super fácil haciendo que en pocos minutos tengamos la distro instalada y lista para usar.
De esta distribución también se destaca una aplicación hecha especialmente de parte del equipo de desarrollo de ZorinOS llamada “Zorin Look Changer”, la cual en algunos simples clicks podemos tener el aspecto tanto de Windows 7, como Windows XP o también el estilo clásico de Gnome2.

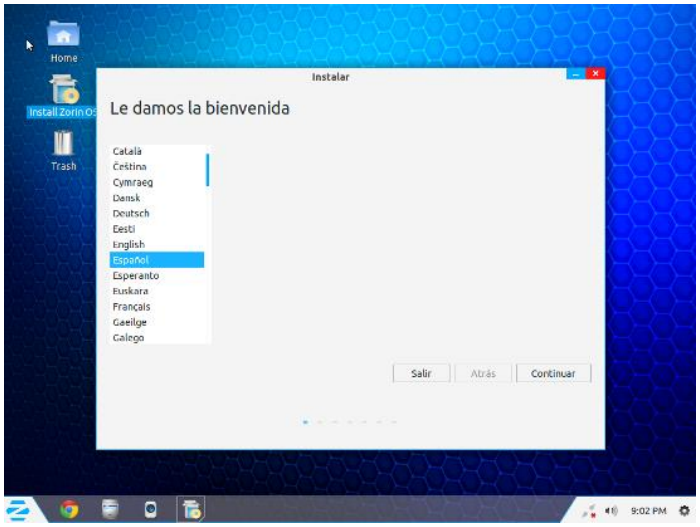
Empecemos con la instalación, después les voy a mostrar el Look Changer funcionando:
Booteamos el sistema y nos aparece el típico menú de instalación. Elegimos que bootee el sistema en modo Live dándole Enter.



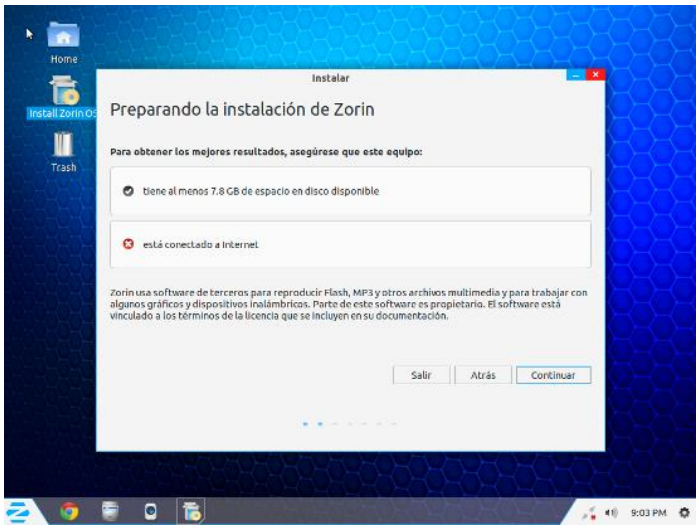
El sistema ya está booteado luego de unos segundos y vemos la pantalla limpia con un entorno muy conocido por muchos y los típicos 3 iconos en el escritorio (Carpeta Home, Icono de instalación, Icono de Papelera).
Como les había dicho el entorno es muy amigable y cualquier persona que nunca haya usado GNU/Linux en su vida puede empezar a usar la distro ya que su parecido con Windows 7 es muy evidente.
Por si alguno lo pregunta, la barra de inicio con sus botones y configuraciones es el conocido dock AWN (Avant Window Navigator).



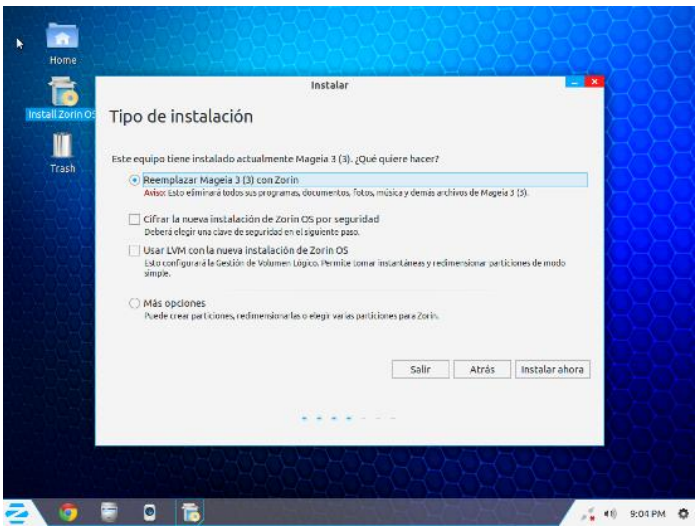
Hacemos doble click en el icono de instalación y va a ejecutarse el instalador ultra conocido por todos, el de Ubuntu. Elegimos el idioma de instalación y hacemos click en “continuar”.



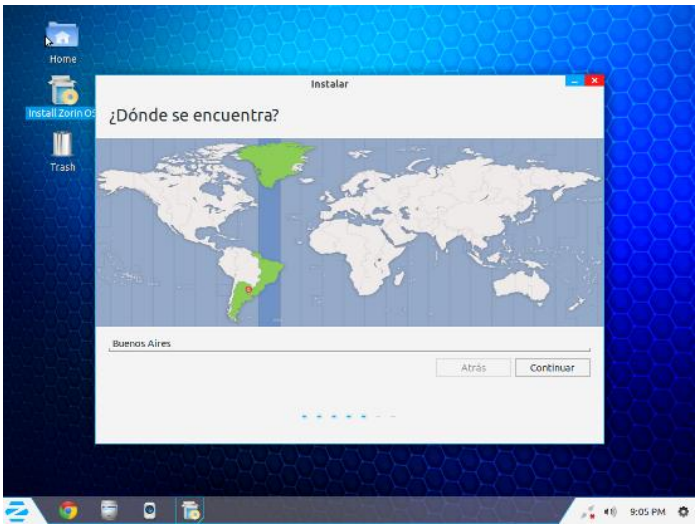
El siguiente paso es verificar, por parte del sistema, si tenemos el espacio suficiente para instalar la distro y si estamos conectados a internet. Esto último lo hace para poder instalar los codecs de terceros privativos (mp3, flash y demás). Como yo estoy en una máquina virtual, no estoy conectado a internet así que no se me van a instalar los codecs y la instalación va a ser más rápida debido a esto.



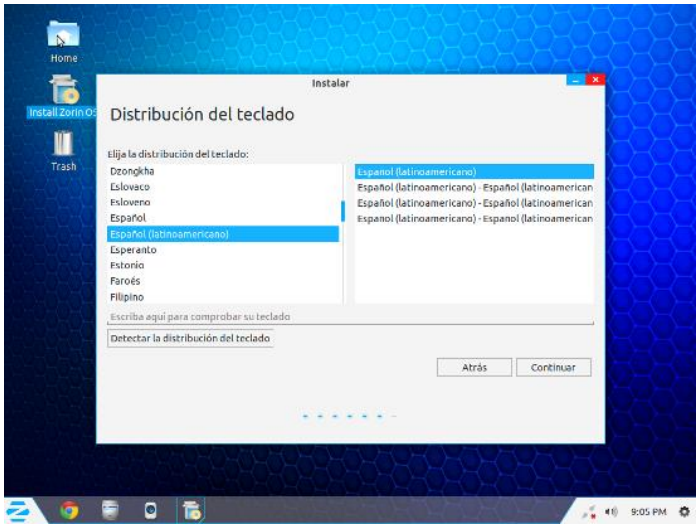
El siguiente paso es elegir el tipo de instalación. Como yo tenía instalado anteriormente Mageia 3, me da la opción de reemplazarlo formateando dicha partición. Si vamos a “Más Opciones” vamos a poder elegir la partición donde queramos instalar ZorinOS, incluso si ya tenemos otro Sistema Operativo instalado. Yo elegí que reemplace el sistema instalado.



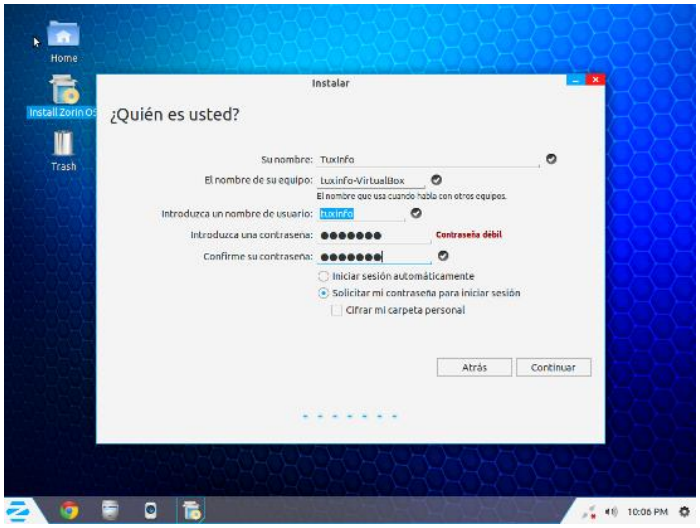
Una vez que hacemos click en “instalar ahora”, vamos a elegir nuestra ubicación regional. La elegimos y hacemos click en “Continuar”.



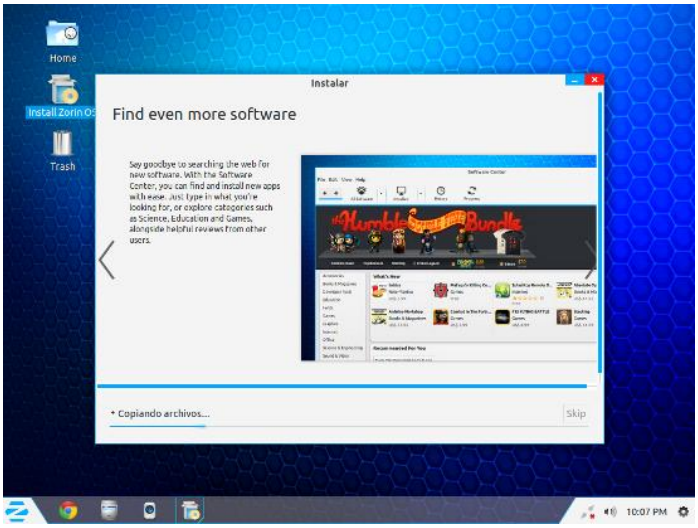
Elegimos la distribución del teclado que tengamos. La mía es Latinoamericano. Si tenemos la duda, podemos probar las teclas y combinaciones en la casilla de abajo. Hacemos click en “Continuar”.



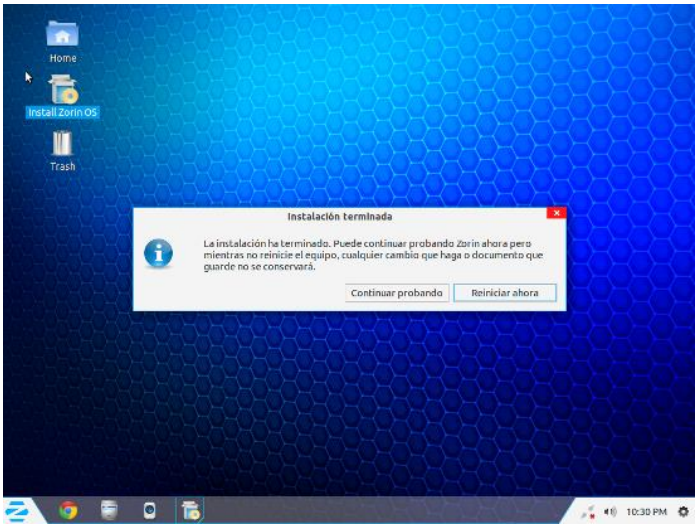
Ahora vamos a completar los datos del usuario que va a ser el administrador del sistema. Si queremos podemos hacer que se inicie sesión automáticamente o por medio de una contraseña. Completamos los datos y hacemos click en “Continuar”.



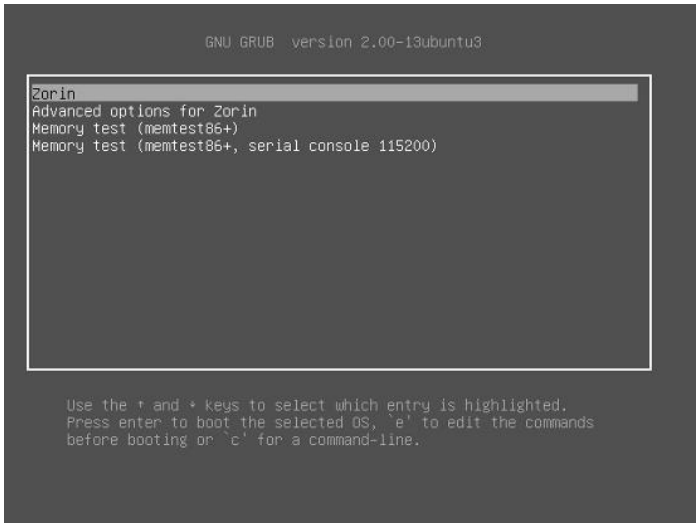
Una vez completados los datos el sistema se va a empezar a instalar. Si ustedes están conectados a Internet, este paso va a durar un poco más ya que va a descargar los codecs privativos para los archivos de audio y video y también el flash, entre otras cosas. Si no están conectados a Ínternet, este paso tendría que durar, más o menos, unos 7 u 8 minutos.



Una vez que el sistema se instaló hacemos click en “Reiniciar ahora” para que el sistema ya instalado inicie por primera vez. Como ven los pasos de instalación son muy sencillos, incluso más sencillos que muchas distros ya conocidas por todos. Esa también es la idea de los desarrolladores, que cualquier novato se sienta a gusto para poder instalar el mismo el sistema.

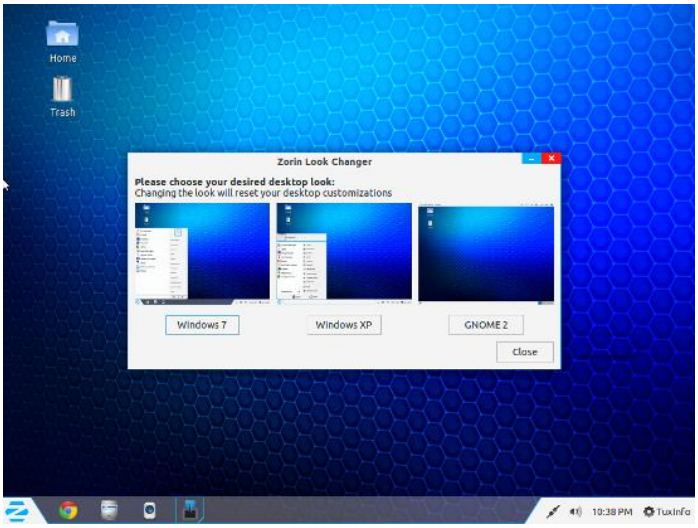


Al reiniciar, nos vamos a encontrar con el GRUB que si es que ZorinOS 7 es el único sistema que está instalado va a mostrar la pantalla igual a esta, sino va a mostrar la lista con ZorinOS y otro sistema que tenga instalado. Elegimos “Zorin” para que bootee nuestro sistema.



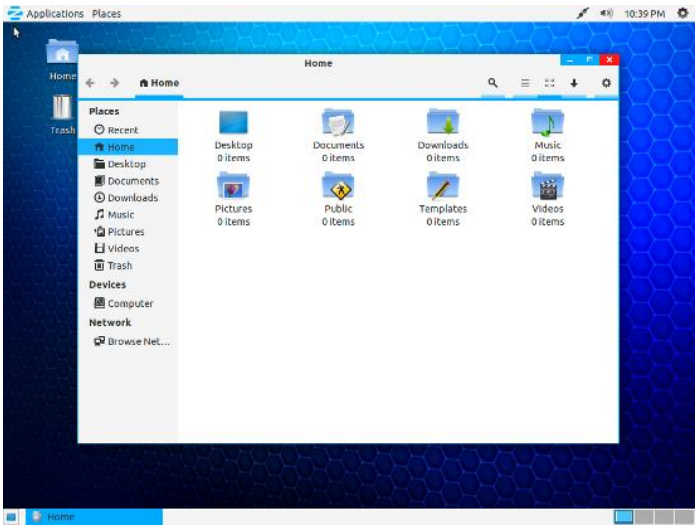
Si alguna vez instalaron Ubuntu, esta pantalla les va a resultar muy conocida ya que es la pantalla de login de Ubuntu 11.04 (en adelante), un poco modificada por el equipo de desarrollo. No se ustedes, pero a mí me gusta más en azul. Ingresamos con el usuario y la contraseña configurados anteriormente.

Nuestro sistema ya está listo para ser usado. Ahora quiero mostrarles el “Zorin Look Changer” que les había comentado al inicio del artículo. En esta versión, la llamada “Free version”, sólo se pueden elegir 3 estilos distintos de configuración de escritorio, como dijimos antes, Windows 7, Windows XP y GNOME2.



En la versión “Premium” (que va desde 8 euros hasta 15, según diferentes versiones tales como “Ultimate”, “Business”, “Multimedia” o “Gaming”), tenemos más estilos para elegir, los cuales son: Windows 7, Windows XP, Windows 2000, Unity, MacOS X, Gnome2.

Acá vemos cómo queda el estilo “Gnome2” de la versión Free.



Espero que les haya gustado este tutorial de instalación + Mini Review de ZorinOS 7 y ya saben, si conocen a alguien que se quiere pasar a GNU/Linux y siempre usó Windows, es bueno que le recomienden esta distro ya que se va a sentir muy a gusto tanto con el entorno y le va a ser muy fácil empezar a conocer cómo funciona un sistema con GNU/Linux.

¡¡Hasta el próximo número!!

Natanael Andrés Garrido
Twitter: @NatanaelGarrido
G+: Natanael Garrido
Web: www.neositelinux.com.ar

Auditando una Red WiFi - Ataque Chop-Chop a WEP

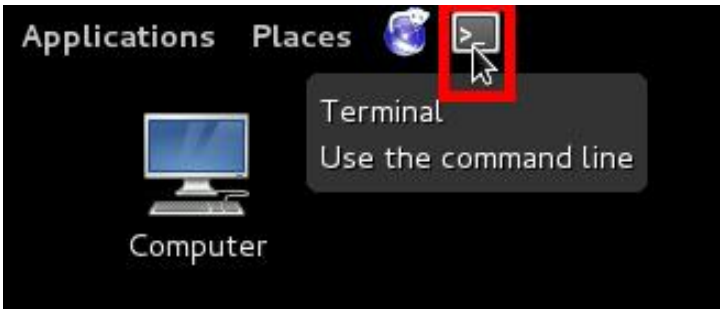
POR **JUAN PABLO LOZANO**

Este artículo tratará con detalles cómo realizar auditorías wireless, en esta ocasión sobre redes WiFi con seguridad WEP (64/128 bits).

Como introducción se hace una breve descripción sobre los tipos de seguridad wireless. Una red Wifi básicamente posee 3 tipos de seguridades: abierta, WEP, WPA/WPA2 y su clasificación es adepta a la encriptación usada. Las redes abiertas son aquellas que conocemos como “libres” es decir, las que no poseen ninguna palabra clave (o contraseña) requerida para conectarse (de ahí su denominación como “Insegura”). En cambio la encriptación WEP es una contraseña de 5 o 10 dígitos según la cantidad de bits y tipo de la clave (alfanumérica o hexadecimal). Por último las redes con mayor seguridad son aquellas con encriptación WPA/WPA2 que varían desde 8 caracteres en adelante.

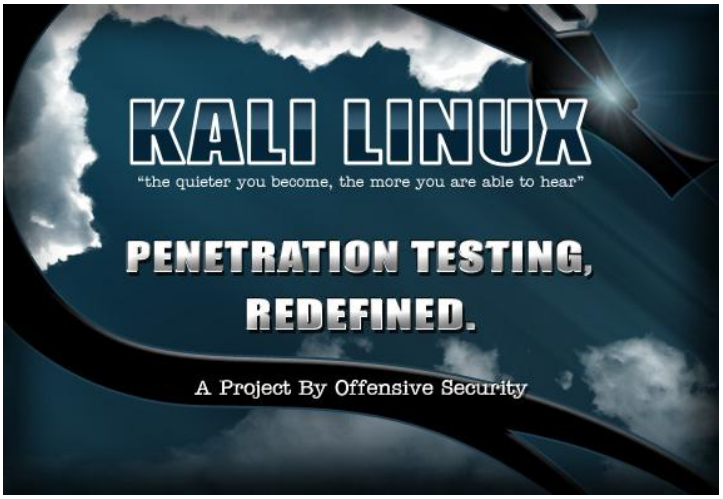
A fines prácticos se utilizará el sistema operativo KALI LINUX, que es una distribución GNU/Linux orientada a la seguridad informática (la cual se trató en el anterior número de TuxINFO). Este sistema tiene la capacidad de iniciarse desde un DVD o USB sin necesidad de instalarlo en el equipo que usaremos para la práctica. Para obtener una copia, puedes descargarlo desde su sitio oficial: http://www.kali.org/downloads/

¡Manos a la Obra!



Para mayor comprensión se usarán marcas ROJAS en las imágenes para apuntar lo más importante e informar sobre algunos términos. Las opciones -h corresponden a la dirección MAC del equipo con el que se realizan las pruebas, mientras que las opciones -b y -a indican la dirección MAC de la red WiFi.

Primeramente lo más obvio y primordial es tener



KALI LINUX ejecutándose preferentemente en un equipo portátil o PC con acceso a una tarjeta de red inalámbrica (WiFi).
Acto seguido pasamos a la auditoría propiamente dicha: Lo que deberán hacer una vez estando en el Escritorio de Kali Linux, es abrir una terminal, para ello basta con hacer clic en el icono de la terminal situado en la parte superior izquierda de la pantalla (junto a los menús de Aplicaciones y Lugares).

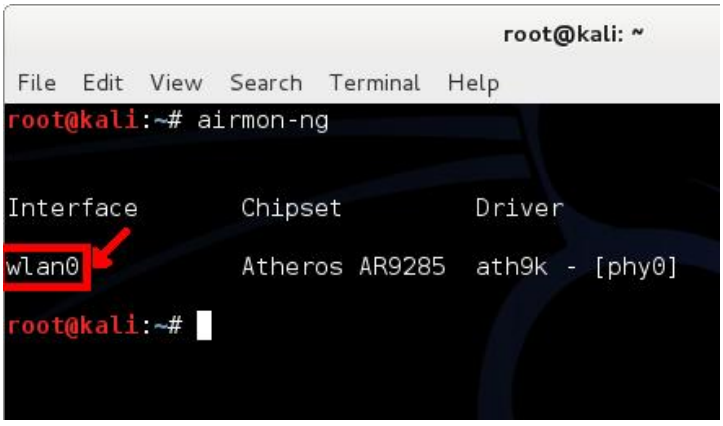
Ahora que estamos en la aplicación de trabajo llamada Terminal, iremos ejecutando órdenes (o comandos) que realizarán determinada tarea, alguna de ellas es vital dejarlas trabajando en una sola terminal y dejar que hagan su trabajo.

Es necesario comprender a líneas generales el proceso que usaremos para auditar para saber lo que iremos haciendo, y en qué consiste.

Analizando de Izquierda a Derecha, uno debe elegir la red WiFi con seguridad WEP que desea auditar, luego asociarse a dicha red, generar un paquete Llave que permitirá generar paquetes para que en alguno de ellos nos devuelva la clave de la red y finalmente desencriptarla.

Hasta ahora es sencillo, pero se debe aclarar que en los pasos de asociarse a la red y en crear el paquete llave y posteriormente generar paquetes válidos, se necesita tener la tarjeta de red inalámbrica en modo MONITOR. Este modo permite hacer lo antes dicho.

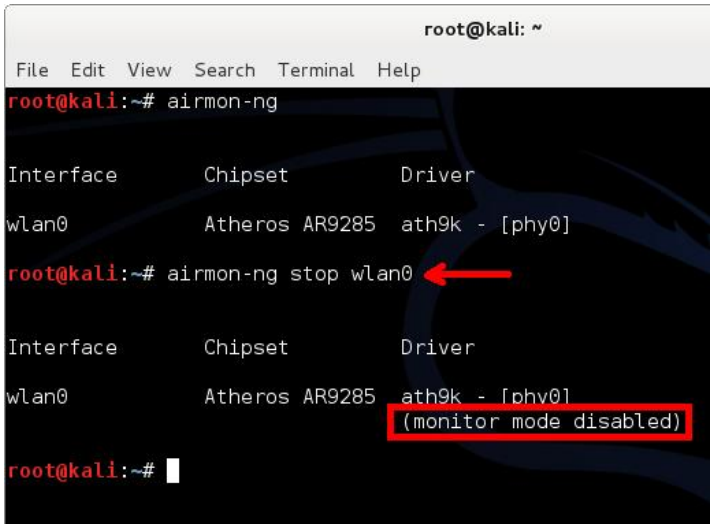
Se procede a colocar en modo monitor la tarjeta de red inalámbrica del equipo que se esta usando. Para ello en la terminal que abrimos en un principio escribimos la siguiente orden:



Como se nota en la imagen, la marca roja indica la interfaz de la tarjeta de red inalámbrica que en la práctica que estamos realizando tiene el nombre “wlan0” se necesita memorizarla (o anotarla) porque se trabajará con ella luego.

Ahora que se sabe el nombre de la tarjeta de red inalámbrica procedemos a pararla (o desactivarla) para realizar cambios, para hacerlo ejecutamos el comando:
airmon-ng stop wlan0

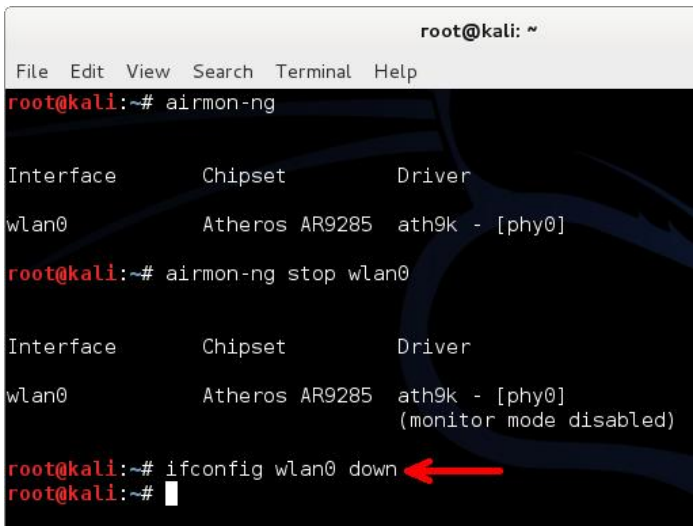
Para entender que es lo que hace, se hace un seguimiento del comando: el programa “airmon-ng” hace un “stop” (parar) a la interfaz “wlan0”. Es necesario ir siguiendo las imágenes correspondientes para guiarse:



Una vez deshabilitada la interfaz de la tarjeta de red, se necesita tener “baja” esta interfaz para que los cambios que se apliquen surtan efecto correctamente. Esto se logra ejecutando la siguiente orden:

ifconfig wlan0 down

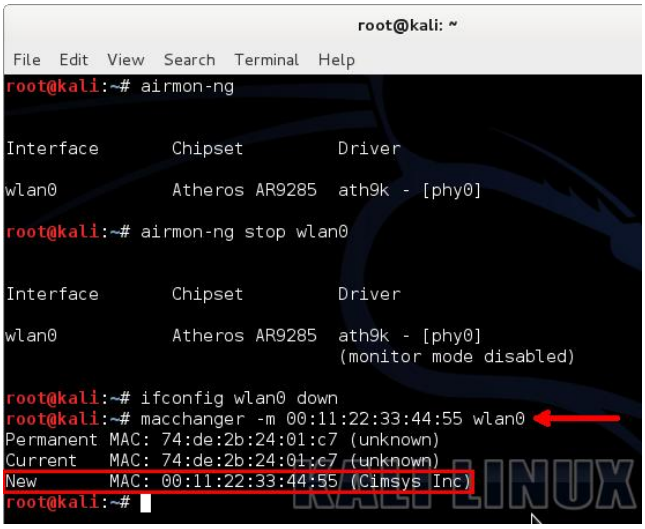
Con esto el programa “ifconfig” hace que la interfaz “wlan0” quede “baja” (down):



En este momento se necesita que la dirección MAC de la tarjeta de red inalámbrica del equipo sea fácil de recordarla, esto se modifica escribiendo:

macchanger –m 00:11:22:33:44:55 wlan0

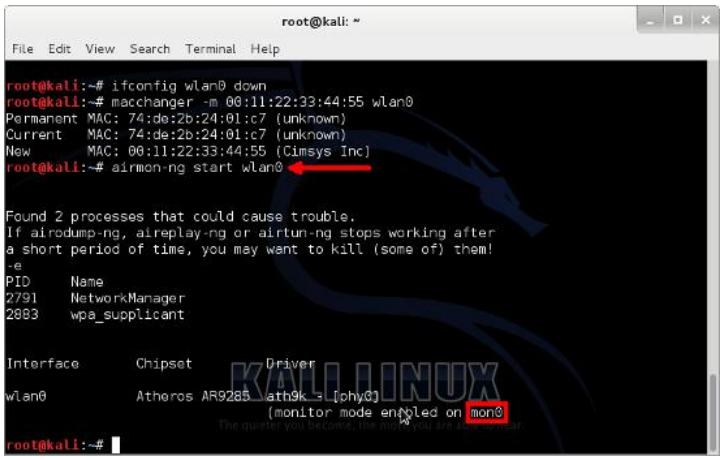
En esta sentencia, el programa “macchanger” hace uso de la opción “m” para modificar la MAC del dispositivo “wlan0” a “00:11:22:33:44:55” (del 0 al 5, números dobles es algo fácil de recordar y rápido de escribir).



Y finalmente se logra que la tarjeta de red inalámbrica funcione en modo monitor ejecutando la orden:

airmon-ng start wlan0

Hasta entonces, se ha logrado colocar el dispositivo inalámbrico en modo monitor (lo cual es de suma importancia para continuar). Lo que se debe rescatar del proceso es el nombre final de la interfaz en modo monitor (en la práctica es “mon0”):

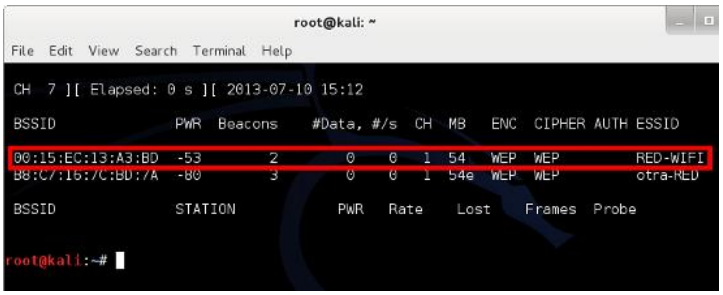


Empezando la auditoria en sí:

Llegado este momento hay que tomar las riendas y elegir la red WiFi con seguridad WEP más optima (esto es, mejor señal y tráfico de paquetes). Para ello se dispone en la terminal a ejecutar el comando:

airodump-ng mon0

Nótese el uso del nombre “mon0” en vez de “wlan0” ya que la interfaz “mon0” es la que se encuentra entonces en modo monitor. Esto arroja una lista de redes de la cual se hace la elección:
A modo de ejemplo, se usará la red Wifi “RED-WIFI” para auditarla. De lo que marca la imagen, se

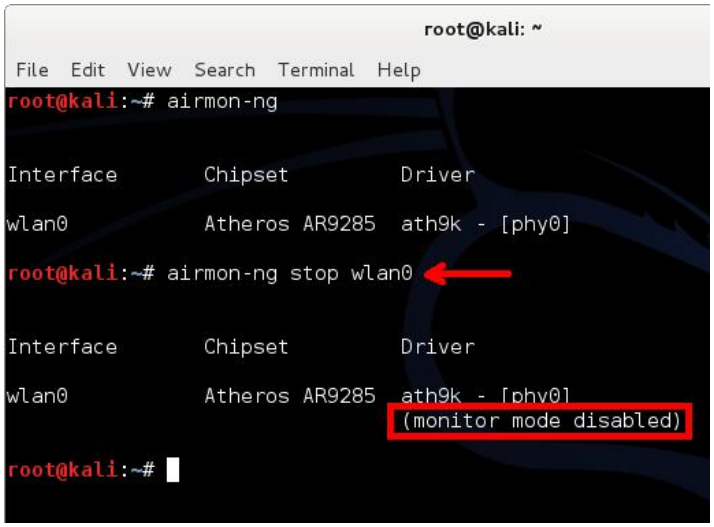


necesitan los siguientes datos acerca de la red: BSSID, CH y ESSID que en castellano sería: BSSID la dirección MAC del Router/Modem, CH el Canal por el cual se distribuye la señal de la red y el ESSID el nombre de la red. Se necesitan anotar estos datos ya que se usarán luego.

Iniciamos la captura de datos usando la siguiente orden:

airodump-ng -c 1 -w archivo --bssid 00:15:EC:13:A3:BD mon0

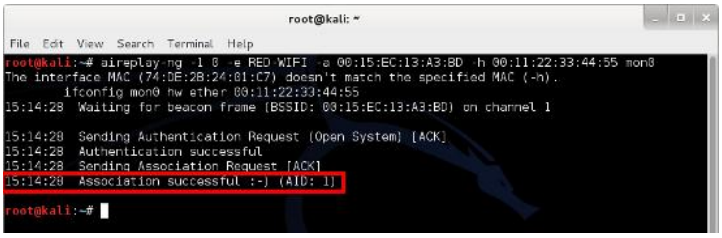
Esta orden es la principal y usa el programa “airodump-ng” para capturar paquetes del canal 1, de la red con MAC elegida en “bssid” guardando paquetes en el archivo llamado “archivo” (opción –w) usando la interfaz wireless “mon0” en modo monitor.



El dato más importante es “#Data” , esta terminal se debe dejar trabajando sin cerrarla, hay que ejecutar otra terminal aparte (separada de la anterior) y proseguir con la autenticación de la red WiFi. Esta autenticación se logra con el comando:

aireplay-ng -1 0 -e RED-WIFI -a 00:15:EC:13:A3:BD -h 00:11:22:33:44:55 mon0

El comando anterior, haciendo uso de “aireplay-ng” ejecuta una autenticación de la red de nombre “RED-WIFI” (opción –e) que usará la dirección MAC de destino (opción –a) y la asociará con la MAC de origen (opción –h) usando la interfaz “mon0” (la cual está habilitada en modo monitor).



Si al ejecutar la orden, no se llega a autentificar correctamente se debe ejecutar tantas veces sea necesario. ¿Cómo saber cuándo autenticó correctamente?, se logra ver en la marca roja la palabra “Successful” y una carita feliz a modo ASCII “:-) (AID: 1)” con eso se da por hecho la correcta autenticación.

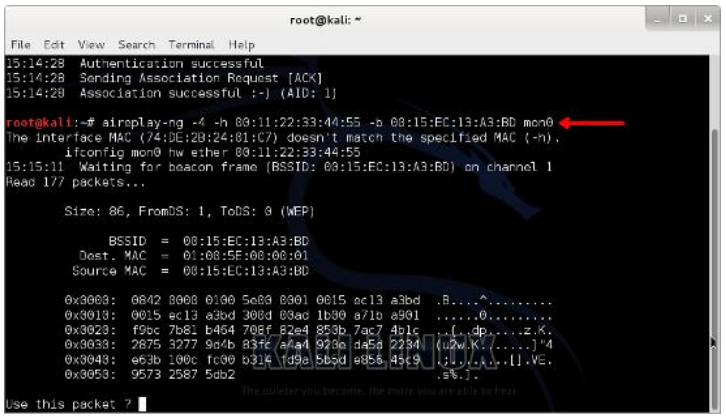
Creando el paquete Llave

Este paquete permite generar una cantidad finita de paquetes con el fin de que en alguno de ellos se retorne la contraseña de la red WiFi.

Para crearlo se debe ejecutar la instrucción:

aireplay-ng -4 –h 00:11:22:33:44:55 –b 00:15:EC:13:A3:BD mon0

Esta instrucción usa “aireplay-ng” para crear el paquete llave, asociando varias conexiones entre la dirección MAC de destino (opción –b) y la dirección MAC de origen (opción –h).

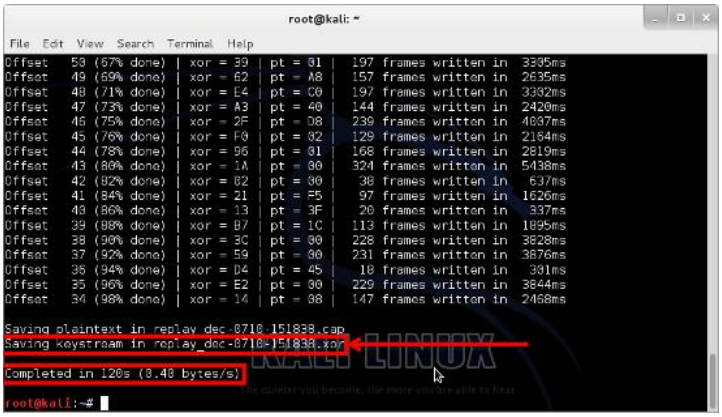


Cuando la terminal se queda parada en esa pregunta, se debe ingresar la letra “y” aceptándola con la tecla “ENTER” del teclado.

Este proceso tiene que completar el 100% de la tarea, si esto no es posible… debe usarse un segundo dispositivo con WiFi incorporado para realizar intentos de conexiones con contraseñas estándares, esto con el fin de estimular el uso de la decodificación. En la práctica se realizó esta tarea desde un Smartphone con Android usando contraseñas genéricas como:

12345
54321
abcde
a1b2c3
1a2b3
Etc..

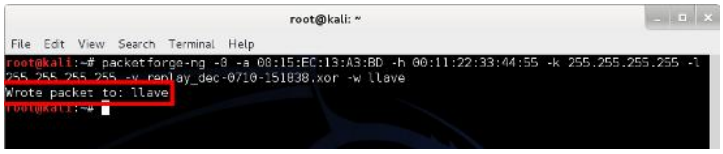
Si el proceso tuvo éxito se debe observar que los cambios se completaron y fueron escritos a un archivo de extensión XOR:



El archivo indicado con la flecha roja, se usará para generar el archivo llave, y para ello se necesita introducir el siguiente comando:

packetforge-ng -0 –a 00:15:EC:13:A3:BD –h 00:11:22:33:44:55 –k 255.255.255.255 –l 255.255.255.255 –y replay_dec-0710-151838.xor –w llave

Analizando el comando: se usa “packetforge-ng” para asociar los paquetes enviados desde una dirección MAC de origen (opción –h) hacia una dirección MAC de destino (opción –a) validando direcciones IP entre rangos lógicos (opción “-k” y “-l”) con el archivo antes generado “replay_dec-0710-151838.xor” guardando todo el trabajo en el archivo de nombre “llave”.



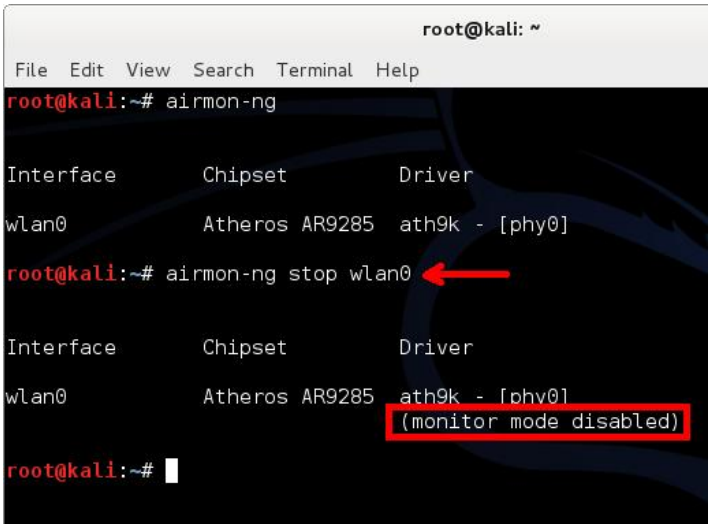
La marca roja indica que se escribieron los cambios en “llave”. Esto quiere decir que se han hecho las cosas bien.

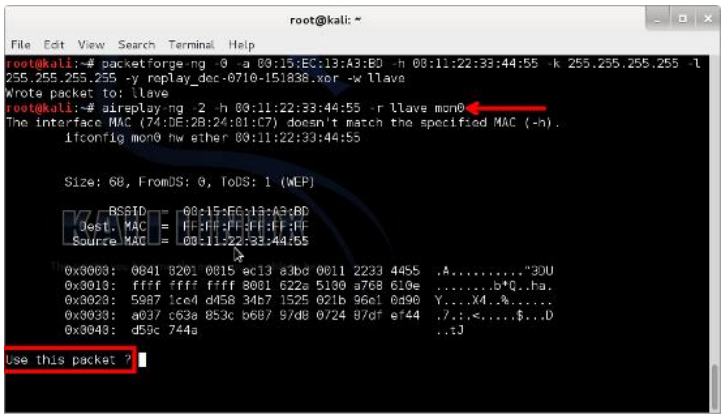
Generando Paquetes:

Para ir terminando, se empieza a generar paquetes válidos con la tarjeta de red inalámbrica del equipo en modo monitor hacia el dispositivo de destino (red WiFi). Para ello escribimos esto en la terminal:

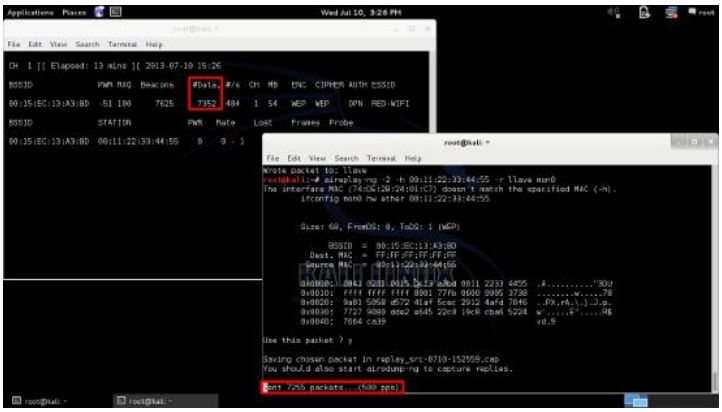
aireplay-ng -2 –h 00:11:22:33:44:55 –r llave mon0

Acá se usa “aireplay-ng” para emitir paquetes con la dirección MAC de origen (opción –h) usando el archivo llave creado anteriormente con la opción “-r”





Nos pregunta si usaremos el paquete llave o no, ingresamos “y” para responder “yes” (SI) y damos ENTER para que empiece a generar paquetes válidos:



Como se logra apreciar, la primera marca roja, indica la cantidad de paquetes capturados, se necesita un mínimo de 15.000 paquetes para empezar a descifrar la contraseña de la red WiFi.

En segundo lugar, la otra marca roja nos marca cuantos paquetes lleva generando. También se logra ver cómo funcionan ambas terminales a la par.

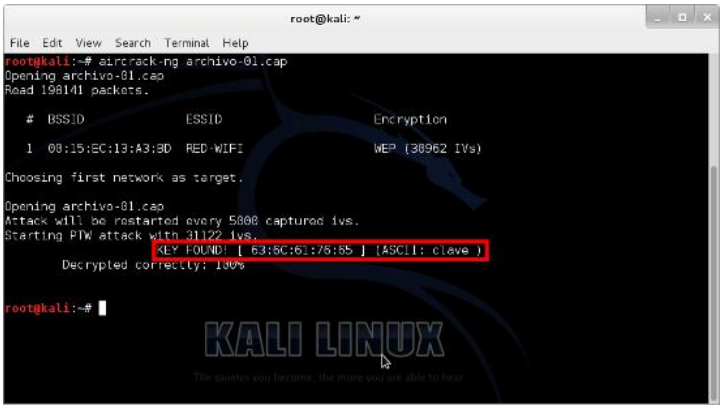
¡El toque final!

Para finalizar, debemos abrir una 3er terminal donde se ejecutará el último comando que nos dará la contraseña de la red WiFi. Estando en una nueva terminal, escribimos la orden:

aircrack-ng archivo-01.cap

Para descifrar la contraseña se usa “aircrack-ng” el cual escaneará el archivo en donde guardamos los datos de la 1ra terminal el cual se llamaba “archivo” por cuestiones de orden, automáticamente agrega

una numeración de forma ascendente “01, 02, 03, etc.” Por lo cual pasaría a llamarse “archivo-01.cap” que es donde está la información de la contraseña.

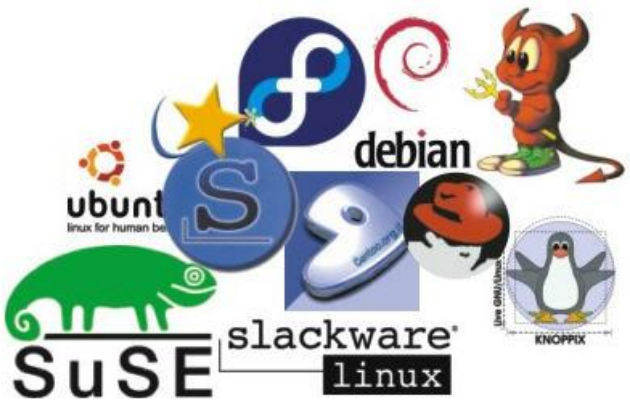


Y como se puede ver en la imagen, la terminal dice “KEY FOUND!” (Contraseña encontrada) la cual podemos usar para conectarnos a la red WiFi.

Los primeros 10 números (de a pares en 5 grupos) es el formato hexadecimal y el último (ASCII) es la contraseña común que se usa para conectarse, que en este caso por motivos prácticos la contraseña es “clave”.

Lo que hace el equipo cuando se conecta a la red WiFi, es transformar la cadena de 5 dígitos (clave) en su forma hexadecimal (63:6C:61:76:65), por ende, usar la contraseña hexadecimal o ASCII es exactamente lo mismo.

Lozano Juan Pablo
lozanotux@gmail.com
twitter: @lozanotux



¿Y como elegir una distribución? - Nuestros lectores escriben

POR **RAMON JARAMILLO**

¿Cuántas veces los principiantes del software libre se han enfrentado a este problema y luego regresan a Windows™?. Si eres de estas personas, este artículo puede ser para ti.

Ensayo y Error

En esta, mi primera colaboración para TuxInfo, traigo a propósito un problema poco tratado como es el de elegir un sistema operativo alterno, distribución o distro para una computadora. Que no se hable de ese tema puede ser ventajoso o no. Lo primero, implica que tanto las comunidades, como las individualidades que apoyan al software libre (y me cuento) no buscan influir a los principiantes con propagandas. La desventaja radica en que, habiendo más de 300 distribuciones basadas en diferentes núcleos (GNU/Linux, BSD, etc.) no hay una manera de elegir que parezca válida.

Cuando aprendí GNU/Linux solía quedarme con la distribución que veía en clases y la usaba en las PCs del aula y de mi casa y experimentaba con ella. Usé después, distros que veía en revistas, en Internet o en otros cursos, guiándome por la facilidad de uso...o por su aspecto vistoso. Al final, elegí Ubuntu, con sus fallas y virtudes. Pero, en el camino me quedé con discos que no botaré, para no contaminar más el planeta, de distros que deseché.

Como no me parece justo que otras personas pasen por las mismas situaciones que yo, se me ocurre que deberían ser establecidos parámetros para elegir una distribución y ahorrarnos tanto esfuerzo.

Criterios

El idioma

En la actualidad, el mundo de las distros, dominado de un lado por el software libre y del otro por el que no lo es tanto, está lleno de diferentes alternativas de programas en varios idiomas. Si tuviéramos una lista con todas las distros existentes, deberíamos buscar aquellas en español o con soporte multilingüe. Pero las que cumplen ese requisito deben ser casi tantas como 100.

Popularidad

Si una distribución es muy popular, se supone que es posible hallar soporte técnico al trabajar con ella. Para conocer cuáles son las distros más populares, existe una Web de la empresa Unsigned Integer Limited de Hong Kong llamada DistroWatch, que funciona como un “observatorio de distribuciones“ y que presenta en los ocho idiomas más populares del mundo, análisis con los detalles técnicos de las registradas. Al ingresar a su página en Internet http://www.distrowatch.com, a media página hallamos al lado derecho una lista de las 100 más populares, que se muestra con diferentes períodos de tiempo (una semana, 1, 3, 6 y 12 meses). De esta lista, propongo tomar las distros que ocupan las 10 primeras posiciones. Pero si haces clic en Distribuciones principales obtienes un resumen técnico con lo que desees saber de 11 de ellas. Todas estas distros son muy conocidas y tienen amplio soporte aunque no son totalmente libres bien porque su licencia no es compatible con la GPL o por sus componentes. Sin embargo, se siguen usando por su buen desempeño en diversas tarjetas madres.

Software 100% libre

Aunque ningún sistema operativo garantiza la privacidad total, hay quienes desean una distribución totalmente libre. Distrowatch y la Free Software Foundation, mencionan las pocas que cumplen ese criterio. Aunque es posible que no funcionen correctamente con dispositivos Wi-Fi y Bluetooth, probé la distro española Trisquel sin instalarla, sorprendiéndome de que detectara correctamente una impresora multifuncional, aunque su desempeño para ver videos en Youtube con el reproductor de HTML 5 no fuera muy bueno. También vi que

Firefox era sustituido por Abrowser, que para el caso, es casi la misma cosa. Es fácil ver, mediante las descripciones de Distrowatch, si estas distros son multilingües o están en español.

Instalación en PCs de empresas privadas

Institutos educativos

En las instituciones educativas privadas, es posible usar el criterio de popularidad o el del software 100% libre para elegir una distribución. De los dos criterios anteriores, podrían ser extraídas distros con interfaces parecidas a las de Windows (ya que muchos se acostumbraron a su aspecto), como los entornos de escritorio KDE y MATE. Luego, la distro debe ser instalada en algunas computadoras a las cuales tenga acceso el alumnado y el personal administrativo y docente del instituto, durante un período de prueba luego del cual se harán encuestas electrónicas o escritas para determinar cómo es la experiencia de uso con la distro elegida.

Otras empresas privadas

Las empresas privadas pequeñas se adaptan rápidamente a los cambios y favorecen el uso de las tecnologías alternativas y libres. En cambio, en las corporaciones, eso no es tan posible pues la alta gerencia y las juntas directivas están compuestas por personas que, desde hace años, piensan que el software producido por los gigantes de la informática es el único que realmente funciona. No pienso hacer que entren en razón tales personajes: es más productivo, recomendar otras alternativas. Las listas de Distrowatch tienen, al menos, 5 opciones de distribuciones empresariales. La primera que menciono es Solaris, desarrollado inicialmente por Sun Microsystems y hoy bajo control de ORACLE.

Esta empresa eliminó a su contraparte abierta, OpenSolaris y el Solaris actual tiene una licencia restrictiva, así que podemos olvidarla. Los 3 sistemas que siguen son Oracle Linux, Red Hat Enterprise Linux y SuSe, de los cuales los dos últimos se pueden instalar pero tienen un período de evaluación limitado, luego del cual hay que pagar el soporte y las actualizaciones...

Las dos últimas distros de la lista son OpenIndiana del Proyecto Illumos, de origen británico y construida a partir del desaparecido OpenSolaris y CentOS, de origen estadounidense. Éste último es el más usado por ser basado en GNU/Linux. A propósito de esto, me sorprendió saber que en un supermercado de la ciudad venezolana de Maracay, la Gerencia hizo instalar en sus cajas registradoras computarizadas este sistema operativo, aunque en modo de texto y sobre éste se “montó” un programa propietario de gestión de ventas al detal y devoluciones.

Conclusión

Ahora que has leído este artículo, espero que tengas las ganas de experimentar con las “distros”, aunque lo que he escrito no sea una guía que hay que seguir al pie de la letra. Sin embargo, más vale empezar por algo, ¿no te parece?. Pero antes de empezar, prepara tu disco duro y una memoria USB. Y si necesitas ayuda, en Internet tienes suficiente...

Ramón Jaramillo
Ingeniero Electrónico en Telecom
Cisco Certified Network Associate
En Twitter: @ramoningeniero.


TUX
INFO
WWW.TUXINFO.COM.AR