

Interfaz gráfica para nftables

José María Caballero Alba

29 de octubre de 2014

Índice

1. Descripción del problema.	3
2. Solución aportada	4
3. Introducción a nftables	5
4. Requisitos de instalación nftables.	5
4.1. Instalación y compilación kernel 3.13.X	5
4.2. Instalación utilidad nft	5
4.3. Prueba simple de nftables con la utilidad nft	6
4.4. ¿Utilidad nft o libnftnl?	7
5. Estudio de utilidades existentes	7
5.1. Entorno de pruebas	8
5.2. Configuración del servidor	8
5.3. Vuurmuur Firewall	9
5.3.1. Instalación de Vuurmuur Firewall	9
5.3.2. Primera toma de contacto	10
5.3.3. Estableciendo nuestras reglas	13
5.4. Fwbuilder	13
5.4.1. Instalación	13
5.4.2. Primera toma de contacto	14
5.5. Ipmenu	15
5.5.1. Instalación	15
5.6. Easy firewall generator	16
5.6.1. Instalación	16
5.6.2. Primera toma de contacto	16
5.7. Turtle firewall project	18
5.7.1. Instalación	18
6. Tabla de ventajas y inconvenientes	18

1. Descripción del problema.

Nftables es un nuevo framework que sustituye al antiguo iptables. Este nuevo software aun no esta desarrollado al 100 % pero ya es operativo en gran parte de sus funciones, tenemos la problemática de la implantación, que aunque se incluye desde el kernel 3.13, las personas aun no están acostumbradas a su uso y por tanto siguen usando iptables o en su defecto la nomenclatura de sus reglas. En pocas palabras:

- Esta disponible desde el kernel 3.13 en adelante
- Trae una la nueva utilidad nft con una sintaxis diferente a la de iptables.
- Tiene compatibilidad con las instrucciones de iptables.
- Infraestructura genérica de conjuntos que permite construir mapas entre asignaciones y acciones para mejorar las búsquedas.
- Aún esta bajo desarrollo.

Las diferencias con iptables son notables, estas serían las proporcionadas por su pagina oficial:

- Maquina de pseudo-estados en el espacio del kernel, nftables interpreta el mapa de reglas proporcionadas por el usuario (con la nueva sintaxis) , esta se compila y entra en la maquina de estados como bytecode y esta misma la transfiere al kernel por la api Netlink's de netfilter.
- La nueva sintaxis permite tener un conjunto de reglas, por ej: (traducir toda la linea)
- Reduce el total del código en el espacio del kernel. Se puede elegir que selectores de paquetes de todos los protocolos existentes puede usar la maquina de pseudo-estados, esto significa que no necesitamos una extensión en espacio de kernel para cada protocolo si queremos soportarlo. Esto supone una ventaja, ya que no necesitamos actualizar el kernel para obtener nuevas características y esto a sido diseñado para trabajar en el espacio lógico de usuario.
- Interfaz unificada para reemplazar las utilidades iptables/ip6tables/arptables/ebtables.

2. Solución aportada

La solución aporta consistirá en una interfaz gráfica escrita en c y usando ncurses para poder manejar nftables y que de esta manera sea mas fácil su uso.

Esto implica una mejora sustancial para aquellas personas que quieran dejar de la iptables y puedan utilizar nftables con todas la características nuevas y así poder sacar mas rendimiento a los sistemas de cortafuegos que usen en la actualidad. La interfaz gráfica deberá de proporcionar una implementación de la utilidad nft para su uso amigable y por tanto su curva de aprendizaje sera menor, esto incluye que pueda explorar todas las posibilidades de la utilidad nft, mejorar tiempos de configuración, etc.

Se usara c por el hecho de que TIOBE (empresa que mide la calidad del software) declara a C como el mejor lenguaje para programar (en su índice actual de 100 lenguajes, c, esta el primero en la lista), cabe destacar que TIOBE comprueba mas de 300 millones de lineas de código de sus clientes en el mundo entero, en tiempo real, cada día.

El de hecho de que TIOBE sea un buen indicador para decidir un lenguaje de programación incluye:

- Que esta basada en el estándar ISO 25010 sobre la calidad del software.
- Mide mas de 350 estandares de aspecto del lenguaje de manera automatizada.
- Da un resultado de 100 (nivel A, el mejor) a 0 (nivel F)
- Usa la metodología TIOBE Quality Indicator

En este caso C esta en el puesto numero uno en el top de TIOBE, siendo el segundo Java, esto indica que Java también podría haberse usado en este proyecto pero debemos de tener en cuenta todo este software esta escrito en C:

- El kernel de linux
- El repositorio Git
- Las bases de datos Mysql, PostgreSQL, SQLite (software libre)
- Las bases de datos Oracle, DB2, Informix, SYBASE (propietarias)
- Muchas de las características de Windows
- El servidor web Apache y nginx
- Los lenguajes Perl, PHP, Python y Lua

Esto nos hace ver que C es un lenguaje altamente usado y testeado para hacer aplicaciones criticas, sabiendo que es un lenguaje altamente maduro y sin fallos aparentes que puedan hacer fracasar el software por ser tecnología nueva o con poco uso y por tanto no testeada.

Se usara un router fisico soekris net550* para su implementación y simulación en entornos laborales.

3. Introducción a nftables

Como ya dijimos en la descripción del problema, nftables es un nuevo framework que sustituye al antiguo iptables para el filtrado de paquetes y clasificación de estos en Linux. Nftables es una combinación del núcleo de linux y una utilidad de espacio de usuario (lo que seria la antigua iptables, nft). Usa la infraestructura de Netfilter, como el connection tracking system.

nftables

4. Requisitos de instalación nftables.

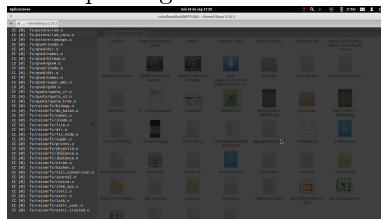
Nftables esta operativo desde el kernel 3.13, para poder usarlo necesitamos compilar este kernel o un kernel superior. Además de la utilidad nft para poder usar el framework la cual necesita las librerías libmnl (normalmente en los repositorios) y libnftnl.

4.1. Instalación y compilación kernel 3.13.X

Como hemos dicho en el apartado anterior, necesitamos instalar un kernel superior al 3.13, en este caso vamos a usar el ultimo kernel estable de kernel.org, siendo este a la fecha actual el 3.16.5

Por lo tanto, para la instalación debemos de:

```
wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.16.3.tar.xz
tar -xvJf linux-3.16.tar.xz
cd linux-3.16
make menuconfig*
make
sudo make modules_install install
sudo update-grub2
```



Una vez hecho esto, reiniciamos y ya tendríamos el kernel disponible en la selección de inicio de grub, para comprobar que esta todo correcto, basta con abrir un terminal y poner “uname -r” debería de salir que tenemos un kernel 3.16.3

4.2. Instalación utilidad nft

Para esto instalaremos las 2 librerías necesarias (libmnl y libnftnl), para ambas necesitamos hacer:

```

$ git clone git://git.netfilter.org/lib[nnl o fnl]
$ cd libnftnl
$ sh autogen.sh
$ ./configure
$ make
$ sudo make install
Ahora ya podemos instalar la utilidad nft:
$ git clone git://git.netfilter.org/nftables
$ cd nftables
$ sh autogen.sh
$ ./configure
$ make
$ make install
Y para comprobar su correcto funcionamiento tecleamos:
$ sudo nft
a al cual la terminal debería de responder:
nft: no command specified
confirmando así su correcta instalación

```

4.3. Prueba simple de nftables con la utilidad nft

Ej: denegar la salida (drop output) a una ip especifica:

1. Nos dirigimos al directorio donde clonamos nftables
2. Iniciamos para ipv4 de la siguiente forma
`sudo nft -f files/nftables/ipv4-filter`
3. Denegamos la salida a la ip
`sudo nft add rule ip filter output ip daddr 1.2.3.4 drop`
4. comprobamos que esta correcto:
`ping 1.2.3.4`

```

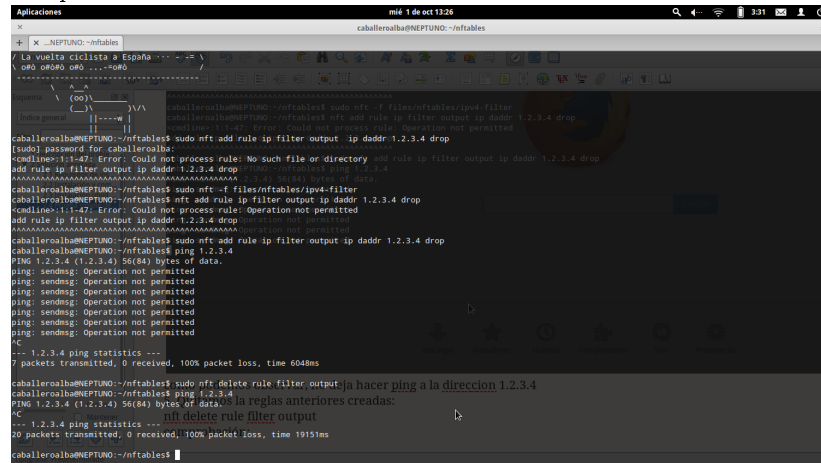
caballero@ubuntu:~$ sudo nft -f files/nftables/ipv4-filter
caballero@ubuntu:~$ sudo nft add rule ip filter output ip daddr 1.2.3.4 drop
caballero@ubuntu:~$ ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data:
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted

```

como podemos observar, no deja hacer ping a la direccion 1.2.3.4

5. Quitamos la reglas anteriores creadas:

sudo nft delete rule filter output
comprobación:



```
caballero@neptuno:~/nftables$ sudo nft add rule ip filter output ip daddr 1.2.3.4 drop
[sudo] password for caballero:
caballero@neptuno:~/nftables$ sudo nft add rule ip filter output ip daddr 1.2.3.4 drop
add rule ip filter output ip daddr 1.2.3.4 drop
caballero@neptuno:~/nftables$ sudo nft add rule ip filter output ip daddr 1.2.3.4 drop
add rule ip filter output ip daddr 1.2.3.4 drop
caballero@neptuno:~/nftables$ sudo nft add rule ip filter output ip daddr 1.2.3.4 drop
add rule ip filter output ip daddr 1.2.3.4 drop
caballero@neptuno:~/nftables$ sudo nft add rule ip filter output ip daddr 1.2.3.4 drop
add rule ip filter output ip daddr 1.2.3.4 drop
caballero@neptuno:~/nftables$ ping 1.2.3.4
PING 1.2.3.4 (0.2.3.4): 56(84) bytes of data:
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
-- 1.2.3.4 ping statistics --
7 packets transmitted, 0 received, 100% packet loss, time 604ms

caballero@neptuno:~/nftables$ sudo nft delete rule filter output
add rule ip filter output ip daddr 1.2.3.4 drop
caballero@neptuno:~/nftables$ ping 1.2.3.4
PING 1.2.3.4 (0.2.3.4): 56(84) bytes of data:
nft delete rule filter output
^C
-- 1.2.3.4 ping statistics --
10 packets transmitted, 0 received, 100% packet loss, time 1915ms

caballero@neptuno:~/nftables$
```

Una vez comprobado el correcto funcionamiento de la utilidad nft podemos indagar en el objetivo de este proyecto.

4.4. ¿Utilidad nft o libnftnl?

No podemos hacer una interfaz gráfica usando nft, si no que debemos usar la librería libnftnl para poder interactuar con nftables a través de libnmnl.

Preguntar esto:

Punto 2.5

Alcance del proyecto (nft o libnftnl, permitir todas las posibles combinaciones VISTA<—> nftables)

Distribución linux operativa en soekris con utilidad integrada en la distribución.

Uso de la librería libnftnl, utilidades definidas en ella para la comunicación

Interacción de bajo nivel (¿ensamblador?)

Escenarios de prueba (sistema chroot)

Problemas con el eclipse.

Libros recomendados

¿curses o ncurses?

5. Estudio de utilidades existentes

Debemos de tener en cuenta de que ya hay utilidades gráficas para iptables pero NO para nftables por cual podemos obtener ideas para la funcionalidad final.

Las utilidades a que vamos a analizar son 5:

1. Vuurmuur firewall

2. fwbuilder
3. Ipmenu
4. Easy Firewall Generator
5. Turtle Firewall Project

De todas estas utilidades deberemos de enumerar sus ventajas/desventajas, formas de instalación y funcionalidades que podamos desear en este proyecto

5.1. Entorno de pruebas

Para poder analizar todas la utilidades vamos a usar una maquina virtual basada en kvm con una instalación de Ubuntu Server en la cual instalaremos todo lo necesario para poder probar estas utilidades.

Características de la maquina:

- 1 CPU
- 512M de RAM
- 10GB de disco duro
- Ubuntu Server 14.04

ESPACIO PARA LA INSTALACIÓN DEL SERVIDOR y configuración del servidor

5.2. Configuración del servidor

Una vez instalado el servidor, necesitamos instalar las dependencias básicas para para poder compilar y instalar paquetes que vayamos a usar, no es mas que usar aptitude o apt-get con las siguientes instrucciones:

- sudo apt-get update
- sudo apt-get install build-essential automake make checkinstall dpkg patchutils autotools-dev debhelper quilt fakeroot xutils lintian cmake dh-make libtool autoconf git-core subversion libncurses5-dev

Esperados unos segundos ya tenemos las herramientas básicas para poder compilar los paquetes correctamente.

En el caso de Ubuntu Server, ya tienes instalado un servidor ssh por lo cual no es necesario instalarlo. Ya que no tenemos interfaz gráfica, nos conectaremos a nuestra maquina virtual por ssh. En este caso tenemos la ip fija 192.168.122.178


```

x
Archivo Editar Ver Borrar Terminal Ayuda
caballeroalba@NEPTUNO: ~
/ En casa extraña, no se junta las \
\ pestañas.
-----
      _ _ _
      \ (oo)\_____/
       (_____)  \/
        ||----w |
         ||     ||
caballeroalba@NEPTUNO:~$ ssh caballeroalba@192.168.122.187
^C
caballeroalba@NEPTUNO:~$ ssh caballeroalba@192.168.122.178
The authenticity of host '192.168.122.178 (192.168.122.178)' can't be establishe
d.
ECDSA key fingerprint is 8e:f1:9d:e3:58:83:63:c8:02:52:f2:d9:ae:54:5e:8c.
Are you sure you want to continue connecting (yes/no)? yes
```

5.3. Vuurmuur Firewall

Vuurmuur Firewall es una interfaz gráfica usando ncurses de iptables que nos permite mediante una interfaz sencilla poder configurar iptables de manera agradable y pudiendo llegar a reglas complejas, al usar ncurses es posibles administrarla de manera remota usando por ej ssh. Es software libre y esta bajo la licencia GNU/GPL

5.3.1. Intalación de Vuurmuur Firewall

Como siempre, nos descargamos el tar de la pagina oficial y procedemos a descomprimir y compilar

- mkdir vuurmuur
- cd vuurmuur
- wget ftp://ftp.vuurmuur.org/releases/stable/Vuurmuur-0.7.tar.gz
- tar xzf Vuurmuur*.tar.gz
- cd Vuurmuur – completar nombre carpeta
- sudo ./install –install

Y iremos aceptando las opciones por defecto de instalación.

```

x caballeroalba@pruebas: ~/vuurmuur/Vuurmuur-0.7
Archivo Editar Ver Buscar Terminal Ayuda
Configurando libserf-1.1:amd64 (1.3.3-1ubuntu0.1) ...
Configurando libsvn1:amd64 (1.8.8-1ubuntu3.1) ...
Configurando subversion (1.8.8-1ubuntu3.1) ...
Processing triggers for libc-bin (2.19-0ubuntu6) ...
caballeroalba@pruebas:~/vuurmuur/Vuurmuur-0.7$ sudo ./install.sh --install
svn: E155007: «/home/caballeroalba/vuurmuur/Vuurmuur-0.7/install.sh» no es una copia de trabajo

Vuurmuur installation
=====

Welcome to the installation of Vuurmuur. First you will be
asked a couple of questions about the location to install the
various parts of Vuurmuur. It is recommended that you choose
the defaults, by pressing just enter.

Please enter the installation dir (/usr).

Installdir: /usr ...

Please enter the directory where the config is going to be stored (/etc).

NOTE!!! in this directory a directory 'vuurmuur' will be created. This behaviour
has been changed in 0.5.65 (so for '/etc/vuurmuur' choose '/etc' here).

Examples: /etc, /usr/local/etc, /opt/vuurmuur/etc

Using Etcdir: '/etc/vuurmuur'.

Please enter the directory where Vuurmuur will store it's logs (/var/log/vuurmuur/).

Using Logdir: '/var/log/vuurmuur/'.

Ok, thank you. Going to build Vuurmuur now. Depending on your hardware
this process will take about 2 to 10 minutes.

Testing for the installation files...
Going to extract the files...
Extracting the files done...
Going to build libvuurmuur... (common code for all parts of Vuurmuur).
█

```

5.3.2. Primera toma de contacto

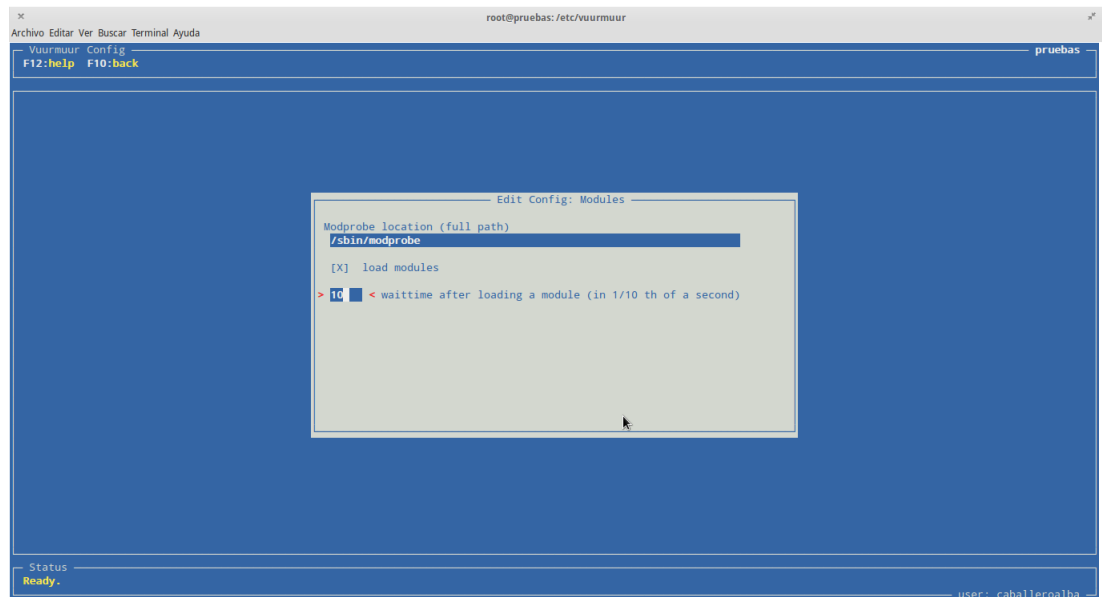
Para ejecutar vuurmuur simplemente:

```
sudo vuurmuur
```

En el caso de que nos salte el error:

Error: checking for iptables-capabilities failed. Please see error.log.

Debemos de ejecutar vuurmuur-conf , ir a vuurmuur config -> Modules y establecer 'waittime after loading a module' a 10



Además debemos de establecer reglas iniciales si queremos iniciar el demonio. Si en el menu principal pulsamos F5 para ver los detalles de los errores, no muestra que hay que configurar tanto las interfaces, zonas y es redes:

```
- No interfaces are active. Please make sure that at least one of
the interfaces is active (warn).

- No zones are defined. Please define one or more zones, and at
least one network (warn).

- No networks are defined. Please make sure that you define at
least one network. See the Zones Section (warn).
```

Por tanto, lo configuramos de la siguiente manera:

- Nos dirigimos al menú de interfaces.
- Pulsamos ins para para crear una nueva
- Añadimos el nombre de la interfaz, en este caso wlan0

Interfaces

<RET> edit

<INS> new

 remove

New Interface

Please enter the name

wlan0

Note: whitespaces not allowed.

Seguimos rellenando:

- Ponemos a activa
- La ip (podemos poner que es dinámica, como nuestra maquina kvm la tiene fija no es necesario)
- Y por ultimo indicamos el dispositivo, en este caso, wlan0.

Edit Interface

Name: wlan0

Active

Yes

Is interface up?

No

IP address

10.100.18.220

Dynamic IP Address

[]

Device

> wlan0 <

Una vez configurado esto ya podemos monitorizar este interfaz, por ejemplo, ver las conexiones activas pulsando en connections (c) en el menu principal:

2: dns	firewall(wlan0)	-> 150.214.186.69	ESTA
1: http	108.160.167.159	-> firewall(wlan0)	ESTA
1: https	firewall(wlan0)	-> 74.125.230.54	ESTA
1: 17500 ->...	firewall(wlan0)	-> 255.255.255.255	CONN
1: 17500 ->...	firewall(wlan0)	-> 10.100.255.255	CONN

5.3.3. Estableciendo nuestras reglas

Una vez que se ha configurado

5.4. Fwbuilder

Fwbuilder consiste en una GUI basada en gtk en la cual podemos configurar gráficamente iptables y exportar reglas. Intenta extraer al administrador de sistemas la tarea de hacerlo todo con la terminal, también soporta otros formatos de reglas a parte de iptables, por ej: ipfilter, ipfw, OpenBSD pf, Cisco PIX y FWSM.

5.4.1. Instalación

Fwbuilder tiendes paquetes .deb ya compilados, así que solo necesitamos usar dpkg:

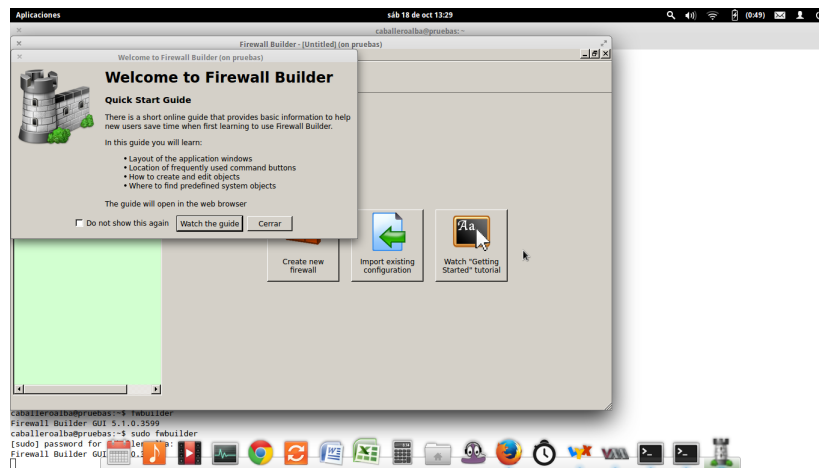
- `wget http://downloads.sourceforge.net/project/fwbuilder/Current_Packages/5.1.0/fwbuilder_5.1.0.3599ubuntu-precise-1_amd64.deb?r=http%3A%2F%2Fsourceforge.net%2Fprojects%2Ffwbuilder%2Ffiles%2Ffwbuilder_5.1.0.3599ubuntu-precise-1_amd64.deb`
- `sudo dpkg -i fwbu*`

```
Aplicacionesláb 18 de oct 13:12🔍 📶 🔋 100% ⌨️ 👤
```

```
caballero@ubuntu-pruebas:~/f-builder$  
Archivo Editar Ver Buscar Terminal Ayuda  
[ ] 43.446.104, 1, 23MB/s T.  
44% [=====] j 43.686.968, 1, 23MB/s T.  
44% [=====] j 43.910.912, 1, 22MB/s T.  
45% [=====] j 44.163.360, 1, 22MB/s T.  
45% [=====] j 44.424.952, 1, 22MB/s T.  
45% [=====] j 44.681.248, 1, 22MB/s T.  
45% [=====] j 44.975.688, 1, 22MB/s T.  
46% [=====] j 45.234.384, 1, 21MB/s T.  
46% [=====] j 45.521.584, 1, 21MB/s T.  
100% [=====] 97.834.618, 1, 29MB/s en 74s  
2014-10-18 13:10:02 (1,26 MB/s) ~ fbuilder_5.1.0.3599-ubuntu-precise-1_amd64.deb~http://httk3AKZP2Sourceforge.net/tz2ProjectsXZFfbuilderXZFfilesXZFCurrent_PackagesXZF5.1.0.3599-debian.tar.gz~gurdado [97834618/97834618]  
caballero@ubuntu-pruebas:~$ ls  
fbuilder_fbuilder_5.1.0.3599-ubuntu-precise-1_amd64.deb~http://httk3AKZP2Sourceforge.net/tz2ProjectsXZFfbuilderXZFfilesXZFCurrent_PackagesXZF5.1.0.3599-debian.tar.gz  
[1]- Hecho wget http://downloads.sourceforge.net/project/fbuilder/Current/Packages/5.1.0.fbuilder_5.1.0.3599-ubuntu-precise-1_amd64.deb~http://httk3AKZP2Sourceforge.net/tz2ProjectsXZFfbuilderXZFfilesXZFCurrent_PackagesXZF5.1.0.3599-debian.tar.gz  
[2]- Hecho mv fbuilder_5.1.0.3599-ubuntu-precise-1_amd64.deb ~/fbuilder/  
caballero@ubuntu-pruebas:~$ cd /home/caballero/fbuilder/  
fbuilder $ cd /home/caballero/fbuilder/  
caballero@fbuilder:~/fbuilder$ dpkg -i fw*  
[sudo] password for caballero:  
Selecciono el paquete fbuilder previamente no seleccionado.  
(leyendo la base de datos ... 65572 ficheros o directorios instalados actualmente.)  
Desempaquetando los paquetes...  
Unpacking fbuilder (5.1.0.3599-ubuntu-precise-1) ...
```

Una vez instalado y haciendo un export DISPLAY=:0 en nuestra sesión ssh y previamente habiéndonos conectado con el parametro -X podremos ejecutarlo graficamente

- ssh -X caballeroalba@192.168.122.189
- export DISPLAY=:0
- sudo fwbuilder

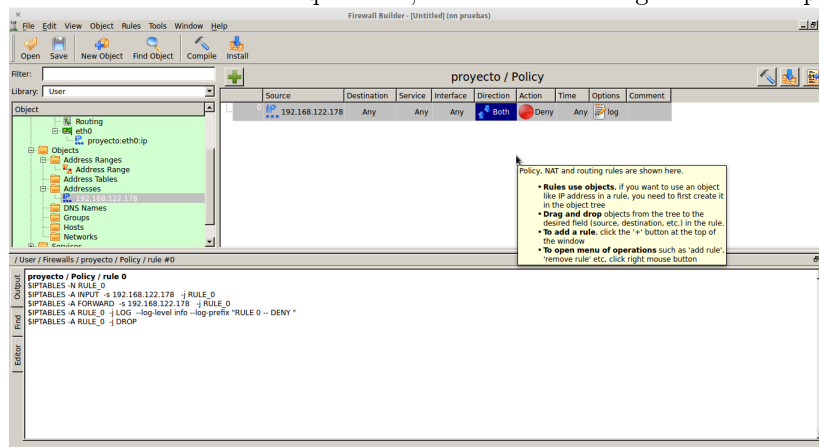


5.4.2. Primera toma de contacto

Una de las grandes ventajas de fwbuilder es que podemos usar direcciones ip (o rango de direcciones) como objetos que podemos “arrastrar” a los campos source y destination de la reglas de iptables. Un ejemplo sencillo de primera toma de contacto seria:

- Creamos un nuevo firewall con el nombre proyecto
- creamos una nueva interfaz (aunque se refiere a nuestras interfaces de red, no las usa en si, solo usa su nombre para después imprimir las reglas)
- Creamos la ip 192.168.122.178 (la de nuestra maquina kvm)
- Creamos una simple regla que no deje pasar ningún paquete
- Compilamos la regla

De esta manera en la consola que tiene, nos devolverá la reglas a usar en iptables



Basta entonces con simplemente usar estos comandos en nuestra consola ssh en la maquina virtual para comprobar que funciona (no deja pasar ningún paquete que tenga como source la ip 192.168.122.178)

- `sudo iptables -N RULE_0`
- `sudo iptables -A INPUT -i eth0 -s 192.168.122.178 -j RULE_0`
- `sudo iptables -A FORWARD -s 192.168.122.178 -j RULE_0`
- `sudo iptables -A RULE_0 -j LOG --log-level info --log-prefix "RULE 0 - DENY "`
- `sudo iptables -A RULE_0 -j DROP`

//completar

5.5. Ipmenu

Ipmenu es una interfaz de consola para el trafico de control de iptables esta escrito en cursel, con el se pueden editar reglas y configurar el cortafuegos para “marcar” los paquetes según la política de routing. Aunque puede parecer que la interfaz esta hecha en ncurses, es cursel el que da vida a este programa. Cursel es software libre que hace de interprete para formularios y menús (en sí es un lenguaje). Este interpreta descripciones de menús en archivos de texto simple escribiendo la GUI (menús, formularios, ficheros de texto) en el terminal para mas información sobre este lenguaje se puede visitar freecode.com/projects/cursel.

5.5.1. Instalación

Para instalar Ipmenu basta con bajarlo y ejecutarlo (no hace falta compilar) y por supuesto tener cursel instalado en el sistema. Por tanto no queda nada mas que bajar Ipmenu y instalar cursel

- `wget http://users.pandora.be/stes/ipmenu-0.0.3.tar.gz`
- `tar -xzf ipmenu-0.0.3.tar.gz`
- `wget http://freecode.com/urls/aff713c84c9e32111c9b907aba558774`
- `tar -xzf cursel-0.1-3`
- `cd cursel-0.1-3`
- `./configure`
- `make`
- `sudo make install`

//preguntar por error de compilación paquete cursel (esta en versión de debian woody hacia abajo, se elimino por error grave de seguridad en ipmenu con cursel, hacia referencia a bases de datos de seguridad, error 244709, pero sigue estando en la pagina de iptables como gui) o mejor usar otro programa.
//continuar aqui

5.6. Easy firewall generator

Easy firewall generator es un software que permite generar reglas de iptables, proporciona un rango de opciones, aunque según describe la pagina web del proyecto, no intenta abarcar todas la posibles opciones.

Easy firewall generator presenta la ideas generales presentadas en el tutorial Oskar Andreasson's iptables-tutorial

Consiste en un pequeño formulario web en el cual podemos introducir los datos y seleccionar las opciones deseadas, esto presenta un problema, y es que debemos de tener conexión a internet para poder usar esta herramienta.

5.6.1. Instalación

Como se trata de una pagina web con un formulario, no es necesaria instalación, basta con visitar la pagina del proyecto <http://easyfwgen.morizot.net/gen/>

5.6.2. Primera toma de contacto

Una vez en la pagina web nos encontraremos con lo siguiente:

Internet Interface: [Help](#)

Select Type of Internet Address [Help](#)

- ☐ Static Internet IP Address
- ☒ Dynamic Internet IP Address

Single System or Private Network Gateway? [Help](#)

- ☒ Single System
- ☐ Gateway/Firewall

☐ Allow Inbound Services [Help](#)

☐ Log entries in a Fireparse format? [Help](#)

☐ Do you use Internet Relay Chat (IRC)? [Help](#)

Esto nos genera un pequeño script el cual podemos ejecutar con las reglas proporcionadas

```
#!/bin/sh
#
# Generated iptables firewall script for the Linux 2.4 kernel
# Script generated by Easy Firewall Generator for IPTables 1.15
# copyright 2002 Timothy Scott Morizot
#
# Redhat chkconfig comments - firewall applied early,
#                               removed late
# chkconfig: 2345 08 92
# description: This script applies or removes iptables firewall r
#
# This generator is primarily designed for RedHat installations,
# although it should be adaptable for others.
#
# It can be executed with the typical start and stop arguments.
# If used with stop, it will stop after flushing the firewall.
# The save and restore arguments will save or restore the rules
# from the /etc/sysconfig/iptables file. The save and restore
# arguments are included to preserve compatibility with
# Redhat's or Fedora's init.d script if you prefer to use it.
```

y dentro del fichero, podemos ver algunas reglas generadas:

```
#####
#
# Flush Any Existing Rules or Chains
#

echo "Flushing Tables ..."

# Reset Default Policies
$IPT -P INPUT ACCEPT
$IPT -P FORWARD ACCEPT
$IPT -P OUTPUT ACCEPT
$IPT -t nat -P PREROUTING ACCEPT
$IPT -t nat -P POSTROUTING ACCEPT
$IPT -t nat -P OUTPUT ACCEPT
$IPT -t mangle -P PREROUTING ACCEPT
$IPT -t mangle -P OUTPUT ACCEPT

# Flush all rules
$IPT -F
$IPT -t nat -F
$IPT -t mangle -F

# Erase all non-default chains
$IPT -X
$IPT -t nat -X
```

Aunque solamente en el resto del fichero, solo hay reglas básicas, no deja muchas opciones para poder elegir (interfaz de red, ip dinámica o estática, sistema

único o gateway de la red).
// completar

5.7. Turtle firewall project

Turtle firewall project es un software que permite realizar un cortafuegos linux según “una manera rápida y simple” , consiste en una interfaz web proporcionada por webmin (aunque se puede editar con xml) que permite hacer las diferentes acciones. Esta escrito en perl.

Tiene las siguientes características:

- Definición de grupos, redes, hosts y zonas
- Reglas de filtrado basadas en servicios
- NAT
- Enmascaramiento

5.7.1. Instalación

La pagina del proyecto nos proporciona un .deb, así que la instalación se resume a:

- `wget http://downloads.sourceforge.net/project/turtlefirewall/turtlefirewall/1.38/turtlefirewall_1.38-1_all.deb?r=http%3A%2F%2Fsourceforge.net%2Fprojects%2Fturtlefirewall%2Ffiles%2Fturtlefirewall_1.38-1_all.deb`
- `sudo dpkg -i turtlefirewall_1.38-1_all.deb`

Es necesario tener instalado webmin, perl, libreria expat, modulos de netfilter ip_tables, ip_conntrack, reenvío de ip activado

6. Tabla de ventajas y inconvenientes

Software/Funcionalidad	vuurmuur	ipmenu	fwbuilder	easy firewall generator	turtlefirewall	Pu
Facilidad de uso						
Funciones avanzadas						
Antigüedad del proyecto						