FIREWALL BÁSICO

autor: sergio.gonzalez.duran@gmail.com

El siguiente es un script muy básico de iptables que puedes usar para proteger un solo equipo conectado a Internet a través de un modem o de una línea dedicada como adsl (algo parecido es lo que yo uso). Lo básico no es sinónimo de inseguro, de hecho este pequeño firewall es un excelente ejemplo de la potencia de iptables, el firewall de Linux, que con unas cuantas líneas es posible establecer un cortafuegos bastante seguro y eficaz.
Tan solo copia y pega lo siguiente en cualquier editor, guárdalo con el nombre que gustes, ejemplo: fw_equipo, después cambia sus permisos para que pueda ser ejecutado:

```
#> chmod 700 fw_equipo
```

y después ejecútalo: (tienes que ser root para ejecutarlo)

```
#> ./fw_equipo
```

Si no manda errores, listo, tu firewall esta protegiéndote de ataques y de accesos indeseados.

```
#
-------------------------------------------------------------------------------
-
# www.linuxtotal.com.mx
# firewall para un solo equipo conectado a traves de modem o adsl
# por: sergio.gonzalez.duran@gmail.com

# (1) se eliminan reglas previas que hubiera y cadenas definidas por el usuario
iptables -F
iptables -X

# (2) se establecen politicas "duras" por defecto, es decir solo lo que se
autorice
# explicitamente podra ingresar o salir del equipo
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# (3)a la interface lo (localhost) se le permite todo
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# (4) evitamos ataques syn-flood limitando el acceso de paquetes nuevos
# desde internet a solo 4 por segundo y los demas se descartan
iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 4 -j DROP

# (5) se evitan paquetes tcp que sean nuevos y que no tengan el flag SYN
# es decir, hay ataques o escaneos que llegan como conexiones nuevas
```

```
# pero sin ser paquetes syn, definitivamente no nos interesan
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

# (6) todo lo que sea icmp (ping) y que intente entrar, se descarta
# con esto bloqueamos cualquier tipo de paquetes con protocolo icmp
# evitando ataques como el del ping de la muerte, aunque esta regla
# podria provocar problemas de comunicacion con algunos ISP.
iptables -A INPUT -p icmp -j DROP

# (7) por ultimo las dos siguientes reglas permiten salir del equipo
# (output) conexiones nuevas que nosotros solicitamos, conexiones establecidas
# y conexiones relacionadas, y deja entrar (input) solo conexiones establecidas
# y relacionadas.
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
#
-------------------------------------------------------------------------------
-
```

Eso es todo. Aunque los comentarios (#) son bastante explícitos, para aquellos que se inician o quieren aprender mas de iptables, demos un repaso.

El paso (1) hace un Flush (borrado) de las reglas que ya hubiera, muchas distribucciones se instalan con un juego previo de reglas de iptables que para este caso no nos sirven, por eso las eliminamos, asi como la opción X que permite eliminar las cadenas de reglas personalizadas. Si quieres ver que reglas tienes actualmente en tu firewall usa la opción -L, o también -L -n que con esta opción 'n' muestra los puertos en formato numérico:

```
#> iptables -L
```

En el segundo paso (2) establecemos las políticas del firewall, hay dos tipos de políticas ACCEPT y DROP, en el primer caso, estariamos aceptando TODO lo que entre y salga del equipo y después se tendría que negar lo que no se quiera, cosa bastante tediosa e insegura. Lo mejor en firewalls es siempre establecer políticas DROP, con esto por default, todo, absolutamente todo es prohibido de entrar o salir, asi que tenemos que ir poniendo reglas que abran los puertos o conexiones que si queramos. La desventaja de crear un firewall con políticas DROP es que suelen ser mas complejos de crear y mantener, pero valen la pena, son por mucho más seguros.

Se establecen entonces para los paquetes que entran al equipo INPUT, para los paquetes que salen del equipo OUTPUT y para paquetes que atraviesan el equipo FORWARD, aunque esta última política en el caso de una PC con una sola tarjeta de red es innecesaria, o mas bien esta sobrando.

El paso (3) es necesario e importante ya que como estamos negando todo por default eso abarca también a nuestro dispositivo de red local o virtual, localhost, y varios servicios que trabajan de manera local en nuestro equipo como el sistema de ventanas X, hacen uso de este dispositivo para trabajar. Localhost no se conecta de ninguna manera al exterior a la LAN o Internet, por lo que es seguro simplemente decirle al firewall que acepte todo de entrada y salida que provenga de localhost.

Los pasos (4), (5) y (6) podrían parecer redundantes o inecesarios ya que como se observa su target es DROP lo que quiere decir que si el paquete se cumple en alguna de esas tres reglas se descartará. Realmente no sobran, añaden un extra de seguridad al firewall, ya que recordemos que las reglas son checadas contra los paquetes que entran o salen en un orden estrictamente secuencial a como las introducimos. Las dos últimas reglas permitirán el paso de paquetes (tal vez), asi que lo que hacemos es depurar los paquetes en los pasos (4), (5) y (6) para que llegan un poco mas controlados a la decisión del paso (7) si son aceptados o no.

Finalmente en el paso (7) se permiten un par de reglas que forman un firewall de estado completo, donde se analiza el estado de los paquetes, sin importar su protocolo, destino, etc. En la regla de INPUT es decir lo que entra al equipo desde Internet solo se permiten paquetes cuyo estado ya este registrado en la tabla de conexiones del kernel. Esto implica que ningún paquete que sea nuevo será permitido, solo se aceptará lo que previamente se haya solicitado desde nuestro equipo y que es precisamente la última regla, la de OUTPUT, donde establecemos que pueden salir paquetes nuevas, como una petición de página web, que son paquetes NEW, paquetes previamente establecidos (ESTABLISHED), como comunicaciones de chat y conexiones relacionadas a una establecida (RELATED), por ejemplo conexiones ftp que establecen conexión de control y de datos en los puertos 20 y 21.

Recuerda que este pequeño script es solo para un equipo que no ofrece ningún servicio al exterior, al Internet, por lo que es relativamente fácil con unas cuantas reglas configurarlo y dejarlo seguro, para un servidor Web por ejemplo, o un equipo Linux con funciones de NAT (que este ubicado entre una LAN e Internet) tendría que ser un firewall muy distinto al aqui presentado.

Listo, ojalá que este pequeño firewall te sea de utilidad y te ayude a comprender como funciona iptables.