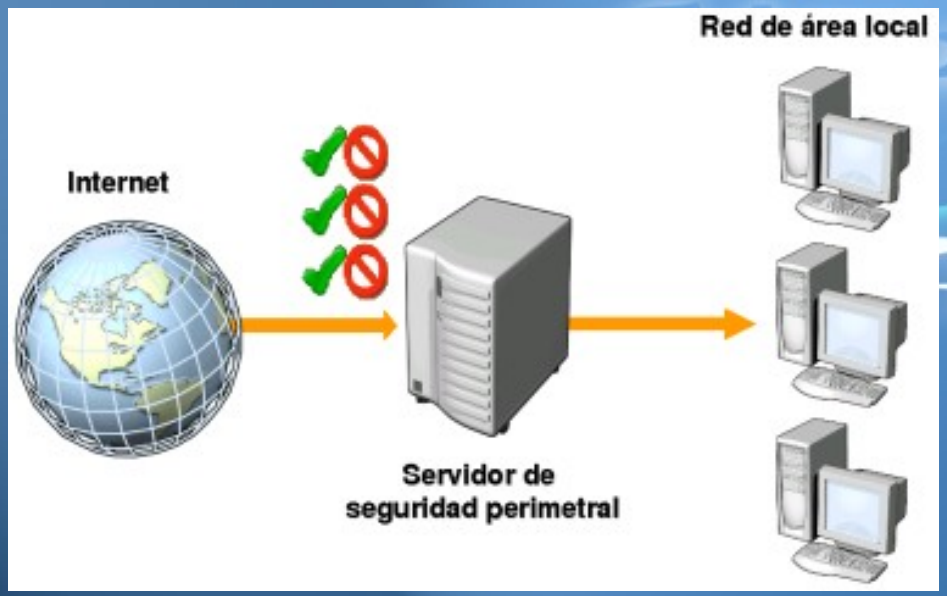
Servidor de Seguridad Perimetral

Por. Daniel Vazart P.

Introducción



- Es un equipo que se ubica entre la red de área local y de frente a Internet, haciendo las veces de router.
- actúa como Firewall.
- Utiliza varios filtros para aplicaciones especificas de Internet.

El Firewall: iptables

- Iptables es una aplicación en linea de comandos que gestiona el filtrado de paquetes en sistemas Linux, en base a las reglas que hayamos definido.
- La estructura de un comando iptables es la siguiente :

  iptables -t [tabla] -[ALFP] [regla] [criterio] -j [acción]

  Donde:

- **-t** es la tabla a modificar, que pueden ser **NAT** para las conexiones que serán modificadas por el firewall como el enmascaramiento de IP. **FILTER** que son las relaciones de los filtros que aplicará el firewall.

El Firewall: iptables

- **-ALFP** estas cuatro opciones básicas son para: **A** es para añadir una de las siguientes reglas **INPUT, FORDWARD** y **OUPUT**. **L** es para listar las reglas. **F** para borrar las reglas y **P** para establecer las reglas por defecto.
- **[criterio]** aquí es donde se especifica el tipo de paquete que sera restringido por esta regla del firewall:

  iptables -A FORWARD **-p [protocolo] -s [ip/red fuente] --sport [puerto fuente] -d [ip/red destino] --dport [puerto destino]** -j DROP

El Firewall: iptables

- **-ALFP** estas cuatro opciones básicas son para: **A** es para añadir una de las siguientes reglas INPUT, FORDWARD y OUPUT. **L** es para listar las reglas. **F** para borrar las reglas y **P** para establecer las reglas por defecto.
- **[criterio]** aquí es donde se especifica el tipo de paquete que sera restringido por esta regla del firewall:

  iptables -A FORWARD **-p [protocolo] -s [ip/red fuente] --sport [puerto fuente] -d [ip/red destino] --dport [puerto destino]** -j DROP

El Firewall: iptables

- **-j** aquí se define que es lo que se va hacer con el paquete, las opciones son:
  ACCEPT aceptara el paquete. **DROP** o **REJECT** rechazará el paquete. **REDIRECT** lo redirige hacia donde indique el criterio y por ultimo **LOG** lo registrará para analizarlo mas tarde.

Ejemplo de una regla sencilla

```
ono:~# ping localhost
PING localhost.localdomain (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=1 ttl=64 time=0.054 ms
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=4 ttl=64 time=0.150 ms

--- localhost.localdomain ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.041/0.072/0.150/0.045 ms
ono:~# iptables -A INPUT -p icmp -j DROP
ono:~# ping localhost
PING localhost.localdomain (127.0.0.1) 56(84) bytes of data.

--- localhost.localdomain ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 1999ms

ono:~#
```

Fabricando el firewall

- Ya que las reglas del firewall son simples comandos en Linux, es necesario crear un script que se ejecute cada vez que se enciende la maquina.
- Se sobre entiende que un firewall es un servidor 24/7, por eso cada vez que se reinicia el servidor las reglas se resetean.
- En el momento que se crea el script, es necesario eliminar cualquier tipo de basura o de reglas antiguas que pueden crear conflictos, eso se logra con la opción -F:

```
ono:~# iptables -F
ono:~# iptables -t nat -F
ono:~#
```

Fabricando el firewall

- Luego debemos establecer las políticas por defecto de nuestro firewall, para eso utilizamos la opción -P. Tenemos 2 tipos de políticas:
- La restrictiva (todo será negado): **iptables -P (INPUT FORWARD OUTPUT) DROP**
- **La permisiva (todo será aceptado): iptables -P (INPUT FORWARD OUTPUT) ACCEPT**

```
ono:~# iptables -P INPUT DROP
ono:~# iptables -P FORWARD DROP
ono:~# iptables -P OUTPUT ACCEPT
ono:~#
```

Reglas Utiles

- En este punto podemos comenzar a establecer las reglas del firewall según nuestras necesidades, a continuación una lista de reglas útiles:
- Cerrar conexiones entrantes desde eth0 y hacia un puerto (local)

  iptables -A INPUT -p tcp -i eth0 --dport <puerto> -j DROP
- Redireccionar al puerto 3128 (proxy) todos los paquetes que entran por eth1 y con destino puerto 80 (HTTP), de esta manera conseguimos un proxy transparente.

  iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128

NAT – IP fordward

- En primer lugar, es importante habilitar la opción del kernel que permite el fordward de IP para que NAT funcione correctamente:

  echo 1 > /proc/sys/net/ipv4/ip_forward

- Aquí esta la regla del firewall para hacer el NAT:

  iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth1 -j MASQUERADE

Ejemplo de un script terminado

```
ono:~# nano firewall
```

```
 GNU nano 1.2.4                    Fichero: firewall

# Eliminar las reglas anteriores
iptables -F
iptables -t nat -F

# Politicas por defecto

iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Aceptar todo el trafico por la interface local
iptables -A INPUT -i lo -p all -j ACCEPT

# Rechazar paquetes fragmentados
iptables -A INPUT -p udp -f -j DROP
iptables -A INPUT -p tcp -f -j DROP

# Prevencion de ataques DoS
iptables -A INPUT -p icmp -j DROP
iptables -A INPUT -p igmp -j DROP
iptables -A INPUT -m state --state INVALID -p tcp -j DROP

# Aceptar DNS, TALK y NTALK
iptables -A INPUT -i eth0 -m multiport -p udp --sport 53,517,518 -j ACCEPT
iptables -A INPUT -i eth0 -m multiport -p udp --dport 53,517,518 -j ACCEPT

# Evitar spoofing
iptables -A INPUT -i eth0 -p all -s 192.168.1.0/24 -j REJECT --reject-with icmp-ho$
iptables -A INPUT -i eth0 -p all -s 127.0.0.0/8 -j REJECT --reject-with icmp-host-$

# Con esta regla pongo a volar a los escaneadores de puertos :)
iptables -A INPUT -m multiport -p tcp -m state --state NEW,ESTABLISHED --dport 25,$
                          [ 48 líneas leídas ]
^G Ver ayuda ^O Guardar   ^R L Fichero ^Y Pág Ant   ^K CortarTxt ^C Pos act
^X Salir     ^J Justificar^W Buscar    ^V Pág Sig   ^U PegarTxt  ^T Ortografía
```

Ejemplo de un script terminado

```
ono:~# chmod 700 firewall
ono:~# ./firewall
ono:~#
```