Firewall con iptables.

Angel Alonso Párrizas

1.-Reglas de firewalling.

#ACTIVAMOS EL IP_FORWARD que nos pemirte hacer routing (NAT) a las otras máquina de la #red.

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

#ACTIVAMOS EL TCP_SYNCOOKIES para evitar ataques TCP SYN que nos desbore la #memoría

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

LIMPIAMOS LA LISTA DE REGLAS
/usr/sbin/iptables --flush
/usr/sbin/iptables -t nat --flush

ACTIVAMOS EL NAT PARA LA RED EXTERNA (eth0)
/usr/sbin/iptables -t nat --append POSTROUTING --out-interface eth0 -j MASQUERADE

ACEPTAMOS TODOS LOS PAQUETES CON ORIGEN LA RED LOCAL (eth1)
/usr/sbin/iptables -t filter -A FORWARD --in-interface eth1 -j ACCEPT
/usr/sbin/iptables -t filter -A INPUT -p tcp -s 192.168.0.0/24 -d 0/0 -j ACCEPT

#HACEMOS ROUTING para el puerto del messenger, para poder enviar y recibir ficheros. El #rango de puertos para el messenger va del 6891 en adelante. Ponemos hasta al 6899 como máximo #porque no se dará el caso de envíar más de 10 fichero a la vez.

/usr/sbin/iptables -t filter -A INPUT -p tcp -s 0/0 -d 0/0 --dport 6891:6899 -j ACCEPT
/usr/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6891:6899 -j DNAT –to destination 192.168.0.2

#Aquí viene lo bueno. Dado que mi servidor tiene FTP, SSH y HTTP corriendo hacia el exterior y #SMTP corriendo para la LAN, lo que hago es logear todo los accesos o intentos de acceso a esos #puertos (ejemplo scan de puertos que normalmente son paquetes TCP). Pondremos una etiqueta #que nos identifique el intento de acceso, en nuestro caso Hax0rs
#para eso lo único que hay que hacer es poner en la regla **-j LOG -log-prefix.** También logeo los #intenos de acceso al puerto 139 por ser los jeikerman sensibles a usar los recursos compartidos #para jeikar máquinas.

/usr/sbin/iptables -A INPUT -p tcp --dport 21 -j LOG –log-prefix Hax0rs
/usr/sbin/iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix Hax0rs
/usr/sbin/iptables -A INPUT -p tcp --dport 80 -j LOG --log-prefix Hax0rs
/usr/sbin/iptables -A INPUT -p tcp --dport 25 -j LOG --log-prefix Hax0rs
/usr/sbin/iptables -A INPUT -p tcp --dport 139 -j LOG --log-prefix Hax0rs

#Con estas reglas filtro absolutamente todo que no sean servicios que quiera dar.

#Del puerto 1 al 20 todo cerrado.
/usr/sbin/iptables -t filter -A INPUT -p tcp -s 0/0 -d 0/0 --dport 1:20 -i eth0 -j DROP

#dejo abierto al 21,22 abierto para poder hacer ftp y ssh,, 23 y 24 como los servicios están cerrados
#no hay problema.Todos cerrados con la siguiente regla hasta el 80 (hhtp)

/usr/sbin/iptables -t filter -A INPUT -p tcp -s 0/0 -d 0/0 --dport 26:79 -i eth0 -j DROP

#cierro desde el 80 al 1025. Estos puertos los cierros porque son los más atacados.
/usr/sbin/iptables -t filter -A INPUT -p tcp -s 0/0 -d 0/0 --dport 81:1025 -i eth0 -j DROP

#ahora dejo abiertos puertos como el 6667 para IRC, y y los 4xxx porque los usa el mldonkey.

#de ahí para arriba todos chapados
/usr/sbin/iptables -t filter -A INPUT -p tcp -s 0/0 -d 0/0 --dport 6900:5000 -i eth0 -j DROP

2.-Creando script para logear en tiempo real. (mon.sh)

Este script cortesía de Laco, nos permite en tiempo real ver los accesos y los intentos de acceso sobre nuestra máquina. Se debe tener instalado el grc.tar.gz que colorea las ip's y lahora y tal para hacerlo más visual.

#!/bin/bash
grc tail -f /var/log/messages /var/log/syslog /var/log/apache2/access_log /var/log/apache2/error_log /var/log/proftpd.log /var/log/xferlog

Es en syslog donde se nos almacenano los intentos de juanking anteriormente logeados con iptables, nos quedaría algo como esto

Dec 15 21:07:57 cocos kernel: Hax0rsIN=eth0 OUT=
MAC=00:50:fc:5a:c8:d8:00:08:e2:30:d8:a8:08:00 **SRC=80.33.169.191** DST=63.43.98.167
LEN=60 TOS=0x00 PREC=0x00 TTL=50 ID=33259 DF PROTO=TCP SPT=44943 **DPT=139**
WINDOW=5840 RES=0x00 SYN URGP=0
Dec 15 21:07:57 cocos kernel: Hax0rsIN=eth0 OUT=
MAC=00:50:fc:5a:c8:d8:00:08:e2:30:d8:a8:08:00 **SRC=80.33.169.191** DST=63.43.98.167
LEN=60 TOS=0x00 PREC=0x00 TTL=50 ID=3945 DF PROTO=TCP SPT=44948 **DPT=139**
WINDOW=5840 RES=0x00 SYN URGP=0

3.-Automatizando el envio de correo con el log del intento de juanking.

Yo lo que he hecho ha sido un script llamado **logear** (falta mejorarlo) que se ejecuta cada día mediante crontab.

El script es tan sencillo como esto:

root@cocos:~#cat /bin/logear
#!/bin/bash
tail -300 /var/log/syslog | grep Hax0rsIN | mail -s "Security mail from cocos" parrizas@wanadoo.es

Ahora solo creamos un fichero para root con la frecuencia de envio del fichero al mail
ej:cron

28 0 * * * /bin/logear

y ahora
root@cocos:~# crontab cron