

Seguridad en Asterisk-es-RSP sobre CentOS 5.5

El presente documento se muestra únicamente como guía y punto de inicio para implementar la seguridad en un sistema Linux y en modo alguno se establece como norma obligatoria el seguir los pasos que aquí comentamos. Constituye simplemente una guía de prácticas recomendables que permite a los administradores disfrutar de cierta tranquilidad en los sistemas que gestionan.

1. Desactivación de servicios instalados por defecto

El sistema operativo CentOS activa en su instalación por defecto varios servicios que, de no utilizarse, podrían ser aprovechados por atacantes externos para obtener acceso al equipo. Por lo tanto, desactivaremos aquellos servicios que no sean imprescindibles para el funcionamiento del sistema:

La lista de servicios activados en el arranque en una distribución CentOS por defecto con Asterisk-es-RSP y FreePBX con tarjetería Sangoma (servicio wanrouter), servidor Apache (paquete httpd), servidor mysql (paquete mysql-server), servidor dhcp (paquete dhcp), servidor ntp (paquete ntp) y servidor tftp (paquete tftp-server) instalados serían los siguientes (marcados en amarillo):

```
[root@centralita ssh]# chkconfig --list
NetworkManager 0:off 1:off 2:off 3:off 4:off 5:off 6:off
acpid           0:off 1:off 2:on 3:on 4:on 5:on 6:off
anacron         0:off 1:off 2:on 3:on 4:on 5:on 6:off
apmd            0:off 1:off 2:on 3:on 4:on 5:on 6:off
asterisk       0:off 1:off 2:on 3:on 4:on 5:on 6:off
atd            0:off 1:off 2:off 3:on 4:on 5:on 6:off
auditd         0:off 1:off 2:on 3:on 4:on 5:on 6:off
autofs         0:off 1:off 2:off 3:on 4:on 5:on 6:off
avahi-daemon   0:off 1:off 2:off 3:on 4:on 5:on 6:off
avahi-dnssconfd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
bluetooth      0:off 1:off 2:on 3:on 4:on 5:on 6:off
capi           0:off 1:off 2:off 3:off 4:off 5:off 6:off
conman         0:off 1:off 2:off 3:off 4:off 5:off 6:off
cpuspeed      0:off 1:on 2:on 3:on 4:on 5:on 6:off
crond         0:off 1:off 2:on 3:on 4:on 5:on 6:off
cups          0:off 1:off 2:on 3:on 4:on 5:on 6:off
dahdi         0:off 1:off 2:on 3:on 4:on 5:on 6:off
dc_client     0:off 1:off 2:off 3:off 4:off 5:off 6:off
dc_server     0:off 1:off 2:off 3:off 4:off 5:off 6:off
dhcpd       0:off 1:off 2:on 3:on 4:on 5:on 6:off
dhcrelay      0:off 1:off 2:off 3:off 4:off 5:off 6:off
dnsmasq       0:off 1:off 2:off 3:off 4:off 5:off 6:off
dund          0:off 1:off 2:off 3:off 4:off 5:off 6:off
firstboot     0:off 1:off 2:off 3:on 4:off 5:on 6:off
gpm           0:off 1:off 2:on 3:on 4:on 5:on 6:off
haldaemon    0:off 1:off 2:off 3:on 4:on 5:on 6:off
hidd         0:off 1:off 2:on 3:on 4:on 5:on 6:off
httpd       0:off 1:off 2:on 3:on 4:on 5:on 6:off
ibmasm       0:off 1:off 2:off 3:off 4:off 5:off 6:off
ip6tables    0:off 1:off 2:on 3:on 4:on 5:on 6:off
iptables     0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

irda	0:off	1:off	2:off	3:off	4:off	5:off	6:off
irqbalance	0:off	1:off	2:on	3:on	4:on	5:on	6:off
isdn	0:off	1:off	2:on	3:on	4:on	5:on	6:off
kudzu	0:off	1:off	2:off	3:on	4:on	5:on	6:off
lm_sensors	0:off	1:off	2:on	3:on	4:on	5:on	6:off
lvm2-monitor	0:off	1:on	2:on	3:on	4:on	5:on	6:off
mcstrans	0:off	1:off	2:on	3:on	4:on	5:on	6:off
mdmonitor	0:off	1:off	2:on	3:on	4:on	5:on	6:off
mdmpd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
messagebus	0:off	1:off	2:off	3:on	4:on	5:on	6:off
microcode_ctl	0:off	1:off	2:on	3:on	4:on	5:on	6:off
multipathd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
mysqld	0:off	1:off	2:on	3:on	4:on	5:on	6:off
netconsole	0:off	1:off	2:off	3:off	4:off	5:off	6:off
netfs	0:off	1:off	2:off	3:on	4:on	5:on	6:off
netplugd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
network	0:off	1:off	2:on	3:on	4:on	5:on	6:off
nfs	0:off	1:off	2:off	3:off	4:off	5:off	6:off
nfslock	0:off	1:off	2:off	3:on	4:on	5:on	6:off
nscd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
ntpd	0:off	1:off	2:on	3:on	4:on	5:on	6:off
odddjobd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
pand	0:off	1:off	2:off	3:off	4:off	5:off	6:off
pcscd	0:off	1:off	2:on	3:on	4:on	5:on	6:off
portmap	0:off	1:off	2:off	3:on	4:on	5:on	6:off
psacct	0:off	1:off	2:off	3:off	4:off	5:off	6:off
rawdevices	0:off	1:off	2:off	3:on	4:on	5:on	6:off
rdisc	0:off	1:off	2:off	3:off	4:off	5:off	6:off
readahead_early	0:off	1:off	2:on	3:on	4:on	5:on	6:off
readahead_later	0:off	1:off	2:off	3:off	4:off	5:on	6:off
restorecond	0:off	1:off	2:on	3:on	4:on	5:on	6:off
rpcgssd	0:off	1:off	2:off	3:on	4:on	5:on	6:off
rpcidmapd	0:off	1:off	2:off	3:on	4:on	5:on	6:off
rpcsvcgssd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
saslauthd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
sendmail	0:off	1:off	2:on	3:on	4:on	5:on	6:off
smartd	0:off	1:off	2:on	3:on	4:on	5:on	6:off
snmpd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
snmptrapd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
sshd	0:off	1:off	2:on	3:on	4:on	5:on	6:off
syslog	0:off	1:off	2:on	3:on	4:on	5:on	6:off
tcsd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
wanrouter	0:off	1:off	2:on	3:on	4:on	5:on	6:off
wpa_supplicant	0:off	1:off	2:off	3:off	4:off	5:off	6:off
xinetd	0:off	1:off	2:off	3:on	4:on	5:on	6:off
ypbind	0:off	1:off	2:off	3:off	4:off	5:off	6:off
yum-updatesd	0:off	1:off	2:on	3:on	4:on	5:on	6:off

xinetd based services:

```

    chargen-dgram:  off
    chargen-stream: off
    daytime-dgram:  off
    daytime-stream: off
    discard-dgram:  off
    discard-stream: off
    echo-dgram:     off
    echo-stream:    off
    eklogin:        off
    ekrb5-telnet:   off
    gssftp:         off
    klogin:         off

```

```
krb5-telnet:    off
kshell:         off
rsync:          off
tcpmux-server: off
tftp:          on
time-dgram:    off
time-stream:   off
```

Podríamos desactivar algunos de los servicios que se encuentran activos por defecto (siempre y cuando no los requiramos expresamente para alguna función o característica del sistema) como por ejemplo:

- autofs: servicio de automontaje de sistemas de ficheros
- avahi-daemon: implementa el protocolo ZeroConf compatible con Rendezvous, Bonjour de Apple para la configuración automática de servicios de red
- bluetooth: gestiona las comunicaciones con dispositivos bluetooth
- cups: controla el servicio de impresión unificada CUPS
- hidd: servidor de gestión bluetooth H.I.D.
- mcstrans: gestiona el funcionamiento del servicio SELinux (el cual desactivamos al instalar Asterisk-es-RSP)
- netfs y nfslock: controlan el servicio de ficheros NFS.
- pcsd: implementa el reconocimiento automático de lectores de tarjetas de inteligentes.
- portmap: controla las comunicaciones RPC del sistema
- rawdevices: controla la gestión de dispositivos de acceso a dispositivos raw.
- restorecond: gestiona el funcionamiento del servicio SELinux (el cual desactivamos al instalar Asterisk-es-RSP),
- rpcgssd: controla el funcionamiento del servicio NFS (paquete nfs-utils).
- rpcidmapd: controla el funcionamiento del servicio NFS.

Para deshabilitar los servicios, ejecutamos como superusuario los comandos siguientes:

```
chkconfig autofs off
chkconfig avahi-daemon off
chkconfig bluetooth off
chkconfig cups off
chkconfig hidd off
chkconfig mcstrans off
chkconfig netfs off
chkconfig nfslock off
chkconfig pcsd off
chkconfig portmap off
chkconfig rawdevices off
chkconfig restorecond off
chkconfig rpcgssd off
chkconfig rpcidmapd off
```

y detenemos los servicios sin reiniciar el sistema:

```
service autofs stop
service avahi-daemon stop
service bluetooth stop
service cups stop
service hidd stop
```

```
service mcstrans stop
service netfs stop
service nfslock stop
service pcsd stop
service portmap stop
service rawdevices stop
service restorecond stop
service rpcgssd stop
service rpcidmapd stop
```

2. Yum-updatesd. Demonio de Gestión de actualizaciones

El demonio de gestión de actualizaciones, `yum-updatesd`, se comunica con los repositorios de CentOS para comprobar el estado de actualización del sistema. Modificaremos la configuración por defecto para ser informados a través de correo electrónico de la disponibilidad de nuevas actualizaciones para tomar medidas al respecto siempre y cuando lo consideremos necesario.

Por defecto, `yum-updatesd` comprueba las actualizaciones cada hora (sin aplicarlas) y se comunica a través de `dbus` para notificar la disponibilidad de actualizaciones. Editaremos el fichero `/etc/yum/yum-updatesd.conf` para realizar los cambios necesarios (en negrita):

```
[root@centralita ~]# vim /etc/yum/yum-updatesd.conf
[main]
# Cambiamos a 12 horas entre comprobaciones de actualización
run_interval = 43200
# how often to allow checking on request (in seconds)
updaterefresh = 600

# Seleccionamos email como método de notificación
emit_via = email
# should we listen via dbus to give out update information/check for
# new updates
# Evitamos la escucha de dbus
dbus_listener = no

# Enviamos las notificaciones por correo electrónico
# Destinatario de correo de notificación
email_to = direccion_destinatario@domino.com
# Remitente del correo de notificación
email_from = direccion_remitente@dominio.com
# Servidor de correo SMTP
smtp_server = IP_SERVIDOR_SMTP

# automatically install updates
do_update = no
# automatically download updates
do_download = no
# automatically download deps of updates
do_download_deps = no
```

y reiniciamos el servicio:

```
service yum-updatesd restart
```

3. SSH. Protección de accesos

Restringir el acceso al sistema mediante el protocolo SSH es uno de los pasos principales para preservar la seguridad de nuestro sistema Asterisk. En primer lugar debemos crear en el sistema un nuevo usuario sin privilegios de administración. Para ello, introducimos lo siguiente en la consola local del sistema como superusuario del sistema:

```
useradd setup
```

Añadimos además al usuario al grupo `wheel`. De esta manera, únicamente los miembros de este grupo tendrán la posibilidad de convertirse en superusuario del sistema mediante el comando `su`:

```
usermod -G wheel setup
```

Por último, modificamos el fichero `/etc/pam.d/su` para habilitar la autenticación del usuario al invocar el comando `su` y su pertenencia obligatoria al grupo `wheel`:

```
[root@centralita ~]# vim /etc/pam.d/su
#%PAM-1.0
auth                sufficient      pam_rootok.so
# Uncomment the following line to implicitly trust users in the "wheel" group.
#auth               sufficient      pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the "wheel" group.
auth               required        pam_wheel.so use_uid
```

El comportamiento del servicio SSH puede modificarse editando su fichero de configuración asociado `/etc/ssh/sshd_config`. Las líneas resaltadas en negrilla representan las modificaciones recomendadas:

```
[root@centralita ~]# vim /etc/ssh/sshd_config

.. .. .
SyslogFacility AUTHPRIV
LogLevel INFO
.. .. .

AllowUsers setup
PermitRootLogin no
LoginGraceTime 5m
MaxAuthTries 2
```

De este modo, registramos todos los accesos al sistema a través de SSH (tanto accesos correctos como incorrectos) en el fichero `/var/log/secure`, habilitamos el acceso SSH únicamente al usuario `setup` creado anteriormente, evitamos el acceso externo del superusuario `root` y limitaremos a tres intentos de autenticación con un período de gracia entre intentos fallidos de 5 minutos.

Para actualizar los cambios, reiniciamos el servicio SSH mediante el comando:

```
service sshd restart
```

4. Fail2Ban. Prevención de ataques SIP y SSH

Fail2Ban es un sistema de detección y prevención de intrusiones relativamente simple. Funciona inspeccionando los ficheros de registro (logs del sistema) y tomando las acciones pertinentes basándose en dichos registros. Utilizaremos Fail2Ban con una configuración que nos permita prevenir ataques de fuerza bruta contra nuestra centralita Asterisk.

Es posible descargar la última versión disponible de Fail2Ban a través de su página web oficial, en <http://www.fail2ban.org>. En el momento de esta guía de instalación, la versión disponible es la 0.8.4. Utilizaremos Fail2Ban en combinación con el firewall iptables, para proteger de ataques de fuerza bruta SIP en una centralita Asterisk.

En nuestro caso, descargaremos el código fuente de la versión 0.8.4 de Fail2Ban, disponible en la página web oficial del proyecto.

```
cd /usr/src
wget http://netcologne.dl.sourceforge.net/project/fail2ban/fail2ban-
stable/fail2ban-0.8.4/fail2ban-0.8.4.tar.bz2
```

Instalaremos además el paquete python e iptables para satisfacer dependencias y descomprimos las fuentes:

```
yum install python iptables -y
tar jxvf fail2ban-0.8.4.tar.bz2
```

Para comenzar la instalación, ejecutamos el instalador de Fail2Ban:

```
python setup.py install
```

Instalamos además el script de inicio en tiempo de arranque a /etc/init.d:

```
cp ./files/redhat-initd /etc/init.d/fail2ban
chmod 755 /etc/init.d/fail2ban
```

Configuración de Fail2Ban

Crearemos un filtro específico de Fail2Ban para Asterisk. De esto modo, Fail2Ban comprenderá el modo en que Asterisk registra los ataques y podrá tomar decisiones al respecto. Accedemos a la ruta de acceso de filtros de Fail2Ban y creamos un nuevo filtro de configuración para Asterisk:

```
cd /etc/fail2ban/filter.d
touch asterisk.conf
```

El contenido del fichero /etc/fail2ban/filter.d/asterisk.conf sería el siguiente:

```
# Fail2Ban configuration file
#
```

```

#
# $Revision: 250 $
#

[INCLUDES]
# Read common prefixes. If any customizations available -- read them from
# common.local
#before = common.conf

[Definition]
#_daemon = asterisk
#_Option: failregex
# Notes.: regex to match the password failures messages in the logfile. The
#         host must be matched by a group named "host". The tag "<HOST>" can
#         be used for standard IP/hostname matching and is only an alias for
#         (?:::f{4,6}:)?(?P<host>\S+)
# Values: TEXT
#

failregex = NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Wrong
password
          NOTICE.* .*: Registration from '.*' failed for '<HOST>' - No
matching peer found
          NOTICE.* .*: Registration from '.*' failed for '<HOST>' -
Username/auth name mismatch
          NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Device
does not match ACL
          NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Peer is
not supposed to register
          NOTICE.* <HOST> failed to authenticate as '.*'$
          NOTICE.* .*: No registration for peer '.*' \ (from <HOST>\)
          NOTICE.* .*: Host <HOST> failed MD5 authentication for '.*' (.*)
          NOTICE.* .*: Failed to authenticate user .*@<HOST>.*

# Option: ignoreregex
# Notes.: regex to ignore. If this regex matches, the line is ignored.
# Values: TEXT
#
ignoreregex =

```

Posteriormente añadiremos al fichero `/etc/fail2ban/jail.conf` la sección siguiente con el objetivo de activar el filtro anterior. En este caso, se prohibirán las conexiones provenientes de una IP atacante durante 3 días:

```

[asterisk-iptables]
enabled = true
filter = asterisk
action = iptables-allports[name=ASTERISK, protocol=all]
        sendmail-whois[name=ASTERISK, dest=root, sender=alertas-
asterisk@picon-networks.com]
logpath = /var/log/asterisk/messages
maxretry = 5
bantime = 259200

```

En el fichero `/etc/fail2ban/jail.conf`, podremos activar también las reglas de detección relacionadas con otros protocolos, como ataques contra el servicio de acceso remoto SSH:

```

[ssh-iptables]
enabled = true
filter = sshd
action = iptables[name=SSH, port=ssh, protocol=tcp]
        sendmail-whois[name=SSH, dest=root, sender=fail2ban@picon-

```

```
networks.com]
logpath = /var/log/secure
maxretry = 5
```

Evitar bloqueos en nuestra red

Para evitar bloquear nuestra propia red, podemos editar el fichero `/etc/fail2ban/jail.conf` y modificar la directiva `ignoreip` en la sección `[DEFAULT]`. Añadiremos, separadas por espacios, las direcciones IP, nombres de host, o segmentos de red en las que confiemos y no deseemos bloquear, por ejemplo:

```
ignoreip = 127.0.0.1 192.168.0.0/24 172.26.0.20 nombre_FQDN_sistema
```

Modificación del sistema de Registro (Log) de Asterisk

Fail2Ban utilizará el sistema de registro de Asterisk como medio de detección de ataques SIP. Por lo tanto, debemos modificar el sistema de registro de Asterisk para adaptarlo a Fail2Ban.

La versión 0.8.4 de Fail2Ban es compatible con el módulo de registro logger de Asterisk por lo que debemos modificar el fichero `/etc/asterisk/logger.conf` descomentando la directiva `messages`, dentro de la sección `[logfiles]`:

```
;
; Logging Configuration
;
.. .. .. ..

[logfiles]
messages => notice
full => notice,warning,error,debug,verbose
```

y recargamos el modulo de registro de Asterisk mediante el comando:

```
asterisk -rx "logger reload"
```

Activación y arranque de iptables y Fail2Ban

Para completar la configuración, activamos los servicios iptables y fail2ban para que se inicien durante el arranque del sistema e iniciamos los servicios:

```
chkconfig iptables on
chkconfig fail2ban on
service iptables start
service fail2ban start
```

Comprobación del funcionamiento de Fail2Ban

Tras iniciar los servicios iptables y fail2ban podemos comprobar si las reglas se han cargado correctamente, mostrando las tablas de iptables:

```
[root@centralita ~]# iptables -L -v

Chain INPUT (policy ACCEPT 5217 packets, 380K bytes)
 pkts bytes target      prot opt in      out     source      destination
  19 1332 fail2ban-SSH tcp -- any    any     anywhere    anywhere
tcp dpt:ssh
  85 8510 fail2ban-ASTERISK all -- any    any     anywhere    anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target      prot opt in      out     source      destination

Chain OUTPUT (policy ACCEPT 5107 packets, 502K bytes)
 pkts bytes target      prot opt in      out     source      destination

Chain fail2ban-ASTERISK (1 references)
 pkts bytes target      prot opt in      out     source      destination
  85 8510 RETURN      all -- any    any     anywhere    anywhere

Chain fail2ban-SSH (1 references)
 pkts bytes target      prot opt in      out     source      destination
  19 1332 RETURN      all -- any    any     anywhere    anywhere
```

Comprobamos que efectivamente Fail2Ban ha añadido correctamente sus filtros de entrada fail2ban-ASTERISK y fail2ban-SSH a la espera de incorporar posibles direcciones IP atacantes.

Mediante un softphone, realizamos un registro incorrecto a través del protocolo IAX2. En el fichero /var/log/asterisk/messages, aparecen los intentos de conexión:

```
[Oct 12 14:48:27] NOTICE[6718] chan_iax2.c: No registration for peer 'test' (from XXX)
[Oct 12 14:48:27] NOTICE[6719] chan_iax2.c: No registration for peer 'test' (from XXX)
[Oct 12 14:48:27] NOTICE[6720] chan_iax2.c: No registration for peer 'test' (from XXX)
```

Afortunadamente, Fail2Ban ha interceptado estos intentos, creando una regla en iptables que corta el tráfico con el host atacante:

```
[root@centralita ~]# iptables -L -v

Chain INPUT (policy ACCEPT 3952 packets, 282K bytes)
 pkts bytes target      prot opt in      out     source      destination
 480 33084 fail2ban-SSH tcp -- any    any     anywhere    anywhere
tcp dpt:ssh
 182 19681 fail2ban-ASTERISK all -- any    any     anywhere    anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target      prot opt in      out     source      destination

Chain OUTPUT (policy ACCEPT 3983 packets, 331K bytes)
 pkts bytes target      prot opt in      out     source      destination

Chain fail2ban-ASTERISK (1 references)
 pkts bytes target      prot opt in      out     source      destination
   7   316 DROP        all -- any    any     nombre_FQDN_equipoatacante anywhere
 175 19365 RETURN      all -- any    any     anywhere    anywhere
```

De esta forma, podemos bloquear los registros incorrectos SIP/IAX en el sistema de forma automática. Hacemos las mismas pruebas para el protocolo SSH. Nos intentamos conectar al puerto SSH (TCP/22) y realizamos tres intentos de acceso con credenciales incorrectas. El fichero /var/log/secure registra la actividad maliciosa:

```
Oct 12 15:42:34 centralita sshd[7464]: Invalid user test from XXX.XXX.XX.X
Oct 12 15:42:34 centralita sshd[7465]: input_userauth_request: invalid user test
Oct 12 15:42:41 centralita sshd[7464]: pam_unix(sshd:auth): check pass; user unknown
Oct 12 15:42:41 centralita sshd[7464]: pam_unix(sshd:auth): authentication failure;
logname= uid=0 euid=0 tty=ssh ruser= rhost=rivas-01.picon-networks.com
```

```

Oct 12 15:42:41 centralita sshd[7464]: pam_succeed_if(sshd:auth): error retrieving
information about user test
Oct 12 15:42:43 centralita sshd[7464]: Failed password for invalid user test from
XXX.XXX.XX.X port 3414 ssh2
Oct 12 15:42:44 centralita sshd[7464]: Failed password for invalid user test from
XXX.XXX.XX.X port 3414 ssh2
Oct 12 15:42:48 centralita sshd[7464]: pam_unix(sshd:auth): check pass; user unknown
Oct 12 15:42:48 centralita sshd[7464]: pam_succeed_if(sshd:auth): error retrieving
information about user test
Oct 12 15:42:50 centralita sshd[7464]: Failed password for invalid user test from
XXX.XXX.XX.X port 3414 ssh2
Oct 12 15:42:50 centralita sshd[7465]: Disconnecting: Too many authentication failures
for test
Oct 12 15:42:50 centralita sshd[7464]: PAM 1 more authentication failure; logname= uid=0
eid=0 tty=ssh ruser= rhost=rivas-01.picon-networks.com

```

Tras la desconexión del servidor SSH por demasiados intentos fallidos, Fail2Ban ya ha añadido la regla en iptables, como se puede observar:

```

[root@centralita ~]# iptables -L -v

Chain INPUT (policy ACCEPT 9504 packets, 722K bytes)
 pkts bytes target     prot opt in     out     source            destination
  369 27864 fail2ban-SSH tcp  --  any    any    anywhere          anywhere    tcp dpt:ssh
 1512 108K fail2ban-ASTERISK all  --  any    any    anywhere          anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source            destination

Chain OUTPUT (policy ACCEPT 9377 packets, 1048K bytes)
 pkts bytes target     prot opt in     out     source            destination

Chain fail2ban-ASTERISK (1 references)
 pkts bytes target     prot opt in     out     source            destination
 1512 108K RETURN    all  --  any    any    anywhere          anywhere

Chain fail2ban-SSH (1 references)
 pkts bytes target     prot opt in     out     source            destination
  18  4336 DROP      all  --  any    any    nombre_FQDN_atacante anywhere
 351 23528 RETURN    all  --  any    any    anywhere          anywhere

```

por lo que ya no es posible acceder mediante SSH al sistema hasta transcurridos 10 minutos, tiempo de rechazo por defecto.

Además de estas acciones automáticas, Fail2Ban nos informa a través de correo electrónico de ambas incidencias:

```

fail2ban@localhost Tue Oct 12 14:49 91/3060 "[Fail2Ban] ASTERISK: banned 192.168.0.20"
fail2ban@localhost Tue Oct 12 15:43 91/3050 "[Fail2Ban] SSH: banned 192.168.0.20"

```

5. Firewall AFP. Mercado ToS y gestión de conexiones

To configure APF is pretty easy and I will touch on a few of the config file options in this document. All of the options are covered in great detail on their website and well commented in the conf.apf file.

The config file for APF lives in /etc/apf and is called conf.apf. We will need to edit the conf.apf file, I like to use vi but any command line editor will work.

If you have multiple interfaces on your trixbox setup, you will want to set the IFACE_IN and IFACE_OUT to your external interface. This is the, untrusted network interface that is connected to the internet. If you have a second card eth1, that is used for internal network, trusted network, you can set the IFACE_TRUSTED to this interface. Please see the comments in the conf.apf if you are uncertain.

The setup script will try to properly determine which interface is used for untrusted network and place it in the appropriate field. I like to change the value of the SET_TRIM to 0. This value sets the total amount of rules allowed inside of the deny trust system. It is designed to save memory and start time. With the default value of 50, the system will start to purge old rules once this number is met. With the inclusion of BFD, this number will generally climb past 50.

Setting this value to 0 will disable this feature.

```
SET_TRIM="0"
```

APF has the ability to do QoS on packets, this is defined with the TOS values in the conf.apf. For SIP and IAX, I set the following.

```
TOS_8="21,20,80,4569,5060,10000_20000"
```

This also requires a small tweak to one of the config files that I will discuss later in the document in order to tag UDP packets.

Since we changed the SSH port to a different number, we have to tweak the conf.apf to match this new port.

```
HELPER_SSH_PORT="2222"
```

Make sure to place the correct port number that you decided to run SSH on. Ingress filtering is used to open inbound ports for access, both TCP and UDP have separate settings. For a trixbox setup, the following ports should be open, both TCP and UDP are listed. If you are not using tftp, then do not have port 69 open. Do not forget to change the SSH port from 22, to the port you choose to run SSH on.

Otherwise you will be locked out, here we are using port 2222 from our example above. I have not included IAX ports in this setup. There is a easy way to ensure that only specific hosts can use IAX that I will cover later. This is handy if you use IAX to do

interoffice trunks like I do, but don't want IAX ports open for the world to see.

```
IG_TCP_CPORTS="2222,69,80,5060,6600,10000_20000"
```

```
IG_UDP_CPORTS="69,5060,10000_20000"
```

Egress filtering is used to allow outbound filtering. I don't use egress filtering and it will not be covered in this document. It is set to `EGF="0"`, or disabled by default. In the section of the `conf.apf` file called Imported Rules, there are settings for various feeds. Some of these feeds are very handy and I use them all. I have even setup my own custom feed that allows me to tweak all of my servers with global deny rules. You can disable or enable this feature with the `USE_DS` setting. A "1" is enabled, a "0" is disabled.

We are now ready to start APF for the first time. Double check that the SSH port is set correctly to the one you are using. If you start APF right now and something is wrong, it will disable itself in 5 minutes. This is called `DEVEL_MODE` and is the first setting in the `conf.apf` file. Leave this set to "1" until you are certain you can get in via ssh and things are working.

To see a list of command line options run `apf` without any flags.

```
[trixbox1.localdomain apf]# apf
apf(3402): {glob} status log not found, created
APF version 9.6 <apf@r-fx.org>
Copyright (C) 1999-2007, R-fx Networks <proj@r-fx.org>
Copyright (C) 2007, Ryan MacDonald <ryan@r-fx.org>
This program may be freely redistributed under the terms of the GNU GPL
usage /usr/local/sbin/apf [OPTION]
-s|--start ..... load all firewall rules
-r|--restart ..... stop (flush) & reload firewall rules
-f|--stop..... stop (flush) all firewall rules
-l|--list ..... list all firewall rules
-t|--status ..... output firewall status log
-e|--refresh ..... refresh & resolve dns names in trust
rules
-a HOST CMT|--allow HOST COMMENT ... add host (IP/FQDN) to allow_hosts.rules
and
immediately load new rule into firewall
-d HOST CMT|--deny HOST COMMENT .... add host (IP/FQDN) to deny_hosts.rules
and
immediately load new rule into firewall
-u|--remove HOST ..... remove host from [glob]*_hosts.rules
and immediately remove rule from firewall
-o|--ovars ..... output all configuration options
```

To start APF we issue the following command.

```
[trixbox1.localdomain apf]# apf -s
apf(3445): {glob} activating firewall
apf(3489): {glob} determined (IFACE_IN) eth0 has address 192.168.1.31
apf(3489): {glob} determined (IFACE_OUT) eth0 has address 192.168.1.31
apf(3489): {glob} loading preroute.rules
apf(3489): {resnet} downloading http://r-fx.ca/downloads/reserved.networks
apf(3489): {resnet} parsing reserved.networks into
/etc/apf/internals/reserved.networks
apf(3489): {glob} loading reserved.networks
apf(3489): {glob} SET_REFRESH is set to 10 minutes
apf(3489): {glob} loading bt.rules
apf(3489): {dshield} downloading http://feeds.dshield.org/top10-2.txt
apf(3489): {dshield} parsing top10-2.txt into /etc/apf/ds_hosts.rules
```

```
apf(3489): {dshield} loading ds_hosts.rules
apf(3489): {sdrop} downloading http://www.spamhaus.org/drop/drop.lasso
apf(3489): {sdrop} parsing drop.lasso into /etc/apf/sdrop_hosts.rules
apf(3489): {sdrop} loading sdrop_hosts.rules
apf(3489): {glob} loading common drop ports
.....trimmed for this document.....
apf(3489): {glob} default (ingress) input drop
apf(3445): {glob} firewall initalized
apf(3445): {glob} !!DEVELOPMENT MODE ENABLED!! - firewall will flush every 5
minutes.
```

We can see that APF has started, downloaded some rules from dshield.org and spamhaus.org and then told us it is in DEVELOPMENT MODE. Now test connecting to your server over SSH to ensure that you have setup the correct port number ingress. If you can't connect, you will have to wait 5 minutes and then APF will shutdown. Once you are sure you can get in with SSH we can change the conf.apf file from DEVEL_MODE="1" to DEVEL_MODE="0" and restart/start APF. APF will start and not warn you about being in DEVELOPMENT MODE, your firewall should be good to go.

APF additional tweaks. This setup might not be ideal for everyone. If you connect to your provider over IAX then you will definitely want to add the IAX ports to the conf.apf. However if you have two or more systems that you connect to each other over IAX for interoffice connections, then this is the way to go. This will work with static IP addresses and DYNDNS setups alike.

You can use a fully qualified DNS hostname or IP address. One of the flags for the apf command is -a, which is allow. This will globally allow a host to connect to this system, bypassing the firewall rules. I can't stress how handy this is. Some examples are allowing a SNMP query, IAX connections, or other ports that you do not want open, but need to allow specific hosts to connect to. To do this just issue the following command, substitute your remote system IP address with the one I have here.

```
apf -a 192.168.1.216
```

This will allow the system 192.168.1.216 to connect to any port on the firewalled server, thereby bypassing the firewall rules. If you are running APF on both systems, be sure to do the same thing on the other host using the correct IP address. APF also allows a system admin to block a host or a complete subnet. This is handy if you see someone attempting to connect to your machine over ftp, telnet, ssh, etc.. To block a specific host use the following, be sure to use the IP you want to block.

```
apf -d 192.168.1.216
```

To block a complete subnet (CIDR) the command is very similar.

```
apf -d 202.86.128.0/24
```

This will block the entire subnet. You can sometimes get the subnet (CIDR) listing using a whois on the IP address. You can also lookup a CIDR by ip on google or ripe.net. Be sure that the subnet is not one you are in or you could lock yourself out. TOS for UDP packets are not defined for APF. Only TCP packets have the TOS bit set. There is a easy way to fix this. In the /etc/apf/internals folder is a file called, functions.apf. We need to edit this file manually. It is pretty straight forward as to what

we need to changed, so don't worry.

There are several places we have to add a single line. Look for the TOS_ section in the functions.apf. It will look like this.

```
if [ ! "$TOS_0" == "" ]; then
for i in `echo $TOS_0 | tr ',' ' '`; do
i=`echo $i | tr '_':`
$IPT -t mangle -A PREROUTING -p tcp --sport $i -j TOS --set-tos 0
done
fi
```

We have to add the settings for UDP. We copy one line and change tcp to upd. A sample is below, highlighted in red.

```
if [ ! "$TOS_0" == "" ]; then
for i in `echo $TOS_0 | tr ',' ' '`; do
i=`echo $i | tr '_':`
$IPT -t mangle -A PREROUTING -p tcp --sport $i -j TOS --set-tos 0
$IPT -t mangle -A PREROUTING -p udp --sport $i -j TOS --set-tos 0
done
fi
```

This additional line has to be done for all the TOS bits you are using. If you are only using TOS_8 , then only worry about doing it for those. Make sure you do the tospostroute and tospreroute sections.

6. Recomendaciones de seguridad

Las siguientes recomendaciones resumen los puntos imprescindibles para garantizar un nivel mínimo de seguridad en Asterisk:

- ***Evitar las contraseñas suministradas por defecto en metadistribuciones.***

Es muy común encontrarse con sistemas basados en metadistribuciones de Asterisk, por ejemplo, la distribución Elastix, con juegos completos de contraseñas en los valores suministrados de fábrica. Estos valores resultan extremadamente sencillos de averiguar con una simple búsqueda en Internet:

Interfaz Web Elastix:

Username: admin

Password: palosanto

Sugar CRM:

Username: admin

Password: password

A2Billing

Username: admin / root

Password: mypassword / myroot

Flash Operator Flash Panel (FOP):

Password: eLaStIx.2oo7

Freepbx:

Username: admin

Password: admin

vtigerCRM use:

Username: admin

Password: admin

Acceso directo a MySQL:

Username: root

Password: eLaStIx.2oo7

Por lo tanto, es IMPERATIVO cambiar las contraseñas por defecto del sistema.

- ***Utilizar contraseñas fuertes para los servicios y extensiones SIP/IAX***

Se recomienda utilizar combinación de caracteres especiales (#, %, &, /, @, etc.), números, mayúsculas y minúsculas y evitar el uso de secuencias de números o palabras que se puedan encontrar en un diccionario. Deberán tener estas características no solo la contraseña de root, la contraseña de gestión de administrativa de FreePBX, MySQL o FOP, sino también las contraseñas de las extensiones SIP e IAX.

- ***Limitar las direcciones IP de comunicación vía SIP/IAX***

Si es posible, impedir las conexiones SIP desde cualquier dirección IP (la configuración por defecto). Para ello, es necesario editar el fichero “sip.conf” o “sip_general_custom.conf” y añadir las directivas “permit=” y “deny=”.

Evidentemente, esta solución representa un problema si tenemos equipos remotos como portátiles que se conectan vía SIP al sistema y cuyo direccionamiento IP cambia con el tiempo. En este caso, es más que recomendable la utilización de VPNs para implementar esta conectividad.

- ***Restringir el acceso al equipo Asterisk***

Si es posible, restringir el acceso a puertos clave del sistema Asterisk desde direcciones IP conocidas, descartando el resto. Resulta buena idea utilizar iptables para restringir el acceso a ciertos puertos desde ciertas direcciones. La misma premisa podría aplicarse a los servicios SSH y Web.

- ***Prevenir fuga de información sobre extensiones SIP***

Por defecto, Asterisk puede dejar escapar información útil sobre las extensiones que podría resultar de utilidad para herramientas de ataque y reducir así el tiempo necesario para asaltar nuestro sistema. A partir de la versión 1.2 de Asterisk es posible controlar este comportamiento.

La directiva “alwaysauthreject=yes” en el fichero “sip.conf” o “sip_general_custom.conf” impide la fuga de información sobre las extensiones.

- ***Restringir la comunicación el Interfaz de Administración de Asterisk, Asterisk Manager Interface, AMI, únicamente a la red local.***

La instalación por defecto de FreePBX configura el AMI de Asterisk para denegar cualquier conexión salvo aquellas provenientes del interfaz local (directivas “deny=0.0.0.0/0.0.0.0” y “permit=127.0.0.1/255.255.255.0”) en el fichero “manager.conf”. Resulta extremadamente importante no redefinirlas en otros ficheros adicionales como los ficheros “manager_additional.conf” o “manager_custom.conf”.

- ***Impedir tráfico SIP anónimo***

FreePBX (y también Elastix) ofrecen la posibilidad de permitir tráfico SIP entrante anónimo a través de trunks. La directiva “Permitir llamadas entrantes SIP anónimas” en el panel de configuración de FreePBX, sección “Configuración general”, hace posible recibir llamadas SIP entrantes de fuentes anónimas. Este parámetro resulta útil únicamente para realizar pruebas de funcionamiento SIP pero deberá establecerse en “NO” en TODO momento.

7. Bibliografía

- Securing Trixbox CE by Tim Yardley
http://engineertim.com/wp-content/uploads/2008/12/securing_trixbox_ce_ver1.pdf
- Fail2Ban (with iptables) And Asterisk
<http://www.voip-info.org/wiki/view/Fail2Ban+%28with+iptables%29+And+Asterisk>
- Seguridad en Asterisk con fail2ban
<http://revistalinux.net/articulos/seguridad-en-asterisk-con-fail2ban/>
- Instalar y configurar fail2ban para Asterisk <http://www.voztovoice.org/?q=node/135>
- Recomendaciones sobre seguridad en Elastix
<http://blogs.elastix.org/es/2009/10/22/recomendaciones-de-seguridad-en-elastix/>