

CONFIGURACION PIX by

Angel Alonso Párrizas

NOTA: a) Los PIX siempre trabajan haciendo NAT.

b) Un PIX deja pasar el trafico de un nivel de seguridad mayor a uno menor (no a la inversa)

c) Para que un PIX deje pasar el trafico de un nivel menor a uno mayor hay que hacerlo mediante listas de accesos.

d) Si desde un nivel de seguridad X se abre un conexión TCP/UDP hacia un nivel menor, el firewall crea una especie de lista dinamica que permite el trafico a la inversa, desde la zon menor hacia la mayor, por ejemplo para una conexión telnet. Esto no ocurre con el ICMP, así pues si se manda un ICMP Echo request hacia un host de nivel de seguridad menor el ICMP echo reply desde el nivel de seguridad menor hacia el mayor no funcionará, excepto si se crea una lista de acceso para paquetes ICMP para que deje pasar los paquetes ICMP.

E) Las listas de acceso se saltan las zonas de seguridad (como ni no existiera)

Si no tiene configuración, inicialmente nos preguntará

Pre-configure PIX Firewall now through interactive prompts [yes]?
no Type help or '?' for a list of available commands.

```
pixfirewall>
```

```
pixfirewall >enable
```

```
Password: <Enter>
```

```
pixfirewall #configure terminal
```

```
pixfirewall (config)#
```

Borramos la configuración por defecto y rearrancamos.

```
pixfirewall (config)# write erase
```

```
Erase PIX configuration in flash memory? [confirm] <Enter>
```

```
pixfirewall (config)# reload
```

```
Proceed with reload? [confirm] <Enter>
```

```
Pre-configure PIX Firewall through interactive prompts [yes]? Ctrl  
+ Z pixfirewall>
```

Los comandos disponibles en modo usuario, los podemos ver con:
pixfirewall> ?

Ahora pasamos al modo privilegiado:

```
pixfirewall> enable
```

```
Password: <Enter>
```

```
pixfirewall#
```

Ver configuración firewall:

```
pixfirewall# write terminal (o sh run)
```

Ver memoria, CPU, version:

```
pixfirewall# show memory
```

```
pixfirewall# show version
```

```
pixfirewall# show cpu usage
```

Configuración Interfaces:

```
pixfirewall# configure terminal -> Modo configuracion
```

```
pixfirewall(config)# hostname PIX -> Poner nombre al PIX
```

```
PIX(config)# names -> poner nombres para el DNS local
```

```
PIX(config)# name 172.16.1.2 bastionhost
```

```
PIX(config)# name 10.0.1.11 insidehost
```

Asignar nivel de seguridad y nombre a una interfaz.

```
PIX(config)# nameif e2 dmz security50 -> el interfaz eth2 conecta la zona DMZ y tiene seguridad 50
```

Asignar velocidad y modo full.

```
PIX(config)# interface e0100full
```

Ver configuración de las interfaces

```
PIX(config)# show interface
```

Asignar IPs y subredes

```
PIX(config)# ip address inside 10.0.1.1 255.255.255.0 ->inside está definido como una interfaz, e1, e2..
```

Definición de rutas

```
PIX(config)# route outside 0.0.0.0 0.0.0.0 192.168.1.1 1 -> el uno final indica el coste
```

Configuración de NAT:

NAT DINAMICO

Vamos a definir conexión (mediante NAT) entre la zona inside a outside, así se podrá tener acceso desde los host en la red inside a los host de la red outside.

```
PIX(config)# nat (inside) 1 10.0.1.0 255.255.255.0 ->primer comando  
define las IP's antes del nat
```

```
PIX(config)# global (outside) 1 192.168.1.200-192.168.1.254  
netmask 255.255.255.0 ->segundo comando define las IP's después del NAT, con las  
que saldrá. El "1" en las reglas nos indica que ambas reglas están relacionadas.
```

Ver estado del NAT:

```
PIX(config)# show global
```

```
PIX(config)# show nat
```

Ver la tabla de traducciones:

```
PIX(config)# show xlate
```

```
PIX(config)# show xlate debug
```

Configuración para que desde inside se tenga acceso a la zona DMZ.

```
PIX(config)# global (dmz) 1 172.16.1.200-172.26.1.254 netmask  
255.255.255.0
```

NAT ESTATICO

```
PIX(config)# static (dmz,outside) 192.168.1.11 bastionhost ->El host  
bastion saldrá en outside con la ip 192.168.1.11
```

listas de acceso:

Las listas de acceso anulan las zonas de seguridad. Una lista de acceso permite todo el tráfico definido en la propia lista.

```
PIX(config)# access-list outside_access_in permit icmp any any ->  
definimos un regla outside_access_in que nos indica que se permite todo el tráfico ICMP con  
origen ANY y destino ANY
```

```
PIX(config)# access-group outside_access_in in interface outside ->  
definimos un grupo que se asocia a la lista de acceso anterior
```

lista de acceso para el servicio de telnet a un host que en la zona outside tiene una IP determinada, y que por NAT se reenviarán las peticiones del servicio a su ip real.

```
PIX(config)# access-list outside_access_in permit tcp any host  
192.168.1.11 eq telnet
```

Mostras las listas de acceso creadas:

```
PIX(config)# show access-list
```

Ver opciones de depurado:

```
PIX(config)# debug packet inside
```

Quitar opciones de depurado:

```
PIX(config)# no debug all
```