PRÁCTICA: Configuración básica de un cortafuegos (cortafuegos PIX 515E de *Cisco Systems*)

Autor: Santiago Felici

1.- Objetivo

En esta práctica vamos a introducir la configuración de un cortafuegos comercial, concretamente de *Cisco Systems*, llamado PIX (*Private Internet eXchange*). Durante el desarrollo de la práctica configuraremos sus interfaces y definiremos niveles de seguridad (*inside, dmz, outside*), habilitando NAT (*Network Address Traslation*) entre ellos. Analizaremos las conexiones permitidas según los niveles de seguridad y realizaremos reglas específicas, mediante listas de acceso para permitir/denegar cierto tráfico.

2.- Introducción

La seguridad perimetral es una de las piezas clave de la política de seguridad de una empresa. Es uno de los elementos de seguridad más críticos de una empresa cuando se conecta a Internet y por regla general es implementada en cortafuegos o "*firewall*", un dispositivo hardware capaz de analizar todas las conexiones entrantes/salientes simultáneamente. Su configuración debe ser acorde según la política de seguridad definida. Además, dado que sobre este dispositivo se analizan las reglas de seguridad, en ocasiones se puede convertir en el cuello de botella, situación que hay que evitar.

El PIX analiza y registra las conexiones entrantes y salientes implementando NAT. Con esta técnica, al PIX se le permite en todo momento tener control de las conexiones establecidas y rechazar conexiones no permitidas.

El PIX define niveles de seguridad o zonas de seguridad, que en el caso de la presente práctica son tres: *inside, dmz, outside*, las cuales quedan asociadas con niveles 100, 50 y 0 respectivamente. La regla que aplica el PIX por defecto es: "*no se puede pasar de nivel de seguridad menor a uno mayor*". En base a esto, vamos a desarrollar toda la práctica.

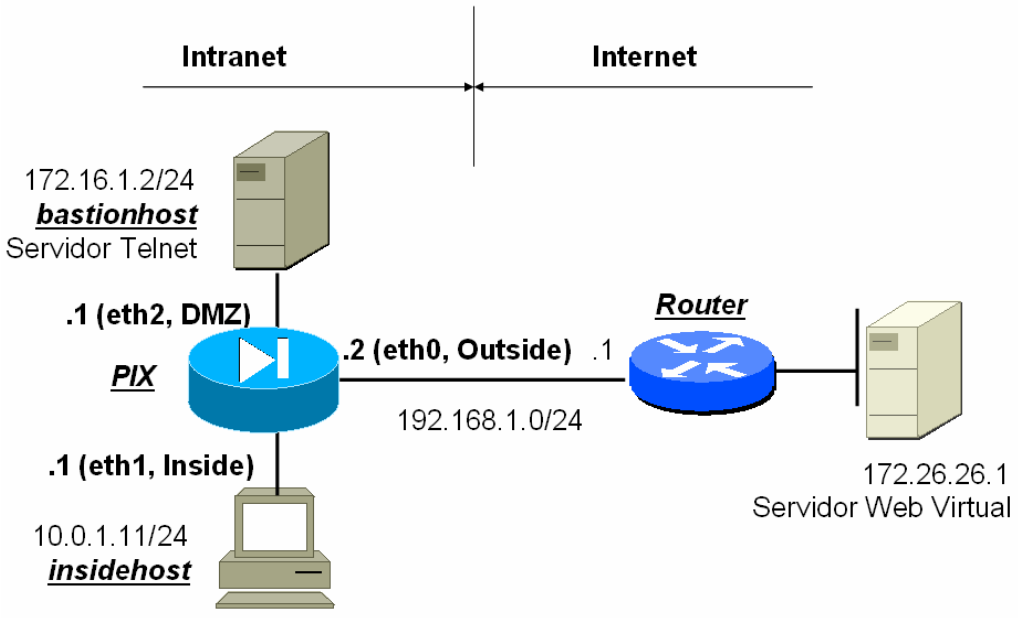

***Figura 1**. Esquema de la maqueta donde se define dos zonas Intranet e Internet, separadas por un cortafuegos con 3 niveles de seguridad definidos (Inside, DMZ, Outside) conectando al exterior (Internet) por un router.*

Además, los PIX permiten la configuración de listas de acceso, de forma similar a los routers. En el caso de tener configuradas listas de acceso, el PIX ignora los niveles de seguridad preestablecidos entre zonas, para el tráfico comprobado por las listas de acceso.

Además, estas listas de acceso se pueden crear de forma transparente al usuario, es decir los PIX implementan un mecanismo automático de listas de acceso para permitir tráfico de entrada cuando se genera tráfico de salida, acorde con los patrones del tráfico de salida, es decir que si generamos conexiones *http* de salida, no podemos tener tráfico *ftp* de entrada. Por tanto una vez permitido el tráfico de vuelta, puede pasar sin problema de un nivel de menor seguridad a uno de mayor. Este mecanismo se implementa de forma automática en los PIX y se conoce como CBACs "*Context Based Access*". Este mecanismo propietario de Cisco Systems sólo funciona en caso de establecimiento de conexiones y/o pseudoconexiones, como TCP/UDP respectivamente, pero no para ICMP. Por tanto, si generemos tráfico TCP/UDP por ejemplo de "*inside*" a "*outside*", CBACs de forma automática crea y/o genera permisos (como listas de acceso temporales), para permitir tráfico de "*outside*" a "*inside*", respetando que el tráfico sea asociada a la misma conexión.

3.- Realización de la práctica

Los siguientes pasos se han de realizar de forma coordinada por todas las parejas. Previamente, se supone que el alumno debe estar familiarizado con la práctica o haberse leído este documento.

Al final de la práctica, en el ANEXO II, se muestra la configuración final que debe tener el PIX. Esta configuración será utilizada en prácticas posteriores.

Los pasos a realizar en esta práctica son:

Paso 0: cableado de la maqueta

En primer lugar comprobar la conectividad física de toda la maqueta tal como indica en la figura 1. Los equipos que la componen son un *router* genérico de *Cisco Systems* y un cortafuegos PIX modelo 515E.

Las interfaces de los cortafuegos son iguales que las de los *routers*, por tanto para conectar directamente ordenador (*host* o servidor) al PIX por una *Ethernet*, debemos utilizar cables UTP-5 cruzados.

El servidor web 172.26.26.1 es un servidor virtual que configuraremos en el router.

Paso 1: configuración IP del router, host y servidores. No configuramos el PIX todavía

Los pasos para configurar los *routers* son:

NOTA: En ningún momento, se indica a lo largo de la práctica guardar la configuración de los routers. Con ello se simplifica el último paso de la práctica, que consiste en dejar la maquetas tal como estaba en el inicio.

a.- utilizaremos la conexión de consola mediante el programa de emulación de terminal "minicom" (comando '**minicom –s**' u otro programa terminal). La configuración del programa de emulación debe ser la siguiente:

- Velocidad 9600 bits/s
- Sin paridad
- 8 bits de datos
- bit de parada (8N1)
- Dispositivo de entrada: ttyS0

b.- encender el *router*. Debe aparecer la secuencia de mensajes de arranque. Esto nos confirma que la comunicación por el puerto de consola es correcta.

c.- Una vez ha arrancado el *router* debe aparecer el *prompt* `'Router>'`; teclear el comando **`'enable'`** para pasar a modo Privilegiado. En caso de que pida una *password* consultar al profesor.

d.- Ejecutar el comando "`show ip interface brief`" y tomar nota de los nombres asignados a las interfaces en cada modelo de *router*. El nombre de las interfaces depende del modelo y se utilizarán a continuación.

e.- Una vez en modo Privilegiado entraremos en modo Configuración Global para introducir la configuración que corresponde a cada router, siguiendo el siguiente modelo. Utilizad en cada router, el nombre para identificar a las diferentes interfaces, tal como se ha indicado anteriormente.

Nota sobre la configuración: Este *router* no define ni routing estático ni dinámico, sólo directamente conectadas, dado que la configuración de la maqueta tiene que funcionar con NAT utilizando para ello direcciones de la red que interconecta el PIX con el *router*, 192.168.1.0/24. La interfaz de *Loopback* simulará el servidor Web de la Internet, conectado directamente al *router*.

```
Router>enable
Router#configure terminal
Router(config)#hostname Internet-router
Internet-router (config)#no ip domain-lookup
Internet-router (config)#ip http server

Internet-router (config)#interface fastethernet[1] 0[2]
Internet-router (config-if)#ip address 192.168.1.1 255.255.255.0
Internet-router (config-if)#no shutdown
Internet-router (config-if)#exit

Internet-router (config)#interface loopback 0
Internet-router (config-if)#ip address 172.26.26.1 255.255.255.0
Internet-router (config-if)#no shutdown
Internet-router (config-if)#exit

Internet-router (config)#line vty 0 4
Internet-router (config-line)#password cisco
Internet-router (config-line)#exit
```

Los pasos para configurar los hosts/servidores son:

En este paso vamos a configurar los hosts/servidores conectados. Para configurar los hosts/servidores debemos de conectarlos en cada una de las interfaces del cortafuegos y asignar una IP y ruta por defecto. La configuración en los hosts es[3]:

```
ifconfig eth0 inet d.d.d.d netmask 255.255.255.0
route add default gw r.r.r.r
```

donde "*d.d.d.d*" es la dirección del host acorde con la figura anexa y "*r.r.r.r*" es la ruta por defecto, que será la IP asignada a la interfaz del PIX.

Paso 2: inicialización del cortafuegos PIX

Inicialmente el PIX se encuentra sin configuración pero de todas formas vamos a reinicializarlo previamente con lo cual debemos ir introduciendo los siguientes comandos.

Los comandos del PIX guardan mucha relación con los comandos del router y por tanto nos resultarán bastante familiares. Para ver una explicación más detallada de los comandos, ver el ANEXO I "Comandos básico del PIX de *Cisco Systems*".

NOTA: En ningún momento, se indica a lo largo de la práctica guardar la configuración del PIX. Con ello se simplifica el último paso de la práctica, que consiste en dejar la maqueta tal como estaba en el inicio.

a.- utilizaremos la conexión de consola mediante el programa de emulación de terminal "minicom" (comando '**minicom –s**' u otro programa terminal), configurada de la misma forma que para configurar el router.

b.- encender el *PIX* e introducir los siguientes comandos.

Si no tiene configuración, inicialmente nos preguntará

[1] Dependiendo del modelo del router, las interfaces se llamen **`FastEthernet o Ethernet`**
[2] En los routers modulares la designación de interfaces es módulo/slot, por ejemplo 0/0, en los no modulares simplemente slot, por ejemplo 0
[3] Si no funcionara el *Telnet*, desactivar el servicio DNS moviendo el fichero **/etc/resolv.conf** a otro nombre. Acuérdese después de restaurarlo. Si aún así tampoco funcionara, desactive el firewall, "**services iptables stop"**

```
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#
```

Mostramos los comandos disponibles en modo privilegiado, que son diferentes a los vistos anteriormente:

```
pixfirewall# ?
```

Para ver la configuración por defecto cargada en el PIX, podemos utilizar tanto "`write terminal`" o "`show run`". En este caso, mostramos una configuración simplificada. En el ANEXO II podemos observar una configuración completa.

```
pixfirewall# write terminal

nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
…
hostname pixfirewall
…
names
…
interface ethernet0 auto shutdown
interface ethernet1 auto shutdown
interface ethernet2 auto shutdown
…
ip address outside 127.0.0.1 255.255.255.255
ip address inside 127.0.0.1 255.255.255.255
ip address intf2 127.0.0.1 255.255.255.255
…
: end
[OK]
```

Para comprobar la memoria disponible en el PIX, la versión de PIX y la utilización de la CPU utilizamos los siguientes comandos

```
pixfirewall# show memory
pixfirewall# show version
```

```
pixfirewall# show cpu usage
```

¿Qué tamaño dispone de memoria?¿Cuántos están libres?

_____
_____

¿Qué versión de PIX se está ejecutando?¿Qué direcciones MAC disponen las interfaces del PIX?¿Qué versiones de VPN dispone en la licencia instalada?

_____
_____

¿Qué utilización tiene la CPU? Una utilización alto puede suponer pérdida de paquetes y además el PIX se puede convertir en el cuello de botella.

_____
_____

Paso 3: configuración de las interfaces del cortafuegos PIX

Una diferencia entre la configuración de un *router* y un PIX utilizando CLI (*Command Line Interface*) o modo comandos, es que en el PIX no aparecen los mismos modos de configuración, específicos en lo que estamos configurando. Es decir, cuando configuramos interfaces en el router aparece "(`config-if`)" y en el PIX sigue apareciendo "(`config`)".

Pasemos ahora a configurar el PIX

```
pixfirewall# configure terminal
pixfirewall(config)# hostname PIX
```

Vamos a configurar ahora la opción para definir resolución de nombres estática y de forma local, para evitar el uso de direcciones IP directamente y/o consulta a un DNS externo. En primer lugar habilitamos este de modo de resolución y seguidamente indicamos las resoluciones:

```
PIX(config)# names
PIX(config)# name 172.16.1.2 bastionhost
PIX(config)# name 10.0.1.11 insidehost
```

Configuramos las interfaces, asignando nombre y nivel de seguridad para la DMZ (que no tiene configuración por defecto) con la palabra `security50`, que asocia un nivel de seguridad 50, y lo comprobamos:

```
PIX(config)# nameif e2 dmz security50

PIX(config)# show nameif

nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
```

Las interfaces físicas el PIX por defecto están deshabilitados, por ello con los siguientes comandos vamos a activar las interfaces del PIX y las configuramos a nivel físico para que no tengan problemas con la autonegociación de la LAN, fijándolas a 100 Mbps en modo Full –duplex.

```
PIX(config)# interface e0 100full
PIX(config)# interface e1 100full
PIX(config)# interface e2 100full
```

Una vez habilitadas y configuradas las interfaces, pasemos a comprobar su configuración:

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

```
interface ethernet0 "outside" is up, line protocol is up
Hardware is i82558 ethernet, address is 0090.2724.fd0f
IP address 127.0.0.1, subnet mask 255.255.255.255
MTU 1500 bytes, BW 100000 Kbit full duplex
…

interface ethernet1 "inside" is up, line protocol is up
Hardware is i82558 ethernet, address is 0090.2716.43dd
IP address 127.0.0.1, subnet mask 255.255.255.255
MTU 1500 bytes, BW 100000 Kbit full duplex
…

interface ethernet2 "dmz" is up, line protocol is up
Hardware is i82558 ethernet, address is 0090.2725.060d
IP address 127.0.0.1, subnet mask 255.255.255.255
MTU 1500 bytes, BW 100000 Kbit full duplex
…
```

El siguiente paso es configurar las IP en dichas interfaces:

```
PIX(config)# ip address inside 10.0.1.1 255.255.255.0
PIX(config)# ip address dmz 172.16.1.1 255.255.255.0
PIX(config)# ip address outside 192.168.1.2 255.255.255.0
```

Comentar que para la interfaz "*outside"* podríamos haber utilizado configuración dinámica con DHCP. Si habilitamos el modo dinámico y para ello debemos configurar un servidor DHCP en el router o en el ISP en su caso. Por simplificación y seguridad configuramos la interfaz de "*outside*" también de forma estática.

Una vez configuradas las IP de las interfaces, comprobemos su configuración:

```
PIX(config)# show ip address

System IP Addresses:
ip address outside 192.168.1.2 255.255.255.0
ip address inside 10.0.1.1 255.255.255.0
ip address dmz 172.16.1.1 255.255.255.0

Current IP Addresses:
ip address outside 192.168.P.2 255.255.255.0
ip address inside 10.0.1.1 255.255.255.0
ip address dmz 172.16.1.1 255.255.255.0
```

Finalmente para que el PIX pueda encaminar dado que también hace funciones de router, configuraremos una ruta estática por defecto al router de Internet con coste 1

```
PIX(config)# route outside 0.0.0.0 0.0.0.0 192.168.1.1 1
```

¿Qué rutas existen disponibles en el PIX?

```
PIX(config)# show route
```

_____
_____
_____
_____

Ahora, comprobemos la conectividad IP utilizando "ping" desde todos los dispositivos a todos los dispositivos. Indicad en la siguiente tabla con una cruz "OK" si funciona correctamente o "NO" si no funciona la conectividad IP con "ping".

Destacar que la configuración del PIX no está completa y por tanto puede que la conectividad IP no sea completa.

Ping Desde ->	**"insidehost"**	**"bastionhost"**	**"PIX"**	**"Router"**

A “insidehost”				
A “bastionhost”				
A “PIX”				
A “Router/Web”				

Comenta los resultados obtenidos y justifícalo.

_____
_____
_____

Paso 4: configuración de NAT (inside, outside) en el PIX y establecimiento de conexiones

Una vez configuradas las interfaces, vamos a habilitar la comunicación entra las diferentes zonas: *inside, dmz, outside*, según sus niveles de seguridad 100, 50 y 0 respectivamente. La regla que aplica el PIX es: “*no se puede pasar de nivel de seguridad menor a uno mayor*”.

Además, para poder establecer la comunicación, el PIX debe poder realizar registro de las conexiones y para ello utiliza NAT. La configuración de NAT conlleva siempre la configuración de 2 comandos, “`nat`” que se asocia a la interfaz de entrada indicando qué IP interna entra en el NAT y “`global`” que se asocia a la interfaz de salida donde se especifica la IP (o IPs) externa. Es decir, estos comandos especifican por un lado las direcciones “dentro” (pre-NAT) y por otro lado la dirección o direcciones (en el caso de un conjunto o “pool”) “fuera” (post-NAT). Ambos comandos, quedan mutuamente asociados por un identificativo numérico, en nuestro caso “1”. En el caso de utilizar una sola IP en global, el NAT se realizaría “overload” o extendido o también llamado PAT (*Port Address Traslation*). Comentar también, que existe un caso de NAT especial que se llama NAT 0, que es no hacer NAT, pero aun así también se registran las conexiones, pero sin realizar cambios a IP diferentes.

Por tanto, vamos a configurar la traducción con identificativo nº 1 de NAT:

```
PIX(config)#nat (inside) 1 10.0.1.0 255.255.255.0
PIX(config)#global (outside) 1 192.168.1.200-192.168.1.254 netmask 255.255.255.0
```

Para comprobar que la traducción nº 1 de NAT se ha configurado correctamente utilizaremos el comando

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

Ya finalmente, vamos a comprobar la configuración final introducida:

```
PIX(config)# write terminal
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
…
hostname PIX
…
names
name 172.16.1.2 bastionhost
name 10.0.1.11 insidehost
…
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
…
ip address outside 192.168.1.2 255.255.255.0
ip address inside 10.0.1.1 255.255.255.0
ip address dmz 172.16.1.1 255.255.255.0
…
global (outside) 1 192.168.1.200-192.168.1.254 netmask 255.255.255.0
nat (inside) 1 10.0.1.0 255.255.255.0 0 0
route outside 0.0.0.0 0.0.0.0 192.168.1.1 1
…
```

Volvamos a comprobar la conectividad IP utilizando "ping" desde todos los dispositivos a todos los dispositivos. Indicad en la siguiente tabla con una cruz "OK" si funciona correctamente o "NO" si no funciona la conectividad IP con "ping".

Ping Desde ->	**"insidehost"**	**"bastionhost"**	**"PIX"**	**"Router"**
A "insidehost"				
A "bastionhost"				
A "PIX"				
A "Router/Web"				

También podemos comprobar si podemos establecer conexión http desde el "insidehost" con el servidor Web (http://172.26.26.1) y hacer Telnet al "bastionhost"[4].

Desde ->	**"insidehost"**	**"bastionhost"**	**"PIX"**	**"Router"**
Telnet a "bastionhost"				
Http a "Router/Web"				

Comenta los resultados obtenidos y justifícalo.

_____
_____
_____
_____

A simple vista parece extraño que funcionen las conexiones TCP/UDP y no funcione ICMP para el "*ping*". La razón es porque los PIX implementan un mecanismo automático de listas de acceso para permitir tráfico de entrada cuando se genera tráfico de salida, acorde con los patrones del tráfico de salida, es decir que si generamos conexiones *http* de salida, no podemos tener conexiones *ftp* de entrada. Este mecanismo se implementa de forma automática en los PIX y se conoce como CBACs "*Context Based Access*". Este mecanismo propietario de Cisco Systems sólo funciona en caso de establecimiento de conexiones y/o pseudoconexiones, como TCP/UDP respectivamente, pero no para ICMP.

Por tanto, si generemos tráfico TCP/UDP por ejemplo de "*inside*" a "*outside*", CBACs de forma automática crea y/o genera permisos (como listas de acceso temporales), para permitir tráfico de "*outside*" a "*inside*", respetando que el tráfico sea asociada a la misma conexión.

Podemos observar la tabla de traducciones NAT realizada utilizando el siguiente comando:

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

Paso 5: configuración de NAT (de inside a dmz) en el PIX y establecimiento de conexiones

Como hemos comprobado anteriormente, no podemos acceder a la DMZ porque en ella no se ha asignado ningún proceso de NAT.

Desde "*insidehost*" hacer **telnet 172.16.1.2**.
Desde "*insidehost*" hacer **ping 172.16.1.2**.

Por tanto, a continuación vamos a configurar para que los usuarios de la "*inside*" puedan acceder a la DMZ a través de los siguientes comandos:

```
PIX(config)# global (dmz) 1 172.16.1.200-172.16.1.254 netmask 255.255.255.0
```

Y también podemos comprobar la conexión Telnet a 172.16.1.2, es decir que tenemos permiso para conectarnos al puerto 23 del servidor en la DMZ:

[4] Si no funcionara el *Telnet*, desactivar el servicio DNS moviendo el fichero **/etc/resolv.conf** a otro nombre. Acuérdese después de restaurarlo. Si aún así tampoco funcionara, desactive el firewall, "**services iptables stop"**

Desde "*insidehost*" hacer **telnet 172.16.1.2**. ¿funciona?

En este caso, el tráfico de vuelta también ha sido permitido por CBACs y por eso, el ping sigue sin funcionar.

Desde "*insidehost*" hacer **ping 172.16.1.2**. ¿funciona?

Para analizar el estado de la conexión, vamos a comprobar con los siguientes comandos "show":

```
PIX(config)# show arp
```

_____
_____
_____
_____

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

_____
_____
_____
_____

```
PIX(config)# show conn
```

_____
_____
_____
_____

Paso 6: configuración de NAT estático (de dmz a outside) en el PIX y establecimiento de conexiones

Ahora vamos a tratar de conectarnos desde "*outside*" por telnet al "*bastionhost*". ¿Funciona?. ¿Por qué?

_____
_____

Si desde "*outside*" pretendemos realizar una conexión telnet al "*bastionhost*", podemos comprobar que no funciona. La razón es porque no tenemos definida ninguna traducción NAT, o las entradas de NAT no están inicializadas previamente . Para solucionarlo, en primer lugar vamos a definir una traducción NAT estática, de forma que "*bastionhost*" (o la IP 172.16.1.2) desde el exterior tenga asignada la IP 192.168.1.11 de forma estática, con los siguientes comandos:

```
PIX(config)# static (dmz,outside) 192.168.1.11 bastionhost
```

Una vez que tenemos declarada la traducción NAT `(dmz,outside),` podemos comprobar que aun así ni funciona el telnet ni el ping, desde "*outside*" a "*dmz*":

Desde "*outside*" realizar*:* ***ping 192.168.1.11*** ¿funciona?
Desde "*outside*" realizar*:* ***telnet 192.168.1.11*** ¿funciona?

Sin embargo sí funciona el telnet o http desde "*dmz*" a "*outside*", debido a que CBACs, por ser tráfico TCP/UDP da permiso para el tráfico de vuelta. El ping por tratarse de ICMP, de "*dmz*" a "*outside*" tampoco quedará permitido por CBACs.

Desde "*dmz*" realizar*:* ***ping 172.26.26.1*** ¿funciona?
Desde "*dmz*" realizar*:* ***http://172.26.26.1*** ¿funciona?

Por tanto, en segundo lugar y para completar el acceso desde "*outside*", para que nos funcione la conectividad para el "ping", debemos configurar unas listas de acceso que permitan el tráfico ICMP desde un nivel con seguridad 0 a un nivel con seguridad 50.

Cuando definimos listas de acceso, directamente los mecanismos de niveles de seguridad son ignorados, es decir podemos pasar directamente de nivel de seguridad 0 al 50, o al 100. Para ello introducimos la siguiente configuración:

```
PIX(config)# access-list outside_access_in permit icmp any any
PIX(config)# access-group outside_access_in in interface outside
```

Destacar que la sintaxis de las listas de acceso es similar a las de los *routers* IOS de Cisco Systems, sin embargo el número o identificativo ahora puede ser un nombre (ej. `outside_access_in`), las máscaras son normales, no son *wildmasks* y la asignación a una interfaz se realiza directamente con "`access-group`" especificando la interfaz y siempre con sentido de entrada (*in*). En este caso, configuramos que el tráfico ICMP sea permitido desde cualquier origen (primer "*any*") a cualquier destino (segundo "*any*"), aplicado como entrada en la interface de "*outside*".

Ahora ya podemos hacer "*ping*" al "*bastionhost*" desde "*outside*", desde la zona con menor nivel de seguridad.

Desde "*outside*" realizar*:* ***ping 192.168.1.11*** ¿funciona?

Comprobar la nueva tabla de traducciones:

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

Sin embargo todavía no podemos realizar conexión Telnet. ¿Por qué?

Desde "*outside*" realizar: ***telnet 192.168.1.11*** ¿funciona?

_____

_____

Obviamente, por la misma razón anterior, no tenemos definida una lista de acceso que nos permite acceder por TCP desde el nivel de seguridad 0 al 50, por tanto debemos configurar unas listas de acceso que permitan el tráfico TCP y en el puerto deseado, en nuestro caso para el puerto 23 (Telnet). Para ello introducimos la siguiente configuración:

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

Ahora podemos comprobar que el telnet funciona.

Podemos evaluar el funcionamiento de las listas de acceso, realizando la siguiente consulta en el PIX como se muestra a continuación, donde `hitcnt=17` nos indica las ejecuciones de dicha lista de acceso:

```
      PIX(config)# show access-list
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 1024)
            alert-interval 300
access-list outside_access_in; 2 elements
access-list outside_access_in line 1 permit icmp any any (hitcnt=17)
access-list outside_access_in line 2 permit tcp any host 192.168.1.11 eq telnet (hitcnt=1)
```

<u>Paso 7: configuración de NAT estático (de inside a outside) en el PIX y establecimiento de conexiones</u>

Y finalmente, lo que vamos a configurar es permitir el acceso por NAT estático desde "*outside*" a "*inside*", siguiendo los mismos pasos anteriores para esta nueva combinación de interfaces:

```
PIX(config)# static (inside,outside) 192.168.1.10 insidehost
```

Volvamos a comprobar la conectividad IP utilizando "ping" desde todos los dispositivos a todos los dispositivos. Indicad en la siguientes tabla con una cruz "OK" si funciona correctamente o "NO" si no funciona la conectividad IP con "ping".

Desde ->	**"insidehost"**	**"bastionhost"**	**"PIX"**	**"Router"**
A "insidehost"				
A "bastionhost"				

A “PIX”				
A “Router/Web”				

También podemos comprobar si podemos establecer conexión http desde el “*insidehost*” con el servidor Web (http://172.26.26.1) y hacer Telnet al “*bastionhost*”.

Desde ->	**“insidehost”**	**“bastionhost”**	**“PIX”**	**“Router”**
Telnet a “bastionhost”				
Http a “Router/Web”				

Comenta los resultados obtenidos y justifícalo.

_____
_____
_____
_____

Finalmente, también es importante utilizar las opciones de depuración (debug) para comprobar y solucionar problema. Si realizamos un ping desde el router (*outside)* al *insidehost* y activamos en el PIX la depuración en la interface incide, podemos comprobar su funcionamiento:

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

Si ahora queremos desactivar esta opción, ejecutamos

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

Paso 8: realizar esquema de los permisos habilitados y/o configurados en el PIX

Es buena práctica, para poder entender el complejo proceso de la práctica, realizar una especie de resumen donde queden reflejados los permisos habilitados y/o configurados en el PIX.

Paso 9: dejar las cosas como estaban al principio

Finalizada la práctica, dejaremos la configuración de los equipos y la maqueta tal como esta al principio de empezar la práctica.

ANEXO I: Comandos básico del PIX de Cisco Systems

Los comandos del PIX guardan mucha relación con los comandos de los routes de Cisco Systems y por tanto nos resultarán bastante familiares.

Sin embargo, una diferencia entre la configuración de un *router* y un PIX utilizando CLI (*Command Line Interface*) o modo comandos, es que en el PIX no aparecen modos diferentes de configuración, específicos en lo que estamos configurando, mientras que en el router sí. Es decir, cuando configuramos interfaces en el router aparece "(`config-if`)" y en el PIX sigue apareciendo "(`config`)".

Comandos genéricos y básicos:

- **Help o "?"**
- Para eliminar la configuración de algún comando: "**no**" comando.

`show running-config`	Muestra la configuración en ejecución, como en los routers
`write erase`	Borra la configuración de la memoria Flash
`write memory`	Almacena la configuración actual en la memoria Flash, es el comando equivalente a "copy run star" en los routers IOS de Cisco
`write terminal`	Muestra la configuración

`show history`	Permite visualizar comandos anteriores
`show memory`	Muestra la memoria disponible y ocupada en el PIX
`show version`	Permite comprobar los siguientes valores: *versión de software, último rearranque, tipo de procesador, tipo de memoria flash, tipo de tarjetas , ...*
`show cpu usage`	Muestra la utilización de la CPU

Comandos para configurar interfaces

`interface`	Permite definir la velocidad y modo de funcionamiento
`ip address if_name ip_address [netmask]`	Define la dirección IP de cada interfaz
`nameif hardware_id if_name security_level`	El comando **nameif** define el nombre en una interfaz y le asocia su nivel de seguridad. En el PIX, los nombres por defecto para las dos primeras interfaces son "outside" y "inside" que corresponde con "eth0" y "eth1" con niveles de seguridad 0 y 100 respectivamente. El comando **clear nameif** permite volver a la configuración por defecto de nombres y niveles del PIX.

Comandos para configurar y comprobar NAT

`clear xlate`	Borra la tabla de traducciones NAT.
`global [(if_name)] nat_id {global_ip [-global_ip] [netmask global_mask]} | interface`	Crea y/o modifica el conjunto de IP globales para el NAT.
`static [(prenat_interface, postnat_interface)] {mapped_address| interface} real_address [dns] [netmask mask] [norandomseq] [connection_limit [em_limit]]`	Permite definir una traducción uno a uno o NAT estático, mapeando una IP local en IP global.

`show conn`	Muestra estado de las conexiones.
`show xlate`	Muestra las traducciones NAT utilizadas e información de su conexión.

debug icmp trace	Muestra información del tráfico ICMP.

ANEXO II: Configuración del PIX final

Una vez finalizada la práctica, la configuración que debe quedar debe ser la que se muestra a continuación. Esta configuración será utilizada en prácticas posteriores.

```
PIX Version 6.3(3)
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
hostname PIX
fixup protocol dns maximum-length 512
fixup protocol ftp 21
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol http 80
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol sip 5060
fixup protocol sip udp 5060
fixup protocol skinny 2000
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol tftp 69
names
name 172.16.1.2 bastionhost
name 10.0.1.11 insidehost
access-list outside_access_in permit icmp any any
access-list outside_access_in permit tcp any host 192.168.1.11 eq telnet
mtu outside 1500
mtu inside 1500
mtu dmz 1500
ip address outside 192.168.1.2 255.255.255.0
ip address inside 10.0.1.1 255.255.255.0
ip address dmz 172.16.1.1 255.255.255.0
ip audit info action alarm
ip audit attack action alarm
pdm history enable
arp timeout 14400
global (outside) 1 192.168.1.200-192.168.1.254 netmask 255.255.255.0
global (dmz) 1 172.16.1.200-172.16.1.254 netmask 255.255.255.0
nat (inside) 1 10.0.1.0 255.255.255.0 0 0
static (dmz,outside) 192.168.1.11 bastionhost netmask 255.255.255.255 0 0
```

```
static (inside,outside) 192.168.1.10 insidehost netmask 255.255.255.255 0 0
access-group outside_access_in in interface outside
route outside 0.0.0.0 0.0.0.0 192.168.1.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
telnet timeout 5
ssh timeout 5
console timeout 0
terminal width 80
```