Cisco Companion Topics

Chapter I

Configuring Cisco PIX Firewalls

= =

In This Chapter

= =

Sometimes you may have a Cisco PIX 501 firewall protecting your DSL based home network. This chapter covers how to configure it and in addition, there are a number of fully commented sample PIX configurations in the appendix in which each line is explained.

It is important to remember that the PIX 501 has two Ethernet interfaces. The named “outside” should always be connected to the Internet and the one labeled “inside” should be connected to your home network. The “outside” interface may sometimes be referred to as the “unprotected” interface and the “inside” interface is frequently referred to as the “protected” one.

Network Address Translation (NAT)

Network address translation is a method used to help conserve the limited number of IP addresses available for internet purposes. The introduction to networking page explains the concept in more detail in addition to other fundamental topics. We will return to the NAT discussion, specifically how to configure it, later on this page, but first a very basic introduction on how to configure and use the PIX.

Accessing the PIX command line

Via The Console Port

Your Cisco PIX will come with a console cable that will allow you to configure your PIX using terminal emulation software such as Hyperterm. Once you’ve set up all your PIX with an IP address you’ll be able to access it via Telnet.

Via Telnet

- One easy way to get access to any device on your network is using the /etc/hosts file. Here you list all the IP addresses of important devices that you may want to access with a corresponding nickname. Here is a sample in which the PIX firewall "pixfw" has the default IP address of 192.168.1.1 on its inside protected interface:

```
#
# Do not remove the following line, or various programs
# that require network functionality will fail.
#
127.0.0.1 localhost.localdomain localhost
192.168.1.1 pixfw
192.168.1.100 bigboy mail.my-site.com
```

- Once connected to the network you can access the PIX via telnet

```
[root@bigboy tmp]# telnet pixfw
Trying 192.168.1.1...
Connected to pixfw.
Escape character is '^]'.
```

- You'll be prompted for a password and will need another password to get into the privileged "enable" mode. If you are directly connected to the console, you should get a similar prompt too. There is no password in a fresh out of the box PIX and simply hitting the "Enter" key will be enough.

```
User Access Verification

Password:
Type help or '?' for a list of available commands.
pixfw> enable
Password: ********
pixfw#
```

- Use the "write terminal" command to see the current configuration. You will want to change your "password" and "enable password" right after completing your initial configuration.

```
# wr term
Building configuration...
: Saved
:
PIX Version 6.2(2)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password dsjf5sdfgsjrgjwk encrypted
passwd sdffg8324dgrggjd encrypted
hostname pixfw
fixup protocol ftp 21
...
...
```

- ALL PIX configuration commands need to be done in configuration mode, by issuing the "configure terminal" command from enable mode prompt.

```
pixfw# conf t
pixfw(config)# "Enter commands here"
pixfw(config)# exit
pixfw#
```

- You can usually delete commands in the configuration by adding the word “no” to the beginning of the command you want to delete. Some commands that can only have a single value won’t accept a “no” to change them and will just be over-written when you issue the new command.

  In the example below, we change the PIX’s name and then delete one of many access control list (ACL) entries attached to the outside (Internet) interface.

```
pixfw# conf t
pixfw(config)# no access-list inbound permit tcp any any eq www
pixfw(config)# hostname firewall
firewall(config)# exit
firewall#
```

- One of the first things you should do is change the default passwords for the PIX.

```
pixfw# conf t
pixfw(config)# enable password enable-password-here
pixfw(config)# passwd telnet-password-here
pixfw(config)# exit
pixfw#
```

  Note: The console password is the one used to gain access from the console or through telnet.

- When you've finished configuring, you can permanently save your changes by using the "write memory" command:

```
pixfw# wr mem
Building configuration...
Cryptochecksum: 3af43873 d35d6f06 51f8c999 180c2342
[OK]
pixfw#
```

Sample PIX Configuration: DHCP

Configuring DSL PPPoE DHCP

- DHCP and DSL require you to get a pppoe password and username from your ISP. Most ISPs have a homepage where you can register to get the username and password, ask customer service for the URL. You should substitute this username and password for "dsl-username" and "dsl-password" below. The VPDN group statements

just assign a username, password, authentication type to a profile, in this case "ISP". The configuration steps are relatively straight forward. (Remember to be in config mode)

```
ip address outside pppoe setroute
ip address inside 192.168.1.1 255.255.255.0
vpdn group ISP request dialout pppoe
vpdn group ISP localname dsl-username
vpdn group ISP ppp authentication pap
vpdn username dsl-username password dsl-password
```

In this example, the IP address of the PIX is 192.168.1.1. As the PIX will be acting as your default gateway to the internet, you will have to set the default gateway on all your servers to be 192.168.1.1 You **must** be using PIX IOS version 6.2 or greater for this to work.

Configuring Cable Modem DHCP

- DHCP configuration for cable modems is much simpler, there is no password requirement like with regular DSL. The command to let your PIX get a DHCP IP address from your ISP is as follows:

```
ip address outside dhcp setroute
ip address inside 192.168.1.1 255.255.255.0
```

In this example, the IP address of the PIX is 192.168.1.1. As the PIX will be acting as your default gateway to the internet, you will have to set the default gateway on all your servers to be 192.168.1.1

NAT Configuration with DHCP

Here we allow any traffic coming in on the inside (private/protected) interface to be NAT-ted to the IP address of the outside (Public/unprotected) interface of the firewall. If DSL - DHCP has assigned an address of 97.158.253.12 then the traffic passing through the firewall, from your protected PCs, will appear to be coming from address 97.158.253.12.

```
global (outside) 1 interface
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
Dynamic DNS Port Forwarding Entries
```

Here we allow all incoming www traffic (on TCP port 80) destined for the firewall's interface to be forwarded to the web server at 192.168.1.100 on port 80 (www). Once configured, you may be able to hit your website using PCs behind your firewall using the firewall's outside interface's IP address as the destination. eg: http://firewall-outside-ip-address

```
access-list inbound permit icmp any any
access-list inbound permit tcp any any eq www
access-group inbound in interface outside
static (inside,outside) tcp interface www 192.168.1.100 www
netmask 255.255.255.255
```

How To Get Static IPs For DSL Cheaply

Many ISP DSL providers offer cheap DHCP (dynamic IP) service. Due to competition they'll even throw in a DSL modem and even a router for free. This service frequently isn't available for users with static IPs which the ISPs frequently feel are businesses. If you really want static IP addresses and are willing to pay the higher monthly fee, then you can reduce your installation costs by:

- Ordering DHCP DSL first with the free modem and/or router
- Upgrade to static IPs a week later. They probably won't ask about the modem and/or router, and it becomes bundled in free.

Sample PIX configuration: DSL - Static IPs

PPOE authentication is only required for DSL DHCP. Once you go for static IPs, the vpdn statements won't be required. In this example internet subnet that has been assigned is 97.158.253.24 with a mask of 255.255.255.248 (/29). The IP address selected for the PIX is 97.158.253.25, the default gateway is 97.158.253.30

If you are converting from dynamic to static IP addresses, you do not need the **vpdn** PIX command statements for static IPs

```
ip address outside 97.158.253.25 255.255.255.248
ip address inside 192.168.1.1 255.255.255.0
route outside 0.0.0.0 0.0.0.0 97.158.253.30
```

In this example, the IP address of the PIX is 192.168.1.1. As the PIX will be acting as your default gateway to the internet, you will have to set the default gateway on all your servers to be 192.168.1.1

Outgoing Connections NAT Configuration

Here we allow connections originating coming from servers connected to the inside (private/protected) interface with an IP address in the range 192.168.1.0 to 192.168.1.255 to be NAT-ted to the IP address of the outside (Public/unprotected) interface of the firewall which is 97.158.253.25 :

```
global (outside) 1 interface
nat (inside) 1 192.168.1.0 255.255.255.0 0 0
```

Incoming Connections NAT Configuration

Here we allow the firewall to handle traffic to a second IP address, namely 97.158.253.26, we then allow all incoming traffic to be forwarded to the protected web server which has an IP address of 192.168.1.100. Only www and DNS (Port 53) traffic is allowed to access it via an access control list applied to the outside interface. Once configured, you won't be able to hit your website from PCs behind your firewall using the public IP address assigned to your web server as the destination. You'll have to ask a friend t check it out.

```
access-list inbound permit icmp any any
access-list inbound permit tcp any host 97.158.253.26 eq www
```

```
access-list inbound permit tcp any host 97.158.253.26 eq 53
access-list inbound permit udp any host 97.158.253.26 eq 53
access-group inbound in interface outside
static (inside,outside) 97.158.253.26 192.168.1.100 netmask
255.255.255.255 0 0
```

Here are some additional TCP ports you may be interested in:

Protocol	Port
FTP	20, 21
SMTP Mail	25
POP3 Mail	110
HTTPS / SSL	443

How To Configure Your PIX To Accept Telnet

The **telnet** command can be used to configure your PIX to accept telnet sessions. By default, it allows connections on the inside interface from the 192.168.1.0 network, as seen below:

```
telnet 192.168.2.0 255.255.255.0 inside
```

Of course, if you change the IP address of the inside interface, you may have to change the statement above.

You can also allow access to the outside interface with a similar command. In the case below we're allowing access from the network 64.251.19.0. I generally wouldn't recommended this, but in some cases the need to do it is unavoidable.

```
telnet 64.251.19.0 255.255.255.0 outside
```

As an added precaution, you can set the PIX to automatically log out telnet sessions that have been inactive for a period of time. Here is an example of a 15 minute timeout period.

```
telnet timeout 15
```

How To Make Your PIX A DHCP Server

Enabling your PIX to be a DHCP server requires very few statements. First you have to enable the feature on the desired interface, which is usually the "inside" interface. The next step is to set the range of IP addresses the PIX's "inside" interface will manage, and finally, you need to state the IP address of the DNS server the DHCP clients will use.

The default DNS address the PIX provides its DHCP clients is the IP address of the "inside" protected interface. If the PIX is configured to get it's Internet IP address from your ISP, then the PIX will automatically become a caching DNS server for your home network. This means that in this case you don't have to use the DNS statement.

```
dhcpd enable inside
dhcpd address 192.168.1.20-192.168.1.30 inside
dhcpd dns 192.168.1.100
```

Basic PIX Troubleshooting

The “show interfaces” Command

The show interfaces command will show you the basic status of the PIX’s interfaces. I’ve included some sample output below:

```
pixfw# show interface
interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0009.e89c.fdaa
  IP address 97.158.253.25, subnet mask 255.255.255.248
  MTU 1500 bytes, BW 10000 Kbit half duplex
        5776596 packets input, 569192486 bytes, 0 no buffer
        Received 5315835 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0
abort
        435752 packets output, 74618166 bytes, 0 underruns
        0 output errors, 3988 collisions, 0 interface resets
        0 babbles, 0 late collisions, 6978 deferred
        2 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) (0/77)
        output queue (curr/max blocks): hardware (0/53) software
(0/1)
…
…
pixfw#
```

Your basic physical connectivity should be OK if the interfaces are seen as being in an “up” state with line protocol being “up”. If line protocol is down, you probably have your PIX incorrectly cabled to the Internet or your home network.

If the interfaces are seen as “administratively down”, then the PIX configuration will most likely have the interfaces configured as being “shutdown” like this:

```
interface ethernet0 10baset shutdown
```

This can be easily corrected. First use the “write terminal” command to confirm the shutdown state. Then you should enter “config” mode and reenter the “interface” command without the word “shutdown” at the end.

```
pixfw(config)# interface ethernet0 10baset
```

The “show interfaces” is also important as it shows you whether you have the correct IP addresses assigned to your interfaces and also the amount of traffic and errors associated with each.

The “show xlate” Command

This command will show whether the PIX is doing NAT correctly. Double check your configuration if there are no translations immediately after trying to access the Internet. NAT failure could also be due to bad cabling which will prevent Internet bound traffic from reaching the PIX at all.

```
aquapix# sh xlate
3 in use, 463 most used
PAT Global 97.158.253.25(38448) Local 192.168.1.105(3367)
PAT Global 97.158.253.25(25838) Local 192.168.1.105(2971)
PAT Global 97.158.253.25(26306) Local 192.168.1.105(3610)
aquapix#
```

Using syslog

A really good method for troubleshooting access control lists (ACLs) and also to view the types of methods people are using to access your site is to use syslog. The Appendix has sample configurations for the PIX.

Other Things To Check

Always make sure your PIX has a:

- correct default route. The default is the one with the lots of zeros.

```
aquapix# show route
        outside 0.0.0.0 0.0.0.0 97.158.253.30 1 DHCP static
        outside 12.210.24.0 255.255.252.0 12.210.27.161 1 CONNECT
static
        inside 192.168.1.0 255.255.255.0 192.168.1.1 1 CONNECT static
aquapix#
```

- default gateway that you can “ping”. In the case above the gateway is 97.158.253.30.

Chapter 2

Configuring Cisco DSL Routers

= =

In This Chapter

= =

This is a simple guide on how to set up your Cisco DSL router for DHCP using PPPoE. The examples in this chapter also show how to configure NAT so you can also have a home / SOHO based website. This page should be suitable for the following Cisco routers:

With Built In DSL Modems

- 800 series
- 1700 / 2600 / 3600 series with the ADSL WIC installed

With External DSL Modems

- 1700 / 2600 / 3600 series

An Introduction to Network Address Translation (NAT)

Network address translation is a method used to help conserve the limited number of IP addresses available for internet purposes. The introduction to networking page explains the concept in more detail in addition to other fundamental topics. We will return to the NAT discussion, specifically how to configure it, later in this chapter, but first a very basic introduction on how to configure and use Cisco DSL routers.

Introduction to accessing the router command line

Via The Console Port

Your Cisco router will come with a console cable that will allow you to configure it using terminal emulation software such as Hyperterm. Once you've set up your router with an IP address you'll be able to access it via Telnet.

Via Telnet

- One easy way to get access to any device on your network is using the /etc/hosts file. Here you list all the IP addresses of important devices that you may want to access with a corresponding nickname. Here is a sample in which the router "ciscorouter" has the IP address 192.168.1.1:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
#
127.0.0.1 localhost.localdomain localhost
192.168.1.1 ciscorouter
192.168.1.100 bigboy mail.my-site.com
```

- Once connected to the network you can access the router via telnet

```
[root@bigboy tmp]# telnet ciscorouter
Trying 192.168.1.1...
Connected to ciscorouter.
Escape character is '^]'.
```

- You'll be prompted for a password and will need another password to get into the privileged "enable" mode. If you are directly connected to the console, you should get a similar prompt too. There is no password in a fresh out of the box Cisco router and simply hitting the "Enter" key will be enough.

```
User Access Verification

Password:
Type help or '?' for a list of available commands.
ciscorouter> enable
Password: ********
ciscorouter#
```

- Use the "show running" command to see the current configuration. You will want to change your "password" and "enable password" right after completing your initial configuration.

```
ciscorouter# show run
Building configuration...
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log datetime localtime
service password-encryption
!
hostname ciscorouter
!
no logging console
no logging monitor
```

```
logging trap debugging
...
...
...
```

- ALL router configuration commands need to be done in configuration mode, by issuing the "configure terminal" command from enable mode prompt.

```
ciscorouter# conf t
ciscorouter(config)# "Enter commands here"
ciscorouter(config)# exit
ciscorouter#
```

- You can usually delete commands in the configuration by adding the word “no” to the beginning of the command you want to delete. Some commands that can only have a single value, won’t accept a “no” to change them and will just be over-written when you issue the new command.

  In the example below, we change the router’s name and then delete one of its many access control list (ACL) entries.

```
ciscorouter# conf t
ciscorouter(config)# no access-list 150 deny ip host 10.1.2.1 host
10.3.2.5
ciscorouter(config)# hostname soho-router
soho-router(config)# exit
soho-router #
```

- One of the first things you should do is change the default paswords for the router.

```
ciscorouter# conf t
ciscorouter(config)# enable secret "enable password here"
ciscorouter(config)# line con 0
ciscorouter(config-line)# password "console password here"
ciscorouter(config-line)# line vty 0 4
ciscorouter(config-line)# password "telnet password here"
ciscorouter(config-line)# ^z
ciscorouter#
```

- When you've finished configuring, you can permanently save your changes by using the "write memory" command:

```
ciscorouter# wr mem
Building configuration...
Cryptochecksum: 3af43873 d35d6f06 51f8c999 180c2342
[OK]
ciscorouter#
```

Sample Configurations

DSL Router With Built-In Modem - DHCP

- DHCP and DSL requires you to get a pppoe password and username from your ISP. Most ISPs have a homepage where you can register to get the username and password, ask customer service for the URL. You should substitute this username and password for PPP "username" and "password" listed below.
- Cisco IOS doesn't support DHCP DSL and NAT. If this is so, then putting an Internet accessible web server on your home network would be impossible using the routers mentioned above in this configuration.
- Here is a sample configuration for a Cisco home router. Some of the commands listed are part of Cisco's default settings. Do the "show run" command before starting to configure your router to see what commands you'll really need.

- Remember to be in "config" mode to enter these commands and remember to do a "write memory" at the end to permanently save the configuration

Cisco DSL Router With Built-in Modem Configuration (DHCP)

```
!
vpdn enable
no vpdn logging

!--- Configure the router's PPPoE client so that it
!--- can setup a session with the ISP
!
vpdn-group pppoe
 request-dialin
  protocol pppoe

!--- Configure the home / SOHO network interface's
!--- IP address
!--- The "ip nat" statement tells your router that
!--- this interface:
!--- 1) uses NAT
!--- 2) is the inside "private" interface
!
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside

!--- Configure the DSL interface
!--- Your ISP may provide you with a different pvc
!--- value not necesarily "1/1"
!
```

```
interface ATM0
 no ip address
 no atm ilmi-keepalive
 bundle-enable
 dsl operating-mode auto
 hold-queue 224 in
!
interface ATM0.1 point-to-point
 pvc 1/1
  pppoe-client dial-pool-number 1



!--- Cisco prefers to run the PPPoE client on a virtual
!--- "dialer" interface
!--- This is tied to the real ATM DSL interface with
the !--- "dialer pool" command. The default ethernet MTU
!--- size has been reduced from 1500 to accommodate
!--- the PPPoE header overhead.
!
!--- The "ip nat" statement tells your router that
!--- this interface:
!--- 1) uses NAT
!--- 2) is the outside "public" interface
!
interface Dialer1
 ip address negotiated
 ip mtu 1492
 ip nat outside
 encapsulation ppp
 dialer pool 1

!--- Here are the commands to configure authentication
!--- with with your ISP. This example uses the "CHAP"
!--- method.
!--- Commands for using the "PAP" method are included at
!--- the end of this box
!
 ppp authentication chap callin
 ppp chap hostname <username>
 ppp chap password <password>
!


!--- Tells the router to NAT all traffic that passes
!--- through it:
!--- 1) From the inside to the outside,
```

```
!--- 2) And whose IP address is in the 192.168.1.0
network
!---    as given in access list 1
!--- 3) Giving it an outside "public" address that is the
!---    same as interface Dialer1 gets from the PPPoE
!---    connection
!
ip nat inside source list 1 interface Dialer1 overload
ip classless
ip route 0.0.0.0 0.0.0.0 dialer1
no ip http server
!
access-list 1 permit 192.168.1 0.0.0.255
```

- If your ISP tells you that you need to do the PAP, and not the CHAP, type of authentication then you'll have to replace the lines:

```
ppp authentication chap callin
ppp chap hostname <username>
ppp chap password <password>
```

with only these two:

```
ppp authentication pap callin
ppp pap sent-username <username> password <password>
```

DSL Router With Built-In Modem - Static IP

- Here is a sample configuration for a Cisco home router with a built-in modem. Some of the commands listed are part of Cisco's default settings. Do the "show run" command before starting to configure your router to see what commands you'll really need.
- This example also shows how to use NAT so you can have a web server / mail server / FTP server etc. in your home network.
- Remember to be in "config" mode to enter these commands and remember to do a "write memory" at the end to permanently save the configuration

Cisco DSL Router With Built-in Modem Configuration (Static IP)

```
Current Configuration:
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
!
hostname ciscorouter
!
```

```
ip subnet-zero
no ip domain-lookup
!
bridge irb

!--- Configure the home / SOHO network interface's IP
address
!--- The "ip nat" statement tells your router that this
!--- interface:
!--- 1) uses NAT
!--- 2) is the inside "private" interface
!
interface Ethernet0
ip address 192.168.1.1 255.255.255.0
ip nat inside
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 pvc 0/35
 encapsulation aal5snap
 !
 bundle-enable
 dsl operating-mode auto
 bridge-group 1
!
!--- Cisco prefers to run the PPPoE client on a virtual
!--- "BVI" interface
!--- This is tied to the real ATM DSL interface with the
!--- "bridge-group" command above.
!--- (The BVI number always matches the bridge-group
number)

!--- The "ip nat" statement tells your router that
!--- this interface:
!--- 1) uses NAT
!--- 2) is the outside "public" interface
!
interface BVI1
 ip address 97.158.253.25 255.255.255.248
 ip nat outside

!--- Tells the router to NAT all traffic that passes
!--- through it:
!--- 1) From the inside to the outside,
!--- 2) And whose IP address is in the 192.168.1.0
network
```

```
!---    as given in access list 1
!--- 3) Must get an outside "public" address that is the
!--- same as interface BVI1
!
ip nat inside source list 1 interface BVI1 overload

!--- This statement performs the static address
!--- translation for the Web server. With this statement,
!--- users trying to reach 97.158.253.26 port 80 (www)
will be
!--- automatically redirected to 192.168.1.100 port 80
!--- (www), which in this case is the Web server.
!---
!
ip nat inside source static tcp 192.168.1.100 80
97.158.253.26 80 extendable
!--- Set your default gateway as provided by your ISP
!
ip classless
ip route 0.0.0.0 0.0.0.0 97.158.253.30
!
access-list 1 permit 192.168.1.0 0.0.0.255

bridge 1 protocol ieee
bridge 1 route ip
!
end
```

DSL Router With External Modem - Static IP

- Here is a sample configuration for a Cisco home router with an external modem. Some of the commands listed are part of Cisco's default settings. Do the "show run" command before starting to configure your router to see what commands you'll really need.
- This example also shows how to use NAT so you can have a web server / mail server / FTP server etc. in your home network.
- Remember to be in "config" mode to enter these commands and remember to do a "write memory" at the end to permanently save the configuration

Cisco Router Connected to DSL via External Modem Configuration (Static IP)

```
Current Configuration:
!
version 12.1
service timestamps debug uptime
```

```
service timestamps log uptime
!
hostname ciscorouter
!
ip subnet-zero
no ip domain-lookup
!

!--- Configure the home / SOHO network interface's IP
address
!--- The "ip nat" statement tells your router that
!--- this interface:
!--- 1) uses NAT
!--- 2) is the inside "private" interface
!
interface Ethernet0
ip address 192.168.1.1 255.255.255.0
ip nat inside



!
interface Ethernet1
 ip address 97.158.253.25 255.255.255.248
 ip nat outside

!--- Tells the router to NAT all traffic that passes
!--- through it:
!--- 1) From the inside to the outside,
!--- 2) And whose IP address is in the 192.168.1.0
network
!---    as given in access list 1
!--- 3) Must get an outside "public" address that is the
!---    same as interface ethernet1
!
ip nat inside source list 1 interface ethernet1 overload

!--- This statement performs the static address
translation
!--- for the Web server.
!--- With this statement, users trying to reach
97.158.253.26
!--- port 80 (www) will be automatically redirected to
!--- 192.168.1.100 port 80 (www), which in this case
!--- is the Web server.
!---
!
ip nat inside source static tcp 192.168.1.100 80
97.158.253.26 80 extendable
```

```
!--- Set your default gateway as provided by your ISP
!
ip classless
ip route 0.0.0.0 0.0.0.0 97.158.253.30

!
access-list 1 permit 192.168.1.0 0.0.0.255

!
end
```

Other NAT Topics

Commonly Used TCP And UDP Ports

Here are some additional TCP ports you may be interested in for NAT "**ip nat inside source static**" statements:

Protocol	Port	Type
FTP	20, 21	TCP
SMTP Mail	25	TCP
POP3 Mail	110	TCP
HTTPS / SSL	443	TCP
DNS	53	UDP

- So for example, the command for SMTP mail would be:

  ```
  ip nat inside source static tcp 192.168.1.100 25 97.158.253.26
  25
  ```

- DNS requires a UDP type NAT statement such as:

  ```
  ip nat inside source static udp 192.168.1.100 53 97.158.253.25
  53
  ```

- To have all traffic trying to reach 97.158.253.26, regardless of port, to be NAT-ted to 192.168.1.100, then you can use the command:

  ```
  ip nat inside source static 192.168.1.100 97.158.253.25
  ```

How To Verify That NAT Is Working Correctly

You can use the show ip nat translation command to determine whether NAT is actually occurring as expected:

```
ciscorouter> enable
Password: ********
ciscorouter#show ip nat translation
Pro Inside global     Inside local      Outside local      Outside
global
tcp 97.158.253.26:80  192.168.1.100:80  --- ---
tcp
97.158.253.26:80  192.168.1.100:80  67.34.217.6:5698  67.34.217.6:
5698
ciscorouter#
```

Cisco uses the following terms for the various IP addresses you'll find in any NAT translation process.

- The Inside local address is the actual IP address of the local server on your home network.
- The Inside global address is the IP address of the server presented to the Internet after NAT.
- The Outside local the actual IP address of the remote computer on its local network.
- The Outside global the IP address of the remote computer as presented on the Internet.

As you can see, in this case, NAT seems to be functioning properly for the web server 192.168.1.100 on the home network

How To Troubleshoot NAT

To troubleshoot NAT after you have logged into the router via Telnet requires you to first activate logging to the telnet terminal with the terminal monitor command and then using the debug ip nat detailed command to visualize the translation process. The example below shows that translation occurs for port 80 traffic (HTTP / www) from address 97.158.253.26 to 192.168.1.100, and more specifically that remote host 67.34.217.6 was communicating with the inside global address of 97.158.253.26.

```
ciscorouter> enable
Password: ********
ciscorouter#term mon
ciscorouter#debug ip nat detailed
IP NAT detailed debugging is on
ciscorouter#
03:29:49: NAT: creating portlist proto 6 globaladdr 97.158.253.26
03:29:49: NAT: Allocated Port for 192.168.1.100 -> 97.158.253.26:
wanted 80 got 80
```

```
03:29:49: NAT: o: tcp (198.133.219.1, 5698) -> (97.158.253.26, 80)
[0]
...
...
...
```

Basic Troubleshooting Topics

The “show interfaces” Command

The show interfaces command will show you the basic status of the router’s interfaces. I’ve included some sample output below:

```
ciscorouter>show interface
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is 0008.e3a0.7e80 (bia 0008.e3a0.7e80)
  Internet address is 172.16.1.1/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 1/75/0/0 (size/max/drops/flushes); Total output
drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     303 packets input, 19256 bytes, 0 no buffer
     Received 13 broadcasts, 0 runts, 0 giants, 0 throttles
     1 input errors, 1 CRC, 1 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     60718 packets output, 5770201 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
…
…
…

ciscorouter>
```

Your basic physical connectivity should be OK if the interfaces are seen as being in an “up” state with line protocol being “up”. If line protocol is down, you probably have your router incorrectly cabled to the Internet or your home network.

If the interfaces are seen as “administratively down”, then the router configuration will most likely have the interfaces configured as being “shutdown” like this:

```
…
```

```
…
…
interface ethernet0
 shutdown
…
…
```

This can be easily corrected. First use the “show running” command to confirm the shutdown state. Then you should enter “config” mode and enter the “no shutdown” command. Here is an example for interface ethernet0.

```
ciscorouter(config)# interface ethernet0
ciscorouter(config-if)# no shutdown
ciscorouter(config-if)#end
ciscorouter# write memory
```

The “show interfaces” is also important as it shows you whether you have the correct IP addresses assigned to your interfaces and also the amount of traffic and errors associated with each.

Using syslog

A really good method for troubleshooting access control lists (ACLs) and also to view the types of methods people are using to access your site is to use syslog. The Appendix has sample configurations for Cisco routers.

Other Things To Check

Always make sure your router has a:

- correct default route. The default is the one with the lots of zeros.

```
ciscorouter>sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 97.158.253.30 to network 0.0.0.0

     192.168.0.0/24 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Ethernet1
S*   0.0.0.0/0 [1/0] via 97.158.253.30
ciscorouter>
```

- default gateway that you can “ping”. In the case above the gateway is 97.158.253.30.

Chapter 3

Configuring Cisco SOHO VPNs

= =

In This Chapter

= =

Here is a brief explanation on how to configure a “permanent” Small Office / Home Office (SOHO) VPN using low end Cisco routers and PIX firewalls.

There is a sample PIX configuration in the appendix in which remote users can use Windows based VPN software on their notebook computers to access the SOHO site by first dialing into their ISP and then connecting to the PIX with the software such as Cisco’s EasyVPN suite. As you can imagine, this “temporary” VPN setup can be quite useful.

Scenario

In this example we have two SOHO offices.

- A VPN needs to be created between the two sites so that they can communicate with each other without the fear of eavesdropping.
- For simplicity, neither site is site wants to invest in a CA certificate service or RSA infrastructure. They prefer to use pre-shared keys.
- The network administrators at both sites are aware that permanent site – to – site VPNs require fixed Internet IP addresses and have upgraded from their basic DHCP services originally provided by their ISPs.

Site1

- uses a private network of 192.168.1.0
- has a router with an external Internet IP address of 97.158.253.25
- uses a Cisco DSL router with a built in DSL modem like the Cisco 800 series of routers.

Site2

- uses a private network of 192.168.2.0
- uses a Cisco router with an external DSL modem or a PIX firewall.
- uses a router (Scenario A) or firewall (Scenario B) with an external Internet IP address of 6.25.232.1

Other Information

The administrator at Site 1 wants to be able to access all the protected servers at site 2 by using their real IP addresses and vice versa. For example; Site 1 will refer to Site 2 servers with their 192.168.2.X IP addresses, not the Internet NAT addresses on the 6.25.232.X network.

VPN Terminologies

Before we begin, it is best to review some basic VPN terminologies in the Linux Home Networking guide.

Site 1 - Router VPN Configuration Steps

There are a number of steps that need to be done to create the VPN.

IKE

Phase 1 of the creation of a VPN tunnel first requires an exchange of the encryption capabilities of the VPN devices at both ends of the tunnel. The second phase involves encrypting the data by either using either:

- Pre-shared keys known to both VPN devices (This is what we'll be using in all the examples below) or
- Keys generated via the RSA methodology or
- Keys obtained from Certification Authorities (CAs)

Cisco router / firewall devices usually require you to configure each of the various combinations of key encryption capabilities available. The device will then send **all** of the combinations to the remote VPN as part of the negotiation to decide which one to use.

- Create an IKE key policy. The policy number "9" identifies it from all other IKE policies that may be configured. This policy requires a pre-shared key.

```
crypto isakmp policy 9
 hash md5
 authentication pre-share
```

I've chosen only one combination for the sake of simplicity, but you could add more like this. If your device is licensed appropriately, and you intend to establish a connection with a Linux VPN device, then you should consider a **3DES** option which Linux

FreeS/WAN prefers. Here is a snippet that includes 3DES and may other policy capabilities.

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
!
crypto isakmp policy 4
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp policy 5
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
crypto isakmp policy 10
 authentication pre-share
 group 2
!
crypto isakmp policy 12
 authentication pre-share
!
crypto isakmp policy 20
 hash md5
 authentication pre-share
 group 2
!
crypto isakmp policy 23
 encr 3des
 hash md5
 authentication pre-share
```

- You'll then need to configure a VPN shared key that can be used between this site and the VPN site at 6.25.232.1

```
crypto isakmp key VPNsecretPASSWORD address 6.25.232.1
```

IPSec

- Set a lifetime for the IPSec Security Associations. A security association is the equivalent of a site – to – site VPN relationship.

```
crypto ipsec security-association lifetime seconds 86400
```

- Configure an access list to define the valid traffic to be directed through the VPN from 192.168.1.0 to 192.168.2.0

```
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
```

- Define which encryption transformations will be used to shield the VPN traffic as it passes over the Internet with the "crypto ipsec transform-set" command. Each “single line” set can be given its own name. In this case we’ve chosen set **s1s2trans** to use one of the most common combinations, **esp-des** and **esp-md5-hmac**.

```
crypto ipsec transform-set s1s2trans esp-des esp-md5-hmac
```

If the remote site prefers to use the more secure 3DES method, (Linux FreeS/WAN only does 3DES) then you may want to replace the above statement with this one:

```
crypto ipsec transform-set s1s2trans esp-3des esp-md5-hmac
```

You can create multiple transform sets depending on your security requirements. For example; you could create a transform set named “weak” with regular DES encryption and another named “strong” using the better 3DES method.

- Create a crypto-map to match the valid traffic defined by the ACL with the transform set we want to use with VPN peer router/firewall at the other site. This example is creating a map entry of priority “10”.

```
crypto map to-site2 10 ipsec-isakmp
 set peer 6.25.232.1
 set transform-set s1s2trans
 match address 101
```

You can add additional map entries to correspond with tunnels to other remote sites with additional priorities. Just remember to create the appropriate access control lists and pre-shared keys. Here is an example of additional map entries using two different transform sets:

```
crypto map to-site2  150 ipsec-isakmp
 set peer 108.112.44.95
 set transform-set s1s2trans
 match address 101
crypto map to-site2  153 ipsec-isakmp
 set peer 4.21.116.23
 set transform-set s1s2trans-strong
 match address 102
crypto map to-site2  158 ipsec-isakmp
 set peer 223.52.37.25
 set transform-set s1s2trans-strong
 set pfs group2
 match address 103
```

- Bind the crypto-map to the external interface of the router.

```
interface BVI1
 crypto map to-site2
```

This example assumes you are using a router with a built in DSL modem. In such a case, the external Internet facing interface would most likely be called BVI1 with a "sister" interface ATM0. Make sure both are configured correctly.

If you are using a router with an external DSL / Cable modem, then there will only be one Internet facing interface to configure. This interface would be usually named either Ethernet0 or Ethernet1 depending on the type of router. The Site 2 configuration uses an external DSL / Cable modem.

Site 1 – Configuration Example

Our SOHO Router (Site #1)

```
Current Configuration:
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
!
hostname soho1
!
ip subnet-zero
no ip domain-lookup
!
bridge irb

!
! *  Configure IKE properties
!
crypto isakmp policy 9
 authentication pre-share
 hash md5
crypto isakmp key VPNsecretPASSWORD address 6.25.232.1

!
! *  Configure IPSec properties
!
crypto ipsec security-association lifetime seconds 86400
crypto ipsec transform-set s1s2trans esp-des esp-md5-hmac

!
! * If the remote site prefers to use 3DES, (Linux FreeS/WAN only does 3DES) then you may want to
! * replace the above statement with this one:
!
! crypto ipsec transform-set s1s2trans esp-3des esp-md5-hmac
!


!
! *  Define the Site1 to Site2 traffic to be encrypted
!
crypto map to-site2 10 ipsec-isakmp
 set peer 6.25.232.1
 set transform-set s1s2trans
 match address 101

!
! *  Give the protected interface an IP address and
! *  and let it know that it should do NAT as a protected
! *  "inside" interface
```

```
!
interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside


interface ATM0
 no ip address
 no atm ilmi-keepalive
 pvc 0/35
 encapsulation aal5snap
 bundle-enable
 dsl operating-mode auto
 bridge-group 1


! *  Encryption will be done on interface BVI1 according to
! *  the crypto map statement

interface BVI1
 ip address 97.158.253.25 255.255.255.248
 ip nat outside
 crypto map to-site2
 ip mtu 1412


! *  Tells the router to NAT all traffic that passes through it:
! *  1) From the inside to the outside,
! *  2) And whose IP address matches those in route map "nonat"
! *  3) Must get an outside "public" address that is the same as
! *     interface BVI1
! *
! *  Replaces the following command used on the basic DSL router page
! *
! *  ip nat inside source list 1 interface BVI1 overload

ip nat inside source route-map nonat interface BVI1 overload

! *  This statement performs the static address translation
! *  for the Web server.
! *  With this statement, users trying to reach 97.158.253.26
! *   will be automatically redirected to 192.168.1.100
! *  which in this case is the Web server.
!
ip nat inside source static 192.168.1.100 97.158.253.26


! *  Set your default gateway as provided by your ISP
! *  Set a route to Site2 via the Tunnel IP of the
! *  router at Site2
!
ip classless
ip route 0.0.0.0 0.0.0.0 97.158.253.30

! *  Encrypt all traffic passing over the tunnel
! *  interface between the two sites
!
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
!

! *  ACL used by route map "nonat" to exclude traffic
! *  between Site1 and Site2 from NAT process as this
```

```
! *  will pass through the VPN tunnel
!
access-list 150 deny   ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
access-list 150 permit ip 192.168.1.0 0.0.0.255 any

! *  Use a route map to define which traffic from the private
! *  network should be included in the NAT process:

route-map nonat permit 10
 match ip address 150
```

Site 2 - Router VPN Configuration Steps (Scenario A)

IKE

- Create an IKE key policy. The policy number "9" identifies it from all other IKE policies that may be configured. This policy requires a pre-shared key

```
crypto isakmp policy 9
 hash md5
 authentication pre-share
```

- Configure a VPN shared key that can be used between this site and the VPN site at 97.158.253.25

```
crypto isakmp key VPNsecretPASSWORD address 97.158.253.25
```

IPSec

- Set a lifetime for the IPSec Security Associations

```
crypto ipsec security-association lifetime seconds 86400
```

- Configure an access list to define the valid traffic to be directed through the VPN from 192.168.1.0 to 192.168.2.0

```
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
```

- Define which transformations will be used to shield the VPN traffic with the "crypto ipsec transform-set" command. Each set can be given its own name.

```
crypto ipsec transform-set s2s1trans esp-des esp-md5-hmac
```

If the remote site prefers to use the more secure 3DES method, (Linux FreeS/WAN only does 3DES) then you may want to replace the above statement with this one:

```
crypto ipsec transform-set s1s2trans esp-3des esp-md5-hmac
```

- Create a crypto-map to match the valid traffic, the transform set, the security-association lifetime with the VPN peer router/firewall at the other site

```
crypto map to-site1 10 ipsec-isakmp
 set peer 6.25.232.1
 set transform-set s1s2trans
 match address 101
```

- Bind the crypto-map to the external interface of the router

```
interface Ethernet1
 crypto map to-site1
```

Site 2 – Configuration Example (Scenario A)

Their SOHO Router (Site #2)

```
Current Configuration:
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
!
hostname soho2
!
ip subnet-zero
no ip domain-lookup


! *  Configure IKE properties
!
crypto isakmp policy 9
 authentication pre-share
 hash md5
crypto isakmp key VPNsecretPASSWORD address 97.158.253.25

!
! *  Configure IPSec properties
!
crypto ipsec security-association lifetime seconds 86400
crypto ipsec transform-set s2s1trans esp-des esp-md5-hmac

!
! * If the remote site prefers to use 3DES, (Linux FreeS/WAN only does 3DES) then you may want to
! * replace the above statement with this one:
! *
! * crypto ipsec transform-set s2s1trans esp-3des esp-md5-hmac
!

!
```

```
! *  Define the Site1 to Site2 traffic to be encrypted
!
crypto map to-site1 10 ipsec-isakmp
 set peer 97.158.253.25
 set transform-set s2s1trans
 match address 101

!
! *  Encryption will be done according to the crypto
! *  map statement
!
interface Ethernet1
 ip address 6.25.232.1 255.255.255.248
 ip nat outside
 crypto map to-site1

!
! *  Give the protected interface an IP address and
! *  and let it know that it should do NAT as a protected
! *  “inside” interface
!

interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside

!
! *  Tells the router to NAT all traffic that passes through it:
! *  1) From the inside to the outside,
! *  2) And whose IP address matches those in route map "nonat"
! *  3) Must get an outside "public" address that is the same as
! *     interface ethernet1
! *
! *  Replaces the following command used on the basic DSL router page
! *
! *  ip nat inside source list 1 interface ethernet1 overload
!
ip nat inside source route-map nonat interface ethernet1 overload


!
! *  Set your default gateway as provided by your ISP
! *  Set a route to Site2 via the Tunnel IP of the router
! *  at Site2
!
ip classless
ip route 0.0.0.0 0.0.0.0 6.25.232.6

!
! *  Encrypt all traffic passing over the tunnel interface
! *  between the two sites
!
access-list 101 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255

!
! *  ACL used by route map "nonat" to exclude traffic between
! *  Site1 and Site2
! *  from NAT process as this will pass through the VPN tunnel
!
access-list 150 deny   ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
access-list 150 permit ip 192.168.2.0 0.0.0.255 any
```

```
!
! *  Use a route map to define which traffic from the private
! *  network should be included in the NAT process:
!
route-map nonat permit 10
 match ip address 150
```

Site 2 – PIX Firewall VPN Config. Steps (Scenario B)

IKE

- Plan on creating an IPSec policy with a unique identifier number. The PIX will check each set of configured numbered policies for IKE till it achieves success. In this case we'll only use one policy "20".
- Define the type of encryption to be used (DES or 3DES)

```
isakmp policy 20 encryption des
```

- Define the hashing method for authentication (SHA or MD5)

```
isakmp policy 20 hash md5
```

- Define the overall authentication method (Pre-shared key or rsa-sig). We'll use the simpler pre-shared method.

```
isakmp policy 20 authentication pre-share
```

- Define the shared key to be used.

```
isakmp key VPNsecretPASSWORD address 97.158.253.25 netmask
255.255.255.255
```

- Specify how the hosts will identify themselves to one another (By address or hostname). The same method should be used on both ends.

```
isakmp identity address
```

- Enable ISAKMP on the external interface of the PIX

```
isakmp enable outside
```

IPSec

- Configure an access list to define the valid traffic to be directed through the VPN from 192.168.2.0 to 192.168.1.0

```
access-list ipsec permit ip 192.168.2.0 255.255.255.0 192.168.1.0
255.255.255.0
```

- Define which transformations will be used to shield the VPN traffic with the "crypto ipsec transform-set" command. Each set can be given its own name, in this case "s2s1trans".

```
crypto ipsec transform-set s2s1trans esp-des esp-md5-hmac
```

If the remote site prefers to use the more secure 3DES method, (Linux FreeS/WAN only does 3DES) then you may want to replace the above statement with this one:

```
crypto ipsec transform-set s1s2trans esp-3des esp-md5-hmac
```

- Create a crypto map to match the valid traffic, the transform set, the security-association lifetime with the VPN peer router/firewall at the other site.

```
crypto map s2s1ipsec 10 match address ipsec
crypto map s2s1ipsec 10 set peer 97.158.253.25
crypto map s2s1ipsec 10 set transform-set s2s1trans
crypto map s2s1ipsec 10 set security-association lifetime seconds
86400
```

In this case the crypto map is named "s2s1ipsec" and each statement has a sequence number or "ranking" of "10". Statements with lower "sequence numbers" are considered before those with higher values.

Just like the routers, you can add more statements for tunnels to other remote VPN devices. You just have to remember to make sure that:

- the **crypto map** statements referring to each remote site uses a unique sequence number,
- that the shared secrets match and
- that corresponding ACLs are created.

- Bind the crypto-map to the external interface on which VPN traffic will originate

```
crypto map s2s1ipsec interface outside
```

- Let the PIX's ASA always implicitly allow IPSec traffic through

```
sysopt connection permit-ipsec
```

Site 2 – Configuration Example (Scenario B)

Here is a sample configuration for Site 2 when using a PIX firewall. There are a number of fully commented sample PIX configurations in the appendix in which each line is explained.

Our SOHO PIX (Site #2)

```
PIX Version 6.2(2)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password uR0ZSMuMGz09CMpz encrypted
passwd uR0ZSMuMGz09CMpz encrypted
hostname ciscopix
domain-name stcla1.sfba.home.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names

!
! *  Allow IPSec traffic from Site2's private
! *  network to Site1's private network
!
access-list ipsec permit ip 192.168.2.0 255.255.255.0 192.168.1.0 255.255.255.0

!
! *  Do not Network Address Translate (NAT) traffic
! *  originating on Site2's private network destined
! *  to Site1's private network. This ACL is the first
! *  step.
!
access-list nonat permit ip 192.168.2.0 255.255.255.0 192.168.1.0 255.255.255.0

pager lines 25
logging on
logging timestamp
logging trap warnings
logging history warnings
logging facility 22
logging host inside 192.168.2.237
interface ethernet0 10baset
interface ethernet1 10full
icmp deny any outside
mtu outside 1500
mtu inside 1500

! *  Setup the IP addresses of the interfaces

ip address outside 6.25.232.1 255.255.255.248
ip address inside 192.168.2.1 255.255.255.0
ip audit info action alarm
ip audit attack action alarm
```

```
pdm logging informational 100
pdm history enable
arp timeout 14400
global (outside) 1 interface

!
! *  Do not NAT traffic that matches access list "nonat",
! *  NAT everything else
!
nat (inside) 0 access-list nonat
nat (inside) 1 192.168.2.0 255.255.255.255 0 0

route outside 0.0.0.0 0.0.0.0 6.25.232.6 1
timeout xlate 0:05:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
filter java 80 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
filter activex 80 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
filter java 80 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
filter activex 80 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
ntp server 192.168.2.237 source inside
http server enable
http 192.168.2.0 255.255.255.0 inside
snmp-server host inside 192.168.2.237
no snmp-server location
no snmp-server contact
snmp-server community passwdboo
snmp-server enable traps
tftp-server inside 192.168.2.237 /ciscopix-confg
floodguard enable
no sysopt route dnat
telnet 192.168.2.0 255.255.255.0 inside
telnet timeout 15
ssh 192.168.2.0 255.255.255.0 inside
ssh timeout 15
dhcpd address 192.168.2.20-192.168.2.30 inside
dhcpd lease 3600
dhcpd ping_timeout 750
dhcpd auto_config outside

!
! *  IPSec policies:
!
 sysopt connection permit-ipsec
crypto ipsec transform-set s2s1trans esp-des esp-md5-hmac

!
! * If the remote site prefers to use the more secure 3DES method, (Linux FreeS/WAN only does 3DES)
! * then you may want to replace the above statement with this one:
!
! * crypto ipsec transform-set s2s1trans esp-3des esp-md5-hmac
!


crypto map s2s1ipsec 10 set security-association lifetime seconds 86400
crypto map s2s1ipsec 10 ipsec-isakmp
crypto map s2s1ipsec 10 match address ipsec
crypto map s2s1ipsec 10 set peer 97.158.253.25
crypto map s2s1ipsec 10 set transform-set s2s1trans
```

```
crypto map s2s1ipsec interface outside

!
! *  IKE policies:
!

isakmp enable outside
isakmp key VPNsecretPASSWORD address 97.158.253.25 netmask 255.255.255.255
isakmp identity address
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption des
isakmp policy 20 hash md5
isakmp policy 20 group 1

terminal width 80
Cryptochecksum:3af43873d35d6f0651f8c999180c2342
: end
```

Troubleshooting Cisco VPNs

Cisco provides a number of commands to test the status of your site – to – site VPN tunnel. If your tunnel fails to be created you'll need to ensure that all the parameters are set up correctly. The most common failure I've seen is having mismatched isakmp transform sets.

Displaying the Key Exchange Status

The "**show crypto isakmp sa**" command works on both routers and PIX firewalls and is used to determine whether the first phase of the VPN tunnel establishment (isakmp key exchange) was successful. In the example below Site 1 & 2 have a working tunnel with the status output showing the Internet IP addresses of the VPN devices at both ends of the tunnels.

```
soho1# show crypto isakmp sa
Total     : 1
Embryonic : 0
        dst               src      state     pending     created
 6.25.232.1       97.158.253.25    QM_IDLE           0           0
soho1#
```

Displaying the IPSec Tunnel Status

The "**show crypto ipsec sa**" command works on both routers and PIX firewalls and is used to determine whether the second phase of the VPN tunnel establishment (IPSec) was successful. In the example below Site 1 & 2 have a working tunnel with the status output showing the Internet IP addresses of the VPN devices at both ends of the tunnels.

```
soho1# sh crypto ipsec sa

interface: BVI1
    Crypto map tag: to-site2, local addr. 6.25.232.1

   local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/0/0)
   current_peer: 97.158.253.25:500
```

```
      PERMIT, flags={origin_is_acl,}
     #pkts encaps: 871118, #pkts encrypt: 871118, #pkts digest 871118
     #pkts decaps: 917581, #pkts decrypt: 917581, #pkts verify 917581
     #pkts compressed: 0, #pkts decompressed: 0
     #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
     #send errors 99, #recv errors 0

      local crypto endpt.: 6.25.232.1, remote crypto endpt.: 97.158.253.25
      path mtu 1500, ipsec overhead 56, media mtu 1500
      current outbound spi: 95992f5

      inbound esp sas:
       spi: 0xe43e931d(3829306141)
         transform: esp-des esp-md5-hmac ,
         in use settings ={Tunnel, }
         slot: 0, conn id: 6, crypto map: to-site2
         sa timing: remaining key lifetime (k/sec): (4601836/22657)
         IV size: 8 bytes
         replay detection support: Y

     …
     …

      outbound esp sas:
       spi: 0x95992f5(156865269)
         transform: esp-des esp-md5-hmac ,
         in use settings ={Tunnel, }
         slot: 0, conn id: 5, crypto map: to-site2
         sa timing: remaining key lifetime (k/sec): (4605007/22656)
         IV size: 8 bytes
         replay detection support: Y
     …
     …

soho1#
```

Debugging

Cisco has the very useful **debug** set of commands which you can use to follow the sequence of events that occur during the establishment of the VPN tunnel. Unfortunately the use of the debug command is beyond the scope of this book.

Miscellaneous Cisco Topics

= =

In This Chapter

= =

We briefly discuss some miscellaneous topics in this chapter that are beyond the scope of this book with the intention that you will be stimulated to consider utilizing some of the technologies discussed to improve your website and / or your SOHO network.

Syslog Configuration and Cisco Devices

Syslog reserves facilities "local0" through "local7" for log messages received from remote servers and network devices. Routers, switches, firewalls and load balancers each logging with a different facility can each have their own log files for easy troubleshooting. The following examples will show how to have a different log file for each class of device.

If you have a large data center, then you may also want to switch off all logging to **/var/log/messages** as suggested above for the home/SOHO environment. In all the network device configuration examples below we are logging to the remote Linux logging server 192.168.1.100 which we set up in the previous section.

Cisco Routers

By default Cisco routers send syslog messages to their logging server with a default facility of local7. We won't set the facility in this case, but we can tell the router to timestamp the messages and make the messages have the source IP address of the loopback interface.

```
service timestamps log datetime localtime
no logging console
no logging monitor
logging 192.168.1.100
```

Catalyst CAT Switches running CATOS

By default Cisco switches also send syslog messages to their logging server with a default facility of local7. We won't change this facility either, therefore making routers and switches log to the same file.

```
set logging server enable
set logging server 192.168.1.100
set logging level all 5
set logging server severity 6
```

Cisco Local Director

Local Directors use the "syslog output" command to set their logging facility and severity. The value provided must be in the format **FF.SS** (facility.severity) using the numbering scheme below:

Facility	FF Value
local 0	16
local 1	17
local 2	18
local 3	19
local 4	20
local 5	21
local 6	22
local 7	23

Severity	SS Value
System unusable	0
Immediate action required	1
Critical condition	2
Error conditions	3
Warning conditions	4
Normal but significant conditions	5
Informational messages	6
Debugging messages	7

Here we using facility LOCAL4 and logging debugging messages and above.

```
syslog output 20.7
no syslog console
syslog host 192.168.1.100
```

Cisco PIX Filewalls

PIX firewalls use the following numbering scheme to determine their logging facilities.

Facility	Logging Facility Command Value
local 0	16
local 1	17
local 2	18
local 3	19
local 4	20
local 5	21
local 6	22
local 7	23

This configuration example assumes that the logging server is connected on the side of the "inside" protected interface. We're sending log messages to facility LOCAL3 with a severity level of 5 (Notification) set by the "logging trap" command.

```
logging on
logging standby
logging timestamp
logging trap notifications
logging facility 19
logging host inside 192.168.1.100
```

Cisco CSS11000 (Arrowpoints)

This configuration for this is more straight forward. You specify the facility with an intuitive number using the "logging host" command and set the severity with the "logging subsystem" command. This example shows the CSS11000 logging facility LOCAL 6 and severity level 6 (Informational)

```
logging host 192.168.1.100 facility 6
set logging subsystem all info-6
logging commands enable
```

The Sample Cisco syslog.conf File

```
#
# All LOCAL3 messages (debug and above) go to the firewall file
ciscofw
#
local3.debug /var/log/cisco/ciscofw

#
# All LOCAL4 messages  (debug and above) go to the Local Director
file ciscold
#
local4.debug /var/log/cisco/ciscold

#
# All LOCAL6 messages  (debug and above) go to the CSS file
ciscocss
#
local6.debug /var/log/cisco/ciscocss

#
# All LOCAL7 messages  (debug and above) go to the ciscoacl
# This includes ACL logs which are logged at severity debug
#
local7.debug /var/log/cisco/ciscoacl

#
# LOCAL7 messages  (notice and above) go to the ciscoinfo
# This excludes ACL logs which are logged at severity debug
#
local7.notice /var/log/cisco/ciscoinfo
```