

Safepipe interoperability with Cisco Secure Pix 506 Firewall

A tech note prepared by Eicon Networks July 2001

Summary

Safepipe is interoperable with the Cisco Secure Pix 506 Firewall. In a test scenario the following was established:

- *Tunnels can successfully be established between the two devices.*
- *Use of authentication mode (preshared keys) can successfully be negotiated.*
- *Use of authentication algorithm (MD5) can successfully be negotiated.*
- *Use of encryption algorithm (triple-DES) can successfully be negotiated.*
- *Use of Diffie-Hellman group (group 2 (1024bit)) can successfully be negotiated.*
- *Lifetime can successfully be negotiated.*
- *Data packets can successfully be exchanged through the tunnels.*

Structure of this document:

Introduction	4
Aims and objectives	4
Test scope and example network	5
Known limitations	5
Configuration on Cisco Pix	6
Defining an access list	6
Preventing NAT in VPN tunnel	6
Enabling the IPSec system option	7
Defining a transform set for phase 1	7
Defining a transform set for phase 2	7
Complete configuration	8
Configuration on Safepipe	11
Configuring the tunnel	11
Verifying that traffic is able to pass through the tunnel	13
Viewing the Cisco Pix log	13
Viewing the Safepipe tunnel log	13
Conclusion	14

Introduction

As VPN standards evolve not all implementations by the various vendors of VPN devices may be interoperable. One of Safepipe's strengths is its ability to interoperate with several VPN solutions offered by other vendors. This document describes a successful test of Safepipe's interoperability with the Cisco Secure Pix 506 Firewall (hereafter: Cisco Pix), and details the steps needed to configure the two devices for interoperability.

Please note that this tech note assumes a familiarity with Safepipe and the Cisco Pix. It describes VPN tunnelling on a test network; if recreating the scenario, substitute any names, IP addresses, etc. with names and addresses relevant to your own network.

Aims and objectives

The aim of this test described in this document was to determine whether Safepipe and the Cisco Pix were interoperable, i.e. whether a functional tunnel could be established between the two devices.

In more detail, we needed:

1. To determine whether an authentication mode could be negotiated between the two devices.
2. To determine whether an authentication algorithm could be negotiated between the two devices.
3. To determine whether encryption with a triple-DES algorithm could be negotiated between the two devices.
4. To determine whether a Diffie-Hellman group could be negotiated between the two devices.
5. To determine whether lifetime could be negotiated between the two devices.
6. To determine whether exchange of data through the tunnel was possible.

Test scope and example network

The test was performed using a Safepipe 50 with software version 2.3 and a Cisco Secure Pix 506 Firewall device with software version 6.01 (3-DES).

The test network was set up with a workstation on each of the private networks connected to the private interfaces of the two VPN devices. Before the test, the various interfaces were configured with IP addresses.

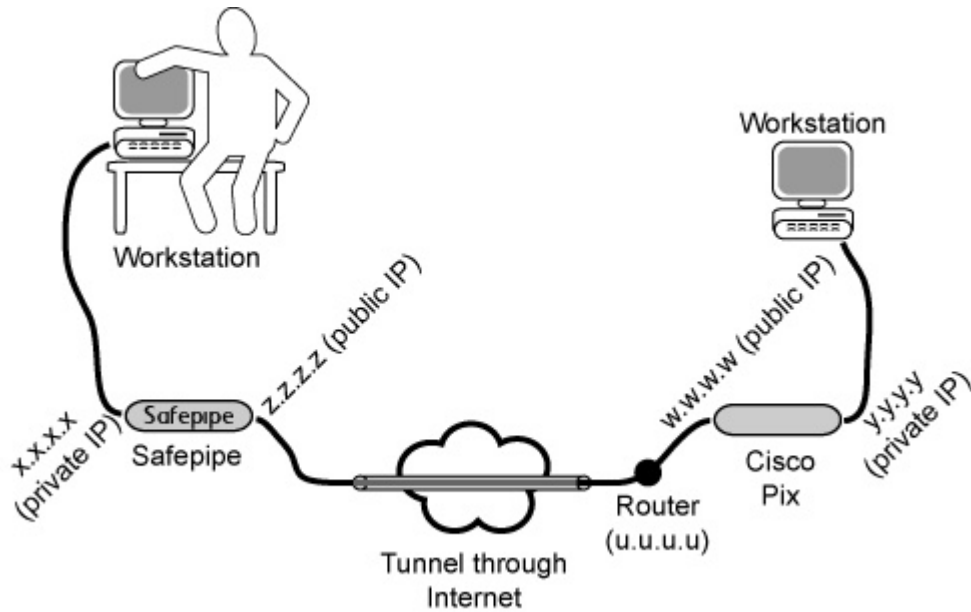


Diagram of test network

Known limitations

It was decided to let Safepipe initiate the VPN tunnel. A test in which the Cisco Pix initiates the VPN tunnel was not carried out. No extensive performance test was carried out.

Configuration on Cisco Pix

The following configuration requires that the Cisco Pix has been initially configured with IP addresses, etc., that it has been connected according to the **example network diagram** in this document, and that the user has console access to the Cisco Pix.

Defining an access list

An access list controls what traffic is allowed. We need to define an access list which specifies that traffic is allowed both ways between the two private networks in the VPN.

In this case the name of the access list is "100". The access list configuration looks like this (an example of the *complete* Cisco Pix configuration is available on page 8 of this document):

```
access-list 100 permit ip y.y.y.0 255.255.255.0 x.x.x.0 255.255.255.0
access-list 100 permit ip x.x.x.0 255.255.255.0 y.y.y.0 255.255.255.0
```

Please note that *y.y.y.0* denotes the private network behind the Cisco Pix (followed by the subnet mask), and *x.x.x.0* denotes the private network behind the Safepipe (followed by the subnet mask). Substitute these IP addresses and subnet masks as necessary. The above access list thus allows traffic both ways.

Preventing NAT in VPN tunnel

NAT may cause problems if it is enabled in the VPN tunnel. We therefore need to make sure that NAT is not used on the addresses specified in the access list. The necessary configuration looks like this:

```
nat (inside) 0 access-list 100
```

Enabling the IPsec system option

We should now enable the IPsec system option (sysoption). This is done with the following configuration:

```
sysopt connection permit-ipsec
```

Defining a transform set for phase 1

A transform set defines how traffic is going to be protected. We need to configure a transform set for phase 1 (which covers initial negotiation and derivation of keys) that will make the Cisco Pix:

- use 3-DES as encryption algorithm
- use MD5 as hashing algorithm
- use IPsec isakmp (Internet Security Association and Key Management Protocol)
- match the access list we created earlier with the name “100”
- use the Safepipe as peer

We will call this transform set “safepipe_set”. The configuration looks like this:

```
crypto ipsec transform-set safepipe_set esp-3des esp-md5-hmac
crypto map safemap 10 ipsec-isakmp
crypto map safemap 10 match address 100
crypto map safemap 10 set peer z.z.z.z
crypto map safemap 10 set transform-set safepipe_set
crypto map safemap interface outside
```

Please note that z.z.z.z denotes the Safepipe’s public IP address. Substitute the IP address as necessary.

Defining a transform set for phase 2

Likewise, a transform set must be defined for phase 2 (which covers negotiation of security associations and keys that protect the actual application data exchanges). In addition to information about encryption and hashing, this transform set will include information about which pre-shared key (a.k.a. “shared secret” on Safepipe), which

Diffie-Hellman group, and which lifetime the Cisco Pix should use. However, it should be said that, as the Safepipe initiated the tunnel in our test, the default lifetime value suggested by the Safepipe would be used under all circumstances. The configuration looks like this:

```
isakmp enable outside
isakmp key ***** address z.z.z.z netmask 255.255.255.255
isakmp identity address
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption 3des
isakmp policy 10 hash md5
isakmp policy 10 group 2
isakmp policy 10 lifetime 1000
```

Please note that ***** denotes the pre-shared key (shared secret), and that z.z.z.z denotes the SafePipe's public IP address. Substitute IP address and the associated subnet mask as necessary.

Complete configuration

For the sake of simplicity we have above described parts of the configuration in isolation. However, the configuration needs to be complete in order to work. The complete configuration for the Cisco Pix used in the test looks as follows:

```
PIX Version 6.0(1)

nameif ethernet0 outside security0
nameif ethernet1 inside security100

enable password AbCdEfGhIjKlMnOp encrypted
passwd aBcDeFgHiJ.kLmN encrypted
hostname pix
domain-name company.com

fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
```

```

fixup protocol skinny 2000

! *** Access lists for VPN
access-list 100 permit ip y.y.y.0 255.255.255.0 x.x.x.0 255.255.255.0
access-list 100 permit ip x.x.x.0 255.255.255.0 y.y.y.0 255.255.255.0

interface ethernet0 auto
interface ethernet1 auto

mtu outside 1500
mtu inside 1500

ip address outside w.w.w.w 255.255.255.224
ip address inside y.y.y.y 255.240.0.0

global (outside) 1 interface

! *** Do not NAT in VPN tunnel
nat (inside) 0 access-list 100

nat (inside) 1 0.0.0.0 0.0.0.0 0 0
route outside 0.0.0.0 0.0.0.0 u.u.u.u 1

!enable sysopt IPSEC
sysopt connection permit-ipsec

! *** Phase 1 ***
crypto ipsec transform-set safepipe_set esp-3des esp-md5-hmac
crypto map safemap 10 ipsec-isakmp
crypto map safemap 10 match address 100
crypto map safemap 10 set peer z.z.z.z
crypto map safemap 10 set transform-set safepipe_set
crypto map safemap interface outside

! *** Phase 2 ***
isakmp enable outside
isakmp key ***** address z.z.z.z netmask 255.255.255.255
isakmp identity address
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption 3des
isakmp policy 10 hash md5
isakmp policy 10 group 2
isakmp policy 10 lifetime 1000

end
[OK]
pix#

```

Lines like `! *** Access lists for VPN` are included as headings for easy overview and can be omitted.

Please note that hostnames, domain names, IP addresses, subnet masks, etc. should be substituted with your relevant values if recreating the scenario. In the above configuration IP addresses are shown in a dummy format (e.g. x.x.x.x). To help you make a correct configuration for your VPN, substitute the dummy values using the following guidelines:

- y.y.y.0 denotes the private network behind the Cisco Pix
- x.x.x.0 denotes the private network behind the Safepipe
- u.u.u.u denotes the IP address of the router between the Cisco Pix and the Internet
- w.w.w.w denotes the public side IP address of the Cisco Pix
- y.y.y.y denotes the private side IP address of the Cisco Pix
- z.z.z.z denotes the public side IP address of the Safepipe
- x.x.x.x denotes the private side IP address of the Safepipe
- ***** denotes the pre-shared key (shared secret)

Configuration on Safepipe

The following configuration requires that the Safepipe device has been initially configured with IP addresses, etc., that it has been connected according to the example network diagram in this document, and that the user has launched and logged on Safepipe's browser-based user interface.

Configuring the tunnel

From Safepipe's menu, select Tunnels > Restricted.

Click the Add Tunnel button. Click the <Not Configured> link that appears.

A tunnel configuration page appears:

Tunnel		
Tunnel Name:	Cisco Pix	Enter a suitable tunnel name.
IP Address of remote device:	www.www	Enter the public interface IP address of the remote Safepipe.
Authentication Method:	<input type="radio"/> x 509 Certificate <input checked="" type="radio"/> Shared secret	Select Shared Secret or X 509. With Shared Secret, the text box must contain the same secret in both ends. With X 509, the text box must contain the ID of the remote end.
	secret	
Local Network:	xxx.x0	Enter the IP subnet and mask of the local private network.
Local Subnet Mask:	255.255.255.0	
Remote Network:	yyy.y0	Enter the IP subnet and mask of the remote private network.
Remote Subnet Mask:	255.255.255.0	
<input type="checkbox"/> IP Compression		Compress data sent through the tunnel.
<input type="checkbox"/> NAT Tunneling		Use NAT tunneling when Safepipe or the remote VPN device is located behind a firewall.
<input checked="" type="checkbox"/> NetBIOS		Enable/Disable NetBIOS.
<input checked="" type="checkbox"/> Initiate tunnel		Initiate tunnel after configuring.

Tunnel configuration page on Safepipe

- Enter a suitable **Tunnel Name** (in this example the name "Cisco Pix" was used).
- Enter the **IP address of the public interface on the Cisco Pix** in the **IP Address of remote device** field.
- Select **Shared secret** as the **Authentication Method** to use, and enter the agreed preshared key (known as "shared secret" on Safepipe) in the associated field.

- Enter the **IP address of the network on the private side of the Safepipe** in the **Local Network** field.
- Enter the associated **Subnet mask** in the **Local Subnet Mask** field.
- Enter the **IP address of the network on the private side of the Cisco Pix** in the **Remote Network** field.
- Enter the associated **subnet mask** in the **Remote Subnet Mask** field.
- Leave the **IP Compression** box **unchecked**.
- Leave the **NAT Tunnelling** box **unchecked**.
- Check the **NetBIOS** box only if you intend to allow NetBIOS traffic, otherwise leave it unchecked.
- Check the **Initiate tunnel** in order for Safepipe to be the initiating party in the tunnel. With the **Initiate tunnel** box checked, Safepipe was able to successfully initiate a tunnel to the Cisco Pix. We did not attempt to have the Cisco Pix initiate a tunnel.

Click the Apply Changes button.

Verifying that traffic is able to pass through the tunnel

Once each of the tunnel endpoint devices had been configured as described, we established that data could successfully be exchanged through the tunnels. This was done by having the workstation located on the private side of the Safepipe successfully ping the workstation located on the private side of the Cisco Pix through the tunnel, and vice versa.

We used the log features of the two VPN devices to verify that tunnels had indeed been established according to configuration and that the ping data traffic we sent was thus able to pass securely through the tunnels.

Viewing the Cisco Pix log

On the Cisco Pix, use the following command to generate a log describing the negotiation process:

```
debug crypto isakmp
```

Viewing the Safepipe tunnel log

On the menu of the Safepipe, select **Tunnels > Restricted**.

Click the **Name** link of the tunnel in question.

Click the **Log** tab to view the log for the tunnel in question.

Conclusion

We were able to successfully establish a VPN tunnel between Safepipe and the Cisco Pix. Safepipe was able to initiate a tunnel to the Cisco Pix. We did not test with the Cisco Pix as tunnel initiator. By monitoring logs on the two devices we were able to verify that negotiation had been successful.

Successful negotiation included: negotiation of authentication mode (preshared keys, a.k.a. shared secret), negotiation of authentication algorithm (MD5), negotiation of encryption algorithm (triple-DES), negotiation of Diffie-Hellman group (group 2 (1024bit)) and negotiation of lifetime (although, with Safepipe being the tunnel initiator, lifetime was fully determined by Safepipe's default lifetime).

By having workstations located on the private networks at each end of the tunnel successfully ping each other through the tunnel, we established that data could be exchanged through the tunnels.

We did not test performance further.