

# *Wind River Documentation Profile for VPNC Interoperability*

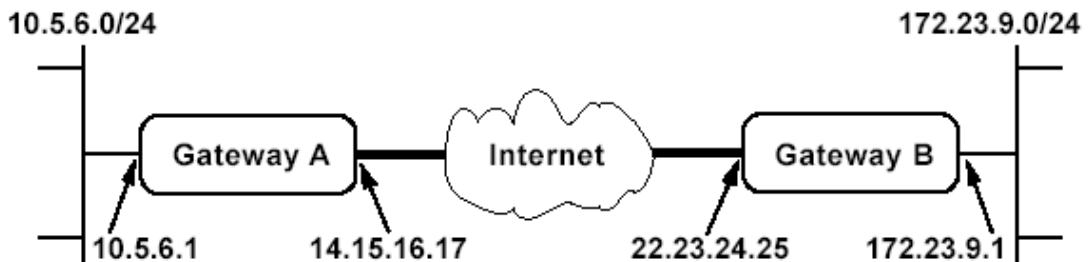
*WIND NET IPsec and IKE, 3.0*

## 1. Overview

This document describes how to configure WIND NET IPsec and IKE 3.0 for interoperability testing by VPNC using “Scenario 1: Gateway-to-Gateway with Preshared Secrets”.

## 2. Scenario 1: Gateway-to-Gateway with Preshared Secrets

The following is a typical gateway-to-gateway VPN that uses a preshared secret for authentication:



Gateway A connects the internal LAN 10.5.6.0/24 to the Internet. Gateway A's LAN interface has the address 10.5.6.1, and its WAN (Internet) interface has the address 14.15.16.17.

Gateway B connects the internal LAN 172.23.9.0/24 to the Internet. Gateway B's WAN (Internet) interface has the address 22.23.24.25. Gateway B's LAN interface address, 172.23.9.1, can be used for testing IPsec but is not needed for configuring Gateway A.

The IKE Phase 1 parameters used in Scenario 1 are:

- Main mode
- TripleDES
- SHA-1
- MODP group 2 (1024 bits)
- pre-shared secret of "hr5xb84l6aa9r6"
- SA lifetime of 28800 seconds (eight hours) with no kbytes rekeying

The IKE Phase 2 parameters used in Scenario 1 are:

- TripleDES
- SHA-1
- ESP tunnel mode
- MODP group 2 (1024 bits)
- Perfect forward secrecy for rekeying
- SA lifetime of 3600 seconds (one hour) with no kbytes rekeying
- Selectors for all IP protocols, all ports, between 10.5.6.0/24 and 172.23.9.0/24, using IPv4 subnets

## 3. Setup

### 3.1 Physical System Setup

The test target is a Pentium box containing a serial port (COM1), a floppy drive, and two network interface cards, elPci0 and elPci1, which together constitute a gateway. Set up the system as follows:

1. Using a serial cable, connect COM1 of the target to a serial port on a host such as a PC. This connection is used to establish a console session from the host to

the target, allowing interaction with the target where commands can be issued and status monitored.

2. Connect the eIPci0 card to the public network (WAN or Internet).
3. Connect the eIPci1 card to the private network (internal LAN).
4. Insert the boot floppy provided into the floppy drive. The target is set up to boot from the floppy. The VxWorks executable containing the IPsec product image and the startup script will be auto-downloaded from an FTP server.

### 3.2 FTP Parameters

As part of the boot process the target will connect to an FTP server to download the VxWorks executable image (containing the IPsec code) and the startup script. The parameters are displayed on the serial console upon bootup. The process for setting or modifying the parameters is described in Appendix A.

### 3.3 Startup Script

As mentioned previously, the startup script runs automatically at target bootup, to set the IPsec policies and IKE parameters. The script is a text file with embedded comments describing the parameters being set. Its contents are shown below.

#### Setup of Network Interfaces

**ikeSetIfAddr("14.15.16.17")**

Binds IKE to an IP address. This is the public address of the gateway.

**ipsecAttachIf("14.15.16.17")**

Binds a network interface to IPsec. This is the public address of the gateway.

**ipAttach(1,"eIPci")**

**ifconfig("eIPci1 10.5.6.1")**

Sets up the secondary network interface card. This is the private address of the gateway.

## Phase 1 Setup

### **ikeSetXform ("ph1\_xform\_1,3DES,SHA")**

Creates a Phase 1 transform named **ph1\_xform\_1** with TripleDES encryption and SHA-1 hash.

### **ikeSetProp ("ph1\_prop\_1,ph1\_xform\_1")**

Creates a Phase 1 proposal named **ph1\_prop\_1** based on the previously defined transform.

### **ikeSetPropAttrib("ph1\_prop\_1,DHGROUP,G2,LIFETIME,28800,UNITOFTIME,SECS")**

Sets attributes of the previously defined proposal using parameter name/value pairs. MODP group2, SA lifetime of 28800 seconds.

### **ikeAddPeerAuth("22.23.24.25,14.15.16.17,ph1\_prop\_1,KEYPFS,PSK,hr5xb84l6aa9r6")**

Specifies the IKE authentication information between the host and a peer. The command format is described below.

### **ikeAddPeerAuth(configurationString)**

*configurationString* is a character string formatted as follows:

*peerIpAddress,interfaceIpAddress,proposalName,PFS,  
authenticationMethod,authenticationInfo*

where

*peerIpAddress* is the IP address of the IKE peer.

*interfaceIpAddress* is the local IP address that is to communicate with the peer.

*proposalName* is the name of a Phase 1 proposal defined by **ikeSetProp()**.

*PFS* is **NOPFS** if perfect forward secrecy is not required, or **KEYPFS** if perfect forward secrecy of keys is required.

*authenticationMethod* is **PSK**, meaning preshared key.

*authenticationInfo* is the preshared key, represented as printable ASCII.

## Phase 2 Setup

**spdSetESPXform("ph2\_esp\_xform\_1,ESP3DES,SHA")**

Creates a Phase 2 ESP transform named **ph1\_esp\_xform\_1** with TripleDES encryption and SHA-1 hash.

**spdSetProp("ph2\_prop\_1,ph2\_esp\_xform\_1")**

Creates a IKE Phase 2 proposal named **ph2\_prop\_1** and binds it with pre-existing transform.

**spdSetPropAttrib("ph2\_prop\_1,DHGROUP,G2,ENCAP,TUNNEL,HARDLIFETIME,3600")**

Sets attributes of an existing IKE Phase 2 proposal using parameter name/value pairs; DHGROUP to MODP group 2, tunnel mode encapsulation, and SA lifetime of 3600 seconds.

**spdSetSA("sa\_1,ph2\_prop\_1,1")**

Creates a security association (SA) proposal in the SPD, named **sa\_1**, based on the pre-existing proposal **ph2\_prop\_1**, and sets the proposal number to 1.

**spdAddTunnel("ANY/ANY/ANY,172.23.9.0/24,10.5.6.0/24,OUT,POLICY,IKE,sa\_1,22.23.24.25")**

Adds a tunnel mode policy to the SPD for any traffic between the two subnets. The command format is described below.

**spdAddTunnel(configurationString)**

*configurationString* is a character string formatted as follows:

*protocolSelector*[/*destinationPort*/*sourcePort*],*destinationAddressSelector*,  
*sourceAddressSelector*,*directionality*,*useSelectors*,*keyManager*,*saProposalName*,  
*tunnelEndpointAddress*

where:

*protocolSelector* is the IANA IP protocol number, decimal value | ANY  
Use 6 for TCP or 17 for UDP.

*destinationPort* and *sourcePort* are decimal value | ANY

*destinationAddressSelector* and *sourceAddressSelector* are  
*ipAddress1*[-*ipAddress2* | /*ipMaskPrefix*]

*directionality* is IN | OUT

If IN, then this policy applies to traffic coming into the current host.

If **OUT**, it applies to traffic going out of the current host. A mirrored policy will automatically be created for the opposite traffic flow.

*useSelectors* is **PACKET | POLICY**

*keyManager* is **MANUAL | IKE** (either manual or key negotiation).

*saProposalName* is the name of an SA proposal (stringValue).

*tunnelEndpointAddress* is the ipAddress of the remote end of the tunnel

## Routing Table

**route("add default 22.23.24.25")**

Adds an entry to the routing table.

## 3.4 VxWorks Image and Startup Script

Place the VxWorks image containing the IPsec product and the startup script on an FTP server that is reachable from the target. The target downloads the VxWorks image, loads it into memory and starts the various IPsec modules. The target will next download and run the startup script which sets the network interface cards, parameters for IKE Phase 1 and Phase 2, and the policies for communications to other hosts.

## 3.5 System Startup

This section assumes that the target is setup as per description above and that an FTP server is running from which the VxWorks executable and startup script can be downloaded.

From a host such as a PC, open a serial console session to COM1 of the target using a terminal emulation program (such as CRT). Switch the target on. This should result in the target booting up, loading of the VxWorks executable image into memory and IPsec initialization followed by running of the startup script. The following status messages are displayed on the console:

- listing of the boot parameters followed by messages about attaching to network interface card eIPci0
- loading of the image from the ftp server
- initialization status of the various IPsec modules

- VxWorks banner
- echo and status of commands run from the startup script

The system is now fully set up and will operate as a secure gateway.

### 3.6 Rebooting the Target

It may be necessary to re-boot the target between tests so that the target can be in a known state. This can be done by pressing “ctrl-x” on the session console. It will take a few seconds before you see the auto-boot message on the console.

## 4. Testing and Diagnostics

Once the gateway is setup the following commands can be used to monitor it.

### **muxShow( )**

Displays configuration of devices and their associated protocols that are registered with the MUX. You should see two devices; elPci Unit: 0 and elPci Unit:1. As well, TCP/IP protocol should be bound to both devices.

### **spdShow( )**

Displays a list of all security policies in the inbound and outbound databases. You should see a TUNNEL mode policy as configured by the **spdAddTunnel()** command of the startup script in the inbound SPD, and one in the outbound SPD.

### **sadbShow( )**

Displays a list of all SAs in the inbound and outbound SADBs. At this point there should not be any SAs present.

### **ikeShowSessions( )**

Displays information for all (active and inactive) Phase1 SAs. For each session, the following items are printed:

- source address
- destination address

- initiator/responder status
- authentication method
- Diffie-Hellman group (DH)
- hard lifetime
- soft lifetime
- encryption algorithm
- hash algorithm

### **ikeShowPS( )**

Displays information for all (active and inactive) protection suites (inbound and outbound IPsec Security Association pairs). For each protection suite, the following items are displayed:

- protocol number
- source IP address and port
- destination IP address and port
- protection suite number (handle)
- initiator/responder status
- Phase 2 policy handle

## **5. Establishing SAs and IPsec Processing**

Once the SPD is filled with the proper security policies, an IPsec tunnel can be established between the gateways to allow communication between the hosts on the subnets being served by the gateways. This is done by pinging a host on the internal LAN of one gateway from a host on the internal LAN of the other gateway. The first ping will cause IKE Phase 1 and Phase 2 to be negotiated, but since the secure connection does not yet exist, this packet is dropped. Subsequent pings will succeed as they will be processed by the established SAs.

Status messages such as the ones below will appear on the console:

```
IKE: Channel Builder (<initiator|responder>) successfully negotiated phase I
IKE: IPsec Negotiator: <Initiator|Responder> Established a QM connection
IKE: Created protection suite 0x1
```

Messages will also be displayed for deletion and renewals of IKE Phase 1 and Phase 2 SAs.

Typing **sadbShow()** at the console should show tunnel mode SAs in the inbound and outbound SADB.

## 6. Appendix: Changing the Target Boot Parameters

This step should only be performed if any of the target boot parameters need to be modified (e.g. the IP address of the FTP server is changed).

Connect to COM1 of the target and turn the target on or if the target is already running, reboot it by pressing ctrl-x on the serial session console. It will take a few seconds before you see the auto-boot message on the console. When prompted with "Press any key to stop auto-boot...", press any key. There is an 8-second window to do this, after which the boot process will continue. Stopping auto-booting enables you to view the boot parameters. At the "[VxWorks Boot]:" prompt enter "p" to view the boot parameters.

The following is a brief description of some of the parameters:

- file name: vxworks executable image to be downloaded into target
- inet on ethernet: public ip address of target
- host inet: ip address of the ftp server
- user: user name for accessing the ftp server
- ftp password: user password for accessing ftp server
- startup script: IPsec configuration script to be run after bootup

To change any of the above parameters, type "c" at the "[VxWorks Boot]:" prompt and then press the "Enter" key on the keyboard to traverse through the parameters. Pressing "Enter" leaves the parameter value as is. To change a parameter's value just enter the new value. After you are done with the modifications and the "[VxWorks Boot]:" prompt appears, type "p" again to view the parameters, making sure the parameter values are as you expect.

Once you are satisfied with the boot configuration, type "@" to continue the boot process. Your changes will automatically be saved to the boot floppy, so the next time you reboot the target the newly added values will be in effect.