Wolverine Firewall Release 1.9
User Documentation

Revision 1.9-04012004

Author: Joshua Jackson <jjackson@vortech.net>

Date: 04/01/2004

Table of Contents

Introduction

Wolverine is a version of Embedded Coyote Linux that is designed to function as a firewall and VPN server. The design goal for this project was to produce a system similar in operation and functionality to the Cisco PIX firewall appliance. While many of the configuration directives were closely modeled after the Cisco PIX, several unique features and options were added to Wolverine to enhance its functionality.

To get the latest information and downloads for Wolverine and other Coyote Linux projects, please visit the Coyote Linux project homepage at http://www.coyotelinux.com or the Vortech Consulting web page at http://www.vortech.net.

Getting started with Wolverine

System Specifications

To prepare a system to run Wolverine, you will need to make sure you have at least the following:

Pentium 75Mhz or better processor
64Mb of RAM
32Mb IDE Drive - ATAPI Flash drives will work nicely.
At least 2 PCI network cards
CDROM Drive

The IDE storage Wolverine is to be installed on to needs to be installed as the IDE primary master device.

As the network interface card (NIC) detection in Wolverine is intended to be automatic, only PCI interfaces are supported. This greatly simplifies the detection and configuration process for both developers and the end users.

While many distributions of Linux will run on lesser systems, the functionality built into Wolverine requires at least the system specifications listed here. In particular, the encryption components (IPSEC, PPTP, SSH) require a fair amount of processing power to perform as expected.

NOTES FOR VPN USERS

If you plan to use the VPN capabilities of Wolverine to establish a large number of concurrent VPN tunnels, your system may need more than the minimum 64Mb of RAM. Failure to provide adequate RAM may result in unexpected behavior as the system is forced to kill off processes that request more memory than is available.

In addition to more RAM, you will also want to increase the processing power of a gateway that is providing IPSEC or PPTP+MPPE128 as strong encryption is very processor intensive. Testing of the IPSEC subsystem (Linux FreeS/WAN) has shown that a Pentium III 866Mhz computer is needed to achieve 30Mbps of sustained encrypted throughput using 3DES encryption or 80Mbps using AES128.

Installing Wolverine

To install Wolverine you will need to either download the CD image and create your own installation CD or you can order one from Vortech Consulting.

Ordering information is available on the Coyote Linux web site or at the end of this document.

If you have downloaded the “wolverine.iso” from the Vortech Consulting download channels, you will need to create a CD using a CD mastering application. Nearly all CD mastering applications support the ISO9660 format.

Once you have booted the target machine from the installation CD, you will be asked to confirm installation of Wolverine. Please note that all data on the primary IDE device will be erased during the installation process. The installation process itself is fully automatic and takes less than a minute on most systems.

If the system you are using for Wolverine is not capable of booting from the CDROM, you can also create boot disks from files contained in the /boot directory. You will need 2 1.44Mb floppy disks to create the boot disks from the boot.img and root.img files. These files need to be written to the floppy using the included rawrite utilities or using the unix utility “dd”. Rawrite is contained in the /utils directory of recent build. See the README file for information on choosing the correct version of the utility for your operating system.

Upon booting Wolverine for the first time, you will be asked some basic questions about the desired configuration for the system. Options such as the hostname, domain name and the addresses for any network interface cards (NICs) that were detected will be automatically placed into an initial system configuration file. This configuration will then be loaded and can be edited from the main menu or using the web administration interface.

Setup Interview

When booting the system for the first time after installation, the initial configuration process will ask for the following information:

Hostname: This option specifies the name this particular firewall is to be given. Note this name should be unique throughout your network. If you have multiple Wolverine firewalls you plan to connect together via VPN tunnels, each system MUST have a unique hostname.

Domain Name: This should be the DNS domain name for your network. If you do not have a domain for your LAN, you can use the domain name of your ISP.

Interface Addresses: During setup, Wolverine will attempt to detect the network cards installed in your firewall. As mentioned in the system specifications, you need to use PCI network cards for them to work with Wolverine. You can specify an address of "*dhcp*" or “*pppoe*” for the first interface if it is to be configured using the DHCP or PPPoE protocols (typically used on Cable modem or DSL networks). If PPPoE is selected you will also be prompted for your PPPoE username and password.

The interface eth0 (the first detected interface) should be used as the public interface. The public interface is connected to the untrusted (typically the Internet) side of your firewall.

When specifying an IP address for an Interface, it should be in the format of *ipaddress/bitmask* (ie: 192.168.0.1/24) If you are not familiar with this format of netmask specification, please see the appendix on netmasks at the end of this document for information on bitmask calculations.

Default Gateway: If you did not specify DHCP address assignment for any of the network interfaces, you will be asked to specify a default gateway address (next hop router).

DNS Server: This will allow you to specify a DNS server for name resolution on your network. If you do not have an internal DNS server configured, you can use the DNS servers for your ISP.

Administrator Password: This password will be stored for use with the users "admin", "root", and "debug". When logging into the Wolverine firewall via either the console or remote SSH/Telnet, the use of "admin" or "root" will place you directly into the administration menu. If you log in as "debug", you will be given an ASH shell prompt instead. Unless you are familiar with the command line and Linux in general, the use of the "debug" login is not recommended.

Remote Telnet Address: This option will allow you to specify an address or network of addresses which will be allowed to connect to the firewall remotely using the telnet or SSH protocols. This address is in the same format as the *Interface Addresses*, specified above.

Timezone and Time server: Setting the firewall timezone and time server is very important if you plan to use the IPSEC functionality provided by Wolverine. The IPSEC subsystem relies on x.509 certificate files which have date/time stamps that indicate when they were created and when they expire. If the date and time in your firewalls are not in sync, it is possible to generate x.509 certificates on one firewall that would be rejected by another.

You will be presented with a list of valid timezones to choose from. The time server can be any Internet time server, or you can use the default of time.vortech.net. The Vortech.net time server is kept in sync with the time.nist.gov stratum 1 time server with updates done on a daily basis.

Update username and Password: You will need to enter the username and password for the account you created when you purchased Wolverine. This account is the same as the one you use to log into the Coyote Linux (www.coyotelinux.com) or the Vortech Customer Service (https://secure.vortech.net/login.php) web sites. If you do not currently have an account, you can sign up for one at the following location:

- https://secure.vortech.net/signup.php

Once you have an account established, you can purchase a download subscription to the Wolverine Firewall and VPN server. For more information on ordering a download subscription, please see the “Ordering Wolverine” section at the end of this manual.

Once the initial configuration has been loaded, your Wolverine firewall should have network connectivity and can now be fine-tuned to meet the specific needs of the network it is protecting.

Using Wolverine

System Boot-up

During boot up, Wolverine will display various messages about the actions it is performing. If your firewall is not performing as expected, be sure to take note of any error messages that are displayed during the boot-up process.

System Shutdown

Unlike most Linux systems, you do not have to prepare a Wolverine firewall for shutdown. The boot filesystem is kept read-only during normal firewall operation. This allows for a cold power-down without risk of damaging the filesystem. The exceptions to this are during the editing of the system configuration, ipsec configuration or during a system image update. You should not power-down the firewall during any of these events.

Logging into Wolverine

Once the firewall has booted, you will be given a login prompt that will appear as follows:

```
Wolverine system login
testfw login:_
```

The above example shows the hostname which was assigned to this firewall was "testfw". From here you can log in using the username "admin" with the password you assigned during system setup. You will then be presented with the system configuration main menu. From this menu you can reconfigure and control nearly all aspects of the firewall.

If you are familiar with Linux and want to access the underlying system, you can log in as the user "debug". This will give you a BASH compatible shell prompt instead of the main menu. Note that the use of the debug login is not recommended nor supported.

System Console

Unlike versions of Wolverine prior to 1.2, 1.3 does not implement multiple virtual terminals. The system logging data is no longer displayed on TTY4. You can still access the recent log data using web administration utility or by sending all logging data to a remote logging host (see the “logging” configuration directive).

If you have a hardware firewall from Vortech Consulting, you should refer to the firewall documentation for additional information on connecting to the console and logging ports.

System Console Main Menu

Once you have logged into Wolverine, you will be presented with the main system menu. From here you can save/restore your system configuration, edit the master config file, reboot or reload the system, and enter the IPSEC configuration or system information menus. The options listed here assume you are connecting to the firewall console directly or via SSH. For more information on using the web based administrator, please see the section entitled “Using the web administrator”.

Configuration backup and restore

The first two options on the main menu are used to save and restore your system configuration to a floppy disk (if it is present in the system). The disk that you use to store your configuration files on should be pre-formatted with a DOS FAT filesystem (these include factory pre-formatted floppies or those formatted in any version of Microsoft Windows or DOS). If you need to format a floppy disk in Linux for use on your Wolverine firewall, you can do so with the following commands:

```
fdformat /dev/fd0h1440
```

The above command will low level format the floppy to 1.44Mb capacity. If this completes without errors, you can create the FAT filesystem using either of the following commands:

```
mformat a:
- or -
mkdosfs /dev/fd0
```

If the first command does not work on your Linux system, try the second one. The use of the “mformat” command requires that the “mtools” package be installed; some distributions do not install this package by default.

Once you have a properly formatted floppy disk, you can back up your configuration using option 2 from the main menu. This will copy your system configuration and encryption identification keys to the disk. This floppy can be used to configure another, identical firewall using option 1 from the main menu.

NOTE: The files contained on the system backup floppy contain security sensitive information about your firewall. This floppy must be kept secure or your firewall and/or VPN tunnel security may be compromised.

If your firewall does not have a floppy device, you can use the web administratior to download a backup of your system configuration. This file can be used later to restore your configuration. See the documentation section for the web administrator for more information.

Updating the firewall software

From the main menu, you can download an update to your firewall software using option 3. This option can be used to download updates from the official Wolverine update server or from an FTP server of your choosing (handy if you have the developer's kit and have made custom modifications to Wolverine).

If you have upgraded a Wolverine 1.0 system to version 1.9, you will be asked to enter your account username and password the first time you attempt to update from the official update server. This account information is the same that you chose on the Coyote Linux web site when you purchased your subscription to the Wolverine download channels. This account information will be used to establish a secure, FTP+SSL connection to the update server.

If you are updating from a server other than update.coyotelinux.com, you will be asked to specify the source directory and authentication information for the server you have selected. The files on the source FTP server are expected to match those that would be present on the official server (a kernel image, ramdrive image, image version information file, and a checksum file).

Before downloading the update, Wolverine will attempt to determine the build number of the update files. If your system is up to date or running a newer build than the one present on the update server, you will be asked to confirm the download action before it is carried out.

After the new files are downloaded, the integrity of the downloaded files will be checked using the MD5 hashes present in the checksum file downloaded during the update. If the files do not pass the integrity check, the update will not be performed.

Editing the main configuration file

From the main menu, you can edit the main system configuration file using option 4. The editor that is opened by default is GNU “nano”. This editor is very similar to “pico” on other Linux or BSD based systems. If you are not familiar with either of these editors, some keyboard shortcuts that you may find handy:

Ctrl-o - Save the current file
Ctrl-x - Exit the editor
Ctrl-v - Page Down
Ctrl-y - Page Up
Ctrl-^ - Set Mark
Ctrl-k - Cut Line / Marked text
Ctrl-u - Paste text

When you edit the master configuration file, your changes do not go into affect immediately. For this, you will need to reload or reboot the system.

Rebooting or Reloading the system

Options 5 and 6 on the main menu allow you to reboot or reload the system. Most changes to the firewall configuration will not require you to reboot. The exception to this would be downloading a system update; you must reboot for the system update to to take affect.

Product Activation

Starting with Wolverine v1.2, you will need to activate your installation to prevent it from expiring. Before activation the system will remain up for 6 hours before automatically halting. You can reboot the system up to 20 times for another 6 hours of trial time after each reboot. After the 20th reboot, the firewall installation will expire. After expiring, the firewall will work for 10 minutes after each reboot to allow time to provide your product activation information.

To activate your copy of Wolverine, you will need to enable the web administrator. The first option on the web administrator main menu will allow you to supply your product activation certificate. This certificate will be provided for you at the time of your initial software purchase.

Wolverine System Configuration

Network Interface Configuration

As of this writing, only PCI Ethernet network interfaces are supported by Wolverine. Support for USB and dial-up serial modems are being considered but ISA and PCMCIA types will not likely be supported by this product.

Wolverine uses the standard Linux naming scheme for any network interface adapters present in the system. The first interface in the firewall will be listed as "eth0", the second as "eth1" and so on. If you want to use a naming scheme other than the default, you can give each interface an alias name with the "nameif" directive in the main configuration file. This allows for a more readable configuration and gives interfaces a more descriptive name. Here is an example of the use of "nameif":

```
nameif eth0 public
nameif eth1 dmz
nameif eth2 private
```

The aliases of "public", "dmz", and "private" can now be used in place of the interface name in configuration statements which require an interface name.

Public Interface

The Wolverine configuration system normally assumes that the first interface in the system is connected to the untrusted network segment (typically the Internet). In most cases, use of the default is recommended. If you must use an interface other than eth0 for your public network connection, you can specify a different public interface with the following:

```
interface eth1 public
```

Note: If you are using PPPoE or DHCP to for address assignment on an interface, this interface will automatically become the public interface.

Address Assignment

Wolverine allows for DHCP, PPPoE and static address assignment for network interfaces. PPPoE support also requires additional authentication information to be specified using the “pppoe” configuration directive. Address configuration types not supported include PPP dialup (as modems are not supported), and PPPoA, and PPTP (client mode).

If your external connection uses a DHCP configured address, you can enable DHCP address assignment with the following statement:

```
interface eth0 address dhcp
```

You should note that only one interface in the firewall can be configured using DHCP or PPPoE. Wolverine will automatically use the default gateway information if supplied by the DHCP/PPPoE server; as such, this interface will automatically be assigned as the public interface for the firewall. This configuration is typically used on cable and DSL networks.

If your ISP uses PPPoE for address assignment, you will also need to add your authentication information. A typical PPPoE configuration would appear as follows:

```
pppoe user testuser testpass
interface eth0 address pppoe
```

In this case, “testuser” is the username and “testpass” is the password that will be passed to the PPPoE server for authentication. Additionally, you can specify an on-demand connection for PPPoE. To use on-demand an on-demand connection, you can add the following statement after your pppoe authentication info:

```
pppoe demand 300
```

In this case, 300 seconds is the amount of time that a connection can remain idle before the link is terminated.

Note: In on-demand PPPoE mode, the firewall VPN services will be automatically disabled. This is due to the VPN daemons and firewalling rules needing to be able to obtain IP and routing information that is unavailable when the link is down.

Any interfaces configured using a static IP address can also have additional addresses assigned using the "secondary" keyword. Here is an example of a static IP address assignment with a secondary address assignment:

```
interface eth0 172.20.0.1/24
interface eth0 172.20.0.2/24 secondary
```

MAC Address spoofing

Some ISP's require a connecting client to have a registered MAC address. This address is encoded into the hardware of every Ethernet network interface adapter. If your ISP requires your client to connect from a specific MAC address that is different than that of your Wolverine firewall's public interface, you can change (or “spoof”) the MAC address. To do so, use the following syntax:

```
interface eth0 mac 00:01:02:03:04:05
```

Where 00:01:02:03:04:05 is the MAC address you what to assign to the interface. While a new MAC address can be specified for any of the firewall interfaces, this is typically only required on the public interface when connecting to ISP's that expect a different MAC address than that of Wolverine's public interface.

Controlling Access to Network Resources

The primary function of any firewalling product it to protect the network resources on the attached network segments. With Wolverine, all traffic passing through or directed at the firewall will be rejected until specifically enabled. In order to allow access to protected resources, the following techniques can be used:

- Access Lists (access-list)
- Automatic Port Forwards (auto-forward)
- Directed Port Forwards (port-forward)
- ICMP Packet Controlling (icmp)

Access Lists

Access lists define a set of rules that control what types of traffic the firewall will be allowed (or not allowed) to route. These rules are implemented using the "access-list" directive. You should note that access-list statements do not control what is allowed or not allowed to access the firewall itself; they only control the behavior of the firewall when dealing with packets passing *through* it.

An example of allowing traffic to pass through the firewall to reach an internal web server would look like:

```
access-list testlist permit tcp any 172.20.0.5 80
```

This access list permits any external traffic passing through the firewall with a destination of 172.20.0.5 on port 80 to be permitted. If you wanted to restrict the hosts or networks that could access this resource, the keyword "any" could be replaced with a host or network address.

➔ If your internal network does not contain Internet routable addresses, the access list alone will not be enough to provide access to internal resources to hosts on the Internet. In this case you will need the options provided in the next two sections (auto-forward and port-forward).

Automatic Port Forwards

In the event the firewall does not have a static IP address or the address may not be known at the time of configuration, an automatic port forward can be used to direct traffic to internal hosts. This would likely be the method used if you have a DHCP or PPPoE configured Internet connection. The following example would forward any traffic arriving on the external interface (in this case, the default of eth0 is used):

```
auto-forward eth0 tcp 80 192.168.0.2
```

If you need to forward a range of ports, you can also specify this range in the format of start:end. An example of forwarding UDP ports 1000-2000 to an internal host would appear as such:

```
auto-forward eth0 udp 1000:2000 192.168.0.2
```

➔ Automatic port forwards create their own access control list to permit traffic to pass through the firewall. You do not need to specify an additional access-list directive when using them.

Directed Port Forwards

If the forwarding interface address is known at the time of configuration, you can achieve greater flexibility with the port-forward directive. This also allows for configurations where multiple external addresses are used for port forwarding. When using this method of port redirection, an access list is also needed to allow the traffic to traverse the firewall. In this example a server running web (http) and email (smtp) servers is forwarded to from the external interface addresses of 172.20.0.1 and 172.20.0.2:

```
interface eth0 address 172.20.0.1/24
interface eth0 address 172.20.0.2/24 secondary
port-forward 172.20.0.1 192.168.0.2 tcp 80
```

```
port-forward 172.20.0.2 192.168.0.2 tcp 25
access-list portfw1 permit tcp any 192.168.0.2 80
access-list portfw1 permit tcp 10.0.0.0/8 192.168.0.2 25
```

With this example, use of the port 80 forward is permitted from any host, whereas the port 25 redirection can only be used by the 10.0.0.0/8 network. You should also note the use of the “secondary” address on the eth0 interface. In order to forward traffic from an external interface address to an internal host, the forwarding address must be assigned to the firewall.

1:1 Inbound NAT Translations

You can also use the port-forward statement to create 1:1 inbound NAT translations. With this type of translation, all traffic reaching the external IP address will be forwarded to an internal host. To specify which traffic should actually be allowed to reach your internal network, use access-list statements. The folllowing is an example of a 1:1 translation with the same ports being forwarded as the above example:

```
interface eth0 address 172.20.0.1/24
interface eth0 address 172.20.0.2/24 secondary
port-forward 172.20.0.1 192.168.0.2
access-list portfw1 permit tcp any 192.168.0.2 80
access-list portfw1 permit tcp any 192.168.0.2 25
```

With this example, all traffic directed to the external interface address of 172.20.0.2 will be redirected to 192.168.0.2. The access-lists specified will only allow port 80 and 25 TCP traffic to reach the internal host.

Both automatic and directed port forwards require that the internal LAN host route returning traffic back through the Wolverine system that is providing the port forwards. Typically, this involve using Wolverine as the default gateway for the protected server.

Controlling ICMP Responses

A frequently asked question is how to get the firewall to respond to ICMP requests. This can be accomplished using the "icmp" directive. Care should be taken when enabling ICMP responses to the Internet or other untrusted networks. A commonly used denial of service (DoS) relies on ICMP responses from a network host. Enabling ICMP responses from the firewall itself or enabling ICMP requests to pass though the firewall to internal network resources can lead to problems if you come under such an attack.

To get the firewall to respond to the typical "ping" request, the following command can be used:

```
icmp permit any echo-request eth0
```

If you want to enable all ICMP message types (should only be performed on a trusted segment), you could use a command such as:

```
icmp permit 192.168.0.0/24 all eth1
```

The above examples only control ICMP messages that are directed at the firewall itself. If you want external hosts to be able to send ICMP requests to hosts on a protected segment, you can use a simple access list for this:

```
access-list icmptraf permit icmp any any
```

In addition to allowing ICMP traffic to be directed at or through the firewall, it is also possible to limit the rate at which ICMP echo (ping) requests will be permitted. When rate limiting is enabled, responses send at and through the firewall will both be limited automatically. To enable rate limiting, use the following directive:

```
icmp limit 10
```

Where *"10"* is the number of ICMP echo requests to allow per second. This option can be used to protect the internal network from ping-flood attacks or to severely cripple back door applications that make use of ICMP packets for data transmission.

Universal Plug-n-Play (UPnP) support

Universal Plug-n-Play support was added to Wolverine in version 1.9. This option allows client computers which are behind NAT to automatically request port-forward creation on the firewall. This allows applications which do not normally work behind NAT to function normally. One of the most common examples of such an application is Microsoft's Instant Messenger client. Particularly for file transfers and voice services.

Enabling UPnP in Wolverine is very simple. The following configuration line is all that is required:

```
option upnp enable eth1
```

The internal interface which should listen for UPnP requests should be substituted for “eth1”.

⚠ **WARNING: Never enable UPnP support on the firewall's external interface. Doing so will severely compromise the security of your network. UPnP should also only be used on networks were the internal hosts are considered “trusted”. UPnP should never be used on networks were internal hosts may be used to run untrusted applications. As such, UPnP is primarily intended for use in personal or home network configurations.**

Network Address Translation

In most SOHO (Small Office, Home Office) or personal network configurations, only a single IP address is provided for access to the Internet from the ISP. In order to share this Internet connection with a network of computers, you will need to use Network Address Translation (NAT). NAT can be configured using the simple statement:

```
nat eth0 192.168.0.0/24
```

This example configures the firewall such that any traffic originating on the 192.168.0.0/24 network which would routed through the external (eth0) interface should be subject to NAT.

In order to allow connections to be made from the internal network, you will also need to add an access-list (or series of them for more complex configurations). The following access-list will provide unrestricted access to the Internet for internal hosts:

```
access-list outbound permit all 192.168.0.0/24 any
```

In addition to the nat and access-list statements, most networks will also want to enable the firewall connection tracking modules available in Wolverine. Currently these modules include support for FTP, IRC, Eggdrop bots, h.323, Microsoft Messenger, PPTP, Quake3, and Amanda backup software. The reason for these is that some portions of these communication protocols have problems communicating though NAT. If you do not need to support any of these protocols, you can leave the fixup directives out of your configuration. To enable the connection trackers, add the following commands to your firewall configuration:

```
fixup ftp
fixup irc
```

For a complete list of the module names available, please see the "fixup" configuration directive.

In certain, more complex configurations there may be exceptions to the NAT rules you are using. If there are times that only certain traffic should be subject to NAT but not everything passing out of a given interface, you can add a destination address to the nat statement:

```
nat eth0 192.168.0.0/24 172.20.0.0/16
```

In this example, only traffic that is destined for the 172.20.0.0/16 network would be subject to NAT. In a reverse case the following configuration could be used to prevent NAT from being performed for a given range of addresses:

```
nat eth0 192.168.0.0/24
nat bypass 192.168.0.0/24 172.20.0.0/16
```

In this case all traffic with the exception of packets destined for the 172.20.0.0/16 network are subject to NAT. Those destined for the 172.20.0.0 network will be subject to standard routing restrictions.

PPPoE Users: If you are using PPPoE to configure your Internet connection, the PPP device name (ppp0) should be used instead of the Ethernet device name. To perform NAT for a network of computer when using PPPoE, your nat statement should look like:

```
nat ppp0 192.168.0.0/24
```

Configuring external logging

By using the "logging" directive, it is possible to send Wolverine's system logging information to a remote host. This requires a syslog server to be running on the remote host which is configured to accept external logging information. To enable remote logging on Wolverine, use the following statement:

```
logging host 192.168.0.3
```

Where 192.168.0.3 is the host running the syslog server. With most Unix implementations, enabling the syslogd daemon to accept external logging information is accomplished by adding a "-r" to the command line. On Red Hat Linux, you can enable this feature by editing the /etc/sysconfig/syslog file and editing the line that reads:

```
SYSLOGD_OPTIONS="-m 0"
```

to

```
SYSLOGD_OPTIONS="-m 0 -r"
```

You will also need to restart the system loggers with the command:

```
service syslog restart
```

⚠ **You should not enable remote logging on a host which can be logged to (uncontrolled) from the Internet. This can lead to a DoS attack against your server. If your logging server is publicly accessible, be sure to at least set the firewall rules to only permit logging data from authorized hosts.**

Configuring the built-in PPTP server

Wolverine is equipped with a PPTP VPN server capable of handling MSCHAPv2 user authentication and MPPE 128 bit encryption. By default, these are the connection options required to successfully connect to a Wolverine VPN server (128 bit encryption and MSCHAPv2 authentication is required or the connection will be rejected).

The login information supplied by a connecting client can be authenticated using either a local or remote Radius database. The most common use for the Radius authentication method is for integration of Wolverine into an existing Windows 2000 or 2003 based Active Directory.

User Authentication

Before enabling the PPTP server, you need to let the firewall know how to authenticate the user information passed to it during login. This is accomplished using the "authentication" directive. An example of its use would be:

```
authentication ppp radius local
```

The above example instructs the firewall to do Radius authentication of incoming PPP connections (PPP is used by PPTP). If Radius fails, the firewall will fall back to the local PPTP username and password database. The order of these options can be reversed or one of them can be omitted if the specific authentication method is not available.

Radius Server Configuration

If a Radius server is to be used for user authentication, you will need to configure Wolverine's Radius server options with the "radius-server" directive:

```
radius-server host 192.168.0.5
radius-server key SeCrEtKeY
```

The above example sets the Radius server to 192.168.0.5 with a shared, secret key of "SeCrEtKeY". The shared secret and client information will need to be configured on the Radius server as well. See the example configurations section for an example of configuring Microsoft Internet Authentication Service (IAS) to integrate Wolverine into an existing Active Directory domain.

PPTP Directives

To enable the PPTP server there are a few statements which need to be present. The first statements are the "pptp local-address" and "pptp address-pool". These directives define how to address each endpoint of a PPTP tunnel.

```
pptp local-address 192.168.0.100
pptp address-pool 192.168.0.101-150
```

These statements set the local (firewall) side of the PPTP tunnels to use the address of 192.168.0.100 while the remote (client) side of the tunnels will be given an address from the pool of 192.168.0.101 through 192.168.0.150.

If the addresses specified in the local and pool statements are part of the same network that one of the trusted network interfaces is connected to, you can also add the following statement:

```
pptp proxyarp
```

This will instruct the firewall to perform proxy-arp on behalf of the connecting client. Simply put, this will allow the remote host to appear as any other node on the internal LAN while connected via the tunnel. This is the most common configuration and simplifies client configuration by removing the need for static routing on each external client.

Optionally, DNS and WINS servers can be specified for the connecting computers as well using the following statements:

```
pptp wins-server 192.168.0.50
pptp dns-server 192.168.0.51
```

If multiple servers of a given type are needed, use multiple statements (do not specify them on a single line).

PPTP Local Users

If you are using local authentication for PPTP, you need to add a statement for each local user as such (where "testuser" is the username and "testpass" is the password to assign for this user):

```
pptp user testuser testpass
```

Configuring IPSEC Support

This section details the configuration and use of the IPSEC subsystem (Linux FreeS/WAN) of the Wolverine firewall. As of the writing of this document, the FreeS/WAN version used is “Super” FreeS/WAN 1.99.8. For additional information on this packages of software, you can visit the project web sites directly at:

Official FreeS/WAN: http://www.freeswan.ca

Super FreeS/WAN: http://www.freeswan.ca

x509 Patch: http://www.strongsec.com/freeswan

As this document does not contain information about every feature of Super FreeS/WAN, you may also want to review the documentation available from these sites.

Getting Started with IPSEC

Before continuing, you should note the hostname of your firewall is used internally by the IPSEC configuration system. Each Wolverine firewall your firewall will communicate with will need a unique hostname assigned to it. Once you have enabled IPSEC, do not change the hostname of the firewall. If you must change the hostname, you will need to regenerate your host certificates and re-establish communications with any external firewalls.

During the initial boot process of your Wolverine firewall, a certificate will be automatically generated for use with both IPSEC and the SSL web administrator. If you need to regenerate the certificates, you can do so from either the web administrator or from the IPSEC certificate management menu.

Note: If you regenerate the host certificates, you should reboot the firewall. The web administrator will not recognize the new certificates until a reboot is performed. If you used the web administrator to reboot the firewall, you should also close your browser and reopen it before attempting to connect to the firewall again. Most browsers will cache the previous certificate and will display an error message when the certificate changes.

Once you have generated your initial certificates, you should be able to view the information for your system from the Server Certificate Menu. The displayed information should appear something like the following (take note of the subject line as it is needed during configuration):

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: CN=testfw
        Validity
            Not Before: Aug 25 15:41:14 2002 GMT
            Not After : Aug 24 15:41:14 2006 GMT
        Subject: CN=testfw
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:9f:45:fe:66:a0:f3:a7:59:a2:53:79:8a:27:58:
                    9f:53:6a:98:37:bd:2c:17:de:08:2c:04:18:69:d4:
                    2f:da:db:0e:94:9e:c5:eb:95:16:90:52:20:ed:63:
                    8b:73:74:e5:86:64:d4:ae:fa:84:ab:3a:18:ea:0f:
                    40:c9:64:dd:67:bc:1e:65:dd:31:78:7a:07:d5:a9:
                    2e:44:7d:13:db:2e:b7:7d:49:c5:8a:27:36:07:20:
                    70:45:2b:2f:da:24:e2:21:e4:a1:a3:d0:10:92:33:
                    09:6f:27:18:a0:a7:bd:08:40:d1:86:b2:c7:54:af:
                    c9:07:a0:46:d3:46:6b:54:3d
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        43:d9:f9:ab:06:88:6b:c4:9b:7d:0b:05:f7:6c:d8:89:bf:53:
        0b:f9:9c:5b:62:5b:71:a4:cf:92:74:7a:cd:ce:23:1b:cc:39:
        92:8c:74:fb:88:88:e0:90:f8:cc:24:ec:87:33:36:c3:fd:c4:
        16:50:bb:6a:bb:16:11:33:7d:3b:49:5a:1d:6c:84:46:28:a6:
        8d:7f:a3:82:5d:d6:ed:a7:e7:04:5b:a4:64:0a:b6:8d:3c:c2:
        cd:4b:23:dd:7e:d4:f2:f3:d8:f1:a8:c7:c1:de:89:6d:db:22:
        e9:d4:19:ef:ab:a3:97:c5:31:b9:69:65:a8:f5:0f:44:28:53:
        51:c8
```

Exchanging certificates with other systems

In order to establish an IPSEC connection with another system, you will need to exchange x.509 certificates with the remote system. This can be accomplished using the import and export option on the Server Certificates menu from the console or by using the web administrator.

Exporting a certificate

If your firewall is equipped with a floppy drive, you can export a certificate from the Server Certificate Menu using the "Export certificate to a remote system" option. You will be asked to insert a floppy disk into the floppy drive on the firewall system. The file exported will be *hostname*_cert.pem. Where "*hostname"* is the hostname for the firewall exporting the certificate. If your system is not equipped with a floppy drive, you can export certificates using the web administrator.

Importing a certificate

From the Server Certificate Menu, select "Import a remote certificate" option. You will be asked to insert a floppy containing the certificate to import. You will be given a list of the certificate files contained on the floppy (files ending in . pem) and asked for the name of the file you wish to import. If your system is not equipped with a floppy drive, the web administrator can also be used to import certificates.

Creating an IPSEC tunnel

Once you have exchanged IPSEC certificates with another firewall, you can create a tunnel between them by adding a few lines to the IPSEC configuration file. This file can be edited from the IPSEC configuration menu.

When you open the configuration file, you will see a section already present entitled "config setup". Leave this section intact and go to the bottom of the configuration file. You will need to add a section that will appear as such:

```
conn conn-name
        right=172.20.32.2
```

```
rightnexthop=172.20.32.1
rightsubnet=192.168.0.0/24
rightcert=fwtest1_cert.pem
rightid="CN=fwtest1"
left=172.20.33.2
leftnexthop=172.20.33.1
leftsubnet=192.168.1.0/24
leftcert=fwtest2_cert.pem
leftid="CN=fwtest2"
auto=add
```

In this example, two firewalls (testfw1 and testfw2) are connected to each other to create a tunnel between the private subnets 192.168.0.0/24 and 192.168.1.0/24. The choice of which system is considered "right" and "left" is arbitrary, but needs to be the same on both firewalls.

Hint: To keep straight which system is right and left on my firewalls, I use alphabetical order. The system with the lower alphabetical position is always assigned to the "right" side of the connection.

The connection name ("*conn*") that is specified on the first line is arbitrary, but should be the same on both firewalls.

For an explanation of what each line means:

conn *conn-name*

This line names the individual connection. Each additional connection will need a unique connection name.

```
right=172.20.32.2
```

This indicates the address of the public interface on testfw1.

```
rightnexthop=172.20.32.1
```

The "nexthop" indicates the gateway

```
rightsubnet=192.168.0.0/24
```

This is used to designate the subnet that is to be shared with the remote network. Any traffic passing from this subnet to the "left" subnet will be encrypted.

```
        rightcert=fwtest1_cert.pem
```

This line indicates the filename of the x.509 certificate file that you want to use for the connection.

```
        rightid="CN=fwtest1"
```

This line is the subject for the x.509 certificate. This can be obtained by viewing the details for the local system certificate (mentioned earlier in this section of the manual). However, you will need to remove the commas from the output of the local certificate display subject.

```
        left=172.20.33.2
        leftnexthop=172.20.33.1
        leftsubnet=192.168.1.0/24
        leftcert=fwtest2_cert.pem
        leftid="CN=fwtest2"
```

These lines are identical to the "right" directives with the exception that they contain the information for the other firewall.

```
        auto=add
```

This line indicates that this side of the connection should be loaded but not initiated. On one of your firewalls, you will want a line that reads:

```
        auto=start
```

A good suggestion is to add the "start" directive to the firewall that gets rebooted the most.

If one of you IPSEC endpoints has a dynamic IP address, you can configure one of the endpoints to accept a tunnel from any remote endpoint with the following configuration on the static gateway:

```
conn dynaconn
        right=172.20.32.2
```

```
                rightnexthop=172.20.32.1
                rightsubnet=192.168.0.0/24
                rightcert=testfw1_cert.pem
                rightid="CN=fwtest1"
                left=%any
                rightsubnet=192.168.1.0/24
                rightcert=testfw2_cert.pem
                rightid="CN=fwtest2"
```

The configuration that is used on the firewall that has a dynamic IP address is sightly different:

```
conn dynaconn
                right=172.20.32.2
                rightnexthop=172.20.32.1
                rightsubnet=192.168.0.0/24
                rightcert=testfw1_cert.pem
                rightid="CN=US CN=fwtest1"
                left=%defaultroute
                leftsubnet=192.168.1.0/24
                leftcert=testfw2_cert.pem
                leftid="CN=fwtest2"
```

Considerations when upgrading from 1.0 to 1.9

If you have upgraded your Wolverine firewall from v1.0 to v1.9, you will need to add some additional directives to the IPSEC configuration file. Under the section “conn %default”, you should have the following statements (if any are missing, you should add them):

```
conn %default

                authby=rsasig
                rightrsasigkey=%cert
                leftrsasigkey=%cert
                ike=”aes128-md5,3des-md5”
                esp=”aes128-md5,3des-md5”
                type=tunnel
```

You may also have a line that reads “keyingtries=0”. This option can be left alone if it is present and should always be present if the firewall has a dynamically configured Internet connection.

Transparent firewall support

New in version 1.3 and later releases is the ability to use Wolverine in “transparent” mode. In this mode, the firewall acts a bridge rather than a router. This type of configuration is useful when you have a very limited number of IP addresses available for your network or you want to firewall certain hosts on a network without breaking it into multiple subnets.

However, it should be noted that with most bridging configurations Wolverine should not be used as a VPN server. In addition, many of the router-centric functions of Wolverine will not function properly. Such options would include port forwarding, network address translation, and VPN tunnels.

Bridging directives

To configure Wolverine for transparent operation, a new set of bridge directives have been added to the configuration language. To place a pair of network interfaces in bridging (transparent) mode, you would instruct Wolverine using the following interface directives:

```
interface eth0 bridge enable
interface eth1 bridge enable
```

This places an Ethernet bridge between eth0 and eth1. However, by default all traffic which would normally traverse the bridge is blocked until specifically enabled using access-lists.

Remotely accessing a transparent firewall

Once an interface has been bound to the Ethernet bridge it can not be assigned an IP address. Without an IP address, it would be impossible to remotely access the firewall. To overcome this problem a directive is available to assign a central IP address to the bridge itself. You can assign an IP to a the transparent firewall using the following:

```
bridge address 192.168.0.100/24
```

This address should not be used as a gateway for clients unless you have additional interfaces in the system that are not bound to the bridge.

Using The Web Administrator

New to versions 1.2 or greater of Wolverine is the web administrator. This utility allows for much of the same functionality available from the console or SSH session. To access the web administrator, you will need an SSL capable web browser. Any current version of Netscape, Mozilla, Microsoft Internet Explorer, Opera, or Konqueror should work fine.

Enabling the web administrator

To connect to the web administrator, you will need a pair of directives in the firewall master config file:

```
http server enable
http 192.168.0.0/24
```

The first line instructs Wolverine to enable the web administrator, the second specifies a network or host that should be allowed to access it. You can also specify a port at the end of the first line if you want to use something other than the default https port of 443. See the http configuration directive documentation for more information.

Accessing the web administrator

Once you have enabled the web administrator, you can connect from your browser using the URL:

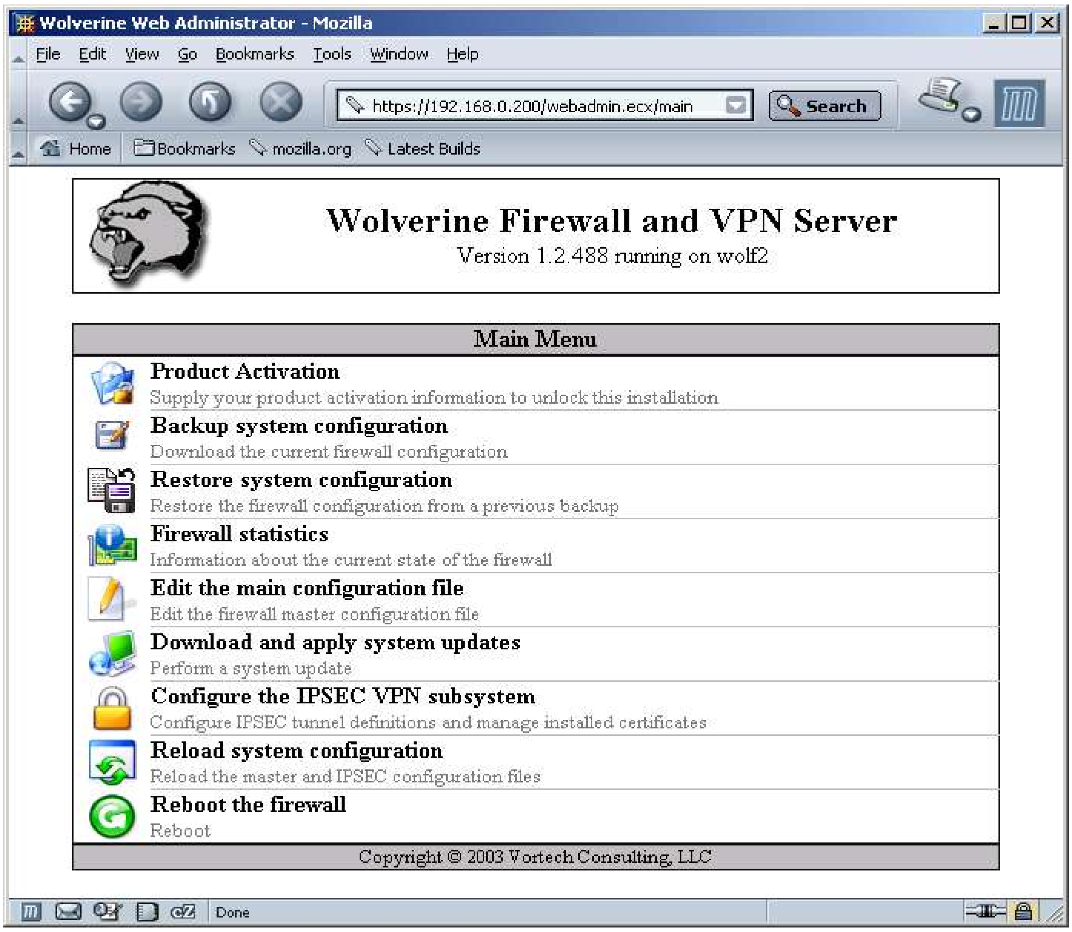
https://192.168.0.1

This assumes you have given your firewall an address of 192.168.0.1. You can specify the address of any of the firewall interfaces as long as you have given your host computer permissions to connect to the web administrator.

When connecting to the web administrator, you will be asked for your login information. The login name is “admin” with the password you specified during the initial setup interview. You may also receive a warning message that the certificate presented to your browser is not from a trusted certificate authority. This message is normal for web servers that generate their own certificates.

Once logged in, you will be given the web administrator's main menu. The following screen shot shows the main menu and its options:

Main Menu


Wolverine Web Administrator - Mozilla
File Edit View Go Bookmarks Tools Window Help
https://192.168.0.200/webadmin.ecx/main
Search
Home Bookmarks mozilla.org Latest Builds
Done

Wolverine Firewall and VPN Server
Version 1.2.488 running on wolf2

Main Menu

Product Activation
Supply your product activation information to unlock this installation

Backup system configuration
Download the current firewall configuration

Restore system configuration
Restore the firewall configuration from a previous backup

Firewall statistics
Information about the current state of the firewall

Edit the main configuration file
Edit the firewall master configuration file

Download and apply system updates
Perform a system update

Configure the IPSEC VPN subsystem
Configure IPSEC tunnel definitions and manage installed certificates

Reload system configuration
Reload the master and IPSEC configuration files

Reboot the firewall
Reboot

Copyright © 2003 Vortech Consulting, LLC

In the header for each page, you can see the version and build number you are running as well as the firewall host you are connected to.

Product Activation: If you have downloaded a trial version of Wolverine, you can purchase a copy of Wolverine to obtain a product activate certificate. This will prevent Wolverine from expiring.

Backup system configuration: Use this option to download a copy of the complete configuration for this Wolverine firewall. The downloaded file is in GNU TAR format and will have a default name of “wolverine.tar”.

⚠ ***WARNING: The downloaded tar file contains sensitive information about the configuration of your firewall. This file should be stored in a secure location.***

Restore system configuration: This option allows you to restrore the firewall configuration from a previous backup. You can also use the backup “wolverine.tar” file written to a floppy disk if the text menu option was used to save your configuration to a floppy.

Note: Only backups from a Wolverine 1.2 or greater system can be used to restore your firewall configuration. Previous version of Wolverine stored their configuration backups in a different format.

Firewall statistics: This option will give you another submenu with options for viewing various bits of information about the firewall's running state.

Edit the main configuration file: This option allows you to change the master firewall configuration file. Information about the format if this file are available in *Configuration Reference* section of this manual. ***Note:*** Changes made to the configuration file will not take effect until until you perform a configuration reload or reboot the firewall.

Download and apply system updates: This feature simply gives you instructions on how to update the firewall software. An update can not be performed directly from the web administrator.

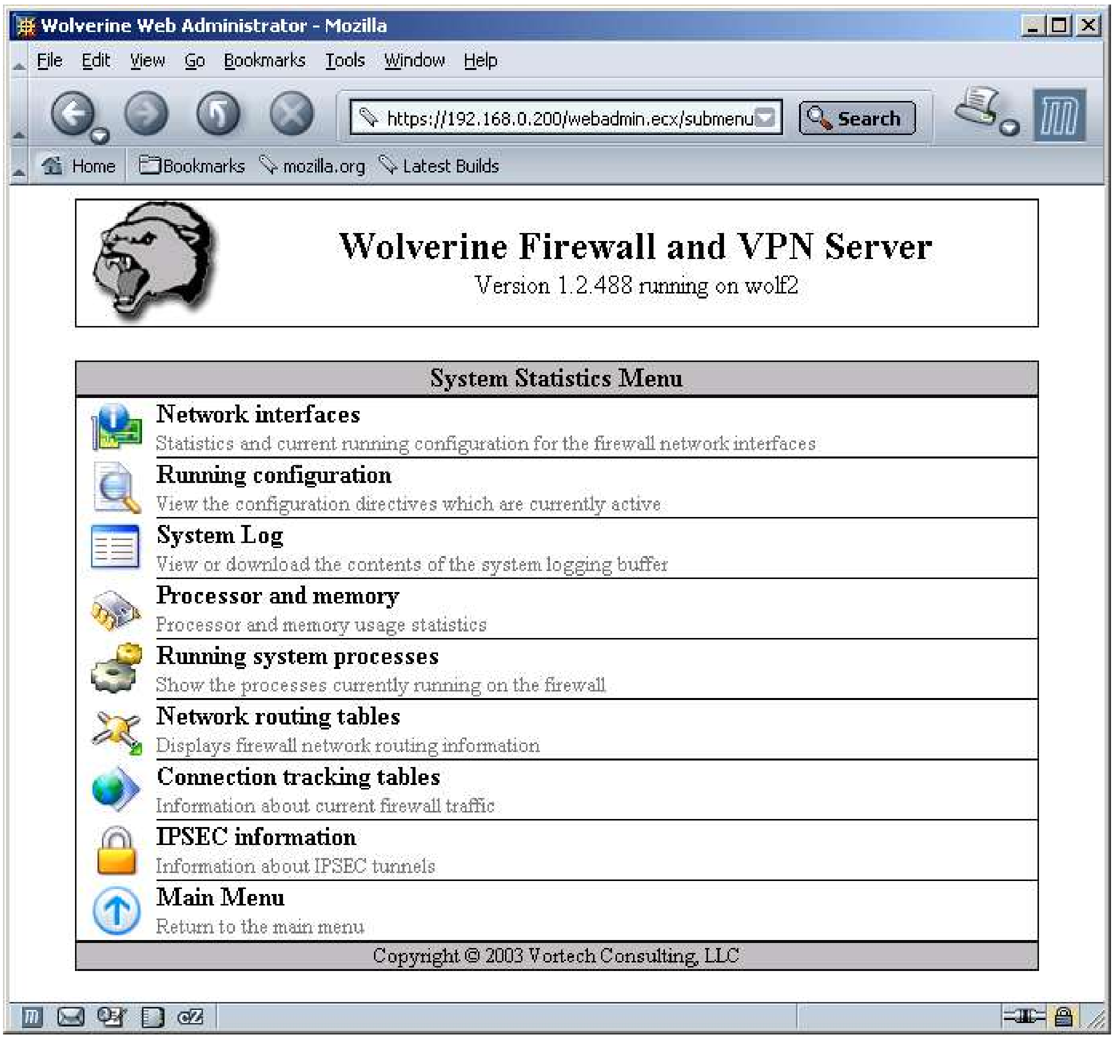
Configure the IPSEC VPN subsystem: This option will give you an additional submenu for configuring Wolverine's IPSEC capabilities.

Reload system configuration: This option will reload the master and IPSEC configuration files. ***Note:*** Any connected VPN clients will have their connections reset during a configuration reload. To apply IPSEC configuration changes without breaking existing tunnels, see the IPSEC configuration submenu.

Reboot the firewall: Reboots the machine Wolverine is running on.

Statistics Menu

The system statistics menu provides access to various information about the actove state of the firewall, its interfaces, memory, processor, and network routing. The following image shows the system statistics menu:



Wolverine Web Administrator - Mozilla
File Edit View Go Bookmarks Tools Window Help
https://192.168.0.200/webadmin.ecx/submenu
Search
Home
Bookmarks
mozilla.org
Latest Builds

Wolverine Firewall and VPN Server

Version 1.2.488 running on wolf2

System Statistics Menu

Network interfaces
Statistics and current running configuration for the firewall network interfaces

Running configuration
View the configuration directives which are currently active

System Log
View or download the contents of the system logging buffer

Processor and memory
Processor and memory usage statistics

Running system processes
Show the processes currently running on the firewall

Network routing tables
Displays firewall network routing information

Connection tracking tables
Information about current firewall traffic

IPSEC information
Information about IPSEC tunnels

Main Menu
Return to the main menu

Copyright © 2003 Vortech Consulting, LLC

Network interfaces: This option displays information about the various hardware and virtual network interfaces in the firewall.

Running configuration: This option displays the configuration the firewall used last to configure itself. This information is the parsed and sorted version of the master configuration file. Note that comments will be removed and the order of the directives may have been changed. ***If you see any comments in the running configuration about parsing problems or find there are missing directives from your master configuration, these lines have been identified has being invalid.***

System Log: This option displays the last 16kB of logging data generated by the firewall. To see more than 16kB of logging data, you will need to specify an external logging server using the “logging” master configuration directive.

Processor and memory: This option displays information about the CPU and memory usage, as well as additional information about system IRQ's, disk IO, and the Linux kernel version.

Running system processes: This option displays a list of processes running on the firewall.

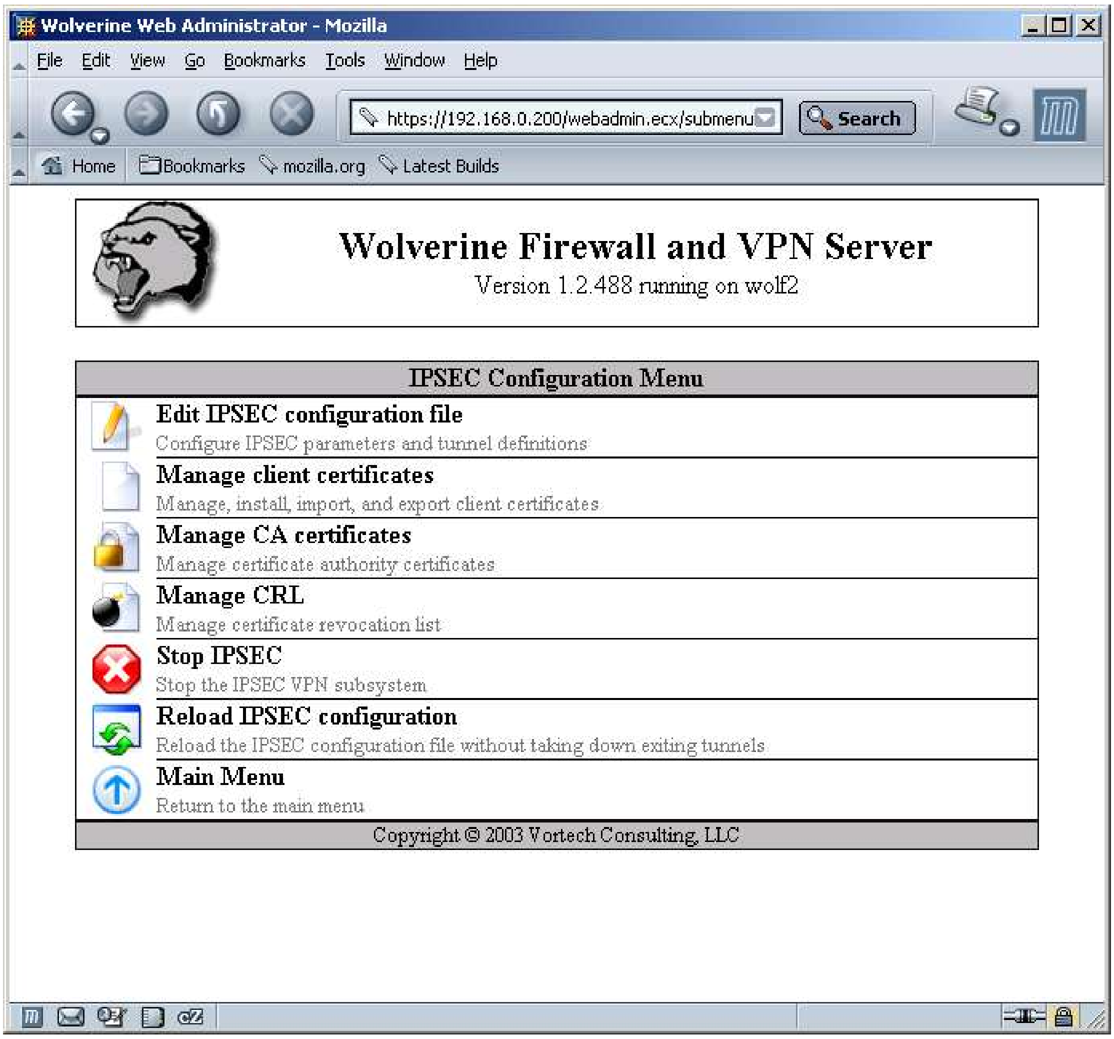
Network routing tables: This option displays the active network routes and any encrypted (IPSEC) routes.

Connection tracking tables: This option shows a list of all connections currently being tacked by the firewall. On busy firewalls, this list can be very large.

IPSEC information: This option displays information about the IPSEC subsystem, including IPSEC virtual interfaces, tunnel definitions, and SA/ESP information.

IPSEC Configuration Menu

From the IPSEC menu you can edit and install certificates, edit the main IPSEC configuration file, and start and stop IPSEC. The menu appears as follows:


Wolverine Web Administrator - Mozilla
File Edit View Go Bookmarks Tools Window Help
https://192.168.0.200/webadmin.ecx/submenu
Search
Home Bookmarks mozilla.org Latest Builds

Wolverine Firewall and VPN Server
Version 1.2.488 running on wolf2

IPSEC Configuration Menu

Edit IPSEC configuration file
Configure IPSEC parameters and tunnel definitions

Manage client certificates
Manage, install, import, and export client certificates

Manage CA certificates
Manage certificate authority certificates

Manage CRL
Manage certificate revocation list

Stop IPSEC
Stop the IPSEC VPN subsystem

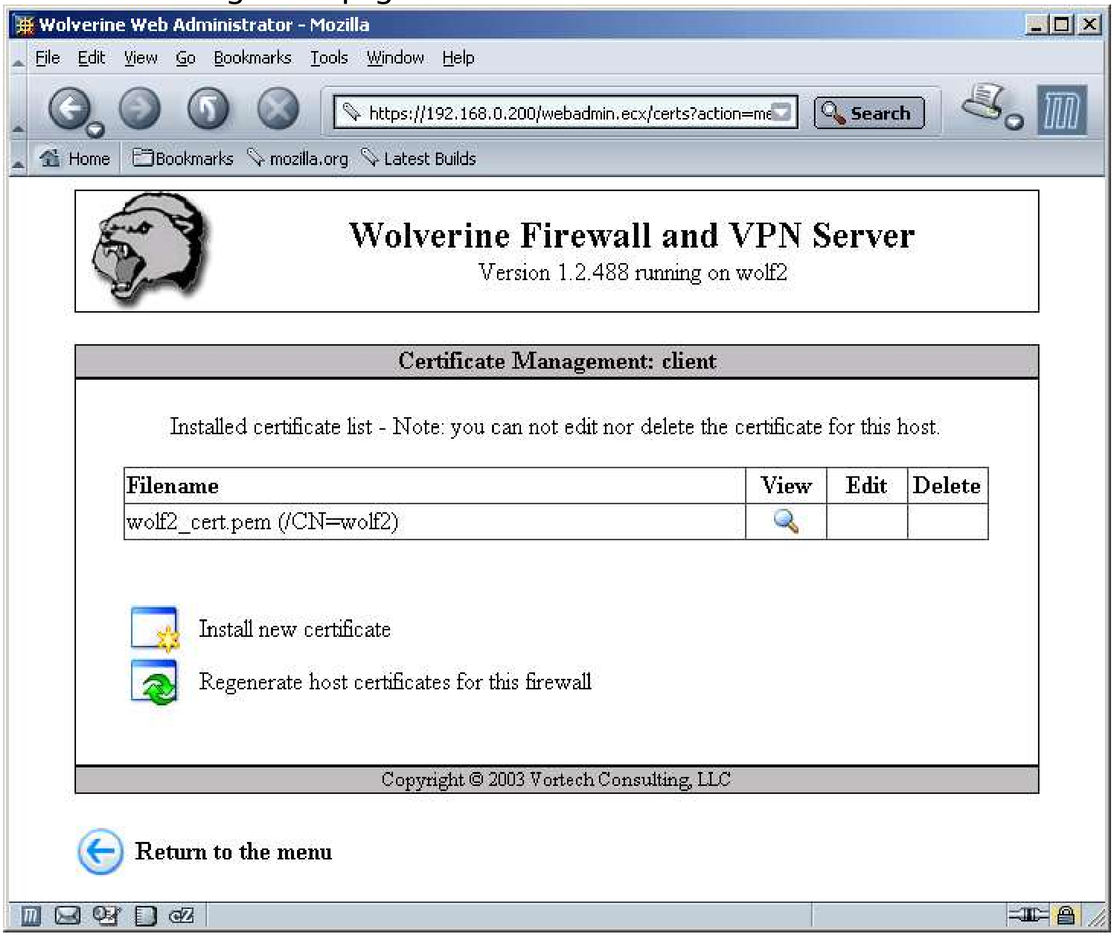
Reload IPSEC configuration
Reload the IPSEC configuration file without taking down exiting tunnels

Main Menu
Return to the main menu

Copyright © 2003 Vortech Consulting, LLC

Edit IPSEC configuration file: This option allows you to edit the main configuration file for the IPSEC subsystem. ***Note:***There is a bit of a quirk when editing this file with the web administrator. The IPSEC configuration file required indented subsection which can not easily be accomplished when editing directly in a browser (due to the browser using the tab key to change fields). You should edit this file in a text editor and paste it back into the edit window.

Manage client certificates: This option lets you edit, view and delete certificates. The certificate for the firewall running the web administrator can not be edited or deleted as it is also used by the web service used to provide the web administration functionality. The following is an image of the client certificate management page:


Wolverine Web Administrator - Mozilla
File Edit View Go Bookmarks Tools Window Help
https://192.168.0.200/webadmin.ecx/certs?action=me
Search
Home Bookmarks mozilla.org Latest Builds

Wolverine Firewall and VPN Server

Version 1.2.488 running on wolf2

Certificate Management: client

Installed certificate list - Note: you can not edit nor delete the certificate for this host.

Filename	View	Edit	Delete
wolf2_cert.pem (/CN=wolf2)	🔍		

Install new certificate

Regenerate host certificates for this firewall

Copyright © 2003 Vortech Consulting, LLC

Return to the menu

You will also notice the "Regenerate host certificates for this firewall" option. This option should not be used unless your certificate/key pair has been compromised or is about to expire. Regeneration of the local certificates will require a reboot for the changes to take effect. You should also re-open your browser after the firewall reboots and before you attempt any further management using the web administrator. Web browsers cache the certificate information for the host they are connected to. Attempting to reconnect to the same host after the certificate changes will produce an error until your browser is restarted.

Manage CA certificates: If your organization uses a certificate authority to create certificates, you can add your CA certificate using this option. This option is not well supported in version 1.2 of Wolverine due to the fact that you can not edit or replace the certificate for the firewall itself. There is also no way to upload a private key file generated outside of the firewall itself. Future releases will include the ability to generate certificate requests for use in organizations that wish to use a central certificate authority.

Manage CRL: This option allows you to add certificates to the Certificate Revocation List (CRL). This option is only required on firewalls using a central CA for client certificate verification. As this is not presently supported, you can simply delete a remote client's certificate from the client certificate list and remove the IPSEC entries to achieve the same effect.

Stop/Start IPSEC: Depending on whether or not IPSEC is currently started, this menu option will give you the option to either start or stop IPSEC.

Reload IPSEC configuration: This option will reload the IPSEC configuration and any changed certificates without bringing down existing tunnels. When the IPSEC configuration is reloaded, you will be simply returned to the menu.

Configuration Reference

This is a list of directives that can be used in the Wolverine configuration file. If you are familiar with the Cisco PIX firewalls, this configuration system will seem very familiar to you.

This configuration file is loaded and parsed during system boot-up or when a reload is forced. The location of this file is /etc/coyote/sysconfig (which is symlinked to /coyote/config on the parent, boot filesystem). When this file is parsed, it is also sorted and invalid directives are removed before it is passed to the system config script. The parsed and sorted version of this file that is actually loaded can be found in /tmp/running-config.

access-list

access-list <list_name> <**permit** | **deny**> <proto> <source[/mask]> <destination[/mask]> [port]

Controls access to resources behind (or in front of) the firewall.

list_name - An arbitrary name for the access list. This name must be alpha-numeric and can not contain spaces or punctuation marks.
permit, deny - Indicates if access should be permitted or denied to the given destination.
proto - The protocol. Typically this would be either "udp" or "tcp". Can also be a protocol number.
source - The source address for the rule. The mask is optional (to apply the rule to a network rather than host).
destination - The destination address for the rule. The mask is optional (to apply the rule to a network rather than host).
port - An optional port number for the rule. If this option is not present ALL ports will be permitted or denied.

Note: The keyword "any" can be used for a source or destination address.

Examples:

```
access-list testlist permit tcp any 123.123.123.1 80
access-list testlist permit tcp 192.168.0.0/24 10.0.0.0/8
```

apply

apply <acl_name> **in** <in-ifname> [**out** <out-ifname>]

Binds a given access control list to the specified interface(s).

acl_name – The access control list name created with an access-list directive.
in-ifname – The name of the inbound interface to match
out-ifname – The (optional) name of the outbound interface to match

Example:

```
access-list myacl permit tcp any 192.168.0.1 22
apply myacl in eth0 out eth1
```

Note: The apply directive is intended for use in transparent firewall configurations when additional fine-tuning of how traffic is matched against access-lists is required. This option may be used in non-transparent configurations, but it not typically required. By default, access-lists are automatically applied by Wolverine and will match any packet being forwarded through the firewall. To enable the use of the apply directive, the following statement must be present in the firewall configuration:

```
option acl-auto-apply disable
```

authentication

authentication <service> <protocol> [protocol2 protocol3 ...]

Specifies the authentication type to use for a given service.

service - The service type.
protocol - The authentication protocol to use

Examples:

```
authentication ppp radius local
```

The above example will attempt radius authentication of an incoming ppp service call (typically a PPTP tunnel) and will fall back to local authentication if radius fails.

auto-forward

auto-forward <interface> <protocol> <ports> <destination>

Forwards all incoming traffic of a given protocol type, destined for a certain port (or range of ports) to the specified host.

interface - The incoming interface. This is typically the same as the public interface.
protocol - The protocol. (tcp or udp).
ports - The port or port range to forward. For a range of ports, the starting and ending port should be separated with a colon. (ie: 2000:3000).

destination - The destination IP address.

Examples:

```
auto-forward eth0 tcp 80 192.168.0.2
auto-forward eth0 udp 900:950 192.168.0.3
```

bridge

bridge <command> <options>

command – specifies the bridge directive command. See below for a valid list of commands and their available options.
options – The options for a given bridge command

Valid bridge commands:

address <ip-address/mask> – Assigns an address to a transparent firewall.
spanning-tree <enable | disable>– Enable or disable the spanning-tree protocol for the bridge. This option would only be useful on a network with multiple bridges.
priority <priority> - Sets the bridges priority. This option would only be useful on a network with multiple bridges.

aging <time> - Sets the bridge aging time in seconds.
hello-interval <interval> - Sets the bridge hello interval in seconds.
garbage-collection <interval> - Sets the bridge garbage collection interval in seconds.
maximum-age <age> - Sets the maximum message age in seconds.
path-cost <interface> <cost> - Sets the path cost for a particular interface.
forward-delay <delay> - Sets the bridge forwarding delay in seconds.
port-priority <interface> <priority> - Sets the priority for a particular interface.

Examples:

```
bridge address 192.168.0.100/24
bridge spanning-tree enable
bridge priority 10
```

clock

clock timezone <zone>

Sets the timezone for the firewall.

zone - Timezone name. Valid entires are:

CST, GMT, GMT+1, GMT+2, GMT+3, GMT+4, GMT+5, GMT+6, GMT+7, GMT+8, GMT+9, GMT+10, GMT+11, GMT+12, GMT-1, GMT-2, GMT-3, GMT-4, GMT-5, GMT-6, GMT-7, GMT-8, GMT-9, GMT-10, GMT-11, GMT-12, UTC, CST, EDT, EST, MST, PST

Note: The timezone names should be specified in upper case (just as listed above).

clock server <host>

host - The host to sync the firewall time to.

Examples:

```
clock timezone EST
clock server time.vortech.net
```

config

config version <xx.xx>

Specifies the configuration version number. This directive will eventually be auto-generated by the configuration editor and should not be altered directly.

dhcpd

dhcpd enable <interface>

interface - Specifies the interface that the DHCP server should be enabled for. Wolverine only supports DHCP assignments for a single network. The public interface can not be used for the DHCP server.

dhcpd address <start> <end>

start - The IP address to use for the start of the DHCP address pool.
end- The address to use for the end of the DHCP address pool.

dhcpd dns | **wins** <address>

address - The address of a DNS or WINS server to send to the client for auto-configuration. If more than one address is needed, use multiple dhcpd statements.

dhcpd lease <time>

time - Specifies the lease time in seconds. If not specified, a 6 hour lease will be used.

dhcpd subnet <subnet-mask>

subnet-mask - Specifies a subnet mask to be used for client auto-configuration.

dhcpd router <address>

address - Specifies a default router address for client auto-configuration.

d**hcpd reserve** <ip-address> <mac-address>

ip-address - An IP address to create a DHCP reservation for.
mac-address – The hardware address of the client to create a reservation for.

A DHCP reservation allows the addition of dynamically assigned static IP address. This means that the DHCP client with a matching MAC address will always be assigned the same IP address.

domain-name

domain-name <domain>

Set the default domain name for the firewall.

domain - The domain name to use

fixup

fixup <protocol>

Enables a firewall connection tracking helper module.

protocol - Name of protocol module to load. The protocols that are currently supported are:

amanda – Amanda backup protocol (http://www.amanda.org)
egg – Eggdrop IRC bots
ftp – File Transfer Protocol
h323 – H.323 Voice over IP protocol
irc – Internet Relay Chat

mms – Microsoft Instant Messenger Service
quake3 – Quake 3 game server protocol
rsh – Remote shell
talk – Talk application protocol
tftp – Trivial File Transfer Protocol

Example:

```
fixup ftp
fixup h323
```

Note: The h.323 module currently only supports outgoing h.323 calls.

hardware

hardware autodetect

Instructs the configuration loader to automatically detect the modules needed to enable the network interface cards in the firewall. The use of this directive will override any “interface module” statements.

hostname

hostname <hostname>

Sets the hostname of the firewall

hostname - The hostname to use for the firewall.

http

http server enable [*port***]**

Enables the internal administrative web server.

http <ip_address>[/mask]

Specifies a source address which is allowed to connect to the administative web services.

Example:

```
http server enable
http 192.168.0.0/24
```

Note: The internal web server only responds to https requests. In order to address the firewall for web access, use the following (where 192.168.0.1 represents the address of the Wolverine firewall):

https://192.168.0.1/

icmp

icmp deny <interface>

This command technically does nothing. The denial of ICMP packets is the default behavior for Wolverine. This command is included simply to more closely mimic the PIX config structure.

icmp permit <ip_address/netmask> <icmp-type> <interface>

Allows ICMP requests on a given interface.

ip_address/netmask - Specifies the host or network that ICMP packets should be accepted from.
icmp-type - The numeric or text (see text type list below) ICMP message type to permit
interface - The interface name or alias

Examples:

```
icmp permit 192.168.0.0/24 echo-request eth0
icmp permit any all eth1
```

ICMP text types that are permitted:

```
echo-reply
destination-unreachable
   network-unreachable
   host-unreachable
   protocol-unreachable
   port-unreachable
   fragmentation-needed
   source-route-failed
   network-unknown
   host-unknown
   network-prohibited
   host-prohibited
   TOS-network-unreachable
   TOS-host-unreachable
   communication-prohibited
   host-precedence-violation
   precedence-cutoff
source-quench
redirect
   network-redirect
```

```
  host-redirect
  TOS-network-redirect
  TOS-host-redirect
echo-request
router-advertisement
router-solicitation
time-exceeded
  ttl-zero-during-transit
  ttl-zero-during-reassembly
parameter-problem
  ip-header-bad
  required-option-missing
timestamp-request
timestamp-reply
address-mask-request
address-mask-reply
```

Note: The use of "any" for the IP address will permit ICMP from any host on the given interface

Note 2: The use of "all" will permit all ICMP types

icmp limit <rate>

Limit the rate of ICMP echo (ping) requests that the firewall will allow. The rate limiting applies to echo requests directed at and through the firewall. The *rate* parameter is the number of echo requests to allow per second.

Example:

```
icmp limit 10
```

This would allow 10 ICMP echo requests per second.

Note: This option does not enable ICMP responses directed at or through the firewall. To enable ICMP responses, you will need to add additional *icmp* or *access-list* statements.

interface

interface <ifname> <**module** | **address | mac** | **public** | **mtu | bridge**> <mod_name | ip_address/netmask | mac_address | **dhcp** | **pppoe** | mtu_size | **enable**> [**down** | **secondary**]

Sets various parameters about the network interfaces that are present in the firewall.

ifname - Specifies the name (eth0, eth1, etc) of a given interface. With the exception of the "module" directive, this can also be an alias specified by a "nameif" statement.
module - Used to indicate the module (driver) needed by the interface. Note – this option is no longer required if the “hardware autodetect” statement is used.
address - Used to indicate the IP address or method of obtaining an IP address for the interface
public - Defines a given interface as "public". This is typically used to indicate the interface that is attached to the Internet.
mtu - Used to specify the MTU size of an interface.
bridge – Used to place interfaces in bridge mode for transparent firewall support.
mod_name - Used in conjunction with the "module" directive. Specifies the actual module name (less the .o).
ip_address/netmask - Used in conjunction with the "address" directive. Specifies the ipv4 address and number of subnet bits for the interface
mac_address – New hardware address to assign to the interface.
dhcp - Used in conjunction with the "address" directive. Indicates that the interface should obtain its address using DHCP.
pppoe - Used in conjunction with the "address" directive. Indicates that the interface should obtain its address using PPPoE.
down - Indicates that the specified interface should be kept offline.
secondary - Used in conjunction with the "address" directive. Indicates that this address is in addition to the "primary" address and should not overwrite any address information that is currently configured. Do not attempt to assign secondary addresses to an interface that was configured using DHCP or PPPoE as the processes responsible for these types of address assignments do not honor the fact that a secondary address has been assigned during address negotiation (in short, the secondary addresses will get wiped out).

Examples:

```
interface eth0 module eepro100
interface eth1 module 3c59x
interface eth0 mac 00:01:02:03:04:05
interface eth0 address 192.168.0.1/24
interface eth0 address dhcp
interface eth0 mtu 576
interface eth0 address pppoe
interface eth0 bridge enable
```

Note: The use of the "module" directive must always precede the use of any other interface directives.

Note 2: The use of PPPoE or DHCP address assignment can only be performed on the public interface. If any interface has the “pppoe” or “dhcp” keywords used, it will automatically become the public interface. During setup, only eth0 can be used for these address types; changing this behaviour is not recommended.

Note 3: When specifying a static IP address, be sure to include the /## prefix-style netmask. If this is not specified, the default IP class mask will be assigned.

Note 4: The use of the pppoe interface directive requires prior pppoe username and password assignements (see the pppoe directive).

Note 5: If a given interface is placed in bridge mode, it can not be assigned an address. To assign addresses to transparent firewalls, see the "bridge" directive.

ip

ip <option> <parameters>
ip conntrack <option> <parameters>

Sets various kernel IP tuning parameters. ***WARNING*** - Do not alter these settings unless you are certain you need to do so. For a description of the available tuning parameters, please see appendix B at the end of this manual.

Examples:

```
ip ecn enable
ip conntrack max-conn 4096
```

logging

logging host <ip_address>

Specifies a remote host to send syslog data to. If no logging host is specified, all data will be logged to the console.

name-server

name-server <server_address>

Adds a DNS server address to the list of servers to use for name resolution.

server_address - The IP address of a server to use for name resolution

nameif

nameif <hardware_name> <alias_name>

Allows an alias name to be specified for a hardware interface name.

Examples:

nameif eth0 external
nameif eth1 dmz
nameif eth2 private

nat

nat <ifname> <source[/mask]> [destination[/mask]]

Starts network address translation for the source address or network.

ifname - The name of the external interface for NAT (eth0, eth1, etc).
source/mask - The source address(es) to translate. If no network mask is supplied, NAT is only performed for the supplied host address.
destination/mask - An optional destination address for the NAT. If specified, only packets destined for the specified address will be NAT'd

nat bypass <source[/mask]> <dest[/mask>

Establishes bypass rules for the NAT tables. This allows connections that would normally be NAT'd to pass through the firewall unaltered.

source/mask - The source host or network address that should be used for the bypass rule
dest/mask - The destination for packets that should not be NAT'd.

nat helper <module_name>

The nat helper directive has been deprecated. See the “fixup” directive.

option

option <option-name> <parameters>

Set various firewall optional parameters.

Valid optional parameter names and values include:

*acl-auto-apply <**enable** | **disable**>* - Enables or disables access-list auto-apply. By default this option is enabled.

*ipsec <**enable** | **disable**>* - Enables or disables the Wolverine ipsec subsystem. By default this option is enabled. If you do not plan to use IPSEC it can be disabled to conserve firewall memory.

Examples:

option acl-auto-apply disable
option ipsec disable
option upnp enable

password

password <**user** | **monitor** | **admin** | **debug**> <data> [encrypted]

Specifies the various passwords for the user, monitor, and admin level security. Currently, only the "admin" and "debug" levels are actually supported.

When logging into Wolverine, the "admin" login will give you the main menu immediately after logging in, while the "debug" login will give a BASH shell prompt.

When using the main menu option to edit the system configuration, any passwords specified with this command will be automatically encrypted when you leave the editor.

Note: When logging is as the admin user, you can also use the login name of "root".

pppoe

pppoe user <username> <password>

Configures a username and password for an outbound PPPoE connection. Note that only one PPPoE username and password can be specified on the firewall. If multiple interfaces use PPPoE connections, they must share the same username and password.

pppoe demand <timeout>

Configures a demand-dial timeout for PPPoE links. The timeout value is the number of seconds a link should be idle before dropping the connection.

pppoe clamp-mss [mss size]

Configures the firewall to clamp the maximum segment size for TCP connections that have the SYN flag set. This helps with PPPoE connections to sites/servers that do not properly process ICMP fragmentation-needed packets. If the [mss size] is not specified, the discovered PMTU – 40 will be used.

Examples:

```
pppoe clamp-mss
pppoe clamp-mss 1412
```

port-forward

port-forward <from_address> <to_address> [protocol [port [dest-port]]]

Allows for port address translation based on the original destination address. In addition to the port-forward command, you will also need to add an access-list statement to allow forwarding of the specified traffic.

from_address - The address that should be redirected.
to_address - The destination of the redirection.
protocol - The protocol to forward. Can be a protocol name (tcp, udp, icmp) or a protocol number.
port - If the protocol being forwarded is tcp or udp, a specific port can be specified. If no port number is specified, all ports for the given protocol will be redirected. If the protocol is not tcp or udp, do not specify a port!
dest-port - For tcp or udp, a separate destination port can be specified. This allows mapping an external port on the firewall to a service on an internal server that is running on a different port

Examples:

This example forwards all connections made to port 80 on 123.123.123.1 to the internal server 192.168.0.1:

```
port-forward 123.123.123.1 192.168.0.1 tcp 80
access-list portfw1 permit tcp any 192.168.0.1 80
```

If the external port is 8080, you could use the following:

```
port-forward 123.123.123.1 192.168.0.1 tcp 8080 80
access-list portfw1 permit tcp any 192.168.0.1 80
```

Note that "portfw1" is an arbitrary name for the access list.

pptp

pptp <**user** | **local-address** | **address-pool** | **wins-server** | **dns-server** | **proxyarp** > <data>

Configures the PPTP server to allow incoming VPN connections.

user - Specifies the name of a user to allow to connect to the PPTP server. The username should be followed by the password to assign for this user account
local-address - Specifies the VPN server-side IP address which should be used in creating the VPN tunnels. This is typically an address in the subnet that you will be accessing via the VPN tunnel.
address-pool - This specifies the range of IP addresses used for the remote end-points of the VPN tunnels. This is specified in the format of 123.123.123.1-123.123.123.10, but can be shortened to the following syntax: 123.123.123.1-10.
proxyarp - If the local-address and address-pool directives specify addresses that are part of a network assigned to one of the firewall's internal interfaces, this directive should be used so that the firewall will perform a proxy-arp on behalf of the remote computer.

Examples:

```
pptp proxyarp
pptp wins-server 192.168.0.100
pptp dns-server 192.168.0.200
pptp user testuser testpass
pptp local-address 192.168.0.2
pptp address-pool 192.168.0.3-20
```

Note: You MUST have an *authentication* directive present for the PPTP server to function properly.

proxyarp

proxyarp <ext_if> <**host**|**network**> <address[/netmask]> <int_if>

Establishes proxy ARP for a give address or network segment.

ext_if - The interface that is to publish the address(es).
host - Indicates that the address will be a host address (/32 netmask)
network - Indicates that the firewall should perform proxyarp for the give subnet
address[/netmask] - The address to publish on ext_if
int_if - The interface that is physically connected to the host or network

Example:

```
proxyarp eth0 host 192.168.0.5 eth1
```

radius-server

radius-server host <address> [port]

address - The hostname or IP address for the Radius Server
port - Specify the port that will be used to communicate with the Radius Server. If this is not specified, port 1812 will be used.

radius-server key <key_string>

Establishes Radius authentication parameters for PPTP VPN connections.

key_string - The text shared secret for radius server authentication

Examples:

radius-server host 192.168.0.3 1812
radius-server key My$ecr3t

route

route <source/mask> <gateway> [**dev** ifname] [**metric** metric_num]

Establishes static routing information.

source/mask - The source network number and bit mask.
gateway - The remote IP address to route traffic through
ifname - Specifies the network interface to use for this route. "ifname" should be the interface identifier (eth0, eth1, etc).
metric_num - Specifies a metric for this route.

snmp

snmp <contact | location | host> <data>

Sets various parameters which are passed to the internal SNMP service.

contact - Specifies the contact name for the firewall.
location - Specifies the system location
host - Indicates a host IP address that is allowed to query the SNMP service.
data - A string containing the text data for the above directives

NOTE: In order to use the SNMP server, the domain name and DNS servers must be specified or the SNMP service will fail to start properly. These can either be specified via the "domain-name" and "name-server" directives, or from the automatic specifications provided by using a DHCP assigned Internet address.

ssh

ssh <address[/mask]>

Sets allowed hosts for ssh access to the firewall

telnet

telnet <address[/mask]>

Sets allowed hosts for telnet access to the firewall. You should only enable telnet access to hosts connecting across a known-secure network segment. Any login information (including passwords) are transmitted in clear-text. For this reason, telnet access should never be enabled for hosts residing on the Internet or any other public/untrusted network.

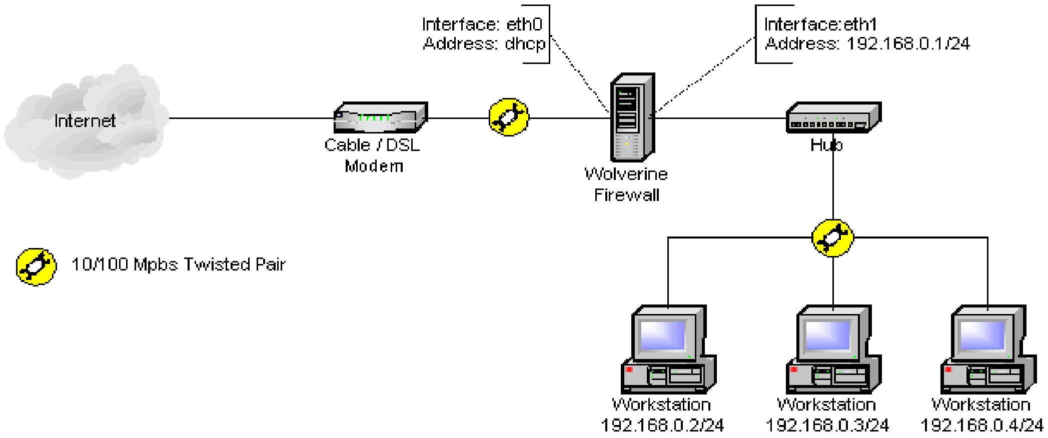
Example Configurations

This section contains some diagrams and sample configurations for typical Wolverine firewall applications. For these examples, some assumptions are:

192.168.x.x are used to indicate private address ranges
172.20.x.x are used to indicate public, Internet routeable addresses.

Even though 172.20.x.x is defined for private use, they represent what would normally be Internet routeable addresses for the purpose of these examples.

Small Office / Home Office (SOHO) with NAT and DHCP server



Configuration:

```
#Embedded Cyote Linux Configuration File
config version 1.3
hostname testfw
domain-name testdomain.com
clock timezone EST
clock server time.vortech.net
name-server 172.20.0.100
password admin xxxxxxxxxxxx encrypted
password debug xxxxxxxxxxxx encrypted
hardware autodetect
interface eth0 address dhcp
interface eth1 address 192.168.0.1/24
dhcpd address 192.168.0.100 192.168.0.200
dhcpd subnet 255.255.255.0
dhcpd lease 86400
dhcpd router 192.168.0.1
dhcpd domain testdomain.com
dhcpd dns 172.20.0.100
dhcpd enable eth1
fixup ftp
fixup irc
nat eth0 192.168.0.0/24
access-list natout permit all 192.168.0.0/24 any
telnet 192.168.0.0/24
ssh 0.0.0.0/0
http server enable
http 192.168.0.0/24
```

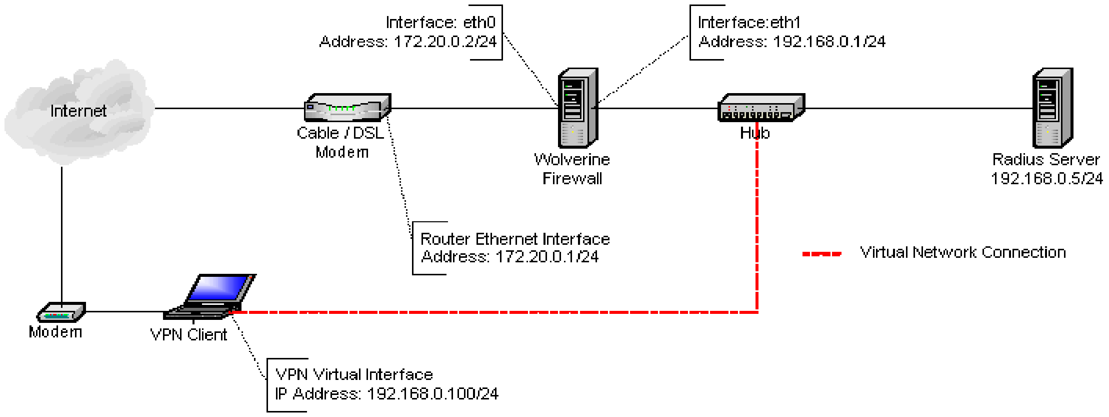
Alternate configuration using PPPoE Internet connection

This example is as identical configuration to the previous example with the exception of the Internet connection type begin PPPoE.

Configuration:

```
#Embedded Cyote Linux Configuration File
config version 1.3
hostname testfw
domain-name testdomain.com
clock timezone EST
clock server time.vortech.net
name-server 172.20.0.100
password admin xxxxxxxxxxxx encrypted
password debug xxxxxxxxxxxx encrypted
pppoe user ispuser isppass
hardware autodetect
interface eth0 address pppoe
interface eth1 address 192.168.0.1/24
dhcpd address 192.168.0.100 192.168.0.200
dhcpd subnet 255.255.255.0
dhcpd lease 86400
dhcpd router 192.168.0.1
dhcpd domain testdomain.com
dhcpd dns 172.20.0.100
dhcpd enable eth1
fixup ftp
fixup irc
nat ppp0 192.168.0.0/24
access-list natout permit all 192.168.0.0/24 any
telnet 192.168.0.0/24
ssh 0.0.0.0/0
http server enable
http 192.168.0.0/24
```

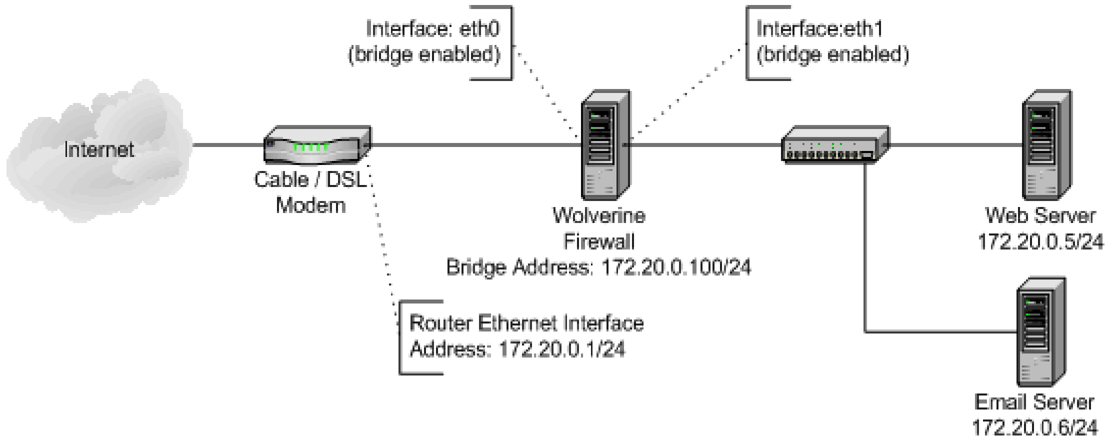
PPTP VPN Server using Radius Authentication with fall-back local accounts



Configuration:

```
#Embedded Cyote Linux Configuration File
config version 1.3
authentication ppp radius local
hostname testfw
domain-name testdomain.com
clock timezone EST
clock server time.vortech.net
name-server 172.20.0.100
password admin xxxxxxxxxxxx encrypted
password debug xxxxxxxxxxxx encrypted
hardware autodetect
interface eth0 address 172.20.0.2/24
interface eth1 address 192.168.0.1/24
route 0.0.0.0/0 172.20.0.1
radius-server host 192.168.0.5
radius-server key SeCrEtKeY
fixup ftp
fixup irc
nat eth0 192.168.0.0/24
access-list natout permit all 192.168.0.0/24 any
pptp local-address 192.168.0.99
pptp address-pool 192.168.0.100-199
pptp proxyarp
pptp user localuser localpass
telnet 192.168.0.0/24
ssh 0.0.0.0/0
http server enable
http 192.168.0.0/24
```

Transparent firewall example



Configuration:

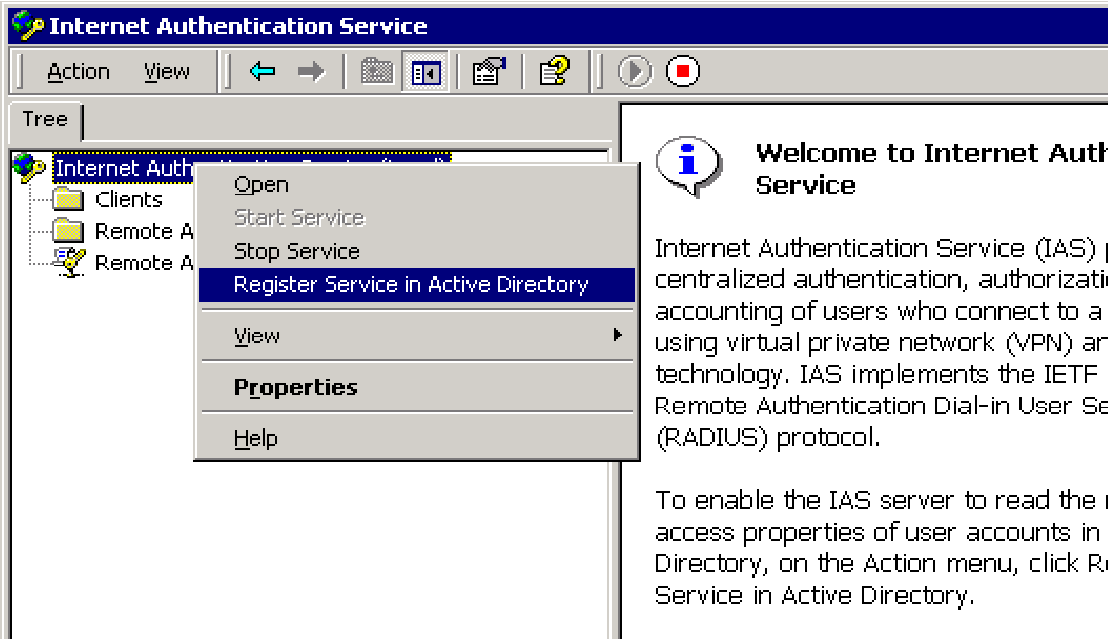
```
#Embedded Cyote Linux Configuration File
config version 1.3
authentication ppp radius local
hostname testfw
name-server 172.20.0.10
domain-name testdomain.com
clock timezone EST
clock server time.vortech.net
name-server 172.20.0.100
password admin xxxxxxxxxxxx encrypted
password debug xxxxxxxxxxxx encrypted
hardware autodetect
interface eth0 bridge enable
interface eth1 bridge enable
bridge address 172.20.0.100/24
route 0.0.0.0/0 172.20.0.1
access-list websrv1 permit tcp any 172.20.0.5 80
access-list emailsrv1 permit tcp any 172.20.0.6 25
access-list emailsrv1 permit tcp any 172.20.0.6 110
# Allow the protected servers to talk to anything, anywhere
access-list outacl permit all 172.20.0.5 any
access-list outacl permit all 172.20.0.6 any
ssh 0.0.0.0/0
http server enable
http 172.20.0.0/24
```

Configuring Microsoft Internet Authentication Service

To integrate the Wolverine PPTP server with a Windows 2000 Active Directory, you can use the Microsoft Internet Authentication Service (IAS). This is basically Microsoft's fancy name for their implementation of Radius. The installation of IAS is part of IIS which is not typically installed. You will need to install it from the Windows Components portion of the Add/Remove programs option in the Windows 2000 control panel.

NOTE: Integration with Active Directory is not required for IAS to function. If the server you are using for Radius authentication is a stand-alone server, the local user database will be used instead.

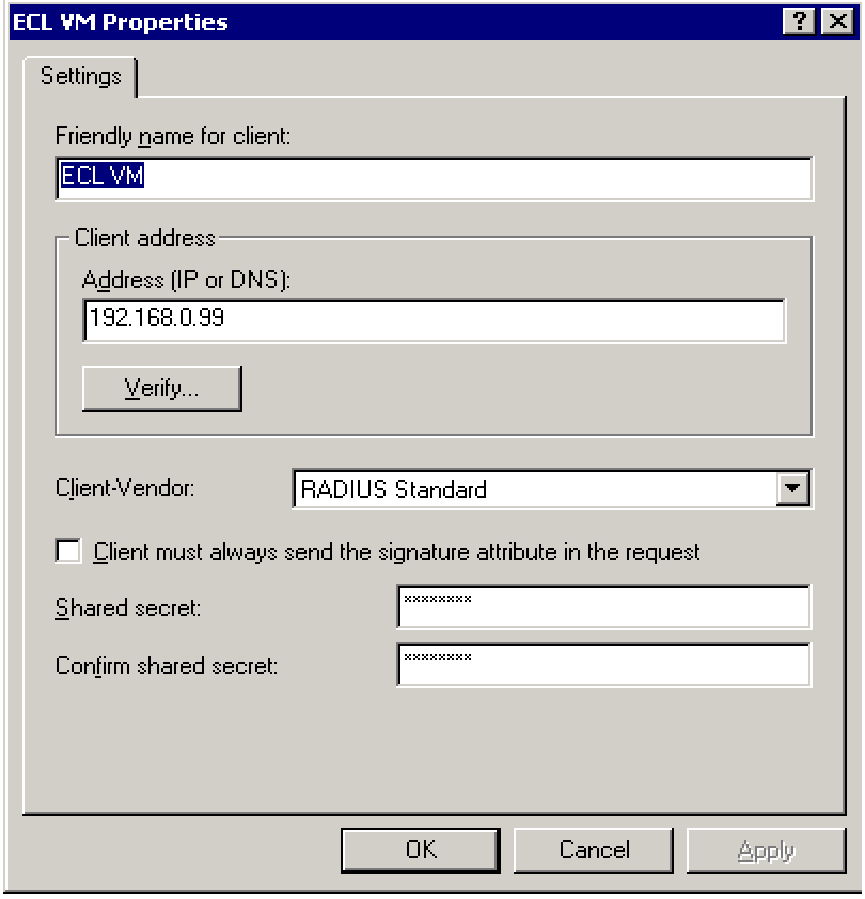
Once installed, you will need to register the IAS server in the Active Directory. This is accomplished using the IAS administration tool (found under Start->Programs->Administrative Tools). Here is a screen shot of the registration process:


Welcome to Internet Auth
Service

Internet Authentication Service (IAS) p
centralized authentication, authorizati
accounting of users who connect to a
using virtual private network (VPN) ar
technology. IAS implements the IETF
Remote Authentication Dial-in User Se
(RADIUS) protocol.

To enable the IAS server to read the r
access properties of user accounts in
Directory, on the Action menu, click R
Service in Active Directory.

This menu is available by right clicking on the root level option in the IAS administration tool. Once the service is registered you will need to let the IAS server know that the Wolverine firewall is permitted to authenticate users. Select the "Clients" folder that can be seen in the above graphic and right click. On the context menu, you will see an option for "New Client". The only information needed about the Wolverine firewall is the IP address of the interface communicating with the IAS server (typically this will be the private interface connected to the segment containing the IAS server) and a shared-secret. The shared secret is basically just a password shared between the IAS server and the Wolverine firewall. For information on configuring the Wolverine firewall's Radius server options, see the "Configuring the built-in PPTP server" section of this manual. The options window for adding a new client can be seen in the following image:



ECL VM Properties

Settings

Friendly name for client: ECL VM

Client address

Address (IP or DNS): 192.168.0.99

Client-Vendor: RADIUS Standard

☐ Client must always send the signature attribute in the request

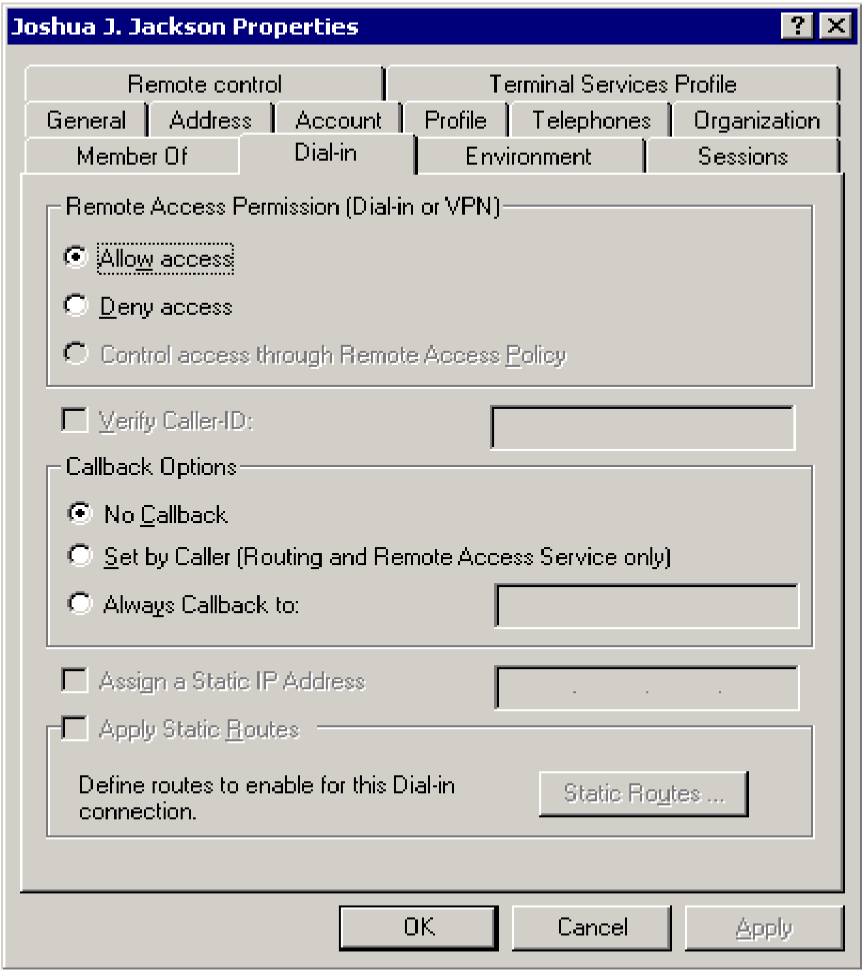
Shared secret: ********

Confirm shared secret: ********

The "Friendly name" is an arbitrary description for the Wolverine firewall. The IP address field should contain the address of the internal interface of the Wolverine firewall. The Client-Vendor should be left at "RADIUS Standard". At this point, the IAS server itself should be properly configured for providing the MSCHAPv2 authentication needed for MPPE128 encrypted PPTP tunnels.

The default access policy for users to be authenticated used IAS simply relies on the dial-in permissions in combination with login hour restrictions. While this can be changed to provide authentication based on Group, Group Policies, or a fully customized set of options, this is beyond the scope of this documentation. To enable dial-in permissions for an Active Directory account, use the Active Directory Users and Computers administration application to edit the options for a specific user. Select the "Dial-in" tab and select "Allow Access" from "Remote Access Permissions (Dial-in or VPN)".

The following is an image of editing the dial-in permissions for a given Active Directory user:



Remote Access Permission (Dial-in or VPN)

● Allow access

○ Deny access

○ Control access through Remote Access Policy

□ Verify Caller-ID:

Callback Options

● No Callback

○ Set by Caller (Routing and Remote Access Service only)

○ Always Callback to:

□ Assign a Static IP Address

□ Apply Static Routes

Define routes to enable for this Dial-in connection.

At this point, you can use an Active Directory username and password for authentication to a Wolverine PPTP server. If you are having troubles with authentication, check the Event Viewer for entries indicating authentication success or failure for Radius requests from the Wolverine Firewall.

Appendix A - Ordering Wolverine

To place an order for Wolverine, you can visit the Coyote Linux (http://www.coyotelinux.com) or Vortech Consulting (http://www.vortech.net) web sites. To jump directly to the Wolverine purchase page, you can use this URL: https://secure.vortech.net/subscriptions/wolverine/subscribe.php.

All purchases come with 6 months of updates to any released version of Wolverine as well as access to the update servers (for direct updates from the firewall itself).

Personal License

This subscription level grants the user a single license to the Wolverine Firewall and VPN server for non-commercial use. Access to download the installation CD-ROM image and documentation are granted for personal subscriptions.

Cost: USD$30.00

Commercial License

With a commercial subscription, the user is given access to the files that are available in the developer subscription, and future versions of the GUI remote administration utilities. This subscription level also grants an organizational license to Wolverine Firewall and VPN server for up to 5 firewalls.

Cost: USD$120.00

Educational or Reseller licensing

If you would like to use Wolverine in an educational facility or would like to become an authorized Wolverine reseller, please contact Joshua Jackson at Vortech Consulting (jjackson@vortech.net) for further information. Educational facilities may qualify for a free site license and resellers can obtain quantity discounts.

Ordering the Wolverine media kits

During your purchase of Wolverine you will also be given the option to purchase the physical CD media and printed documentation. This option is USD$20.00 plus shipping charges.

Appendix B – IP Tuning Parameters

This appendix describes the various IP tuning parameters and what they are used for.

General tuning parameters

Explicit congestion notification

usage: ip ecn <enable | disable>
default: disabled

This option is used to automatically tell the host when there are congestions in a route to a specific host or a network. This can be used to throttle the transmitters to send packets at a slower rate over that specific router or firewall. Explicit Congestion Notification (ECN) is explained in detail in the *RFC 3168 - The Addition of Explicit Congestion Notification (ECN) to IP* document and there is also a performance evaluation of the addition of ECN available in the *RFC 2884 - Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks* document.

Briefly, this document details how a firewall could notify other hosts whether or not it is congested, which can then to choose other routes in preference over the currently used route, or to simply send less data until it no longer receives congestion messages.

> There are still some old firewalls and routers out on the Internet that will discard all IP packets that have the ECN bits set. They are fairly uncommon these days, but it is possible you may run into them. If you experience connection problems to specific hosts, try turning ECN off. If you find the actual host blocking the ECN packets, try getting in touch with the administrators and warn them about this.

SYN backlog

usage: ip syn-backlog <value>
default: 1024 for system with > 128MB of RAM, 128 for systems with less than 128MB of RAM.

This option tells the firewall how many SYN requests should be tracked which have not completed the entire 3-way TCP connection handshake. If your firewall has more than 128MB of RAM, you should not need to increase the default value above 1024. For extremely busy firewalls with less than 128MB of RAM which also implement a large number of port or auto forward statements, this value can be increased if connection problems occur. However, it may be a better idea to simply upgrade the firewall.

Connection Tracking Options

usage: ip conntrack <option> <value>

The connection tracking tuning parameters allow for the adjustment of various settings that deal with the Linux kernel's IP connection tracking code. These parameters are normally set to ideal values unless you have a very busy firewall that needs to keep track of thousands of simultaneous connections.

The various options for tuning the connection tracking features include:

ip conntrack max-conn <value>

This option specifies the maximum number of connections may be tracked by the firewall at any given time. The default value will depend on the amount of RAM present in the firewall. For firewalls with 512MB or more this option will default to the recommended maximum of 32768. It will be lower on firewalls with less memory.

On Wolverine, this value should not be assigned a value larger than 32768. This is due to a fixed kernel connection tracking hash table size. Certain connection tracking options (namely GRE and PPTP tracking) required that they connection tracking module be build directly into the Linux kernel. Unless the module can be loaded dynamically, the hash table can not be resized. Values above 32768 may cause a decrease in overall network throughput performance.

ip conntrack udp-timeout <value>

This option specifies the timeout for initial UDP packets in a connection. When a UDP connection is initialized, the UDP connection enters an NEW and then ESTABLISHED state once it has seen return traffic for the connection. However, it maintains the same timeout until it has seen several packets go back and forth and becomes ASSURED, at which point it is considered a stream (see udp-stream-timeout below).

While this initial state is maintained, the default timeout is 30 seconds. If you are using UDP protocols that send very little data during longer timeframes, you should consider raising this value so that the connection tracking code is able to keep track of your connections properly. It is generally a bad idea to lower this, unless you know that your hosts send UDP packets very often and don't expect a lot of late replies, causing a lot of unnecessary open connection tracking entries.

ip conntrack udp-stream-timeout <value>

This option specifies the timeout value for UDP streams once they have sent enough packets to reach the ASSURED state. This state is normally reached for connections that send a lot of data and relatively often, such as streaming services, voice chats, or ICQ. Examples of streaming services may be certain realplayer servers, or speak freely. This value should always be larger than the initial timeout value for UDP connections (see udp-timeout above).

The default for this option is 180 seconds. If you are having problems with connections timing out, you may try raising this value a bit. It is generally a bad idea to lower this value, since the connection will be destroyed once it times out from this state. Unfortunately, UDP is a stateless protocol, so it is very hard to derive any specific states of the connections. As a result of this, there is no specific timeout for UDP streams that are about to or have already closed.

ip conntrack tcp-close-timeout <value>

This option sets the timeout of the CLOSE state as defined in RFC 793. The CLOSE state is entered if the client sends a FIN and then receive a FIN from the server, and replies with a FIN/ACK to the FIN sent from the server. The CLOSE state is then maintained until a FIN/ACK is received, replying the clients original FIN. When the final FIN/ACK arrives the connection enters the TIME-WAIT state.

The default timeout of this variable is set to 10 seconds. This should be a good value, but if you start noticing a lot of clients gets hung in the CLOSE state, you can raise this value until the CLOSE state problem is no longer observed.

ip conntrack tcp-close-wait-timeout <value>

This option sets the timeout value of the CLOSE-WAIT state as defined in the RFC 793. This state is entered if the client receives a FIN packet and then replies with a FIN/ACK packet. At this point, the connection will enter a CLOSE-WAIT state. This state is then maintained until the client sends out the final FIN packet, at which time the connection will change state to LAST-ACK.

The default value for this option is set to 60 seconds This timeout should in general be large enough to let most connections exit the CLOSE-WAIT state. If you are having problems with connections being improperly timed out, this value can be increased. On some Linux based systems this value defaults to 12 hours.

ip conntrack tcp-established-timeout <value>

This option tells us the default timeout value for tracked connections in the ESTABLISHED state. All connections that have finished the initial 3-way handshake, and have not seen any kind of FIN packets are considered ESTABLISHED. This is the default state for active TCP connections.

Since you never want a connection to be lost on either side of the firewall, this timeout is set extremely high so entries are not accidentally erased which are still used. This option has a default of 432000 seconds, or 5 days.

ip conntrack tcp-fin-wait-timeout <value>

This option defines the timeout values for both the FIN-WAIT-1 and FIN-WAIT-2 states as described in RFC 793. The FIN-WAIT-1 state is entered when the server send a FIN packet, while the FIN-WAIT-2 state is entered when the server receives a FIN/ACK packet from the client in return to the initial FIN packet. If the server receive a FIN while still waiting for the FIN/ACK packet it enters the CLOSE (defined as CLOSING in RFC 793) state instead of FIN-WAIT-2.

The default for this option is 120 seconds. The default here should generally be a good value but may be raised in case clients and servers are left in a FIN-WAIT state because the firewall stopped forwarding the packets due to the entry timing out in advance. If you have problems with running out of connection tracking hash entries, this value may also be lowered, but not more than allowing the full closing handshake to take place on both ends.

ip conntrack tcp-syn-recv-timeout <value>

This option sets the timeout value for the SYN-RECEIVED (also known as SYN-RCVD or SYN-RECV) state as defined in RFC 793. The SYN-RECEIVED state is entered from the LISTEN or SYN-SENT state once the server receives a SYN packet and then replies with a SYN/ACK packet. The SYN-RECEIVED state is left for the ESTABLISHED state once the server receives the final ACK packet in the 3-way handshake.

The default for this option is 60 seconds. This should generally be a good value, but if you do experience timeouts where your server or clients end up stuck in a SYN-RECV or SYN-SENT state, you should consider raising this value a bit. It is generally a bad idea to lower this variable to circumvent problems with connections timing out.

ip conntrack tcp-syn-sent-timeout <value>

This option sets the timeout of the SYN-SENT state as defined in RFC 793. The SYN-SENT state is entered once the client has sent out a SYN packet and is still awaiting the returning SYN/ACK packet. The SYN-SENT state is then left for the SYN-RCVD or ESTABLISHED states once a SYN/ACK packet has been received.

The default for this option is set to 120 seconds. This should generally be a good setting unless you experience problems where the client and server get stuck in a SYN-SENT or SYN-RCVD state. If this is the case you may need to raise this value. It is generally a bad idea to lower this value to get around problems with not having enough connection tracking entries.

ip conntrack tcp-time-wait-timeout <value>

This option defines the timeout value of the TIME-WAIT state as defined by RFC 793. This is the final state possible in a TCP connection. When a connection is closed in both directions, the server and client enter the TIME-WAIT state, which is used so that all stale packets have time to reach the client and/or server. One example of the usage of this may be if packets are reordered during transit between the hosts and end up in a different order at either side. In such a case, the timeout defined with this is used so that these packets may reach their destinations before the connection tracking entry is destroyed. When the timeout expires, the entry is destroyed and the state is set to CLOSED, which means that there is not expected to be any additional traffic for this specific connection.

The default for this option is set to 120 seconds. This value is typically sufficient unless the quality of your connection is extremely poor. If this value is too low you will often experience corrupt downloads or missing bytes in downloaded data. If your firewall only communicates with other hosts on other low-latency networks, this value can be lowered to recycle the connection tracking entry space faster.

ip conntrack generic-timeout <value>

This variable is used to set the firewall connection tracking code's generic timeout in case it can not determine the protocol used or if the protocol does not have a predefined set of timeout options. Any stream or packet that enters the firewall that can not be fully identified as any of the other protocol types in this section will get a generic timeout set to it.

This option takes an integer value and defaults to 600 seconds. If this is not enough for your applications, you should raise it until connections problems no longer occur.

ip conntrack icmp-timeout <value>

This option is used to set the timeout for ICMP packets that will result in return traffic. This includes echo request and reply, timestamp request and reply, information request and reply, and address mask request and reply.

As an ICMP reply is expected to be quite quick with the reply returning to, or through, the firewall within a couple of seconds. The default value is set to 30 seconds. This should generally be a good value, unless you have an extremely poor connection.