OpenVPN

Version: 3573

Copyright 2007-2010 ImageStream Internet Solutions, Inc., All rights Reserved.

Table of Contents

OpenVPN

Using OpenVPN with Imagestream Router

Introduction

Imagestream routers have built in support for OpenVPN tunnels. There are two types of tunnel interfaces, tun and tap. Tun interfaces are point to point interfaces, one interface can support one peer at a time. Tap interfaces are point to multipoint interfaces, this is used in client/server setups. Both types of tunnel interfaces look and act like normal network interfaces. Both Linux and Windows support Peer to Peer(tun) and Client/Server(tap) modes. This document covers setup of the base command-line OpenVPN client in Linux, and a Windows GUI based client. Later revisions will cover GUI configuration in Linux.

Requirements

If you are using Linux, you must have OpenSSH and OpenVPN both installed. Both are available from most distributions. If you are using Windows, you must have PuTTY and PSCP, as well as an OpenVPN client, we use OpenVPN GUI. Furthermore, you must know what IP addresses are in use on your network, as well as have a moderate understanding of TCP/IP networking. The IANA assigned port 1194/UDP to OpenVPN, so you should make sure that your firewall rules allow that traffic through

Peer to Peer mode

To setup a Peer to Peer tunnel, you must have a /30 block of addresses, and know the IP address of an interface on the Imagestream router. In this example we use 192.168.45.0/30 for our tunnel network and the IP address assigned to the router is 192.168.42.42. The router side of the tunnel is assigned 192.168.45.1, and the PC (or other) side of the tunnel is assigned 192.168.45.2.

Side 1: Imagestream router

```
!
interface Tunnel0
 tunnel mode openvpn
 tunnel source 24.34.2.1 1194
 tunnel destination 53.23.15.1 1194
 tunnel key b232892562bde187af65431ecc643147
 pointopoint address 192.168.45.2
 ip address 192.168.45.1 255.255.255.252
#  tunnel options --route 192.168.69.0 255.255.255.0
!
```

Tunnel0 is the name of the interface, Imagestream routers require the device to be named TunnelX where X is a non-negative integer, you will need the name of the interface in step two. `Tunnel mode openvpn` sets a basic openvpn tunnel, client/server setups will have a different setting there.

`Tunnel source 0.0.0.0 1194` sets the router to listen to port 1194 on any interface. `Tunnel key b232892562bde187af65431ecc643147` sets the shared key to use for the tunnel. `Pointtopoint address 192.168.45.2` sets the IP address of the remote side of the tunnel. `Ip address 192.168.45.1 255.255.255.252` sets the IP address and netmask of this side of the tunnel. You can optionally use `tunnel options --route <network> <netmask>` to add a route to the other side of the link. This is useful to let peers talk to other systems in your network.

Side 2: Windows

1. Download and install OpenVPN GUI. Once that is installed, open a command shell with Start>Run(cmd) and perform these commands:
2. `cd \program files\openvpn\config`
3. `edit Tunnel0.ovpn` and type in these lines:

```
dev Virtual_Tap_Device
dev-type tun
ifconfig 192.168.45.2 192.168.45.1
secret Tunnel0.key
remote 192.168.42.42
rport 1194
ping            5
ping-restart    15
ping-timer-rem
persist-tun
persist-key
tun-mtu 1500
comp-lzo
float
```

1. Use pscp to copy /etc/openvpn/key-Tunnel0 to c:\Program Files\OpenVPN\config\Tunnel0.key. The command should be something like `pscp root@<ROUTER_IP>:/etc/openvpn/key-<INTERFACE_FROM_SIDE_ONE> Tunnel0.key`.
2. Alternate-click on the OpenVPN GUI icon in your system tray, select Tunnel0->Connect and your Windows PC will create a connection to the Imagestream Router

Side 2: Generic Linux

1. Login as root.
2. Download and install OpenVPN for your distribution. Then open a text editor and type the following:

```
dev Tunnel0
dev-type tun
ifconfig 192.168.45.2 192.168.45.1
secret /etc/openvpn/Tunnel0.key
# This is the IP address of the router mentioned at the
#   beginning of this section.
remote 205.159.243.85
rport 1194
ping            5
ping-restart    15
ping-timer-rem
```

```
persist-tun
persist-key
tun-mtu 1500
comp-lzo
float
```

1. File->Save(/etc/openvpn/Tunnel0.conf), you will have to be root to do this.
2. Copy the key file from the router to the local system, `scp root@<ROUTER_IP>:/etc/openvpn/key-Tunnel0 /etc/openvpn/Tunnel0.key`
3. As root, run `openvpn --config /etc/openvpn/Tunnel0.conf` to start the connection. Alternatively `/etc/init.d/openvpn start` should work.

Client / Server mode

To setup a client / server tunnel, you must have a netblock not used elsewhere, a username / password combination, and know the IP address of an interface on the Imagestram router. In this example we use 192.168.45.0/24 as the netblock, rjune / test123 as the username / password, and 192.168.45.1 is the IP address assigned to the router.

Side 1: Imagestream Router

```
!
user rjune password test123
!
interface Ethernet0
 ip address 205.159.243.85 255.255.255.0
!
interface Tunnel0
 tunnel mode openvpn server 192.168.45.0 255.255.255.0
 tunnel options --dev-type tap --passtos --push "route 192.168.0.0 255.255.255.0" --push "route 1
 ip address 192.168.45.1 255.255.255.0
!
```

Tunnel0 is the name of the interface, Imagestream routers require the device to be named TunnelX where X is a non-negative integer. `Tunnel mode openvpn server 192.168.45.0 255.255.255.0` sets the tunnel to server mode and tells it that 192.168.45.0/24 is the network to be used on the interface. `Tunnel options --dev-type tap --passtos` sets the interface to tap mode and does TOS magic. Optionally add `--push "route <network> <netmask>"` to add a route to the local network for incoming clients. `Ip address 192.168.45.1 255.255.255.0` sets the IP address and netmask of this side of the tunnel.

Side 2: Imagestream Router

```
!
interface Tunnel0
 description Client Tunnel
 tunnel mode openvpn client username rjune password test123
 tunnel destination 205.159.243.85
 tunnel options --dev-type tap --float
!
```

Tunnel0 is the name of the interface, Imagestream routers require the device to be named TunnelX where X is

a non-negative integer. `Tunnel mode openvpn client username rjune password test123` sets the tunnel to client mode with login credentials of rjune/test13. `Tunnel destination <ROUTER_IP>` tells the router to connect to the OpenVPN server at <ROUTER_IP>. And finally, `tunnel options --dev-type tap` sets the tunnel to a tap style device.

Side 2: Windows

1. Download and install OpenVPN GUI. Once that is installed, open a command shell with Start>Run(cmd) and perform these commands
2. `cd \program files\openvpn\config`
3. *edit Tunnel0.ovpn* and type in these lines:

```
dev Virtual_Tap_Device
dev-type tap
remote 205.159.243.85
client
auth-retry nointeract
auth-user-pass
ca imagestream_ca.crt
cert openvpn.crt
key openvpn.key
ping            5
ping-restart    15
ping-timer-rem
persist-tun
persist-key
tun-mtu 1500
comp-lzo
float
```

1. Copy the server keys to the workstation (`pscp root@<ROUTER_IP>:/etc/rsa-keys/openvpn.* .`)
2. Copy the server certificate to the workstation (`pscp root@<ROUTER_IP>:/usr/share/etc/imagestream_ca.crt .`)
3. Alternate-clicks on the OpenVPN GUI icon in your system tray, select Tunnel0->Connect and your Windows PC will create a connection to the Imagestream Router. You will be prompted for a username and password. rjune is the username, and test123 is the password.

Side 2: Generic Linux

1. Login as root
2. Open a text editor and type the following into it. Then save it as /etc/openvpn/Tunnel1.conf.

```
dev Tunnel1
dev-type tap
remote <ROUTER_IP>
client
auth-retry nointeract
auth-user-pass /etc/openvpn/Tunnel1.pass
ca /usr/share/etc/imagestream_ca.crt
cert /etc/rsa-keys/openvpn.crt
key /etc/rsa-keys/openvpn.key
ping            5
ping-restart    15
```

```
ping-timer-rem
persist-tun
persist-key
tun-mtu 1500
comp-lzo
float
```

1. Make the directory /etc/rsa-keys (`mkdir -p /etc/rsa-keys`)
2. Copy the server keys to the workstation (`scp root@<ROUTER_IP>:/etc/rsa-keys/openvpn.* /etc/rsa-keys`)
3. Make the directory to hold server certificate (`mkdir -p /usr/share/etc/`)
4. Copy the server certificate to the workstation (`scp root@<ROUTER_IP>:/usr/share/etc/imagestream_ca.crt /usr/share/etc`)
5. Open a text editor save the following to /etc/openvpn/Tunnel1.pass

```
rjune
test123
```

1. Only root can read/write the password file. (`chmod 600 /etc/openvpn/Tunnel1.pass`)
2. As root, run `openvpn --config /etc/openvpn/Tunnel1.conf` to start the connection. Alternatively `/etc/init.d/openvpn start` should work.

Configuration Option Reference

These wan.conf configuration options are valid for a Tunnel interface.

bandwidth {*bits_per_second*}

Description

Sets the intended bandwidth in bits per second. This command does not set internal clock speeds. See baud to set internal clocking for interfaces.

When used in conjunction with ATM QoS commands this value is used to calculate the correct ATM QoS settings. When this command is used on a Frame-relay interface this is used to setup rate-limiting on the interface.

Parameter

bits_per_second - Non-negative integer which represents how much data an interface can push.

Examples

bandwidth 2000000 - Sets the interface to 2 Mbps
bandwidth 100000000 - Sets the interface to 100Mbps

Tunnel interfaces set rate-limiting from the bandwidth statement

description *string*

Description

Used to add a comment (description) for tracking what is attached to a particular interface.

Parameter

string - Any alpha-numeric characters are allowed.

Example

description Link to CO - Documents this interface as connecting to CO

ip address *IPv4_Address IPv4_Netmask* [secondary]

Description

To set IP addresses for an interface, use the ip address command.

Parameters

IPv4_Address - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia
IPv4_Netmask - Four integers from zero to 255 separated by periods
secondary - Specifies additional IP addresses (aliases)

Examples

ip address 10.1.1.199 255.0.0.0 - Assigns the interface the IP address 10.1.1.199 with a Class A network mask
ip address 192.168.1.254 255.255.255.0 secondary - Assigns the interface the alias IP address 192.168.1.254 with a Class C network mask

pointtopoint address *IPV4_Address*

Description

Configure the destination address for this interface. For use with PPP connections when the destination router will not negotiate this setting.

Parameter

IPV4_Address - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

Example

pointopointaddress 192.168.2.253 - Sets the far side of a point to point link to 192.168.2.253

tunnel destination *<IPv4_Address> <port>*

Description

Set the IP Address and Port used by the other side of the tunnel.

Parameters

<IPv4_Address> - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia
<port> - The port used by the other side of the tunnel.

Examples

tunnel destination 192.168.42.42 1104 - Connect to 192.168.42.42:1104

tunnel key *<key>*

Description

Set the key used for tunnel encryption.

Parameters

<key> - 32 character hex value used to encrypt the tunnel. Not strictly required for a tunnel, but without it there is no encryption.

Examples

tunnel key 8232f92562b8e187af624312cc643147 - use '8232f92562b8e187af624312cc643147' as the tunnel key

tunnel mode openvpn *<mode>*

Description

Set the tunnel to client, server, or peer to peer mode.

Parameters

<mode> - Choices are 'server', 'client', or blank. Server and client are server and client mode respectively, blank sets peer to peer mode.

Examples

tunnel mode openvpn server - Sets the interface to server mode.
tunnel mode openvpn - Sets the interface to peer to peer mode.

tunnel source *<IPv4_Address> <port>*

Description

Set the IP Address and Port used by this side of the tunnel.

Parameters

<IPv4_Address> - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia
<port> - The port used by the other side of the tunnel.

Examples

tunnel source 192.168.42.42 1104 - Connect from 192.168.42.42:1104

tunnel options *<tunnel_options>*

Description

Specify command line options to OpenVPN, any command line option can be passed this way. You should be familiar with OpenVPN before using this command, as not all options are valid on all types of tunnels.

Parameters

<tunnel_options> - Command-line options to be passed to openvpn. This can be used to circumvent limitations in the wan.conf command structure.

Examples

tunnel options --push "route 192.168.42.0 255.255.255.0" - Tell the other side to send traffic to 192.168.42.0/24 here.

no tunnel compression

Description

Disables LZO compression on the OpenVPN tunnel. By default LZO compression is enabled on OpenVPN tunnels.

Examples

no tunnel compression - Disable LZO compression.

no tunnel compression