Title

SSH Tunnel

Successfully pass the free certification exam at IW Academy and become an Infinet Certified Engineer.

To the certification exam

InfiNet devices allow to create up to 16 independent L2 tunnels over SSH. The tap interface can have an IP address, can be used for routing and be a part of a switch group. In addition, the tap interface can be used as the parent for the vlan, lag and prf interfaces and be used as part of the MINT network.

SSH Tunnel between two InfiNet devices

To configure first device that performs the server function, create the "*tap0*" interface and assign the "*192.168.1.1/24*" IP address to it. Set the "*ssh_tun*" login and "*$ecRet*" password for the created interface. Set the size of the SSH server internal reception window, for maximum performance the "*-window*" parameter on both tunnel sides should be set to 128000 or more. Set the duration of the session activity check to 30 seconds. Enable SSH daemon.

Device 1 (server)

```
ifc tap0 192.168.1.1/24 up
sshd tunnel add ssh_tun $ecRet tap0
sshd -window=128000 -keepalive=30
sshd start
```

Configure the second device same way, create the "*tap0*" interface and assign the "*192.168.1.2/24*" IP address to it. Set the parameters of internal window, check duration and server interface value, if "*tap0*" interface was created on server, "*-remote-if=0*" command should be used. Login and password must match those on the remote side. Optionally, we can specify the encryption algorithm, the key exchange algorithm and other tunnel parameters. A list of supported algorithms is displayed by the command: "*sshtun tap0 -algo-list*", where "*kex*" - SSH key exchange algorithms, "*hostkey*" - authentication, "*cipher*" - data coding, "*hash*" - data verification and "*compress*" - data compression. Enable SSH daemon. In order to enable SSH tunnel, enter the "*sshtun start*" command.

Device 2 (client)

```
ifc tap0  192.168.1.2/24 up
sshtun tap0 -window=128000 -keepalive=30 -remote-if=0
sshtun tap0 ssh_tun:$ecRet@192.168.1.1 start
sshtun tap0 -cipher-algos=aes256-cbc -kex-algos=diffie-hellman-group1-sha1 -hostkey-algos=ssh-rsa  -hash-
algos=hmac-sha1 -comp-algos=none
sshd start
sshtun start
```

- At opposite ends of the tunnel is possible to configure IP addresses from different subnets on tap interfaces (for example 192.168.1.1/24 and 192.168.100.1/24). However, this configuration requires static routes on the both sides configured by the command:

```
route add <net>/<mask> <local_interface_ip_address> -iface
```

When adding a route with the address of the local interface as a gateway and the "*-iface*" option, packets will be sent through this interface (in our case tap0).

- To configure an SSH tunnel using a port other than "*22*".

On the server:

Add the "*sshd -port*" command with port value in range 1...32767.

```
sshd -port 32000
```

On the client:

To the command specifying the server address, login and password, add the port number.

```
sshtun tap0 ssh_tun:$ecRet@10.10.10.1:32000
```

- Between two devices more than one SSH tunnel can be configured. To do this, create additional tap-interfaces.

Device 1 (server)

```
ifc tap0 192.168.1.1/24 up
sshd tunnel add ssh_tun $ecRet tap0
sshd -window=128000 -keepalive=30
sshd start
ifc tap1 192.168.100.1/24 up
sshd tunnel add ssh_tun $ecRet tap1
sshd -window=128000 -keepalive=30
sshd start
```

Device 2 (client)

```
ifc tap0  192.168.1.2/24 up
sshtun tap0 -window=128000 -keepalive=30 -remote-if=0
sshtun tap0 ssh_tun:$ecRet@192.168.1.1 start
sshd start
sshtun start
ifc tap1  192.168.100.2/24 up
sshtun tap1 -window=128000 -keepalive=30 -remote-if=1
sshtun tap1 ssh_tun:$ecRet@192.168.100.1 start
sshtun start
```

SSH Tunnel between the InfiNet device and client running openssh

InfiNet device is configured as described above.

```
ifc tap0 10.10.20.1/24 up
sshd tunnel add TEST QQTEST tap0
sshd start
```

In this example, the client device is a server with Debian OS. Install the uml-utilities package on the client device first.

```
tunctl
ifconfig tap0 up
ifconfig tap0 10.10.20.2/24
ssh -N –o Tunnel=Ethernet –w 0:0 TEST@10.10.20.1
```

After that the system will request a password and a tunnel for data transmission will be established. To make the tunnel run in the background, use the command:

```
ssh -fN –o Tunnel=Ethernet –w 0:0 TEST@10.10.20.1
```

NOTE

In case a password is set on the InfiNet device, without the -N option (do not execute commands) the tunnel, will not be established.