

# VPNs using PPP over SSH

Stephen Warren

NCLUG

2005-03-08

# Outline

Why a VPN?

Attempts at IPSEC

PPP to the rescue

`pppd` & `ssh` features that helped

SSH tunneling advantages/disadvantages

Implementation

# Why a VPN?

I have a Linux file-server at home – photos & docs

Has RAID-1, and backup to another filesystem, and DVDs

Wanted remote backup – new server in Phoenix, AZ

VPN enables clustering of remote systems

Protects LDAP/DNS replication → better than just `rsync/ssh`

`rsync/ssh` over non-VPN connections for data replication (performance)

AZ ISP (Cox Cable) blocks some ports e.g. SMTP

VPN used to re-route connections from remote systems  
through my local system

# Attempts at IPSEC

Attempted IPSEC between two FC3 boxes

Each with their own NAT. One with a dynamic IP

FC3 GUI tools don't support IPSEC setup well

Especially, doesn't enable NAT-T in configuration files

Tools appear only suited to very simple setups

>2 days solid fiddling yielded no useful results

Tried many HOWTOs. Just ended up with corrupted packets upon decapsulation, hence session setup never occurred. Gave up!

# PPP to the rescue

Googling to research IPSEC pointed at alternatives

PPP over SSH was one

Many other alternatives were said to be outdated,  
or a security risk. Don't recall if OpenVPN mentioned!

Already understood `pppd` well from dialup days

Simple example scripts available from the 'net

Up and running in just a couple hours!

# PPP over SSH overview

Client manages connection to VPN server

Client runs wrapper script, which spawns `pppd`

`pppd` spawns `ssh` to connect to server

`ssh` runs VPN wrapper script on server

Server wrapper script runs `pppd`

`pppd` instances send PPP data over the SSH link

# pppd options that helped

`maxfail 0` (client only)

Wrapper scripts just run `pppd` and are done

`pppd` will run forever, and attempt to keep the link up

Will 'hangup' and restart session if a problem is detected

# pppd options that helped

*pty script* (client only)

pppd usually uses character devices for transport

We want to use an SSH session, not a device node

*pty* option tells pppd to allocate a PTY pair, run the given *script* (with STDIN/OUT connected to the PTY slave) and use the PTY master for pppd's I/O

# pppd options that helped

```
lcp-echo-interval 60
```

```
lcp-echo-failure 5
```

pppd typically detects link failure using HANGUP of underlying serial device

This option tells pppd use ping-like packets for link monitoring

Could have ssh monitor the connection in the same fashion, and exit upon link failure

# pppd options that helped

## `ipparam param`

Passes the string *param* to any scripts that PPPD runs

Such as `/etc/ppp/ip-up` and hence `ip-up.local`

I use this for the VPN “name”

unique to each client/server combination

Allows `ip-up.local` script to be common across all hosts,  
and use *param* to determine which routing to setup

# pppd options that helped

`nodetach` (server only)

Tells `pppd` not to daemonize itself (run in the background)

This keeps the SSH session up and running

# pppd options that helped

noauth

Since pppd is running over an SSH session, the client must be already authenticated as a system (or PAM) user

VPN runs as regular user

sudo used to run pppd wrapper scripts as root

So, no need to authenticate again

# ssh options that helped

-i *identity\_file* (client)

Public-key authentication used

to prevent password prompts – easier to script!

-i tells SSH client which ID file to use

a different ID file for each client/server pair

Private key file doesn't have a password

→ Set your file permissions correctly!

# ssh options that helped

`~vpn_arun/.ssh/authorized_keys` (server)

Lists public key that client will present,  
to allow automated password-less login

Can restrict options, such as:

```
no-port-forwarding,no-x11-forwarding,  
no-agent-forwarding
```

Can restrict command that user can run

e.g. allow the user to run only  
`sudo /path/vpn/server_script`

# SSH Tunneling Advantages

ssh and pppd available pretty much anywhere

Less software to download/build/install/configure

Both are reliably implemented, and well understood

No troubles with NAT or dynamic IP

If you can make a TCP connection, it'll just work

Simplicity – easy to configure and run

# SSH Tunneling Disadvantages

SSH is a TCP/IP protocol (reliable)

Tunnels TCP over TCP

Can have strange interactions doing this

Small outages in underlying IP network can cause  
(slightly) longer outages in the VPN connectivity.

Latency can be noticeably higher over VPN than raw

Should use unreliable UDP as base transport fixes these –  
e.g. OpenVPN, IPSEC!

# Implementation

A little large for a slideshow!

Email me if you want the scripts

or just to talk about details

`s-t-nclug-vpn-scripts@wwwdotorg.org`

If this address gets harvested by spammers,  
you'll need to respond to an anti-spam challenge  
the first time you mail this address!

Questions?

The End