

An Introduction to OpenVPN

Brian Kidney
St. John's Linux Users Group
Feb 17, 2005

Outline

- ◆ What are VPNs and why do you need one?
- ◆ What are your VPN options?
- ◆ Why OpenVPN?
- ◆ OpenVPN Configuration Options
- ◆ An example configuration (with demo!)
- ◆ References
- ◆ Questions

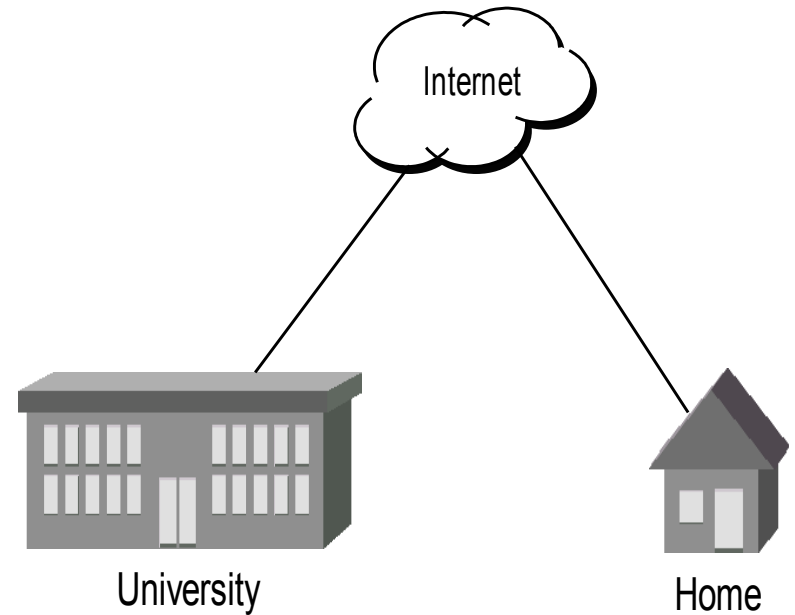
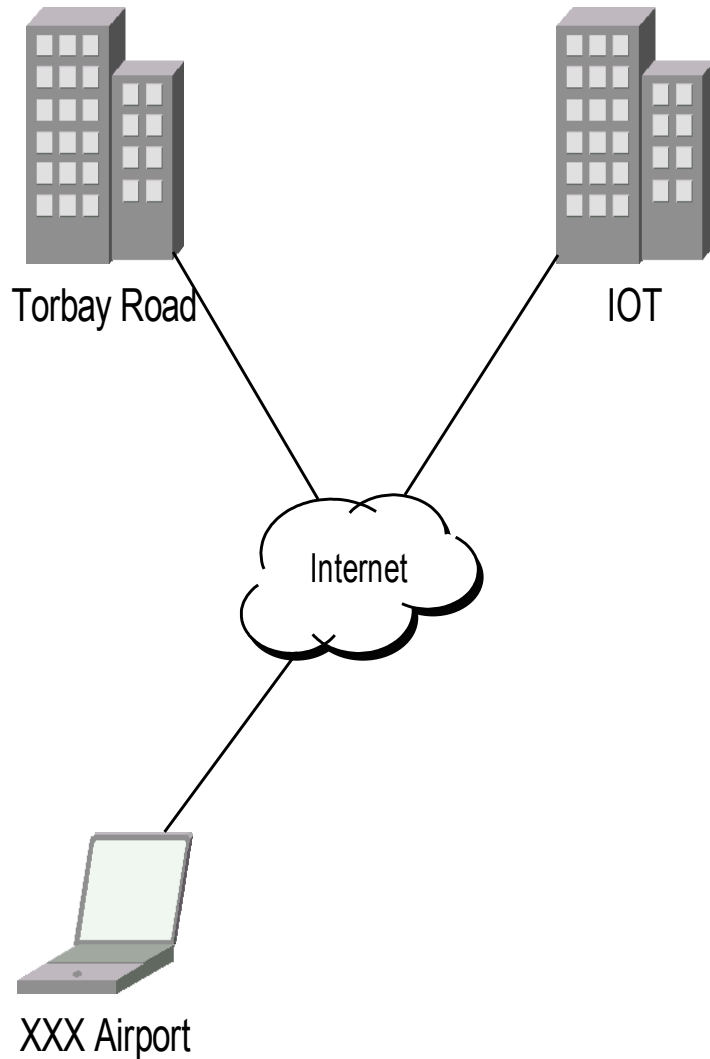
What are VPNs?

- ◆ VPN stands for “Virtual Private Network”
- ◆ Uses semi-permanent “tunnel” to connect two machines
- ◆ Encrypts all data though the tunnel

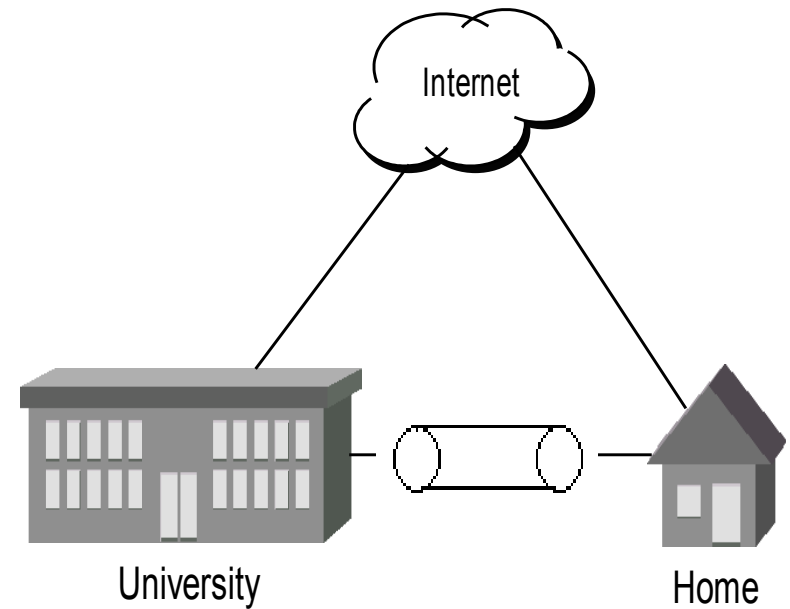
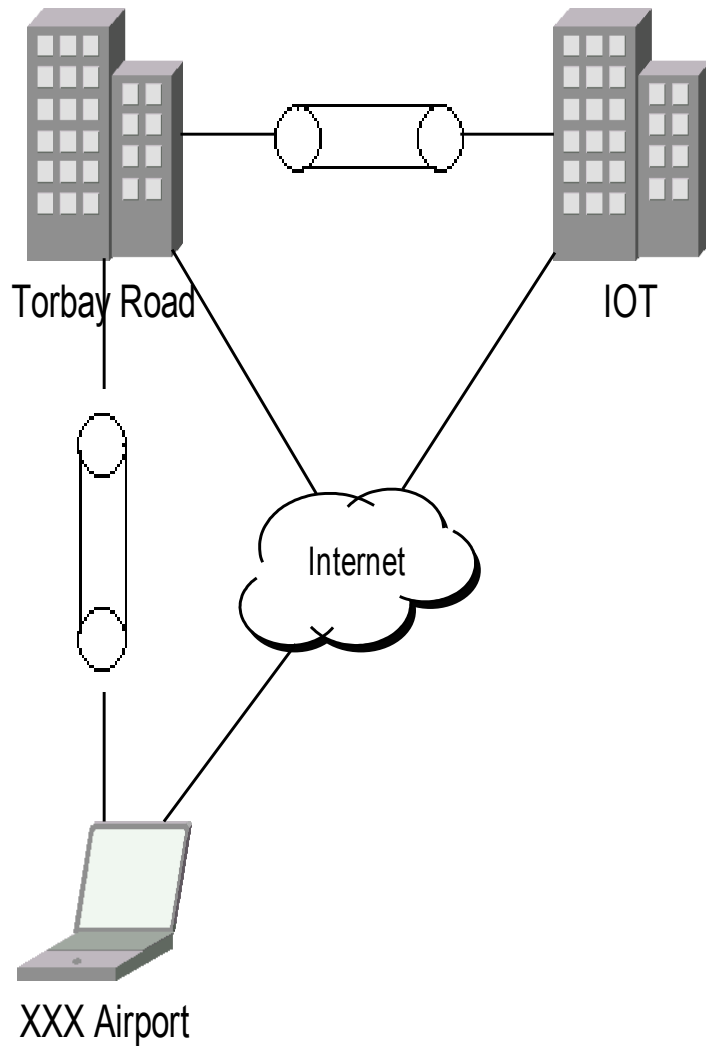
Why do you need a VPN?

- ◆ Provide a secure way to connect:
 - ◆ Two or more locations of a business
 - ◆ Business travelers to the home office
 - ◆ The private networks of friends
 - ◆ Students at school to their home machine
 - ◆ ...

Why do you need a VPN?



Why do you need a VPN?



What are your VPN options?

- ◆ Proprietary hardware (Cisco, Nortel, Linksys...)
- ◆ PPP over SSH
- ◆ CIPE (Crypto IP Encapsulation)
- ◆ IPSEC (L2TP in Windows)
- ◆ SSL/TLS

Why OpenVPN?

- ◆ Open Source Software
- ◆ Built on OpenSSL Library
- ◆ Works on Linux, Windows, Solaris, FreeBSD,...
- ◆ Built in compression (if you want it)
- ◆ No problems with NAT
- ◆ Easy to set up and use
- ◆ Windows GUI (for non-computer types)

OpenVPN Options

- ◆ Two major choices to be made
 - ◆ Symmetric Keys or Public Private Keys
 - ◆ Tunneled connection or Bridged Connection

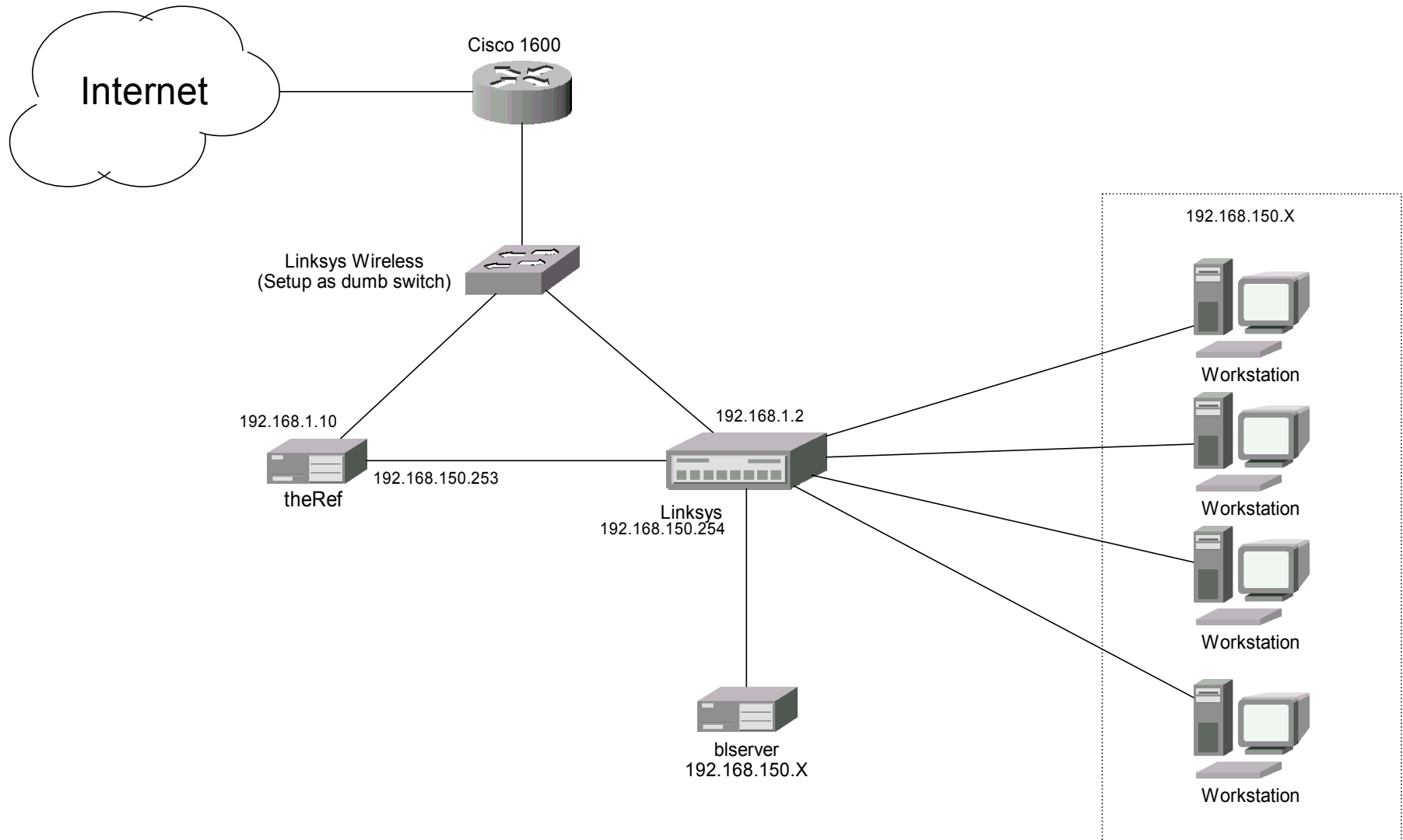
OpenVPN Options

- ◆ Symmetric Keys (Or Shared Secret)
 - ◆ Both parties encrypt/decrypt with same key
 - ◆ Keys must be exchanged in advanced
 - ◆ Simple setup
 - ◆ Harder to maintain
- ◆ Public/Private Keys (or Public Key)
 - ◆ Each party have two keys, a public and private one
 - ◆ Public key contained in a digital signature
 - ◆ Public Key Cryptography used to negotiate symmetric key
 - ◆ More complicated setup
 - ◆ More flexible implementation

OpenVPN Options

- ◆ Tunneled Mode
 - ◆ Point-to-point IP routing
 - ◆ No broadcasts
 - ◆ Efficient
 - ◆ Easy setup (not covered tonight)
- ◆ Bridged Mode
 - ◆ Uses IEEE 802.3 bridging
 - ◆ Layer 2, Protocol Independent (including broadcast)
 - ◆ Harder to setup
 - ◆ Works with iTunes :)

Example Configuration



Example Configuration

- ◆ **Server**
 - ◆ AMD K6-2 400Mhz
 - ◆ Gentoo Linux
 - ◆ OpenSSL 0.9.7d
 - ◆ OpenVPN 2.0_rc6
 - ◆ Version 2 introduced two big changes
 - ◆ One port for multiple clients
 - ◆ Certificate Revocation Lists
- ◆ **Clients**
 - ◆ Many different machines running Win2k, WinXP
 - ◆ OpenVPN GUI v1.0_rc2

Example Configuration

- ◆ Implementation Steps
 - ◆ Recompile Kernel
 - ◆ Requires 802.1d Bridging and TAP/TUN Support
 - ◆ Install OpenSSL
 - ◆ Configure Certificate Authority
 - ◆ See S. Brumbaugh in References
 - ◆ Install OpenVPN
 - ◆ Configure OpenVPN
 - ◆ Create and distribute client certificates
 - ◆ Install client software
 - ◆ Configure client software

Server Config File

```
# openvpn-server.conf
#
# Tunnel mode
dev tap0
proto tcp-server

# Run as a single instance server
mode server

# Specify tls-server for certificate exchange
tls-server

# Diffie-Hellman Parameters (tls-server only)
dh /etc/ssl/CA-DB/dh1024.pem

# Root certificate
ca /etc/ssl/CA-DB/cacert.pem

# Server certificate
cert /etc/ssl/CA-DB/vpn-cert.pem

# Server private key
key /etc/ssl/CA-DB/vpn-key.pem

# Check for revoked client certificates.
crl-verify /etc/ssl/CA-DB/crl/crl.pem

# Specify the log file (stop logging after 20 repeats)
log-append /var/log/openvpn.log
verb 4
mute 20

# Range of IP addresses reserved for clients
ifconfig-pool 192.168.150.50 192.168.150.75
                255.255.255.0

# route setup on the server
#route 192.168.150.0 255.255.255.0

# route command pushed to the client
push "route-gateway 192.168.150.254"

# Set keepalive to keep connections open
# Will timeout if no ping response in 120s
keepalive 10 120

# Drop permissions once server setup
user nobody
group nobody
persist-key
persist-tun
```

Client Config File

```
# openvpn-client.conf

# Set tunnel mode
dev tap
port 1194
proto tcp-client

# Hostname (IP Address) for the VPN server
remote XXX.XXX.XXX.XXX

# This end takes the client role for
# certificate exchange
tls-client

# Certificate Authority file
ca cacert.pem

# Our certificate/public key
cert bkidneycert.pem

# Our private key
key bkidneykey.pem

# Get the rest of our configuration
# from the server.
pull
```

Certificate Gen Script

- ◆ This was created for convenience

```
#!/bin/sh

if [ $# -ne 1 ]
then
    echo "Usage: `basename $0` username"
    exit $E_WRONG_ARGS
fi

echo "Creating You certificate, you will have to enter password for it."
/usr/bin/openssl req -new -keyout /etc/ssl/CA-DB/$1key.pem -out /etc/ssl/CA-DB/$1cert-req.pem

echo "Signing the certificate. You will be prompted for CA password."
/usr/bin/openssl ca -out /etc/ssl/CA-DB/$1cert.pem -in /etc/ssl/CA-DB/$1cert-req.pem
```

Server Initialization Script

- ◆ Runs on boot, sets up bridge for VPN

```
#!/bin/sh
```

```
# Bring down current internal interface  
ifconfig eth0 down
```

```
# Create tap device for openvpn  
/usr/sbin/openvpn --mktun --dev tap0
```

```
# Create bridge and add interfaces  
/sbin/brctl addbr br0  
/sbin/brctl addif br0 tap0  
/sbin/brctl addif br0 eth0
```

```
# Place eth and tap devices in promiscuous mode  
/sbin/ifconfig tap0 0.0.0.0 promisc up  
/sbin/ifconfig eth0 0.0.0.0 promisc up
```

```
# Give new bridge (br0) the internal address once held by eth0  
/sbin/ifconfig br0 192.168.150.253 netmask 255.255.255.0 broadcast 192.168.150.255
```

References

- ◆ OpenVPN: www.openvpn.net
- ◆ Meet OpenVPN, H. Speel, Linux Journal
- ◆ Implementing OpenVPN, F. Andrei, FedoraNEWS.ORG
- ◆ OpenVPN and the SSL VPN Revolution, C. Hosner, Sans Institute
- ◆ VPNs and Public Key Infrastructure, S. Brumbaugh, www.onLamp.org
- ◆ Deploying a VPN with PKI, S. Brumbaugh, www.onLamp.org

Questions