OpenVPN

Tom Eastep
April 29, 2006
Linuxfest NW
http://www.shorewall.net/LinuxFest2006.pdf

Agenda

- About me
- VPNs
  - Why do we need them?
  - VPN Software choices
  - Basics
  - Where can they be used?
- OpenVPN
  - Overview
  - How to install it
  - How to configure it
    - Bridge
    - Tunnel
- Demo
- Q&A

Tom Eastep

- Work for Hewlett-Packard Development Company
  - This presentation is my own and is not sponsored or endorsed by HP
- Creator and Maintainer of Shorewall
  - Open source firewall configuration tool for Linux
- 36+ Years of Software Development and Support
- I have no connection to the OpenVPN project
  - I use it
  - I've added support for it to Shorewall
  - I think that it is really cool
  - I recommend it enthusiastically
  - I am not an expert

VPNs – Why do we need them?

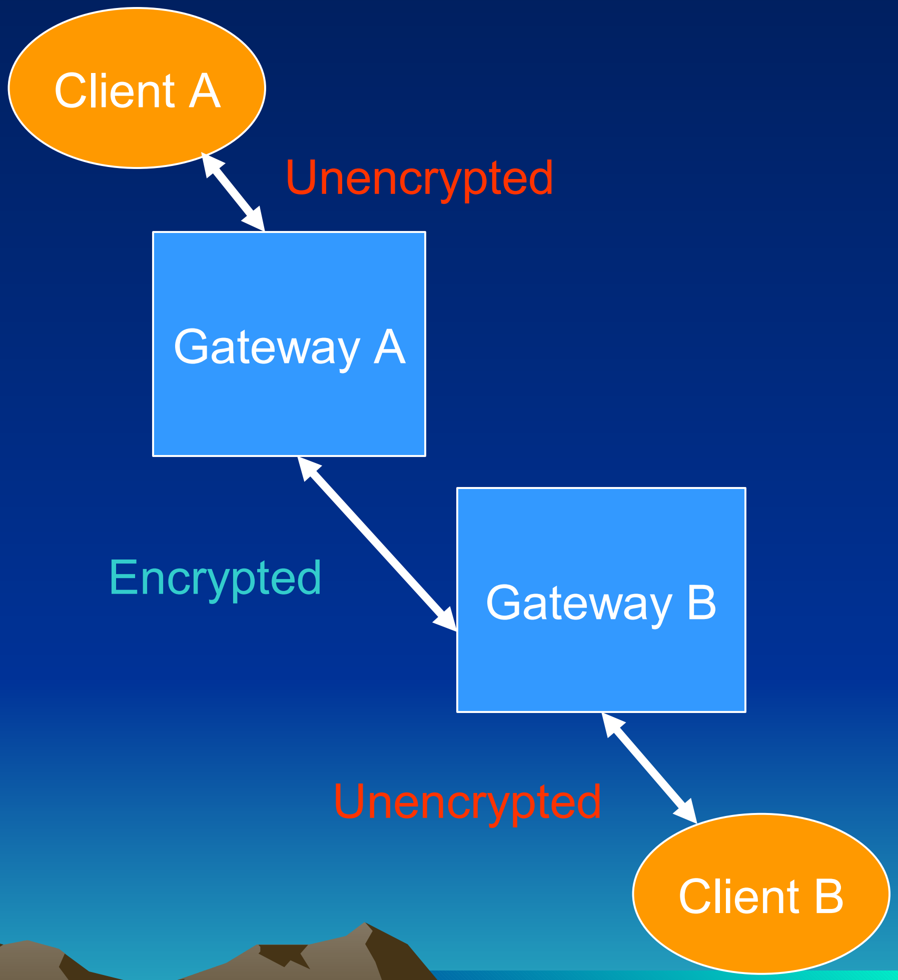
- Secure communication over an insecure network
  - Internet
  - Wireless
- In this environment, we need
  - Authentication
    - Initial authentication (logon)
    - Continuing to insure that packets are not being tampered with in transit
  - Confidentiality
    - Protect against eavesdropping
- Handling “Problem Applications” securely
  - NFS is an example

VPN Software

- Microsoft
  - PPTP (Road-warrior/Telecommuter)
  - IPSEC/L2TP (Road-warrior/Telecommuter)
- Industry Standard
  - IPSEC
    - Developed as part of IPv6
    - “Back-ported” to IPv4
    - A complete IP security framework (not just a VPN solution)
    - Complex to configure (see my LinuxFest NW 2005 presentation at http://www.shorewall.net/LinuxFest2005.pdf)
- Open Source
  - Vtun
  - OpenVPN

VPN Basics

- VPN software runs on *gateways*
- Traffic is sent unencrypted from applications to the nearest gateway (which may be the local system)
- Traffic is encrypted and transmitted to the remote gateway where it is decrypted and forwarded *en clair* to the remote application



VPNs Basics (continued)

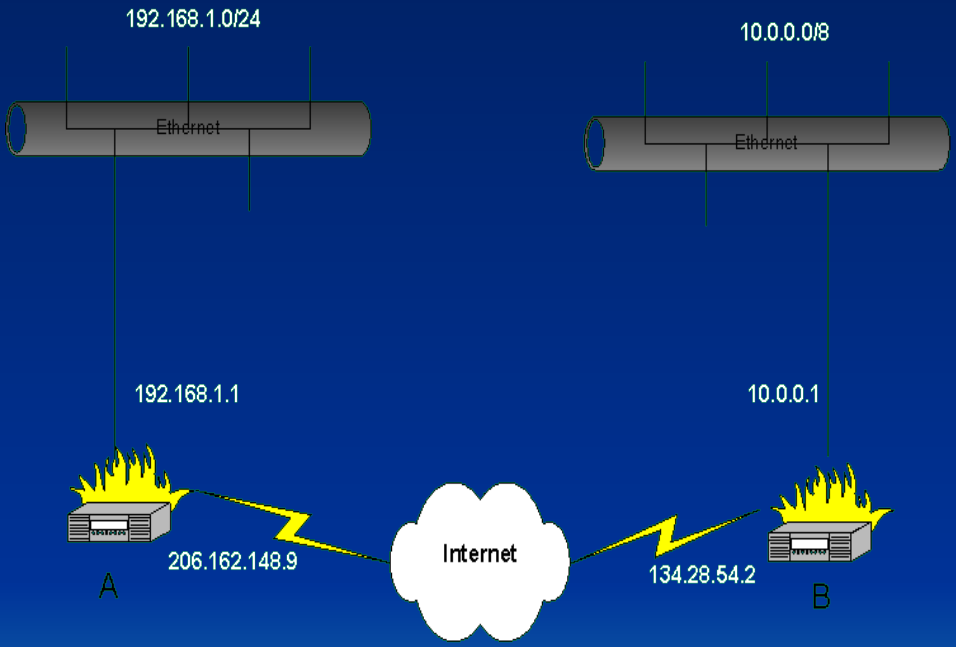
- Under Linux, the VPN software typically creates a *Virtual Network Device* on each gateway
  - PPP creates ppp*n* where n=1,2,…
  - Older IPSEC implementations create ipsec*n*
  - OpenVPN uses either tun*n* (routed) or tap*n* (bridged)
- VPN software performs IP configuration of the device as part of connection establishment
- Routing is used to direct traffic through the VPN
  - Including the default route in some cases

VPNs – Where can they be used?

- Connecting private networks at two or more locations.
- Road-warrior/Telecommuter access to private network
- Wireless Bridge

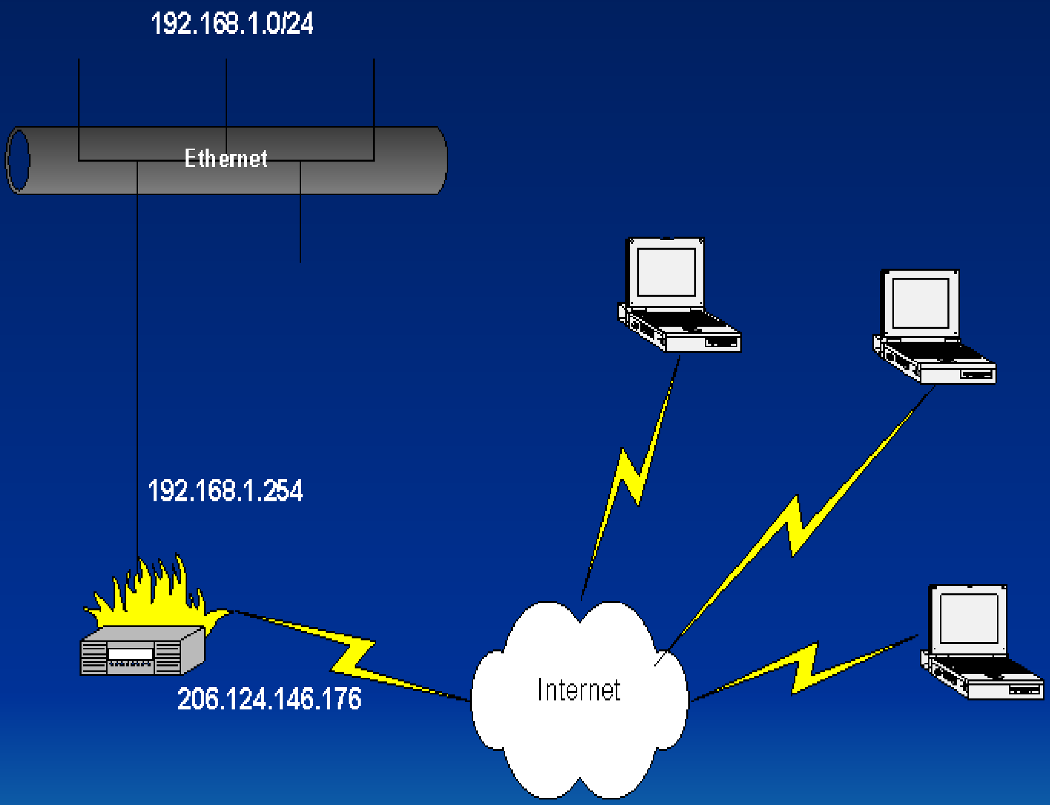
Connecting Private Networks

- Allows secure communication between private networks (even those with RFC 1918 addresses)
- Most straight-forward if OpenVPN runs on your gateway firewall but also works with OpenVPN running on host behind firewall (even if that host has a private address).
- See http://shorewall.net/OPENVPN.htm for configuration details



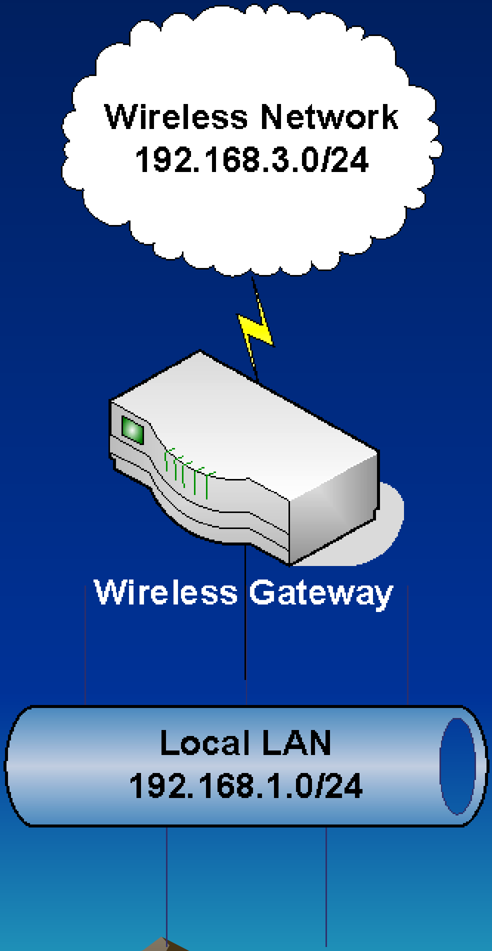
Road-warrior/Telecommuter Access

- Creates a Star Topology
- OpenVPN can be configured to allow client-client connections from within the OpenVPN server.



Wireless Bridge

- Help protect LAN from a Wireless Network
- VPN clients are assigned IP addresses in the local LAN
- Broadcast-based Applications like games and Windows Network Browsing work transparently



OpenVPN – Overview

- Developed and maintained by James Yonan
- Available on a wide range of platforms
  - Linux
  - Windows 2000/XP and higher
  - OpenBSD, FreeBSD and NetBSD
  - Mac OS X
  - Solaris

OpenVPN – Overview (continued)

- Tunnel any IP subnetwork or virtual ethernet adapter over a single UDP or TCP port
  - Default is UDP Port 1194
  - Only use TCP where you cannot use UDP for some reason
- Can use all of the encryption, authentication, and certification features of the OpenSSL library
- Can use any cipher, key size, or HMAC digest (for datagram integrity checking) supported by the OpenSSL Library

OpenVPN – Overview (continued)

- You can choose between static-key based conventional encryption or certificate-based public key encryption
- May use static, pre-shared keys or TLS-based dynamic key exchange
  - I recommend using TLS (Transport Layer Security)
- Includes optional real-time adaptive link compression and traffic shaping to manage link bandwidth utilization

OpenVPN – Overview (continued)

- Can tunnel networks through connection-oriented stateful firewalls (like Netfilter)
- Works over NAT
- Allows creation of secure ethernet bridges using virtual tap devices
- GUIs for configuration and control available on Windows and Mac OS
  - Also some available for Linux but I haven't used them
  - SuSE 10.1 with NetworkManager can configure/control OpenVPN

OpenVPN – Overview (continued)

- Good News – Requires no kernel patching
- Bad News – Because it is implemented in user-space, it generates many user/kernel transitions which limits performance on fast networks.

OpenVPN – Overview (continued)

- OpenVPN 1
  - Point-to-point only – either gateway can initiate the connection
- OpenVPN 2
  - Still supports point-to-point
  - Also supports server mode (both routed and bridged) and client mode (both routed and bridged)
  - Server can handle an arbitrary number of clients
  - Server can be configured to permit client->client connectivity

Routed vs. Bridged

- Routed
  - Gateways act as routers
  - More efficient than bridged (definitely preferred over high-latency networks like the Internet)
  - Generally easier to configure
  - Gateway’s virtual network device is assigned an IP address in a dedicated “VPN” network
  - Routing is used to allow the client to access the network(s) at the remote end.
  - Encapsulated IP packets are sent between the gateways.

Routed vs. Bridged (continued)

- Bridged
  - VPN connection acts as an Ethernet bridge (think of it as a Ethernet switch and a *really* long cable)
  - Harder to set up, especially under Linux (although some distributions such as Debian make it easier than do others)

Routed vs. Bridged (continued)

- Bridged (continued)
  - Preferred when:
    - Need to handle non-IP protocols like IPX,
    - You want to preserve IP addresses when you move laptops from the private LAN to the wireless network or to the Internet
    - You run applications over the VPN which rely on network broadcasts (such as LAN games), or
    - You would like to allow browsing of Windows file shares across the VPN without setting up a Samba or WINS server (weak reason – Samba WINS server is trivial to set up)

Routed vs. Bridged (continued)

- Difference between routed & bridged is primarily on the server side
  - Routed – server routes between the virtual device(s) and other devices on the server
  - Bridged – the virtual device is *bridged* to one of the real network devices on the server. The bridge itself gets the IP configuration

Routed vs. Bridged (continued)

- Bridged (continued)
  - Remote client's virtual network device is assigned an IP address in one of the server's local networks
  - Allows the client transparent access to that local network (including broadcasts, other protocols like IPX, etc).
  - Encapsulated Ethernet frames are sent between the gateways

Installing OpenVPN

- Linux
  - Install your distribution’s OpenVPN package along with any prerequisites.
  - Note: OpenVPN must be installed and run by root
  - Requires OpenSSL

Installing OpenVPN (continued)

- Windows
  - Download the Windows OpenVPN installer from openvpn.net.
  - Run the self-installing .exe on the windows system. The installer also installs the Tap-Win32 driver and creates a virtual network device for use by OpenVPN.
  - If you need additional virtual devices, you can run the tapinstall.exe program included with OpenVPN.
  - Note: OpenVPN must be installed and run by a user that has administrative privileges.

Installing OpenVPN (continued) (Public Key Infrastructure – PKI)

- Disclaimer: I know just enough about Public Key Encryption to make it work.
- OpenVPN includes a toolkit called “easy-rsa” for establishing your own *Certificate Authority* (CA) that can then issue X.509 certificates.
- Very easy-to-follow instructions in the OpenVPN HOWTO (http://openvpn.net/howto.html).

Installing OpenVPN PKI (continued)

- You create a *CA Certificate* and key which can then be used to sign *signing requests* which in turn creates new certificates for your gateways (clients and servers).
  - easy-rsa doesn’t encrypt the CA key by default
- The CA certificate (but not the CA key) needs to be copied to each gateway (on Windows, you do **not** need to install the certificates in the Windows certificate store).
- Create Diffie Hellman parameters using ‘build-dh’ script (required for TLS Servers only).
- Create an empty Certificate Revocation List (CRL)

Installing OpenVPN PKI (continued)

- I recommend creating a separate certificate for each gateway (clients and servers); that way, you can revoke if private key lost or stolen.
- The gateway’s certificate and key must be available on the gateway to start OpenVPN there.
- I don’t recommend assigning a password to the key of the certificate used on your OpenVPN server if you start your server using your distribution’s init scripts.
- I strongly recommend assigning a password on client systems, especially on laptops.
- For added security, you can install the client certificate on a “smart card” or (as I do), keep it on a USB stick.

Configuring OpenVPN

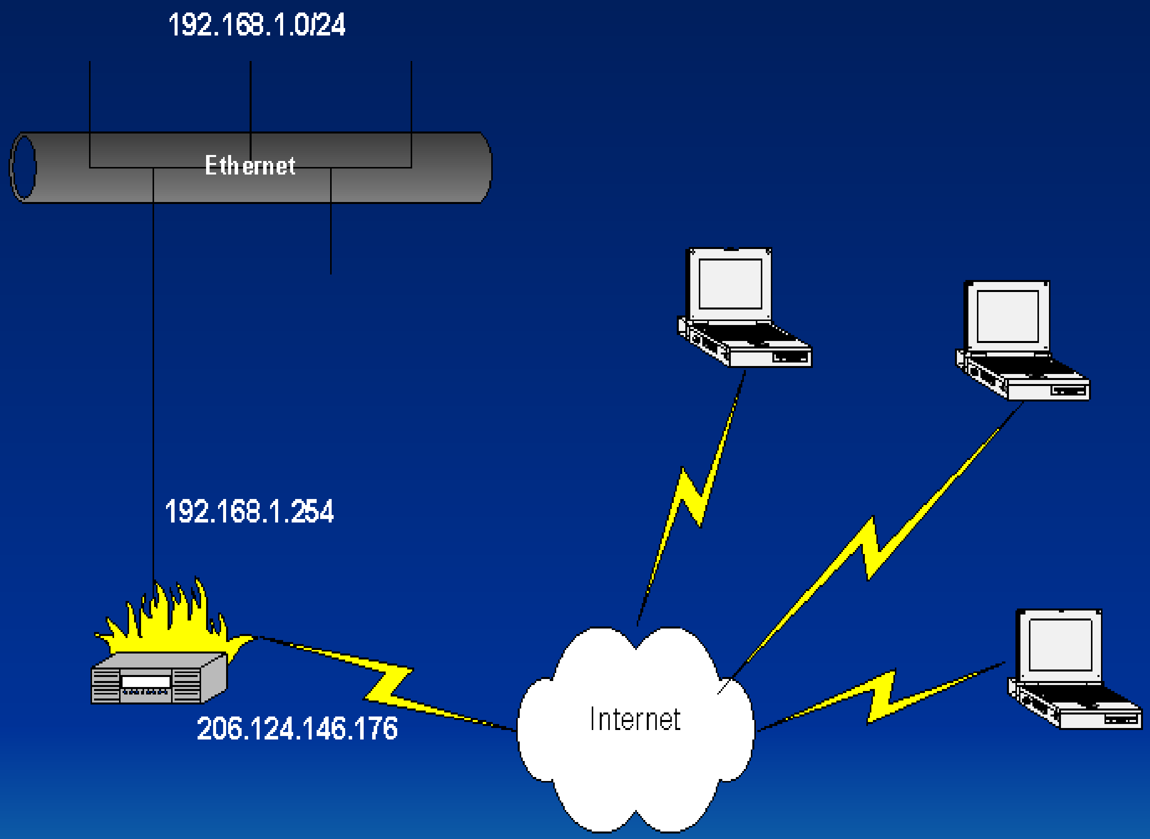
- Each running instance of OpenVPN requires a *configuration file.*
  - Actually, you can specify the configuration on the run-line but that’s pretty cumbersome.
  - “man openvpn” describes the command-line arguments which are prefixed with “--”.
  - In the configuration file, the prefix is omitted.
  - Example:
    - Command line: --push-route
    - Configuration file: push-route

Configuring OpenVPN (continued)

- – On Windows, configuration files have the extension '.ovpn'. I place mine in C:\Program Files\OpenVPN\configs\ (default)
- – On Linux, configuration files have the suffix '.config' and are generally placed in /etc/openvpn/.

Routed Server

- Dual Homed (has two interfaces)
  - Internet
  - Local Network(s)



Example Configuration for a Routed Server

- Server: gateway.shorewall.net
- IP address: 206.124.146.176
- VPN Network: 192.168.2.0/24
  - Because of limitations in the Tap-Win32 driver, each client in a routed configuration needs it’s own /30 network (4 IP addresses).
  - In OpenVPN 2.1, if you don’t have any Windows clients, there is an option to avoid that waste.

Configuration file (Routed Server)

```
dev tun
local 206.124.146.176            #Server’s IP address
server 192.168.2.0 255.255.255.0 #VPN Network
dh dh1024.pem                    #Diffie-Hellman parameters
                                 #Only required on TLS servers
ca /etc/certs/cacert.pem         #CA certificate
crl-verify /etc/certs/crl.pem    #Certificate Revocation List
cert /etc/certs/gateway.pem      #Gateway’s certificate
key /etc/certs/gateway_key.pem   #Gateway’s key
port 1194                        #Default OpenVPN 2.0 Port
comp-lzo                         #Use fast LZO compression
user nobody                      #drop root priv after
group nogroup                    #initialization
```

Routed Server (continued)

```
keepalive 15 45                        #ping every 15 seconds
                                       #restart if no ping
                                       #received in 45 seconds
ping-timer-rem                         #Don’t start ping clock
                                       #until we have a client
persist-tun                            #Don’t close/open tun
                                       #device during
                                       #ping-restart
persist-key                            #don’t re-read key after
                                       #ping restart
client-config-dir /etc/openvpn/clients #Directory where client-
                                       #specific params are kept
ccd-exclusive                          #Require client-specific
                                       #params
client-to-client                       #allow client->client
verb 3                                 #verbosity of the log
```

Sample Configuration for a Routed Client (Windows Roadwarrior)

```
dev tun                              #Routed
remote gateway.shorewall.net         #Server’s Name
tls-remote gateway.shorewall.net     #Common Name in Server’s Certificate
tls-client                           #We are a TLS client
explicit-exit-notify                 #Notify when we exit
pull                                 #Accept server’s pushed parameters
ca "/Program Files/OpenVPN/certs/cacert.pem"
cert “E:/easy-rsa/keys/eastepnc6000.crt"
key “E:/easy-rsa/keys/eastepnc6000.key"
port 1194
comp-lzo
ping-timer-rem
persist-tun
persist-key
mute-replay-warnings
verb 3
```

- Only difference in a Linux config is the file names!

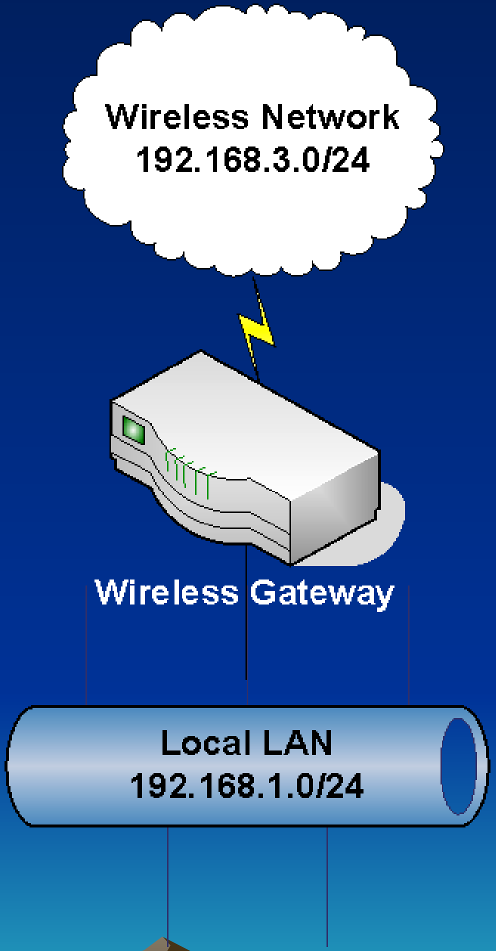
RoadWarrior’s CCD File

- On the server in /etc/openvpn/clients/
- Name is the same as the CN in the client’s certificate

```
#CCD for eastepnc6000.shorewall.net
#Local (server) IP and client IP
ifconfig-push 192.168.2.14 192.168.2.13
#Route to local network
push “route 192.168.1.0 255.255.255.0”
#Route to DNS server
push “route 206.124.146.177.255.255.255.255”
```

Wireless Bridge

- Wireless Bridge is multi-homed:
  - Wireless
  - Local LAN



Sample Configuration of a Bridged Server (Wireless Gateway)

- Server’s Wireless IP address: 192.168.3.254
- Wireless Network: 192.168.3.0/24
- Local Network: 192.168.1.0/24
- Local IP address: 192.168.1.7
- Default Gateway: 192.168.1.254
- Local Interface: eth0
- Server Name: wireless.shorewall.net

Configuration file (Bridged Server)

```
dev tap0                          #Indicates Bridge with pre-
                                  #created device
local 192.168.3.254               #Server address
                                  #Local network plus a pool of
                                  #addresses to assign
server-bridge 192.168.1.7 255.255.255.0 192.168.1.64 192.168.1.71
client-to-client                  #Server handles client->client
                                  #traffic
dh dh1024.pem                     #Diffie Hellman Parameters
ca /etc/certs/cacert.pem          #CA Certificate
crl-verify /etc/certs/crl.pem     #Certificate Revocation List
cert /etc/certs/wireless.pem      #Gateway’s Certificate
key /etc/certs/wireless_key.pem   #Gateway’s Key
port 1194                         #Default port #
comp-lzo                          #Use LZO fast compression
user nobody                       #drop root priv after
group nogroup                     #initialization
```

Bridged Server (continued)

```
keepalive 15 45                         #ping every 15 seconds
                                        #restart if no ping
                                        #received in 45 seconds
ping-timer-rem                          #Don’t start ping clock
                                        #until we have a client
persist-tun                             #Don’t close/open tun
                                        #device during
                                        #ping-restart
persist-key                             #don’t re-read key after
                                        #ping restart
client-config-dir /etc/openvpn/bridge-clients
                                        #Directory where client-
                                        #specific params are kept
ccd-exclusive                           #Require client-specific
                                        #params
verb 3                                  #verbosity of the log
```

Bridged Server (continued)

```
#
# The client supports a “redirect-gateway” option that redirects
# the default gateway through the VPN. I’ve found that to be
# somewhat unreliable whereas this trick works always
#
push “route 0.0.0.0 128.0.0.0 192.168.1.254”
push “route 128.0.0.0 128.0.0.0 192.168.1.254”
```

Bridged Server – Creating the Bridge

- See http://www.shorewall.net/Bridge.html for distribution-specific instructions

```
/usr/sbin/openvpn --mktun --dev tap0 #create dev
/sbin/brctl addbr br0                #create bridge
/sbin/ip link set tap0 up            #Up dev
/sbin/ip link set eth0 up            #Up local IF
/sbin/brctl addif br0 tap0           #Add devs to
/sbin/brctl addif br0 eth0           #to the bridge
```

- br0 is configured using Distribution’s tools

Sample Configuration for a Bridged Client (Windows)

```
dev tap
remote 192.168.3.254
tls-remote wireless.shorewall.net
tls-client
explicit-exit-notify
pull
ca "/Program Files/OpenVPN/certs/cacert.pem"
cert “E:/easy-rsa/keys/eastepnc6000.crt"
key “E:/easy-rsa/keys/eastepnc6000.key"
port 1194
comp-lzo
ping-timer-rem
persist-tun
persist-key
mute-replay-warnings
verb 3
```

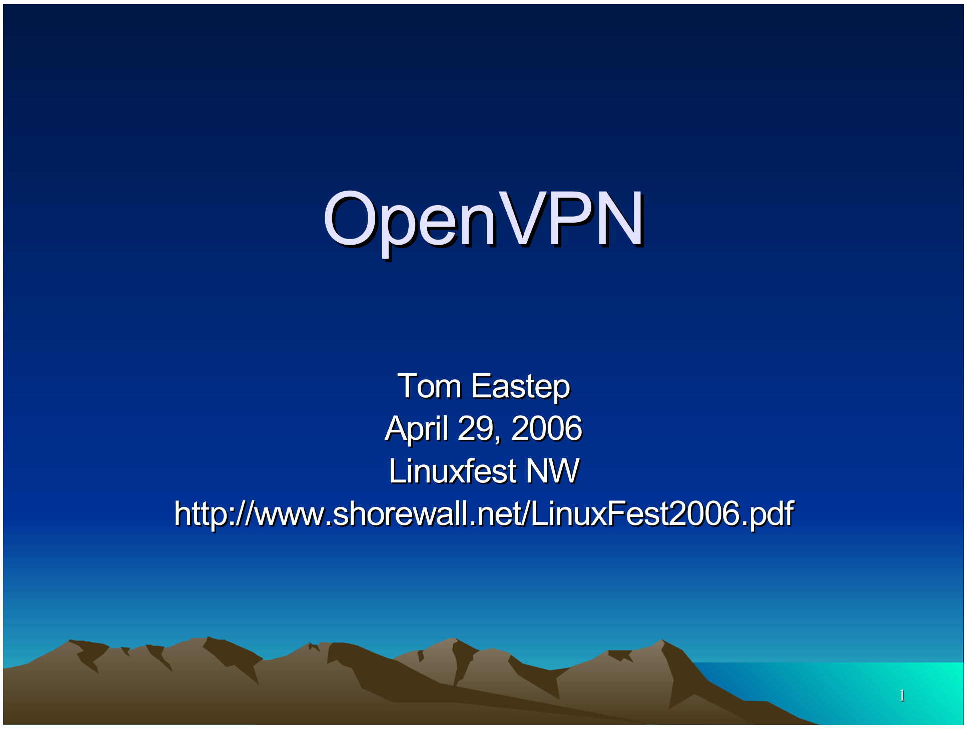
Bridged Client’s CCD File

- On the server in /etc/openvpn/bridged-clients/

```
#CCD for eastepnc6000.shorewall.net
#Client IP
ifconfig-push 192.168.1.6 255.255.255.0
```

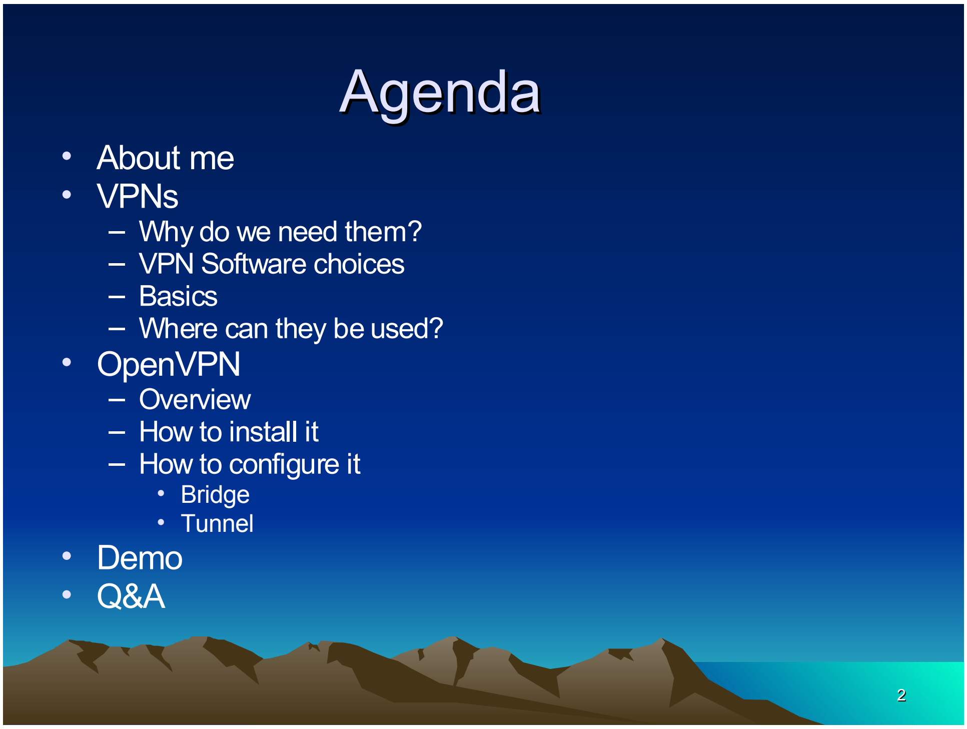
Demo

Q&A



OpenVPN

Tom Eastep
April 29, 2006
Linuxfest NW
http://www.shorewall.net/LinuxFest2006.pdf



Agenda

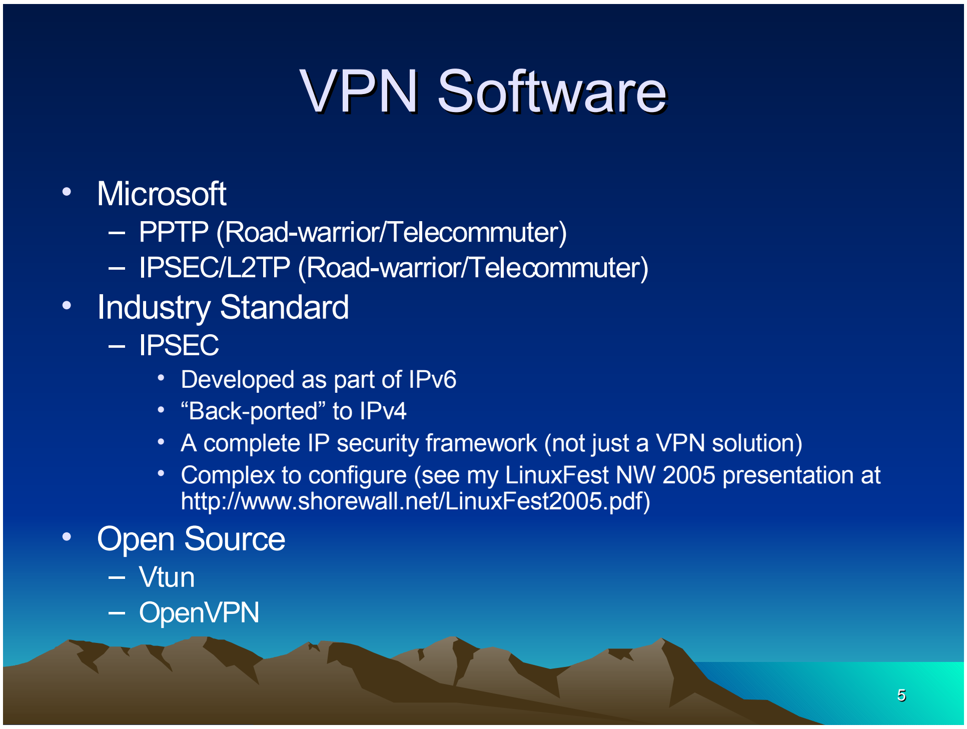
- About me
- VPNs
  - Why do we need them?
  - VPN Software choices
  - Basics
  - Where can they be used?
- OpenVPN
  - Overview
  - How to install it
  - How to configure it
    - Bridge
    - Tunnel
- Demo
- Q&A

Tom Eastep

- Work for Hewlett-Packard Development Company
  - This presentation is my own and is not sponsored or endorsed by HP
- Creator and Maintainer of Shorewall
  - Open source firewall configuration tool for Linux
- 36+ Years of Software Development and Support
- I have no connection to the OpenVPN project
  - I use it
  - I've added support for it to Shorewall
  - I think that it is really cool
  - I recommend it enthusiastically
  - I am not an expert

VPNs – Why do we need them?

- Secure communication over an insecure network
  - Internet
  - Wireless
- In this environment, we need
  - Authentication
    - Initial authentication (logon)
    - Continuing to insure that packets are not being tampered with in transit
  - Confidentiality
    - Protect against eavesdropping
- Handling “Problem Applications” securely
  - NFS is an example

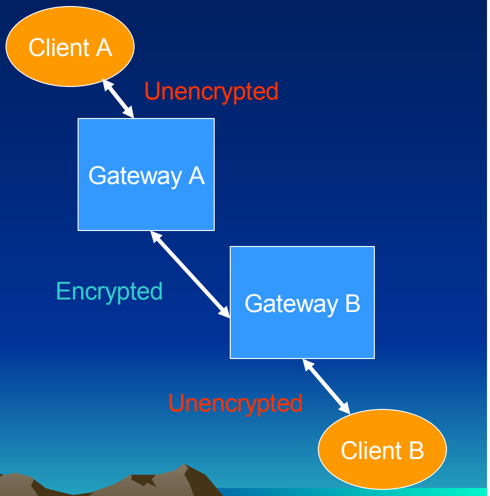


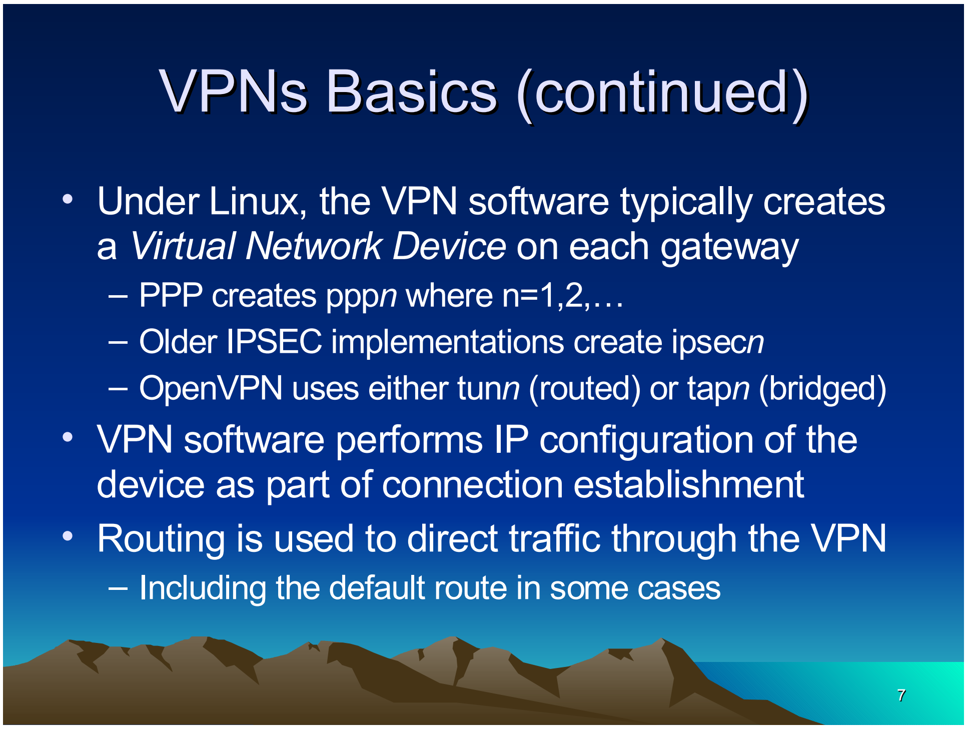
VPN Software

- Microsoft
  - PPTP (Road-warrior/Telecommuter)
  - IPSEC/L2TP (Road-warrior/Telecommuter)
- Industry Standard
  - IPSEC
    - Developed as part of IPv6
    - “Back-ported” to IPv4
    - A complete IP security framework (not just a VPN solution)
    - Complex to configure (see my LinuxFest NW 2005 presentation at http://www.shorewall.net/LinuxFest2005.pdf)
- Open Source
  - Vtun
  - OpenVPN

VPN Basics

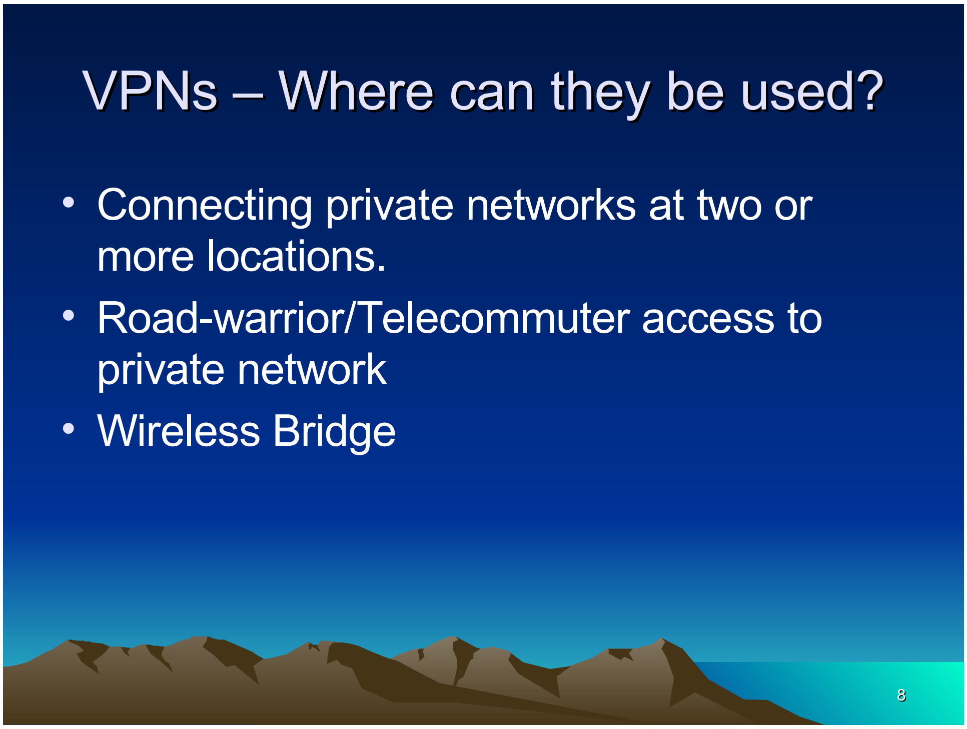
- VPN software runs on *gateways*
- Traffic is sent unencrypted from applications to the nearest gateway (which may be the local system)
- Traffic is encrypted and transmitted to the remote gateway where it is decrypted and forwarded *en clair* to the remote application





VPNs Basics (continued)

- Under Linux, the VPN software typically creates a *Virtual Network Device* on each gateway
  - PPP creates ppp*n* where n=1,2,…
  - Older IPSEC implementations create ipsec*n*
  - OpenVPN uses either tun*n* (routed) or tap*n* (bridged)
- VPN software performs IP configuration of the device as part of connection establishment
- Routing is used to direct traffic through the VPN
  - Including the default route in some cases

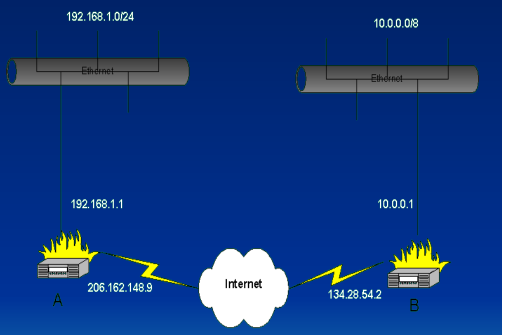


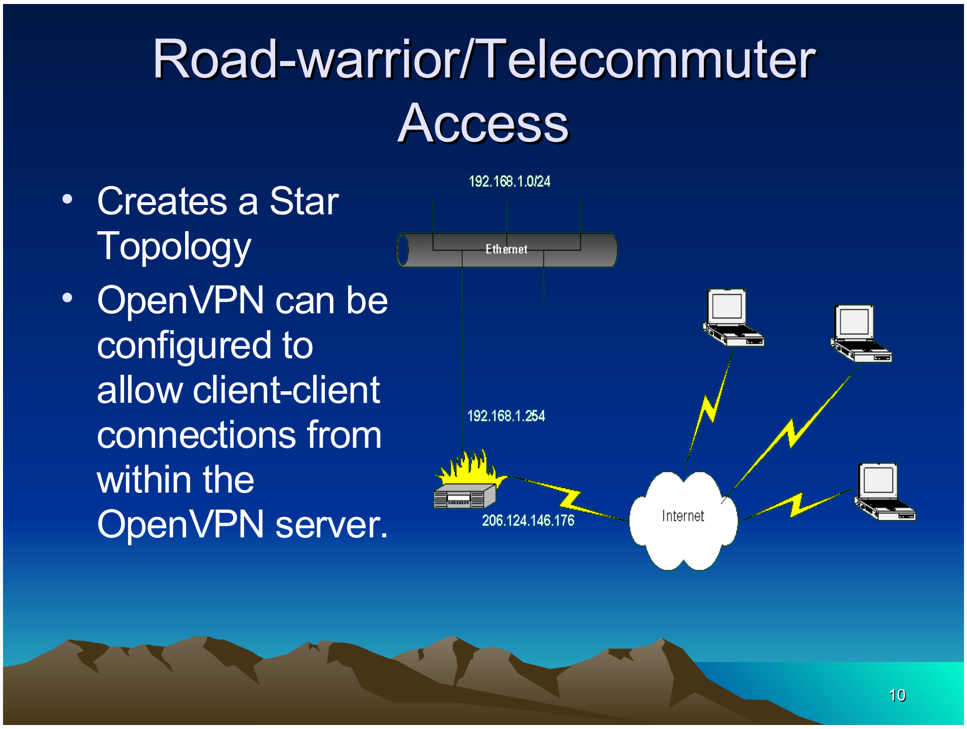
VPNs – Where can they be used?

- Connecting private networks at two or more locations.
- Road-warrior/Telecommuter access to private network
- Wireless Bridge

Connecting Private Networks

- Allows secure communication between private networks (even those with RFC 1918 addresses)
- Most straight-forward if OpenVPN runs on your gateway firewall but also works with OpenVPN running on host behind firewall (even if that host has a private address).
- See http://shorewall.net/OPENVPN.htm for configuration details



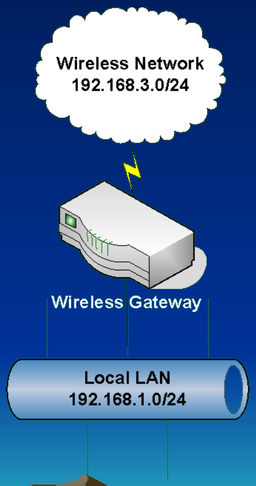


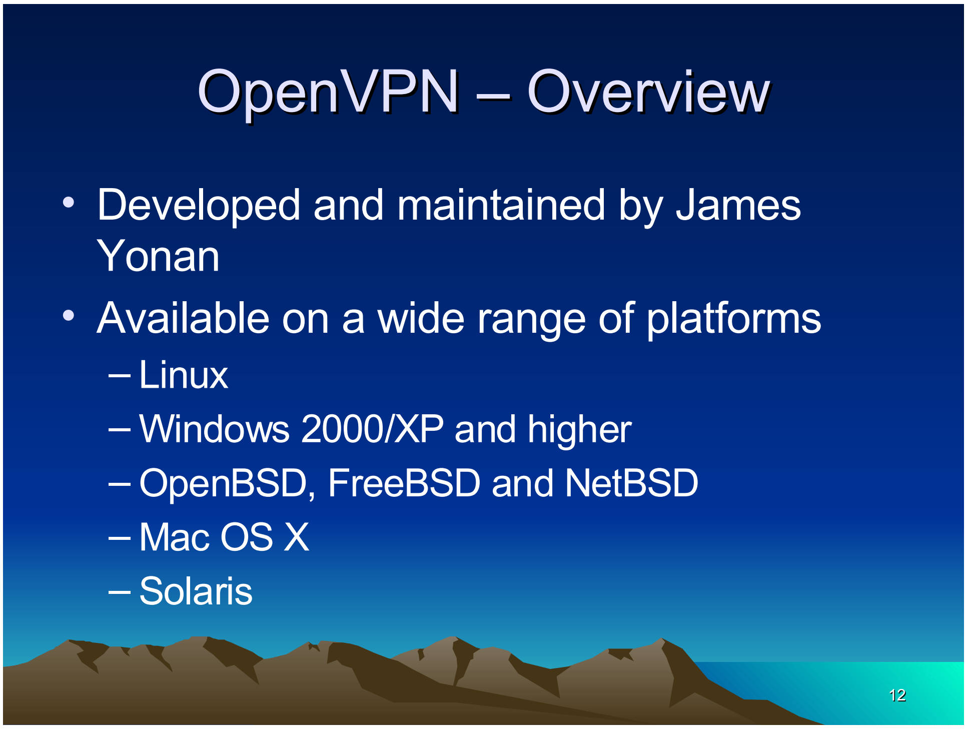
Road-warrior/Telecommuter Access

- Creates a Star Topology
- OpenVPN can be configured to allow client-client connections from within the OpenVPN server.

Wireless Bridge

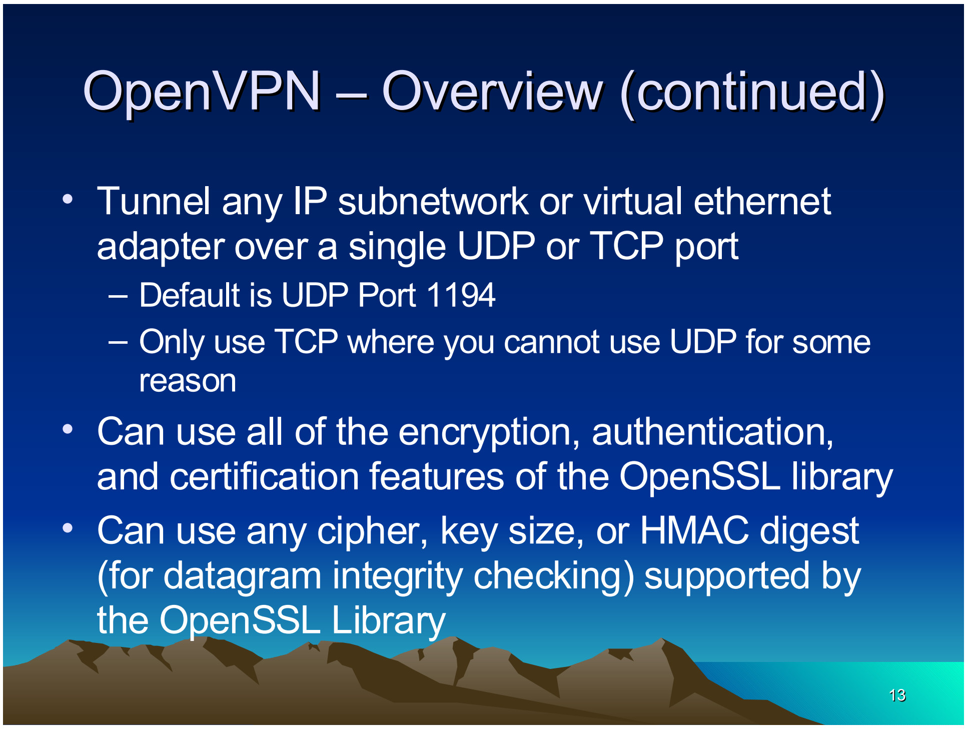
- Help protect LAN from a Wireless Network
- VPN clients are assigned IP addresses in the local LAN
- Broadcast-based Applications like games and Windows Network Browsing work transparently





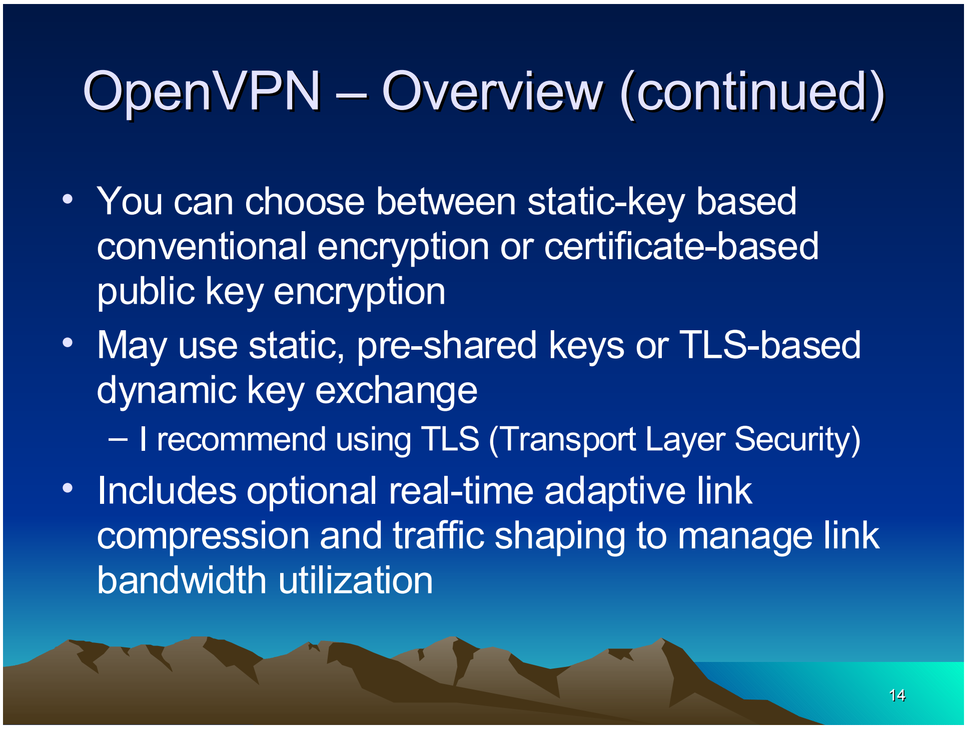
OpenVPN – Overview

- Developed and maintained by James Yonan
- Available on a wide range of platforms
  - Linux
  - Windows 2000/XP and higher
  - OpenBSD, FreeBSD and NetBSD
  - Mac OS X
  - Solaris



OpenVPN – Overview (continued)

- Tunnel any IP subnetwork or virtual ethernet adapter over a single UDP or TCP port
  - Default is UDP Port 1194
  - Only use TCP where you cannot use UDP for some reason
- Can use all of the encryption, authentication, and certification features of the OpenSSL library
- Can use any cipher, key size, or HMAC digest (for datagram integrity checking) supported by the OpenSSL Library

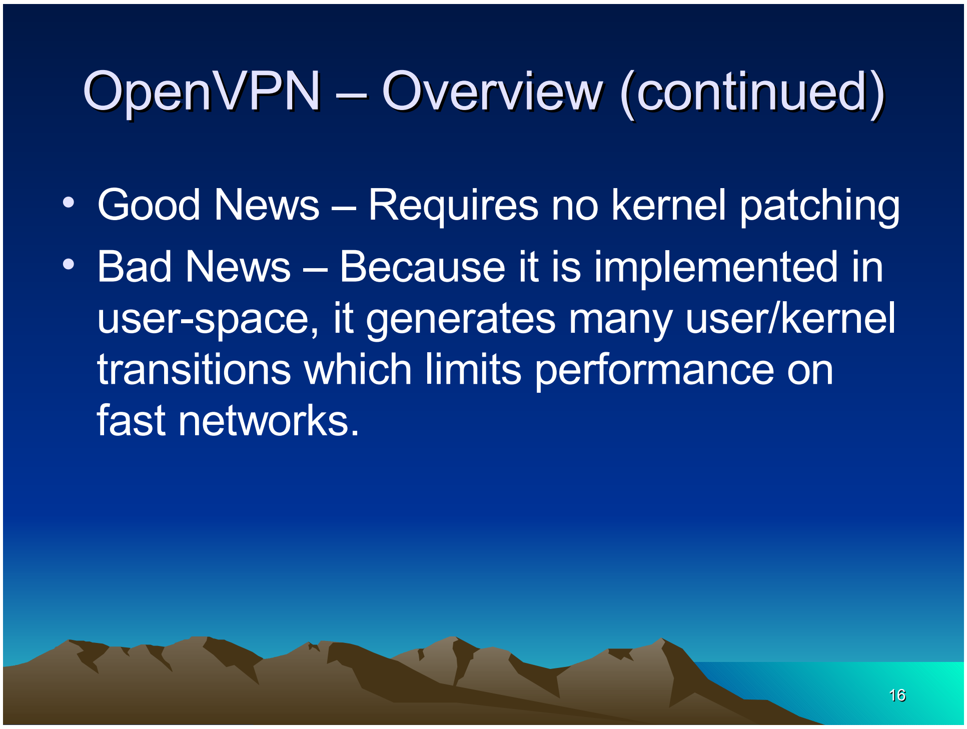


OpenVPN – Overview (continued)

- You can choose between static-key based conventional encryption or certificate-based public key encryption
- May use static, pre-shared keys or TLS-based dynamic key exchange
  - I recommend using TLS (Transport Layer Security)
- Includes optional real-time adaptive link compression and traffic shaping to manage link bandwidth utilization

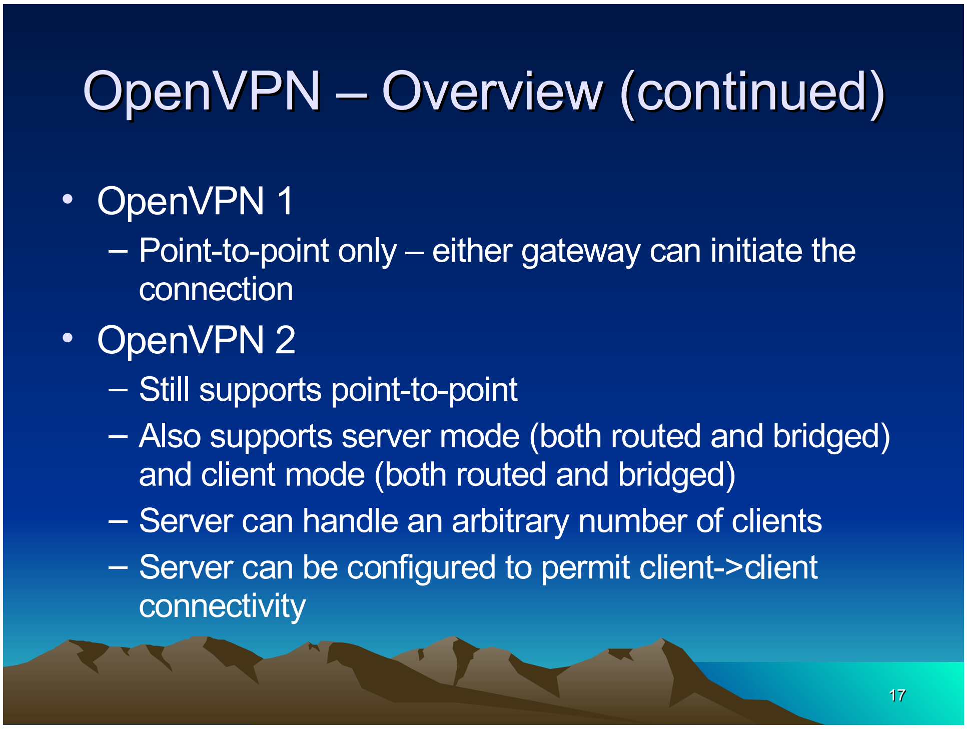
OpenVPN – Overview (continued)

- Can tunnel networks through connection-oriented stateful firewalls (like Netfilter)
- Works over NAT
- Allows creation of secure ethernet bridges using virtual tap devices
- GUIs for configuration and control available on Windows and Mac OS
  - Also some available for Linux but I haven’t used them
  - SuSE 10.1 with NetworkManager can configure/control OpenVPN



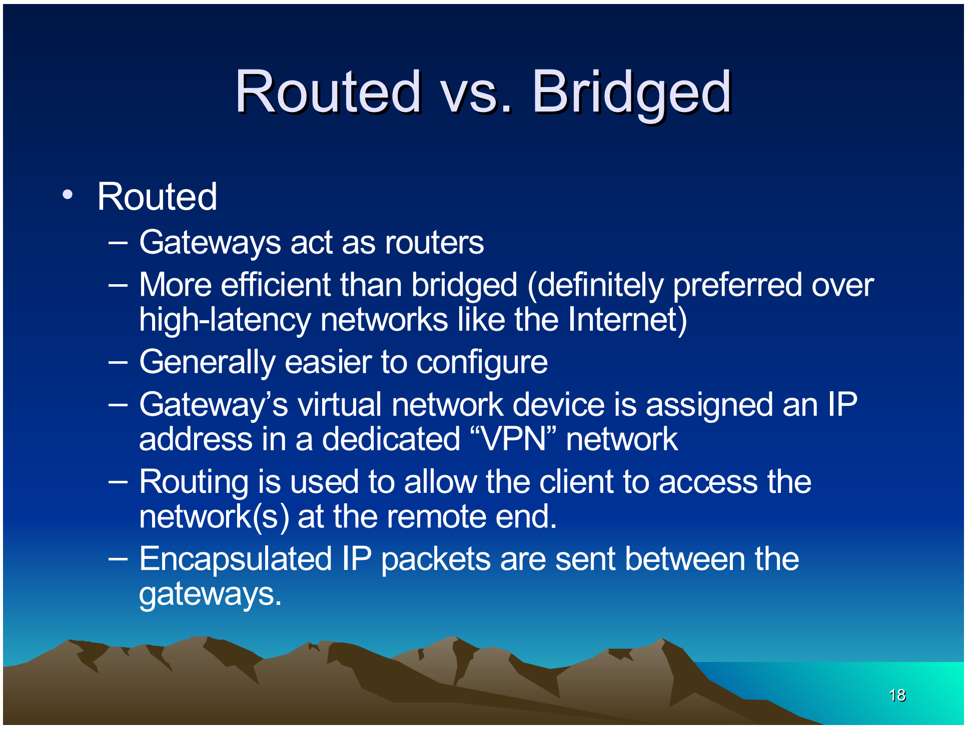
OpenVPN – Overview (continued)

- Good News – Requires no kernel patching
- Bad News – Because it is implemented in user-space, it generates many user/kernel transitions which limits performance on fast networks.



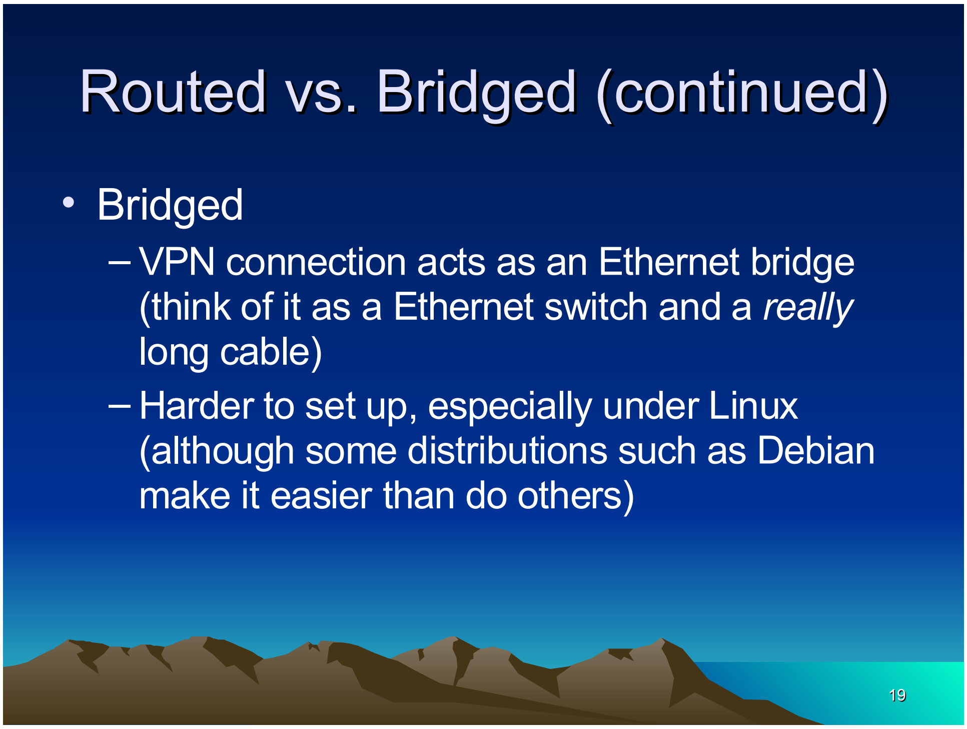
OpenVPN – Overview (continued)

- OpenVPN 1
  - Point-to-point only – either gateway can initiate the connection
- OpenVPN 2
  - Still supports point-to-point
  - Also supports server mode (both routed and bridged) and client mode (both routed and bridged)
  - Server can handle an arbitrary number of clients
  - Server can be configured to permit client->client connectivity



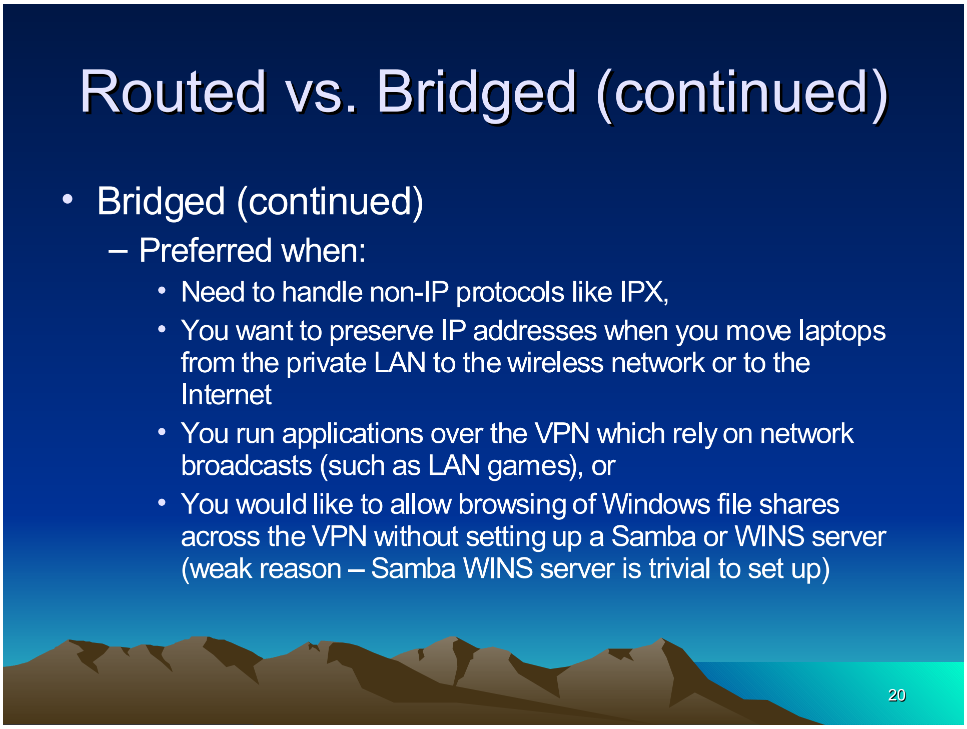
Routed vs. Bridged

- Routed
  - Gateways act as routers
  - More efficient than bridged (definitely preferred over high-latency networks like the Internet)
  - Generally easier to configure
  - Gateway’s virtual network device is assigned an IP address in a dedicated “VPN” network
  - Routing is used to allow the client to access the network(s) at the remote end.
  - Encapsulated IP packets are sent between the gateways.



Routed vs. Bridged (continued)

- Bridged
  - VPN connection acts as an Ethernet bridge (think of it as a Ethernet switch and a *really* long cable)
  - Harder to set up, especially under Linux (although some distributions such as Debian make it easier than do others)

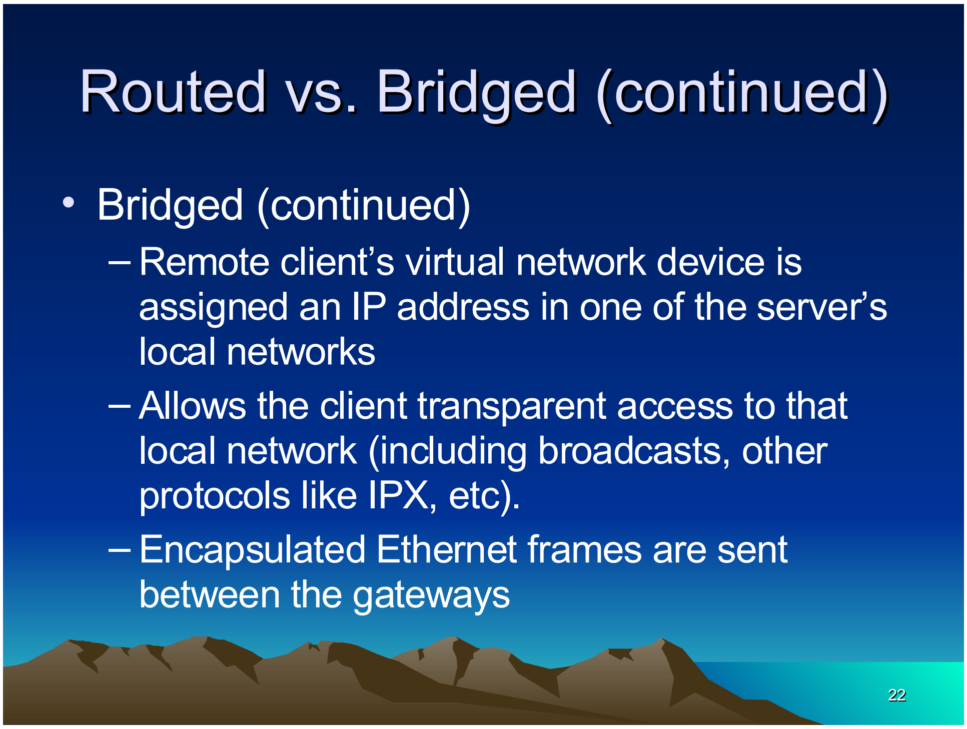


Routed vs. Bridged (continued)

- Bridged (continued)
  - Preferred when:
    - Need to handle non-IP protocols like IPX,
    - You want to preserve IP addresses when you move laptops from the private LAN to the wireless network or to the Internet
    - You run applications over the VPN which rely on network broadcasts (such as LAN games), or
    - You would like to allow browsing of Windows file shares across the VPN without setting up a Samba or WINS server (weak reason – Samba WINS server is trivial to set up)

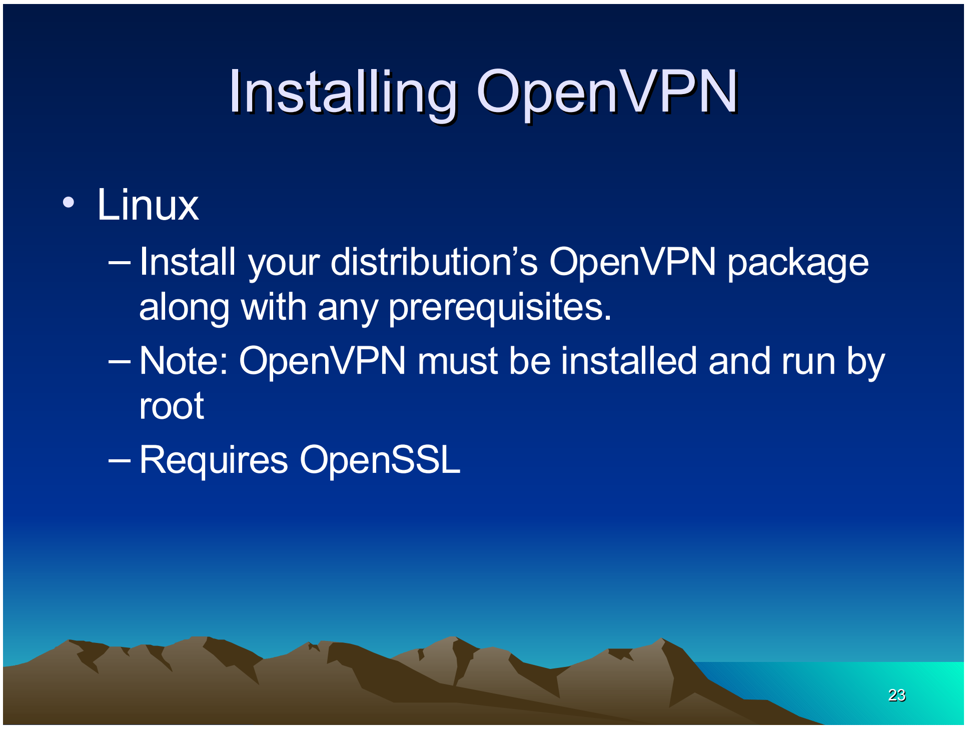
Routed vs. Bridged (continued)

- Difference between routed & bridged is primarily on the server side
  - Routed – server routes between the virtual device(s) and other devices on the server
  - Bridged – the virtual device is *bridged* to one of the real network devices on the server. The bridge itself gets the IP configuration



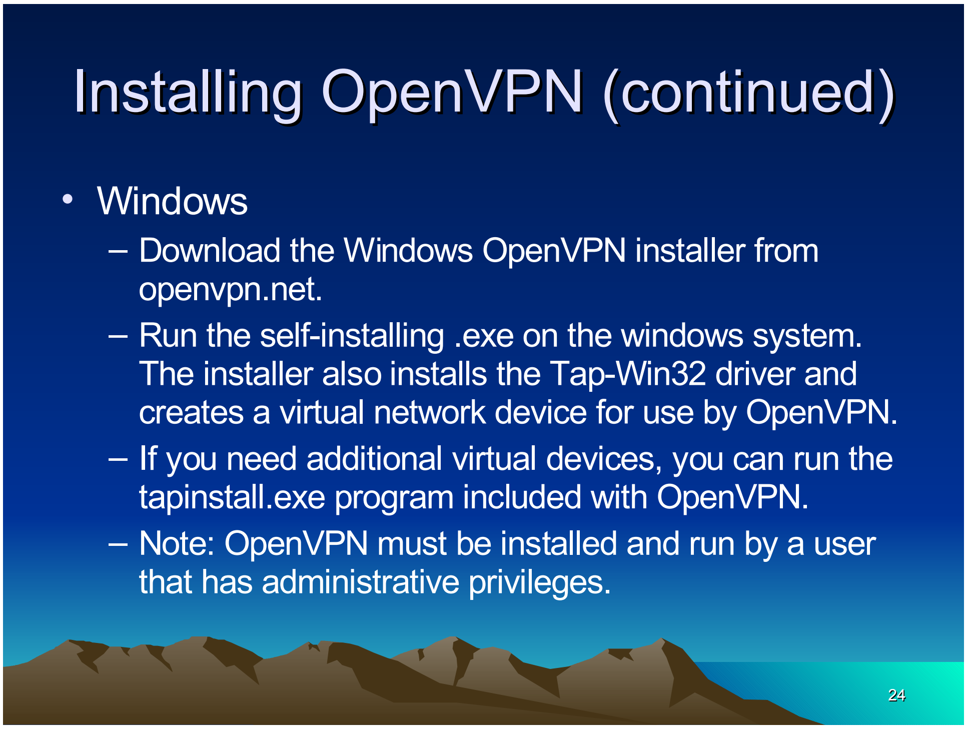
Routed vs. Bridged (continued)

- Bridged (continued)
  - Remote client's virtual network device is assigned an IP address in one of the server's local networks
  - Allows the client transparent access to that local network (including broadcasts, other protocols like IPX, etc).
  - Encapsulated Ethernet frames are sent between the gateways



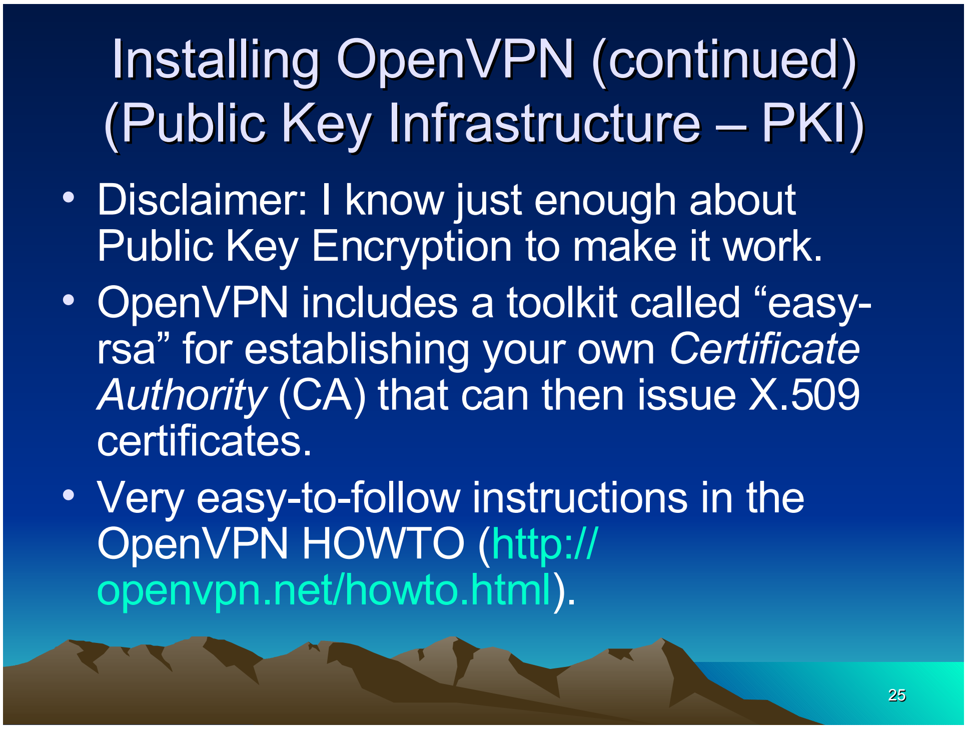
Installing OpenVPN

- Linux
  - Install your distribution’s OpenVPN package along with any prerequisites.
  - Note: OpenVPN must be installed and run by root
  - Requires OpenSSL



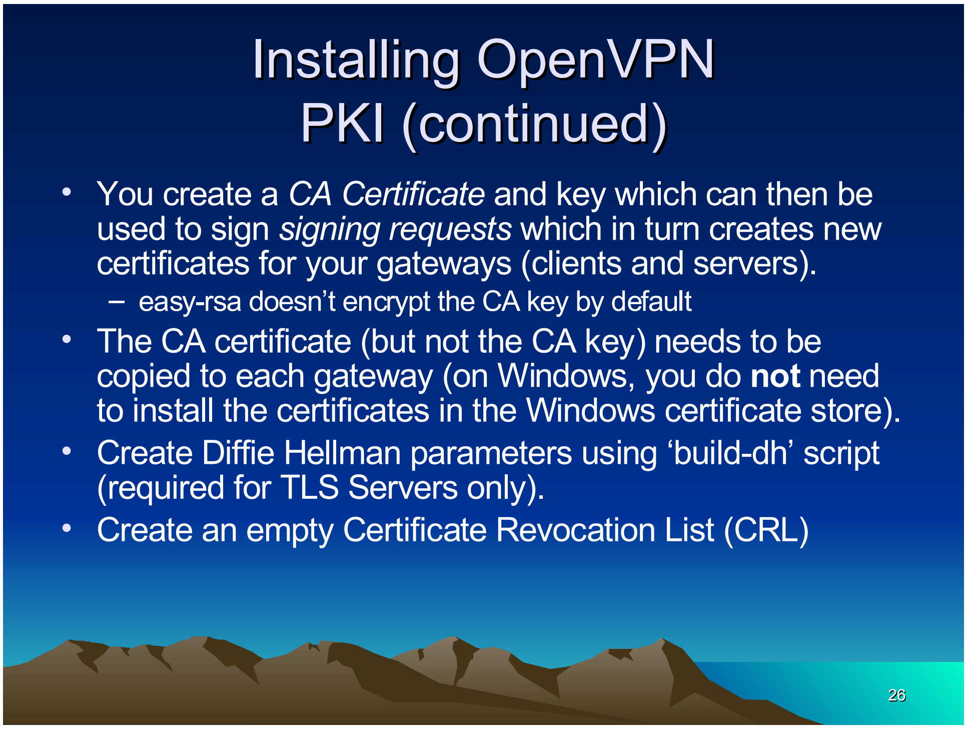
Installing OpenVPN (continued)

- Windows
  - Download the Windows OpenVPN installer from openvpn.net.
  - Run the self-installing .exe on the windows system. The installer also installs the Tap-Win32 driver and creates a virtual network device for use by OpenVPN.
  - If you need additional virtual devices, you can run the tapinstall.exe program included with OpenVPN.
  - Note: OpenVPN must be installed and run by a user that has administrative privileges.



Installing OpenVPN (continued) (Public Key Infrastructure – PKI)

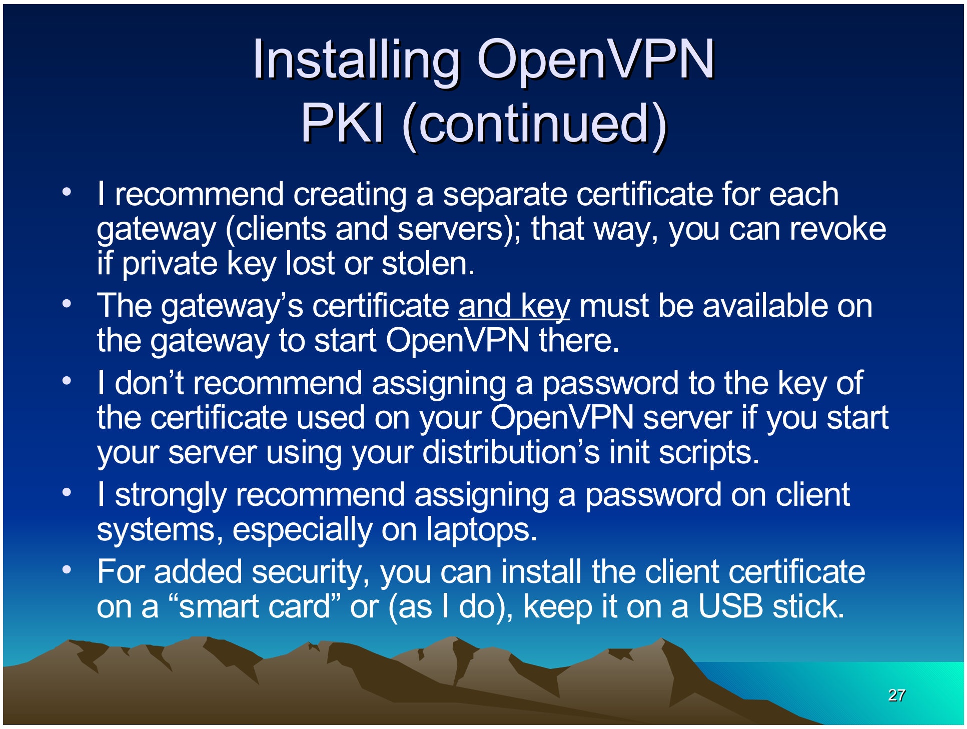
- Disclaimer: I know just enough about Public Key Encryption to make it work.
- OpenVPN includes a toolkit called “easy-rsa” for establishing your own *Certificate Authority* (CA) that can then issue X.509 certificates.
- Very easy-to-follow instructions in the OpenVPN HOWTO (http://openvpn.net/howto.html).



Installing OpenVPN PKI (continued)

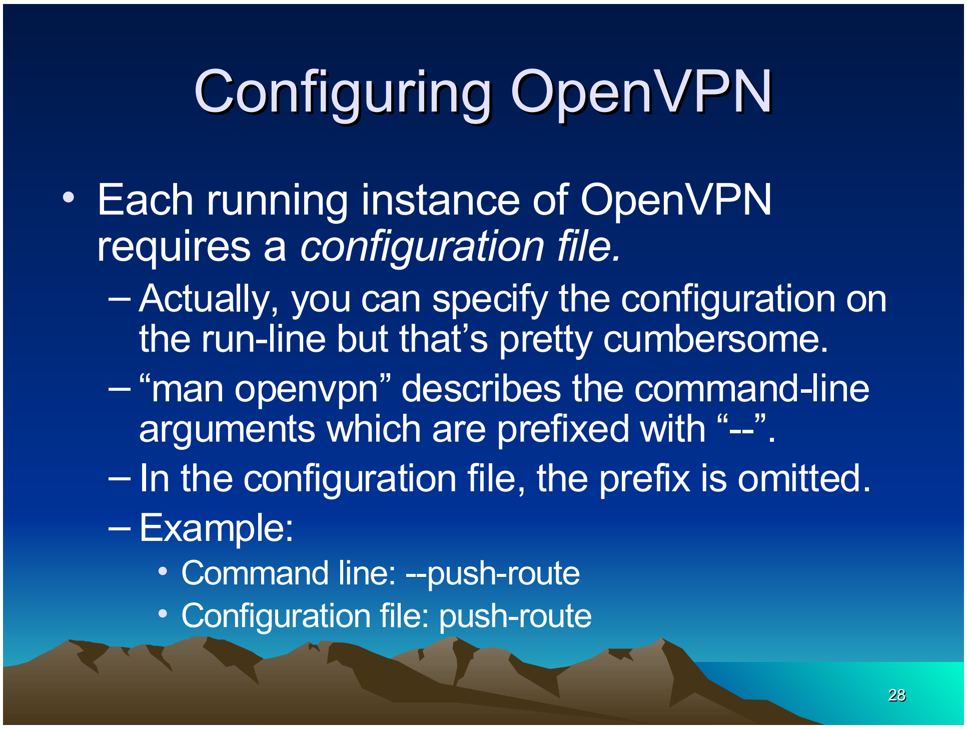
- You create a *CA Certificate* and key which can then be used to sign *signing requests* which in turn creates new certificates for your gateways (clients and servers).
  - easy-rsa doesn't encrypt the CA key by default
- The CA certificate (but not the CA key) needs to be copied to each gateway (on Windows, you do **not** need to install the certificates in the Windows certificate store).
- Create Diffie Hellman parameters using 'build-dh' script (required for TLS Servers only).
- Create an empty Certificate Revocation List (CRL)

Certificates are typically in PEM format although PKCS#12 is appropriate for Smart Cards. PEM is basically <header> <base64 encoded DER> <trailer> where DER == Distinguished Encoding Rules (from ANS.1). PKCS == Public Key Cryptography Standards (RSA Labs).



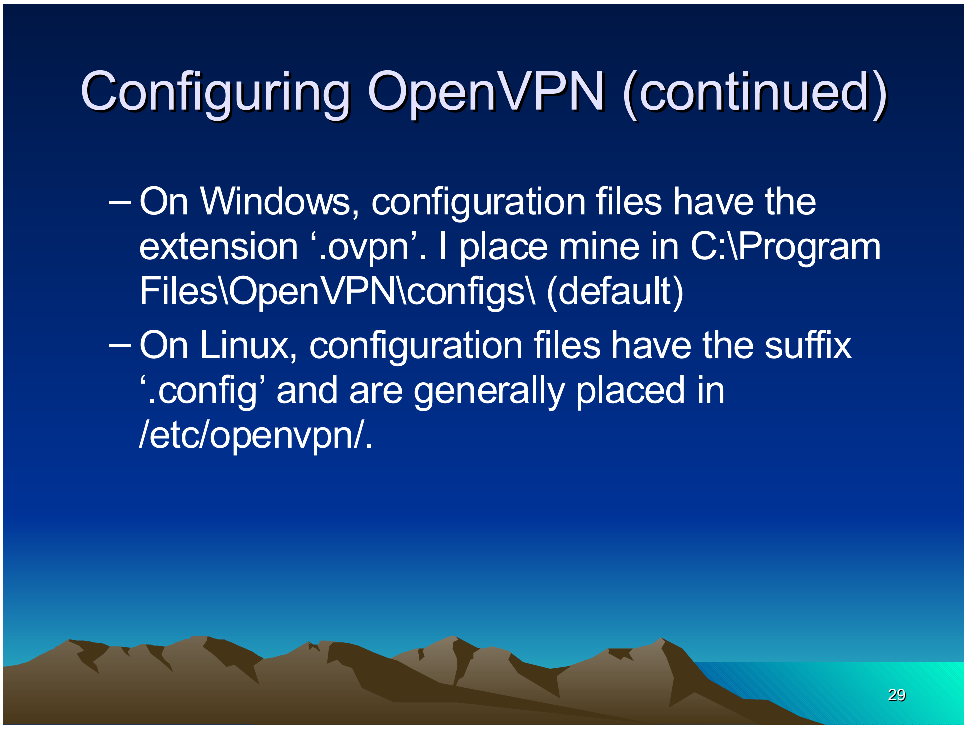
Installing OpenVPN PKI (continued)

- I recommend creating a separate certificate for each gateway (clients and servers); that way, you can revoke if private key lost or stolen.
- The gateway’s certificate and key must be available on the gateway to start OpenVPN there.
- I don’t recommend assigning a password to the key of the certificate used on your OpenVPN server if you start your server using your distribution’s init scripts.
- I strongly recommend assigning a password on client systems, especially on laptops.
- For added security, you can install the client certificate on a “smart card” or (as I do), keep it on a USB stick.



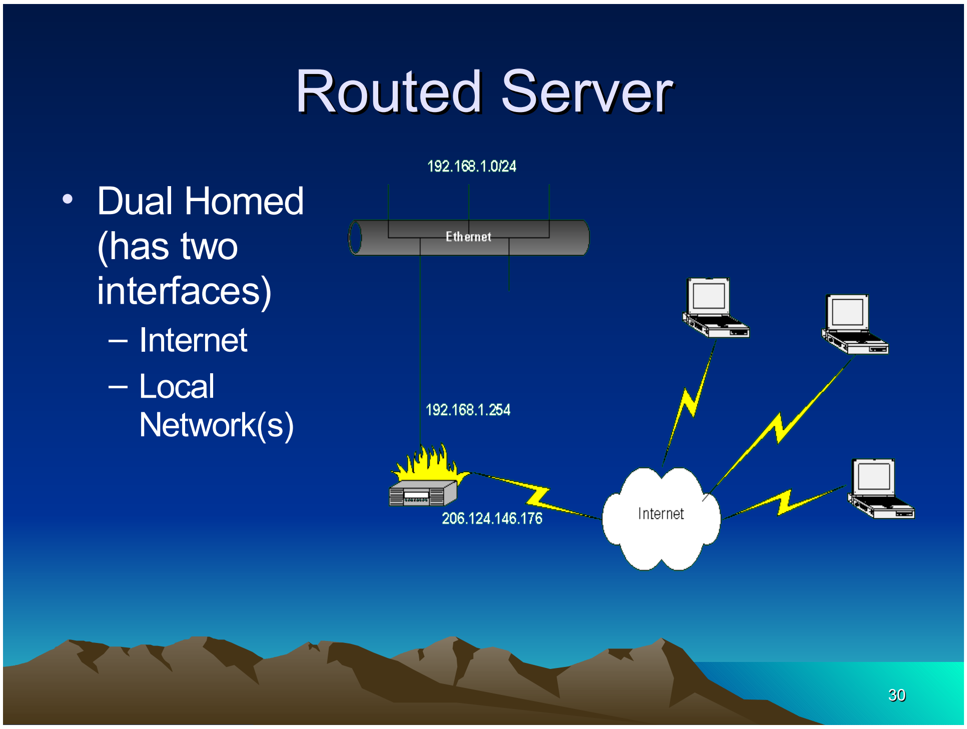
Configuring OpenVPN

- Each running instance of OpenVPN requires a *configuration file.*
  - Actually, you can specify the configuration on the run-line but that's pretty cumbersome.
  - "man openvpn" describes the command-line arguments which are prefixed with "--".
  - In the configuration file, the prefix is omitted.
  - Example:
    - Command line: --push-route
    - Configuration file: push-route



Configuring OpenVPN (continued)

- On Windows, configuration files have the extension ‘.ovpn’. I place mine in C:\Program Files\OpenVPN\configs\ (default)
- On Linux, configuration files have the suffix ‘.config’ and are generally placed in /etc/openvpn/.



Routed Server

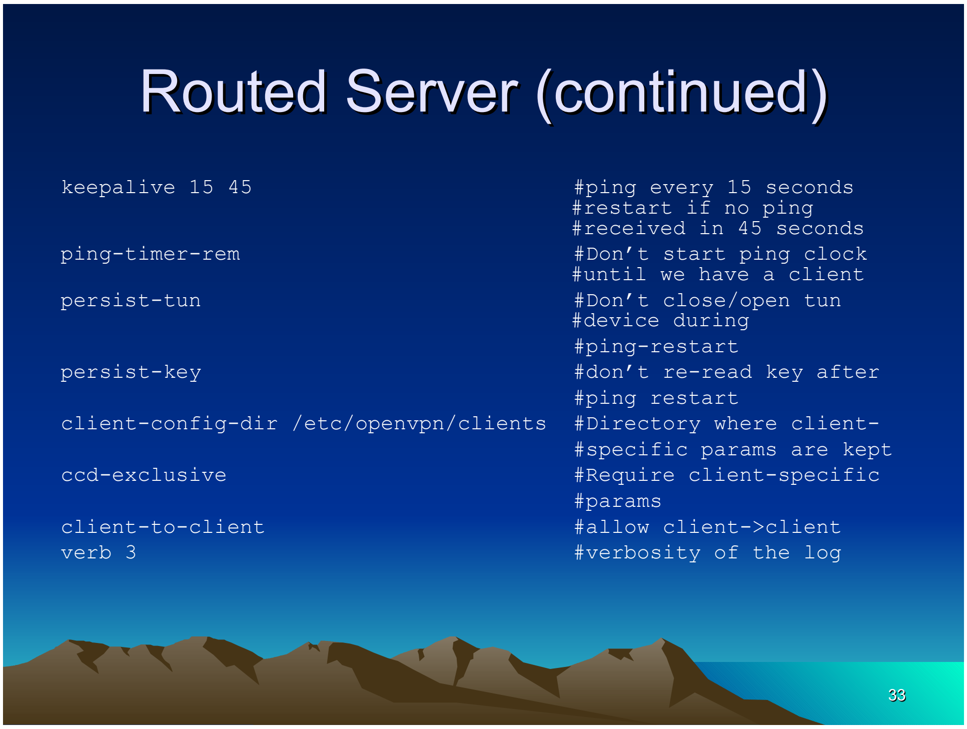
- Dual Homed (has two interfaces)
  - Internet
  - Local Network(s)

Example Configuration for a Routed Server

- Server: gateway.shorewall.net
- IP address: 206.124.146.176
- VPN Network: 192.168.2.0/24
  - Because of limitations in the Tap-Win32 driver, each client in a routed configuration needs it’s own /30 network (4 IP addresses).
  - In OpenVPN 2.1, if you don’t have any Windows clients, there is an option to avoid that waste.

Configuration file (Routed Server)

```
dev tun
local 206.124.146.176             #Server’s IP address
server 192.168.2.0 255.255.255.0 #VPN Network
dh dh1024.pem                     #Diffie-Hellman parameters
                                  #Only required on TLS servers
ca /etc/certs/cacert.pem          #CA certificate
crl-verify /etc/certs/crl.pem     #Certificate Revocation List
cert /etc/certs/gateway.pem       #Gateway’s certificate
key /etc/certs/gateway_key.pem    #Gateway’s key
port 1194                         #Default OpenVPN 2.0 Port
comp-lzo                          #Use fast LZO compression
user nobody                       #drop root priv after
group nogroup                     #initialization
```



Routed Server (continued)

```
keepalive 15 45                         #ping every 15 seconds
                                        #restart if no ping
                                        #received in 45 seconds
ping-timer-rem                          #Don’t start ping clock
                                        #until we have a client
persist-tun                             #Don’t close/open tun
                                        #device during
                                        #ping-restart
persist-key                             #don’t re-read key after
                                        #ping restart
client-config-dir /etc/openvpn/clients  #Directory where client-
                                        #specific params are kept
ccd-exclusive                           #Require client-specific
                                        #params
client-to-client                        #allow client->client
verb 3                                  #verbosity of the log
```

Sample Configuration for a Routed Client (Windows Roadwarrior)

```
dev tun                              #Routed
remote gateway.shorewall.net         #Server’s Name
tls-remote gateway.shorewall.net     #Common Name in Server’s Certificate
tls-client                           #We are a TLS client
explicit-exit-notify                 #Notify when we exit
pull                                 #Accept server’s pushed parameters
ca "/Program Files/OpenVPN/certs/cacert.pem"
cert “E:/easy-rsa/keys/eastepnc6000.crt"
key “E:/easy-rsa/keys/eastepnc6000.key"
port 1194
comp-lzo
ping-timer-rem
persist-tun
persist-key
mute-replay-warnings
verb 3
```

- Only difference in a Linux config is the file names!

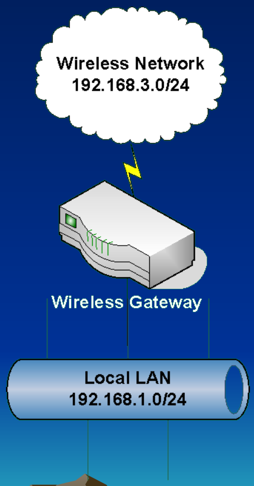
RoadWarrior’s CCD File

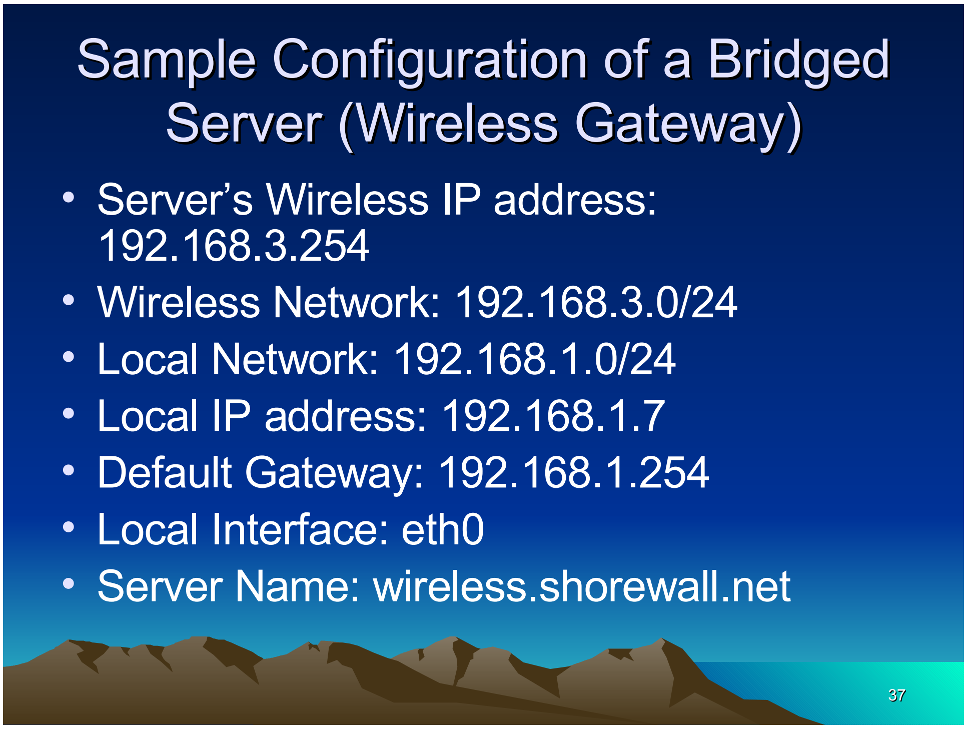
- On the server in /etc/openvpn/clients/
- Name is the same as the CN in the client’s certificate

```
#CCD for eastepnc6000.shorewall.net
#Local (server) IP and client IP
ifconfig-push 192.168.2.14 192.168.2.13
#Route to local network
push “route 192.168.1.0 255.255.255.0”
#Route to DNS server
push “route 206.124.146.177.255.255.255.255”
```

Wireless Bridge

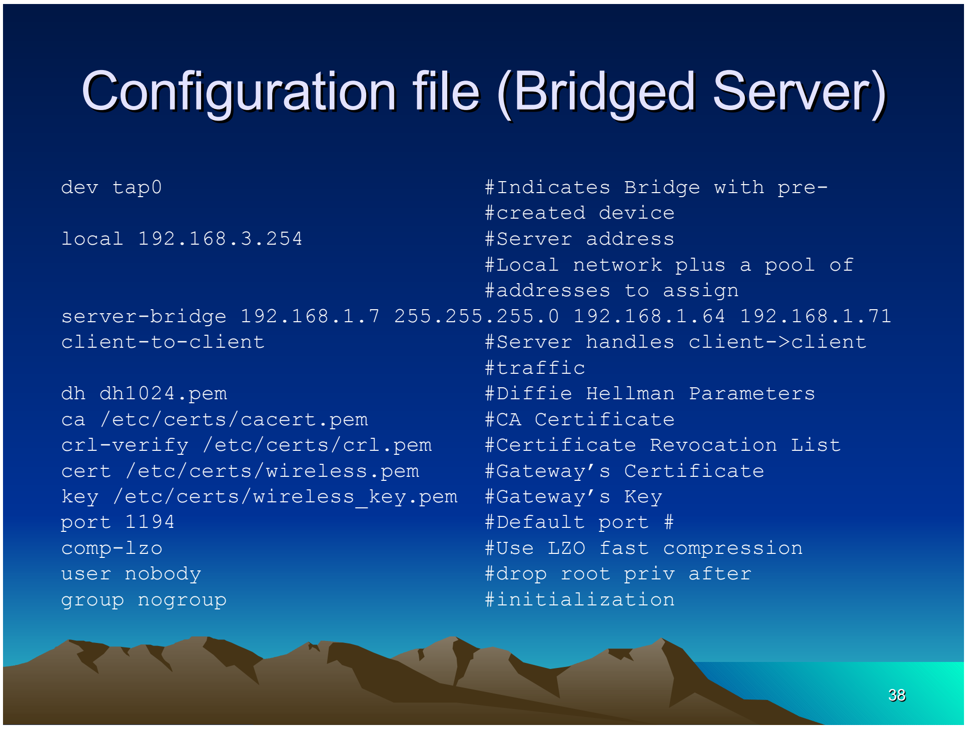
- Wireless Bridge is multi-homed:
  - Wireless
  - Local LAN





Sample Configuration of a Bridged Server (Wireless Gateway)

- Server’s Wireless IP address: 192.168.3.254
- Wireless Network: 192.168.3.0/24
- Local Network: 192.168.1.0/24
- Local IP address: 192.168.1.7
- Default Gateway: 192.168.1.254
- Local Interface: eth0
- Server Name: wireless.shorewall.net



Configuration file (Bridged Server)

```
dev tap0                         #Indicates Bridge with pre-
                                 #created device
local 192.168.3.254              #Server address
                                 #Local network plus a pool of
                                 #addresses to assign
server-bridge 192.168.1.7 255.255.255.0 192.168.1.64 192.168.1.71
client-to-client                 #Server handles client->client
                                 #traffic
dh dh1024.pem                    #Diffie Hellman Parameters
ca /etc/certs/cacert.pem         #CA Certificate
crl-verify /etc/certs/crl.pem    #Certificate Revocation List
cert /etc/certs/wireless.pem     #Gateway’s Certificate
key /etc/certs/wireless_key.pem  #Gateway’s Key
port 1194                        #Default port #
comp-lzo                         #Use LZO fast compression
user nobody                      #drop root priv after
group nogroup                    #initialization
```

Bridged Server (continued)

```
keepalive 15 45                         #ping every 15 seconds
                                        #restart if no ping
                                        #received in 45 seconds
ping-timer-rem                          #Don’t start ping clock
                                        #until we have a client
persist-tun                             #Don’t close/open tun
                                        #device during
                                        #ping-restart
persist-key                             #don’t re-read key after
                                        #ping restart
client-config-dir /etc/openvpn/bridge-clients
                                        #Directory where client-
                                        #specific params are kept
ccd-exclusive                           #Require client-specific
                                        #params
verb 3                                  #verbosity of the log
```

Bridged Server (continued)

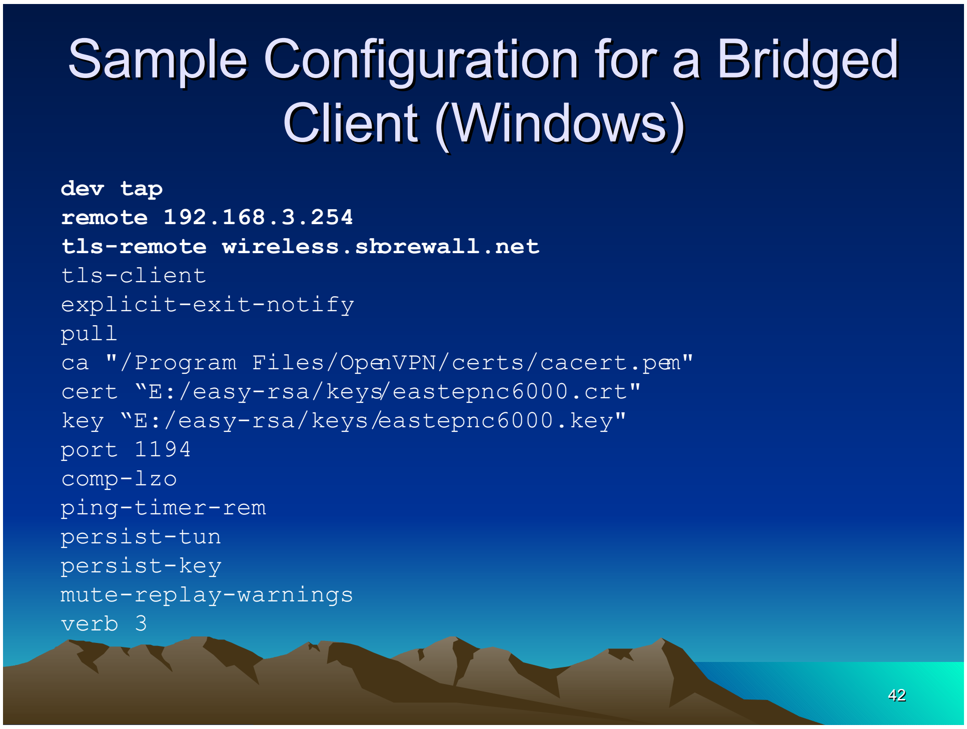
```
#
# The client supports a "redirect-gateway" option that redirects
# the default gateway through the VPN. I've found that to be
# somewhat unreliable whereas this trick works always
#
push "route 0.0.0.0 128.0.0.0 192.168.1.254"
push "route 128.0.0.0 128.0.0.0 192.168.1.254"
```

Bridged Server – Creating the Bridge

- See http://www.shorewall.net/Bridge.html for distribution-specific instructions

```
/usr/sbin/openvpn --mktun --dev tap0 #create dev
/sbin/brctl addbr br0                #create bridge
/sbin/ip link set tap0 up            #Up dev
/sbin/ip link set eth0 up            #Up local IF
/sbin/brctl addif br0 tap0           #Add devs to
/sbin/brctl addif br0 eth0           #to the bridge
```

- br0 is configured using Distribution’s tools



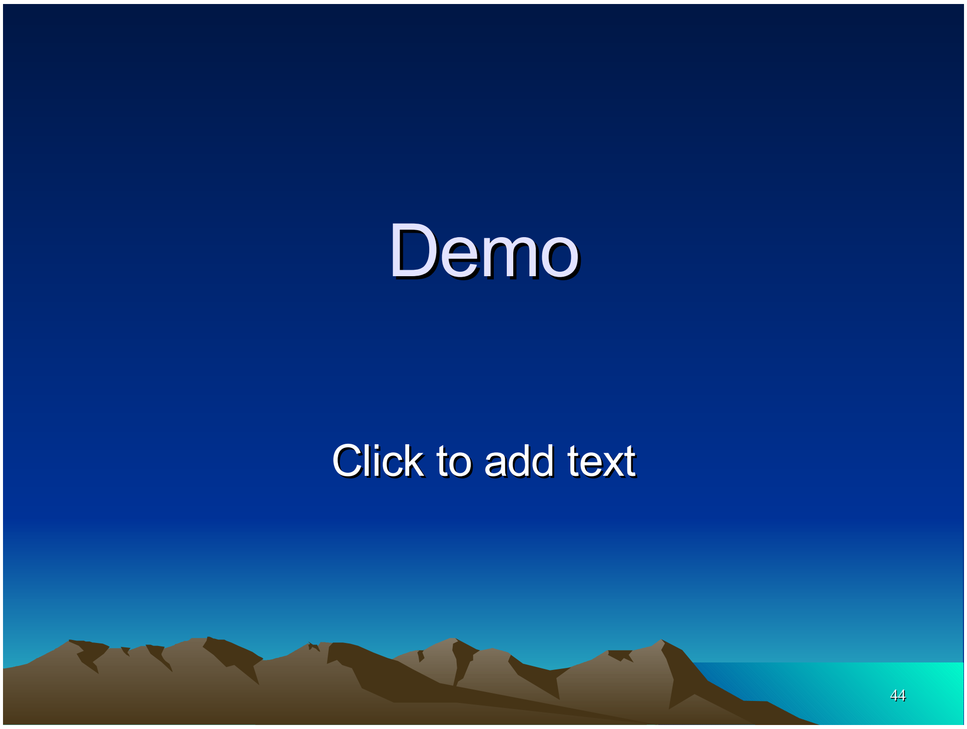
Sample Configuration for a Bridged Client (Windows)

```
dev tap
remote 192.168.3.254
tls-remote wireless.shorewall.net
tls-client
explicit-exit-notify
pull
ca "/Program Files/OpenVPN/certs/cacert.pem"
cert “E:/easy-rsa/keys/eastepnc6000.crt"
key “E:/easy-rsa/keys/eastepnc6000.key"
port 1194
comp-lzo
ping-timer-rem
persist-tun
persist-key
mute-replay-warnings
verb 3
```

Bridged Client’s CCD File

- On the server in /etc/openvpn/bridged-clients/

```
#CCD for eastepnc6000.shorewall.net
#Client IP
ifconfig-push 192.168.1.6 255.255.255.0
```



Demo

Click to add text

Q&A

Click to add text