

# OpenVPN

## Secret Paths on Public Places

Jochen Haemmerle

mail(at)jhaemmerle(dot)org

Unfug.org

June 20, 2006

# Outline

- 1 Introduction
  - What is a VPN?
  - Where OpenVPN comes in!
- 2 OpenVPN
  - Architectures
- 3 Networking
  - Network Devices
  - Tunnel Connection
- 4 Authentication
  - General
  - OpenSSL
  - Shared Object Plugins
  - Script Plugins

# What is a VPN?

## General

- Abbreviation for “Virtual Private Network”
- Creates an embedded overlay network into existing infrastructure (sort of)
- Not limited to networks ;)

## Facets

- Not necessarily encrypted
- Tunnel based (6to4)
- ATM / FrameRelay (Virtual Circuits)
- Multiprotocol Label Switching (MPLS)

# What is a VPN?

## Encrypted VPN

IPSec AH/ESP

PPTP EAP-TLS/MPPE (optional)

CIPE *“out of scope”*

Proprietary ...

OpenVPN ...We will see ;)

# OpenVPN...

## The Basics

- It's Open Source Software (GPL)
- Easy to setup (packages for everything and everybody)
- Cross OS-Plattform (Linux, Solaris, \*BSD, MacOSX, Windows)
- Cross Hardware-Plattform (x86, x86-64, Alpha, ARM, (MIPS))
- Encryption Provided by OpenSSL

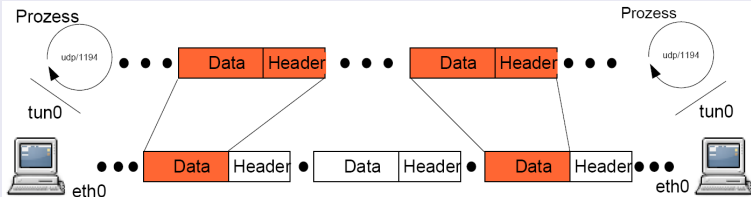
# OpenVPN...

## Compared to other solutions

- “Only” open source implementation that uses SSL
- No extra kernel module necessary (FreeS/WAN)
- Very portable
- Flexible
- Traffic shaping per tunnel
- No specific hardware
- ...

# OpenVPN...

## How OpenVPN works!

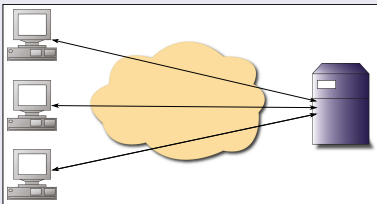


# General Architectures

## Point to Point

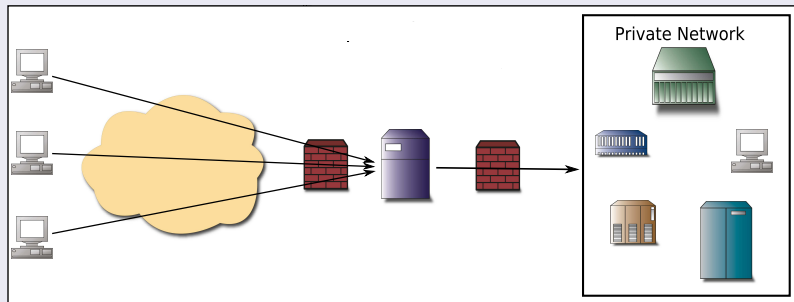


## Point to Multipoint



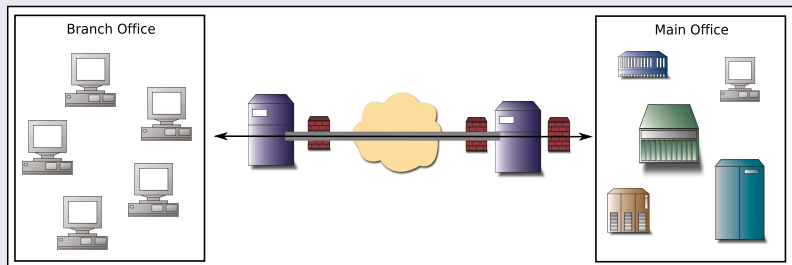
# Common Architectures

## Home Office



# Common Architectures

## Branch Office



# Virtual Network Devices

# Virtual Network Device - TUN

## TUN Device

- TUN is Virtual Point-to-Point network device
- Driver was designed as low level kernel support for IP tunneling
- Provides to userland application two interfaces:
  - `/dev/tunX` character device
  - `tunX` virtual Point-to-Point interface

# Virtual Network Device - TUN

## Advantages and Caveats

- Connecting peers are separated with IP-Subnets
  - Good manageability (firewalling, routing, acl)
  - Has to be managed
- Broadcast traffic is not submitted
  - Less network traffic
  - Some applications don't work (Samba browsing, various games)

# Virtual Network Device - TAP

## TAP Device

- TAP is a Virtual Ethernet network device
- Driver was designed as low level kernel support for Ethernet tunneling
- Provides to userland application two interfaces:
  - `/dev/tunX` character device
  - `tunX` virtual Point-to-Point interface

# Virtual Network Device - TAP

## Advantages and Caveats

- Connecting peers are in one Broadcast Domain
  - Difficult manageability (firewalling...)
  - IP addressing has to be consistent on both sides
  - Even higher management effort
- Broadcast traffic is not submitted
  - More “unwanted” network traffic (Broadcast of any kind e.g. ARP)
  - Application depending on broadcast work
  - Protocols other than IP work (e.g. IPX)

# Tunnel Connection

## UDP/TCP

### UDP

- OpenVPN default
- Default UDP/TCP Port 1194 (source and destination)
- Low Latency
- Depending on the environment - Fragmentation
- Easy integration in firewalls (statefull)

### TCP

- For situations when UDP is not available ;-)
- Default UDP/TCP Port 1194 (source and destination)
- Higher latency (depends on environment)
- Easy integration in firewalls (statefull)

# Tunnel Connection

## HTTP-Proxy

### HTTP-Proxy

- For very restrictive environments
- Using the CONNECT (like HTTPS) to connect to the server.
- Depending on Proxy-Server configuration the OpenVPN server port has to be set to 443
- TCP Connection
- Can be handy sometimes ;-)

# Authentication

## General

### Options

- Everything that is supported by OpenSSL
- Shared Object Plugins
- Script Plugins

# OpenSSL

## Preshared Secret

### Preshared Keys

- Key available on both sides
- Validation correct -> access granted
- Keys have to be transferred to the “other” side (*insecure*)
- No multipoint possible. Each key needs a new server process
- When key is compromised a new key has to be transmitted to all partners

# OpenSSL

## PKI

### Certificates

- Separate keypairs/certificates for client and server signed by a “trustworthy CA”
- If certificate signed by the “right” CA -> access granted
- Bidirectional authentication possible
- Sessionkey exchange using TLS/SSL
- Keys can be generated on client and server side
- Man-in-the-Middle possible if client and server CA are the same
- Revoking of compromised keys (CRL)

# Shared Object Plugins

## Precompiled Plugins

### openvpn-auth-pam

- Plugin to use PAM for authentication

### openvpn-auth-user-pass-verify

- Enables dual checking of certificate and user/password
- Only user/password is also possible (*insecure*)

### openvpn-auth-...

- ...

# Script Plugins

## openvpn-auth-user-pass-verify

- Simple scripts (Perl/python...) handle the authentication
- Return value decides auth was positive
  - 1 failure
  - 0 successful

# The End

Questions?