Tunnels & Network Troubleshooting

Unit 19 - Hands-On Networking - 2018

Prof. Dr.-Ing. Thorsten Herfet, Andreas Schmidt, Pablo Gil Pereira

Telecommunications Lab, Saarland Informatics Campus, 27th Feb. 2018

Recap

- **IPv6**:
  - More Addresses
  - Streamlined Header
  - Prefix Types
  - Auto Configuration (DHCP / SLAAC)
  - Transistion Techniques
- **Packet Analysis:**
  - Wireshark
  - Traffic Visibility
- **Building Networks:**
  - GNS3
  - Virtual Appliances

Definition

Tunnel

A covered passageway through an obstruction

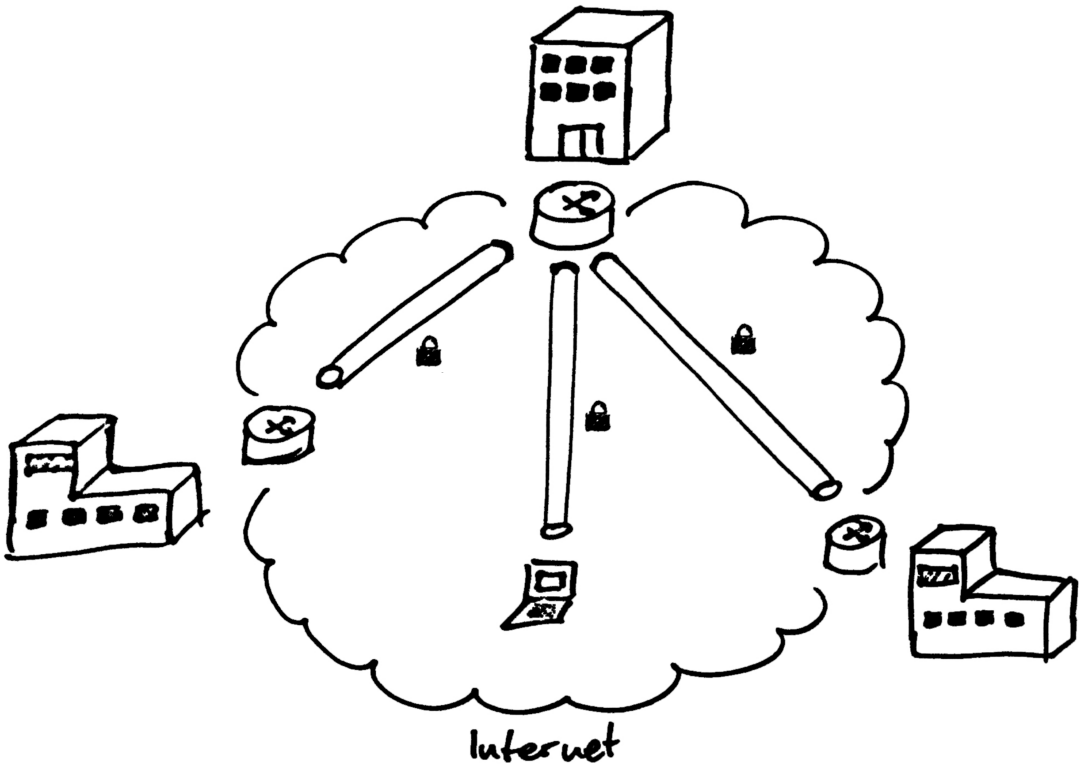
Network Tunnel

A (covered) passageway through a network

A network wormhole.

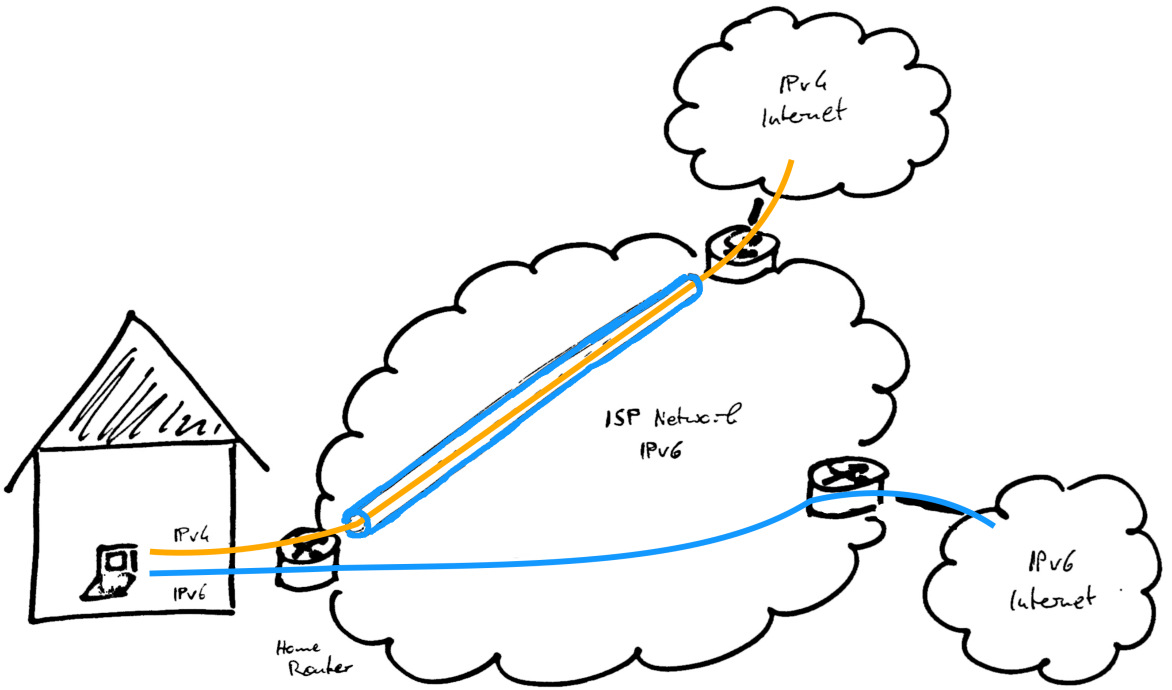
Tunnels | Why?

Create an **encrypted channel across untrusted networks** such as the Internet



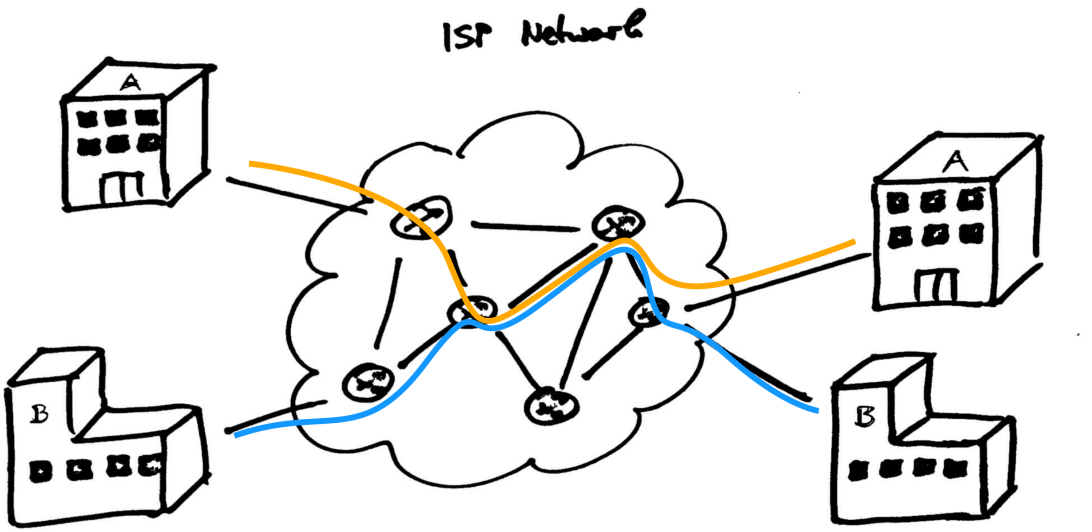
Tunnels | Why?

Transmission of packets between **disjoint networks without a direct routing path**



Tunnels | Why?

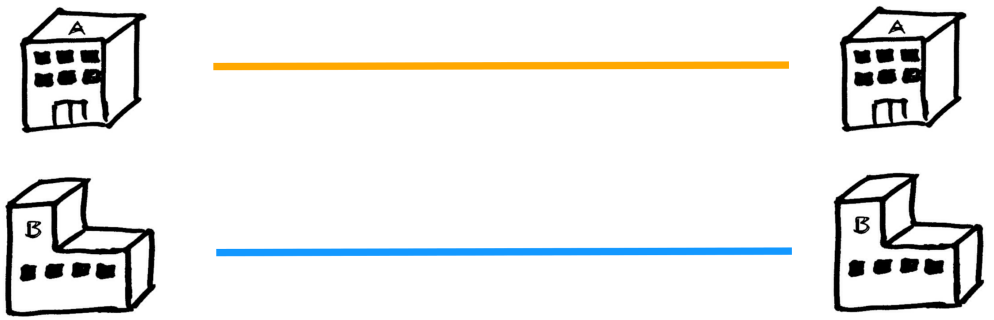
Create a **logical abstraction of the physical network** (**Overlay Network**)



Tunnels | Why?

Create a **logical abstraction of the physical network** (**Overlay Network**)

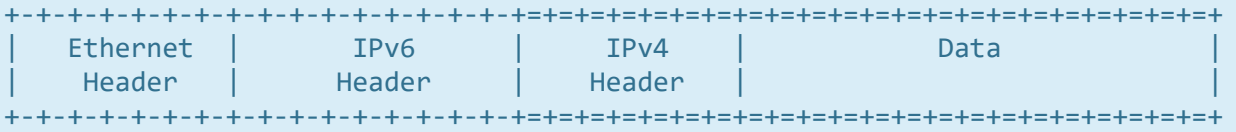
What the customers see:



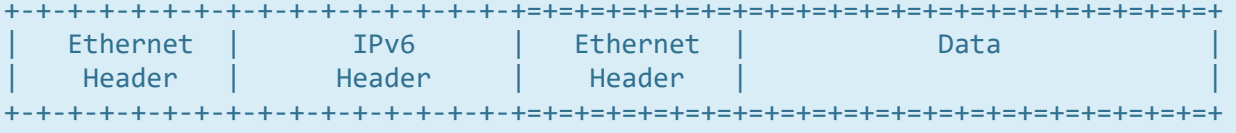
Tunnels | How?

Encapsulate datagrams of one protocol ...

- in another protocol of **the same layer**



- in another protocol of **a higher layer**



Not all tunnels encrypt data, but all **encapsulate** datagrams.

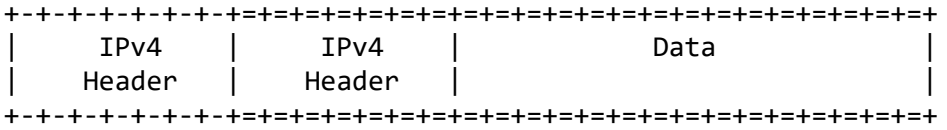
Tunnels in Linux

- Linux supports many tunnel modes out of the box.
- For **(almost) every tunnel endpoint** Linux creates a **virtual network interface**
- Basic tunnels can be managed using **iproute2**

```
$ ip tunnel help
Usage: ip tunnel { add | change | del | show | prl | 6rd } [ NAME ]
          [ mode { ipip | gre | sit | isatap | vti } ] [ remote ADDR ] [ local ADDR ]
          [ [i|o]seq ] [ [i|o]key KEY ] [ [i|o]csum ]
          [ prl-default ADDR ] [ prl-nodefault ADDR ] [ prl-delete ADDR ]
          [ 6rd-prefix ADDR ] [ 6rd-relay_prefix ADDR ] [ 6rd-reset ]
          [ ttl TTL ] [ tos TOS ] [ [no]pmtudisc ] [ dev PHYS_DEV ]

Where: NAME := STRING
       ADDR := { IP_ADDRESS | any }
       TOS  := { STRING | 00..ff | inherit | inherit/STRING | inherit/00..ff }
       TTL  := { 1..255 | inherit }
       KEY  := { DOTTED_QUAD | NUMBER }
```

IP in IP

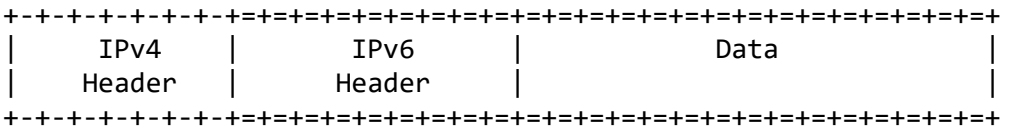
- RFC1853
- Encapsulates IPv4 in IPv4




```
# Create the tunnel interface and enable it
$ ip tunnel add tunnel0 mode ipip remote 198.51.100.3
$ ip link set dev tunnel0 up

# Set an ip address
$ ip address add 10.0.2.1/30 dev tunnel0

# ... or add a direct route to the other side
$ ip route add 192.0.2.0/24 dev tunnel0
```

IPv6 in IPv4

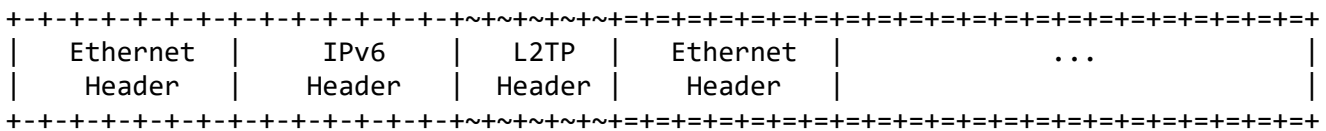
- 6in4 / SIT (Simple Internet Transition) RFC4213
- IPv6 transition technique
- Enable connectivity between disjoint IPv6 networks
- IPv6 datagram encapsulated in IPv4 datagram




```
# Create the tunnel interface and set it UP
$ ip tunnel add tunnel0 mode sit remote 198.51.100.3
$ ip link set dev tunnel0 up

# Set an IP address...
$ ip address add fd28:3bc8:1370::1/48 dev tun0

# ... or set a direct route
$ ip route add fd97:9877:1930:a:/64
```

L2TPv3 (Layer 2 Tunneling Protocol)

- RFC3931
- Tunnels **different link-layer protocols** (e.g. Ethernet, FrameRelay)
- Tunnels also called **pseudowires**
- Supports **multiplexing** (multiple sessions / interfaces per tunnel)
- Choice between **IP or UDP encapsulation**
- Often used to provide **link-layer connectivity in VPNs**




```
# Notice L2TP tunnels are not managed by the tunnel submodule of iproute2

# Create a tunnel using IP encapsulation
$ ip l2tp add tunnel tunnel_id 1 peer_tunnel_id 1 encap ip local 192.0.2.1 remote 198.51.100.3

# Set up a session in the tunnel.
$ ip l2tp add session name l2tp_eth0 tunnel_id 1 session_id 10 peer_session_id 10
```

SSH Tunnels

- Forward packets addressed to the local host to a remote host
- Forward packets addressed to a remote host to a local host
- Technically **not a tunnel**, but a **port forwarding with SNAT**
- Packets encapsulated in the SSH protocol

```
# Redirect port 8080 of the local host to port 80 at 198.51.100.17
# For 198.51.100.17 packets will seem to originate from 192.0.2.67
ssh -L 8080:198.51.100.17:80 root@192.0.2.67

# Redirect port 3000 of host 192.0.2.67 to port 3000 at 203.0.113.44
ssh -R 3000:203.0.113.44:3000 root@192.0.2.67
```

IPSec

IPSec (RFC4301)

- Protocol suite for **Secure IP Communication**
- Open Standard
- First published in 1998

- Open Standard
- State-of-the-art Security
- Highly customizable Security

- Rather complex

IPSec | Functions

- Data Origin Authentication
- Data Confidentiality
- Data Integrity
- Key Management

IPSec | Protocols

AH (Authentication Header)

Integrity Check / Authentication only

Guarantees **integrity** of header and data using **Message Authentication Codes**.

ESP (Encapsulating Security Payload)

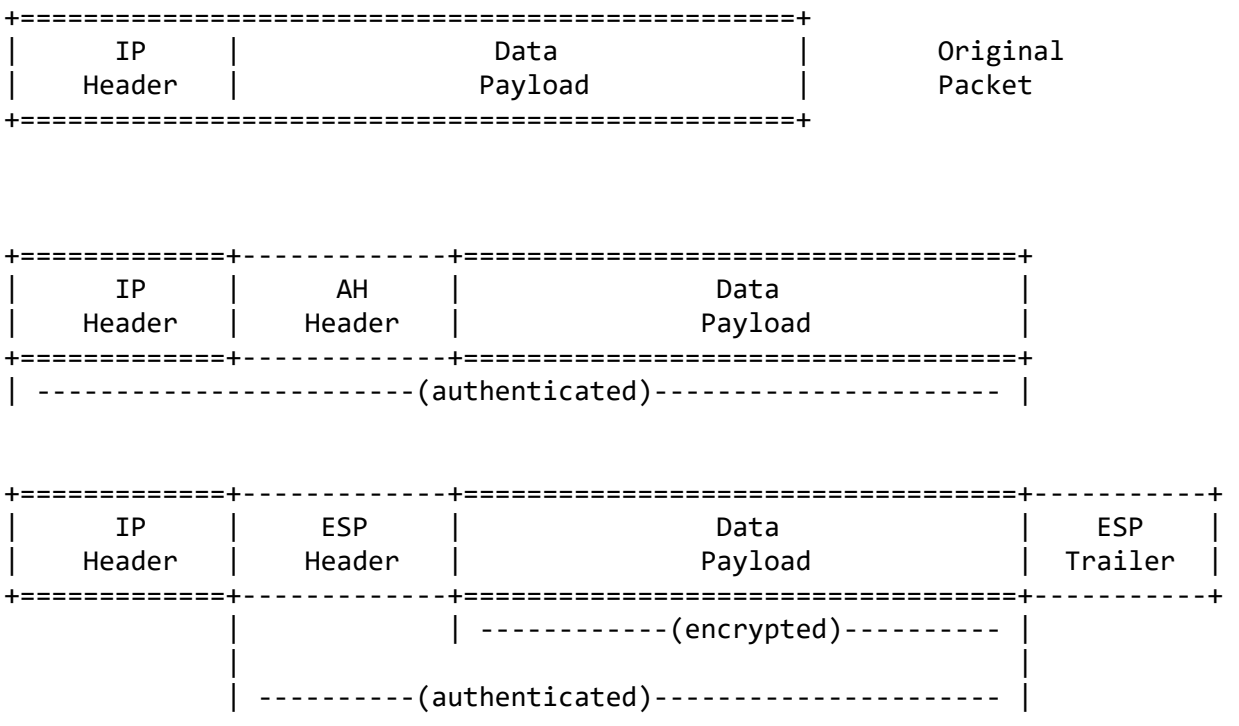
Provides **origin authenticity, integrity** and **confidentiality** of data (but not the header).

IKE (Internet Key Exchange Protocol, RFC4306)

Used to setup security associations between IPSec endpoints.

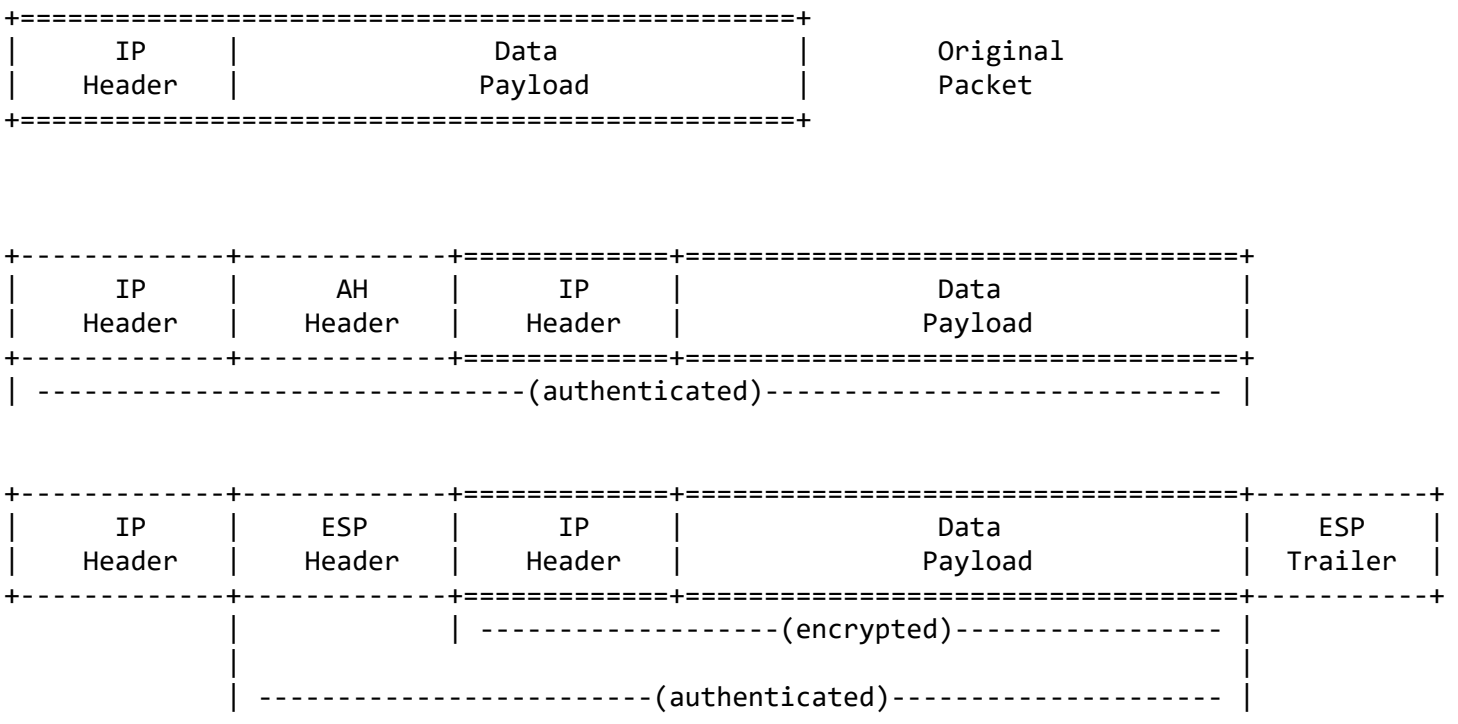
IPSec | Transport Mode

- Authentication / Encryption of the **data payload**
- IP header is not changed



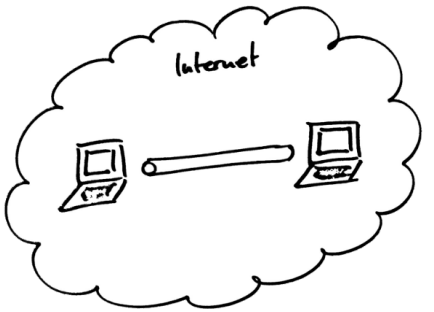
IPSec | Tunnel Mode

- Authentication / Encryption of **the whole packet including headers**
- Adds an additional IP Header

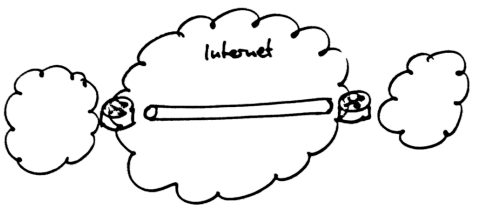


IPSec | Scenarios

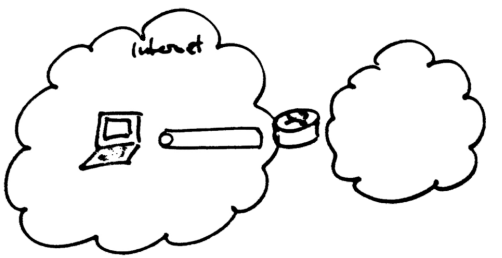
Host-to-Host



Net-to-Net (Site-to-Site)



Host-to-Net (Roadwarrior)



HON

IPSec | Security Policy / Associations

Policy Database

- Contains rules specifying **whether to use IPSec** and in **which mode** (Tunnel / Transport)
- **One-directional**, one needed for sending and one for receiving

Security Association (SA)

- Defines **algorithms**, **cryptographic keys** and **direction** for communication
- Have a unique ID which is included in each IPSec packet (**Security Parameters Index / SPI**)
- Exchanged using **IKE** and stored in local database
- Referenced by policy database entries

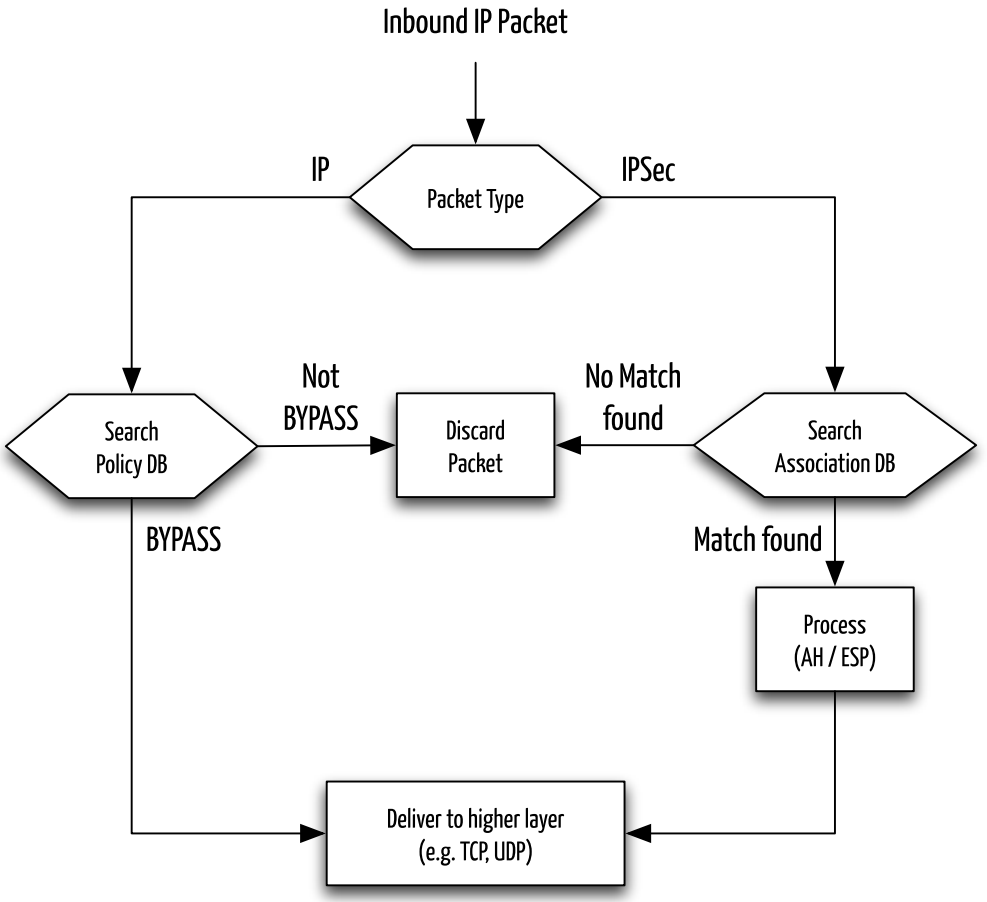
IPSec

In modern Linux kernels IPSec **does not use virtual interfaces**

Packets are transparently encrypted according to the policy database.

IPSec | Outbound Packet Flow

IPSec | Inbound Packet Flow



HON

IPSec in Practice

- IPSec support usually **part of the OS**
- Supported in **Linux**, **MacOS** and **Windows**
- Key exchange and SA management usually handled by userspace software

Implementations (,)

- strongSwan
- Openswan
- FreeS/Wan
- Libreswan
- OpenIKED
- KAME (raccoon)

IPSec in Practice

Example for Strongswan: Static VPN Tunnel using AES128 and pre-shared keys

```
# /etc/ipsec.conf - On the server

conn my-awesome-vpn-server
        keyexchange=ikev2
        left=192.0.8.1
        right=192.0.8.2
        ike=aes128-sha256-modp2048!
        esp=aes128-sha256-modp2048!
        authby=secret
```

```
# /etc/ipsec.conf - On the client

conn my-awesome-vpn-client
        keyexchange=ikev2
        left=192.0.8.2
        right=192.0.8.1
        rightsubnet=192.168.0.0/24
        ike=aes128-sha256-modp2048
        esp=aes128-sha256-modp2048
        authby=secret
```

Authentication keys / certificates are configured in `/etc/ipsec.secrets`.

VPN Software

IPSec is the standardized IP security solution, however there are alternatives.

- OpenVPN
  Open Source SSL VPN software. Supports L2/L3 tunnels.
  Uses TCP/UDP as transport.

- vTun
  Open Source VPN software. Supports L2/L3/PPP/serial tunnels.
  Uses TCP as transport.

- tinc
  OpenSource VPN software. Supports L2/L3 tunnels. Uses TCP/UDP as transport.
  Features mesh routing between VPN nodes.

- Proprietary/non-free VPN software from every network vendor you can think of

Network Troubleshooting


Andy Dwyer: It says you could have Network Connectivity Proble…

Problem Reports

> My Internet isn't working!

What is the problem?

Could be anything.

- Faulty network cable
- Broken Router
- Misconfigured Firewall
- Webserver down
- Broken DNS
- ...

Randomly checking things is not very effective

Get a system!

Methodology

Define the Problem

Isolate Scope

Isolate Network Layer

Isolate the Cause

Fix It

Defining the Problem

> My Internet isn't working!

- User problem descriptions often vague
  ("Nothing works!")
- Description does not necessarily tell you anything about the actual problem
  ("Everytime I try to access a website, I get an error.")
- Users lie (unintentionally)!
  (their favorite cooking blog isn't available, so the internet must be broken...)

- Check if the **problem (still) persists**
- Ask user for **details**
- Try to **be specific**

Defining the Problem

Sometimes the biggest value is in defining the problem, not necessarily in solving it.

Defining the problem ...

- **helps you learn**
- **saves time** in solving the problem
- helps you **describe the problem to others** (e.g. your colleagues, your ISP, ...)

Isolate the Scope

How many users are affected?

Does the problem show up in regular intervals / only at specific times?

Isolating the scope ...

- helps you determine the **urgency**
  If the problem affects other users too the problem probably needs to be handled more quickly. Maybe more important problems have been reported.

- can give you further **details about the problem**
  If nobody can access the internet the problem might be in the network and not a problem with the user's workstation itself.

Isolate the Network Layer

Problems can be **anywhere** in the ISO/OSI stack

Work your way through the layers **one by one** starting with the **physical layer**

Physical Layer (aka "Asking Stupid Questions")

- Is the network cable plugged in?
- Is the link led flashing?
- Has the computer been moved recently?
- Is the laptop connected to the wifi?

- Every layer **depends on the next lower layer**
- If the computer is not plugged in no higher layer can send anything
- Maybe the network cable is faulty

Link Layer

❓ Are other devices on the LAN reachable via the Link-Layer?

```
# Check IP address configuration

$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 70:54:d2:7b:7a:db brd ff:ff:ff:ff:ff:ff
    inet 134.96.86.110/25 brd 134.96.86.127 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::7254:d2ff:fe7b:7adb/64 scope link
       valid_lft forever preferred_lft forever
```

❓ **Why IP? Isn't IP operating on the network layer?**

⚠ **Without IP there is no ARP / NDP!**

Link Layer

```
# ping some hosts on the network
# (the default gateway is always good)

$ ip route show
default via 134.96.86.1 dev eth0  proto static
...

$ ping -c1 134.96.86.1
PING 134.96.86.1 (134.96.86.1) 56(84) bytes of data.
From 134.96.86.110 icmp_seq=1 Destination Host Unreachable

--- 134.96.86.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

```
# Check ARP table afterwards

$ ip neighbor show
... (this will list nothing) ...
```

Possible Causes:

- **Misconfigured VLAN** on the switch?
- Overzealous **switch security feature**?

Network Layer

Can we ping the destination? How far do packets get?

```
$ ping -c1 web-f6.rz.uni-saarland.de
PING web-f6.rz.uni-saarland.de (134.96.7.186) 56(84) bytes of data.
From 134.96.86.110 icmp_seq=1 Destination Host Unreachable

--- web-f6.rz.uni-saarland.de ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

```
$ traceroute www.nt.uni-saarland.de
traceroute to www.nt.uni-saarland.de (134.96.7.186), 10 hops max, 60 byte packets
 1  core13.hetzner.de (213.239.203.221)  0.203 ms core14.hetzner.de (213.239.203.225)  0.177 ms ...
 2  core1.hetzner.de (213.239.245.254)  3.277 ms  3.308 ms core1.hetzner.de (213.239.245.250)  3.295 ms
 3  cr-fra1-be1.x-win.dfn.de (80.81.192.222)  3.838 ms  3.824 ms  3.849 ms
 4  cr-dui1-hundredgige0-6-0-0.x-win.dfn.de (188.1.144.177)  8.549 ms  8.531 ms  8.562 ms
 5  xr-saa1-te2-1.x-win.dfn.de (188.1.146.86)  15.685 ms  15.560 ms  15.595 ms
 6  kr-saa12.x-win.dfn.de (188.1.234.38)  15.431 ms  15.411 ms  15.420 ms
 7  c65evss2win.net.uni-saarland.de (134.96.6.53)  16.574 ms  23.480 ms  23.419 ms
 8  * * *
```

Possible Causes:

- **Wrong IP address / subnet**
- Router **offline** / Routing problem

Transport Layer

Is the specific port open / reachable

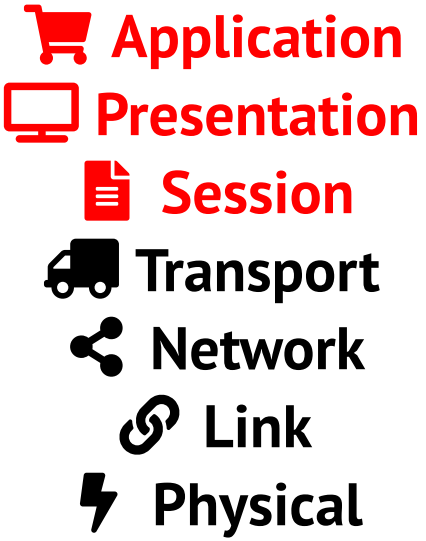
```
# If you have it, use nmap
# (... or use Telnet for TCP)

$ nmap -p 80,443 www.nt.uni-saarland.de

Starting Nmap 6.40 ( http://nmap.org ) at 2016-09-21 15:57 CEST
Nmap scan report for www.nt.uni-saarland.de (134.96.7.186)
Host is up (0.00051s latency).
rDNS record for 134.96.7.186: web-f6.rz.uni-saarland.de
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
```

Possible Causes:

- Server software not running (TCP RST)
- Misconfigured firewall (SYN silently dropped / ICMP prohibited)

All the other Layers

The higher layers are left as an exercise to the reader.



Higher layers can often only be checked with **detailed knowledge of the application** and are **often not diagnosed just by observing the network**.

Isolate the Cause

- Knowing **what the problem is** doesn't necessarily tell you the **source**
- May need **further investigation**
  - Checking **configs**
  - Checking **log files**
  - Perform **more tests** from **different parts of the network**
- Requires **detailed technical knowledge** about the network and network protocols as well as **intuition**
- If **many people are affected**, maybe it is caused by **the same thing**
- Even if you cannot / are not allowed to fix the problem yourself, you can **formulate a theory and pass it on with your findings**

Wrap-Up

Questions?

Take-Home Messages

- **Tunnels** can be helpful in connecting:
  - disjoint networks / islands
  - people on the road
  - non-routable protocols over WANs
- They do so by **encapsulating datagrams**
- There are **many kinds of tunnels**
- If you need **privacy** or **security**, use **IPSec** or another flavour of **VPN**
- IPSec offers **data integrity**, **confidentiality**, **authentication** and **key management**
- IPSec can **selectively handle network traffic** (or not)

- **Troubleshooting network problems** can be **difficult**, but it's **easier if you have a system**.

Further Reading

- Steve Friedl's Illustrated Guide to IPSec