# Communication Wireshark



Document name: Communication Wireshark.docx
Last update: 15.04.2020
Author: Albert Balogh

## Content

Introduction ..... 5
Certification ..... 5
Links / Contacts ..... 6
Commands ..... 6
Windows ..... 6
Unix ..... 6
SHORTCUTS ..... 6
TOOLS ..... 7
Bit-Twist Packet Generator ..... 7
Cable Analyzer ..... 7
Capinfos ..... 7
Dumpcap ..... 7
Editcap ..... 8
iPerf / JPerf ..... 8
Mergecap ..... 8
Rawshark ..... 8
Remote Capture ..... 8
tcpdump ..... 8
Text2pcap ..... 9
Tshark ..... 9
WireEdit ..... 10
Kleine Spielerei mit Excel ..... 10
INSTALLATION ..... 10
Supported File-Extensions ..... 11
Setup GeoIP ..... 12
Setup TCP Key Settings ..... 14
Setup Ugly Colors ..... 14
Wireshark Screen ..... 15
Protocol Data Unit (PDU) ..... 15
SPFB ..... 15
Data (OSI-Layer 5, 6, 7) ..... 15
Segment (OSI-Layer 4) ..... 15
Packets (OSI-Layer 3) ..... 15
Frames (OSI-Layer 2) ..... 16
Bits (OSI-Layer 1) ..... 16
Menue Items ..... 16
File ..... 16
Edit ..... 16
View ..... 16
Go ..... 16
Capture ..... 16
Analyze ..... 16

Statistics ... 17
Statistics → Protocol Hierarchy ... 17
Statistics → IO Graphs ... 17
Statistics → TCP Stream Graphs ... 17
Statistics → WLAN Traffic ... 17
Telephony ... 17
Internals ... 17
The top 10 reasons for a slow network ... 17
No. 1 / Packet-Loss ... 17
No. 2 / Client, Server and Wire Latency ... 17
Client-Latency/Service Response Time (SRT) ... 18
Server-Latency/Service Response Time (SRT) ... 19
Path-Latency/Wire-Latency (RTT) ... 19
No. 3 / Windows Scaling Issues ... 19
Congestion Window (cwnd) ... 19
Receive Window / Window Scaling (rwin) ... 19
No. 4 / Service Response issues and Application behavior ... 20
No. 5 / Network Design Issues ... 20
No.6 / Path Issuse (such as QoS) ... 20
No. 7 / Interference/Noise (WLAN) ... 20
No. 8 / Poor Signal Strength (WLAN) ... 20
No. 9 / Timing Problems ... 20
No. 10 / Users ... 20
STARTING ANALYSIS ... 21
General Analysis Steps ... 21
How to capture on the wire ... 23
Security Tasks for Network Analysts ... 23
Time Display Format ... 23
BASELINING ... 24
Create following baselines: ... 24
Background Noise (_BgNoise) ... 25
IO Graphs ... 25
Network Forensics ... 26
Security Problems ... 28
Remote Procedure Call (RPC) ... 29
Honeypot ... 29
Watch for reconnaissance processes ... 29
IP Scans ... 29
TCP Scans ... 29
UDP Scans ... 29
OS Fingerprinting ... 29
Address Scans ... 29
Application Scans ... 29
Macof Scans ... 29
Packet Handcrafting ... 30
TCP-Problems ... 30
ICMP ... 30
Exceeded Messages: ... 31
Echo Messages: ... 31
Redirect messages: ... 31
Destination Unreachable messages: ... 31
Parameter Problem messages: ... 31
Source Quench messages: ... 31
Timestamp Reply messages: ... 31
Information Request messages: ... 31
TRACERT ... 32
Nagle Algorithm ... 32
Explicit Congestion Notification [ECN] ... 32
What Triggers TCP Retransmission (Note) ... 32
What Triggers Previous Segment Not Captured (Warning) ... 32
What Triggers ACKed Lost Packet (Warning) ... 32
What Triggers Keep Alive (Warning) ... 32
What Triggers Duplicate ACK (Note) ... 33

What Triggers Zero Window (Warning) ..... 33
What Triggers Zero Window Probe (Note) ..... 33
What Triggers Zero Window Probe ACK (Note) ..... 33
What Triggers Keep Alive ACK (Note) ..... 33
What Triggers Out-of-Order (Warning) ..... 33
What Triggers Fast Retransmission (Warning) ..... 33
What Triggers Window Update (Chat) ..... 33
What Triggers Window is Full (Note) ..... 33
What Triggers TCP Ports Reused (Note) ..... 34
What Triggers 4 NOP in a Row (Warning) ..... 34
TCP-Handshake / TCP-Connection ..... 35
SYN (Client) ..... 36
SYN/ACK (Server) ..... 36
ACK ..... 36
TCP Options ..... 37
TCP-Terminating ..... 39
UDP-Problems ..... 39
IPv4-Problems ..... 40
IPv6-Problems ..... 41
ARP/RARP Problems ..... 41
Gratuitous ARP (GARP) ..... 42
DHCP Problems ..... 42
DHCP-Relay Agent Problems ..... 43
DHCPv6 Problems ..... 43
DNS Problems ..... 43
FTP Problems ..... 45
HTTP Problems ..... 45
HTTPS/SSL Problems ..... 46
SSL Settings ..... 47
X.509 ..... 47
IMAP Problems ..... 47
Lotus Notes Problems ..... 47
POP Problems ..... 47
SMB-Problems ..... 48
SMB 1 ..... 48
SMB 2 ..... 48
SMB 3 ..... 48
SMTP Problems ..... 48
SSH Problems ..... 49
VoIP Problems ..... 49
WLAN Problems ..... 49
Application Problems ..... 50
Non-Standard Ports ..... 50
Suspicious Hosts/Traffic ..... 50
Network Scans ..... 51
Dissector ..... 51
Internet Protocol Version 4 ..... 51
Router ..... 51
Switches & Hubs ..... 52
DFILTERS - VIEW FILTERS / DISPLAY FILTERS ..... 52
Filter for Conversations ..... 59
Filter Expressions ..... 60
CFILTERS - CAPTURE FILTERS ..... 62
TECHNIQUES ..... 65
BitTorrent ..... 65
DIG ..... 65
IPv4 Multicast Addresses ..... 65
WHOIS ..... 66
Troubleshooting: Registrar ..... 66
HTTP STATUS CODES ..... 67
PROTOCOLS ..... 68
Abbreviations ..... 79
Table of Figures ..... 80

Tables ..... 81
Index ..... 82

# Introduction

Laura Chappell: www.chappellU.com info@chappellU.com

- Wireshark stores the captures as "**P**acket **Cap**ture Next **G**eneration (**.pcapng**)".
- Wireshark uses two drivers, called ***NPcap*** (Old: WinPcap**)** and **lipcap** to capture data on the "link layer" level.
- The capture syntax format is "**B**erkeley **P**acket **F**iltering (**BPF**-Syntax)".
- "**T**est **A**ccess **P**orts (**TAP**'s)" are used to capture traffic at certain points in the physical network.

**„The Packets never lie!"**


# Certification

Successful completion of the **Wireshark CAN Certificate** Exam indicates you have the knowledge required to capture network traffic, analyze the results and identify various anomalies related to ***performance*** or ***security*** issues. The **WCNA** is an ideal complement to the:

**CISSP**, **CCIE**, **CNP**, **Network+ and Security+ certifications**

The certificate is **3 years valid**. To keep the certificate **20 CPEs/Year** are required.

Wireshark University Certified Training Partners:
www.wiresharktraining.com/iltpartners

# Links / Contacts

www.wireshark.org
www.wiresharkU.org
www.wireshark.org/docs

wiki.wireshark.org
ask.wireshark.org

| | |
|---|---|
| www.wiresharkbook.com | info@wiresharkbook.com |
| www.wiresharktraining.com | |
| www.wiresharktraining.com/certification | info@wiresharktraining.com |
| www.lcuportal2.com | |
| www.pcapr.net | pcap repository! |
| www.emergingthreats.net | Malware patterns |
| http://hak5.org/episodes | Education |

# Commands

## Windows

`Ipconfig`

`ping`

Use special ping to mark sections in the trace.
e.g. `ping -n 1 -|101 <dst_ip>`

`traceroute`

## Unix

`ifconfig`
`dumpcap`
`tcpdump`

# SHORTCUTS

| | |
|---|---|
| Shift + Ctrl + A | Profiles |
| Ctrl + W | Close only Capture |
| Ctrl + M | Mark packet |
| Ctrl + Shift + M | Mark packets |
| Ctrl + Shift + N | Next Marked Packet |
| Ctrl + Shift + B | Previous Marked Packet |
| Ctrl + T | Set time reference *REF* |
| Ctrl + Shift + E | Analyze Enabled Protocols |
| Ctrl + F | Find a packet |

# TOOLS

- LUA-Script language
  Add-On to create FW-Rules from packets.
- Net Optics → www.netoptics.com
- PortableApps Suite → portableapps.com
- Smokeping
- SteelCentral Packet Analyzer (Former: Cascade Pilot)

## Bit-Twist Packet Generator

- Source:http://bittwist.sourceforge.net/
- Sanitize your trace

## Cable Analyzer

DSP-4100 Digital CableAnalyzer www.flukenetworks.com

| | |
|---|---|
| WhatsUP | Grafische Darstellung des gesamten Netzwerkes<br>Überwachen der Devices und Services<br>Meldungen via Pager, SMS, E-Mail optische und akustischeWarnung Fernwartung |
| SurfControl | Wer/Was/Wann/Wie im Internet/Intranet<br>Security features<br>Reports<br>www.mulltiware.ch |
| NetOP | PC-Remote Control<br>- Fernwartung & Admin<br>- Verteilungstool<br>- Zentral. Security |
| Net OP School | www.avatech.ch |
| Imail Server | Skalierbar auf mehrere tausend User<br>POP3, IMAP4, SMTP, LDAP |

## Capinfos

- Prints information's about trace files.

Examples: Page 828

```
capinfos -h
capinfos <input file>
capinfos -t 100pkts.pcap
capinfos -csd <Input File>
```

## Dumpcap

- If Wireshark or tshark doesn't keep up with the traffic, try to use ***dumpcap***.
- DUMPCAP executes directly to save execution time.

```
dumpcap -h
dumpcap -D   Shows the interfaces

dumpcap -f   Capture filter
```

Examples:

```
dumpcap -i1 -f "udp port 53"
```

## Editcap

- To slice pcaps and remove duplicates.

c:\Program Files\Wireshark"\editcap

Examples: Page 831.

```
editcap -h
editcap -c 1000 <input.pcap> <output.pcap>
editcap -csd <Input File>
editcap -d <input.pcapng> <output.pcapng>    ➔ Eliminates duplicate packets!
```

## iPerf / JPerf

- See: Performance Test
- Source:iperf.fr
- Performance Tests
- Iperf: Command Line Tool (CLI)
- Jperf: Is the GUI to Iperf
- iPerf3 is a tool for active measurements of the maximum achievable bandwidth on IP networks.
- It supports tuning of various parameters related to timing, buffers and protocols (TCP, UDP, SCTP with IPv4 and IPv6).
- For each test it reports the bandwidth, loss, and other parameters.
- This is a new implementation that shares no code with the original iPerf and also is not backwards compatible.
- iPerf was orginally developed by NLANR/DAST.
- iPerf3 is principally developed by ESnet / Lawrence Berkeley National Laboratory. It is released under a three-clause BSD license.

***Usage:***

```
Iperf -s                    ➔ Starts the sever
Iperf -c <server IP)        ➔ Starts the client
```

## Mergecap

- To merge two or more trace files.

Examples: Page 835

```
mergecap -h
mergecap -w..
mergecap -w <OFile.pcapng> <IFile1> <IFile2>
```

## Rawshark

- Command Line Tool (CLI)
- Dump and analyze raw pcap data.

## Remote Capture

- rpcapd.exe

Use <rpcapd.ini> for pre configuration.

## tcpdump

There are occasions when you want to capture packets using tcpdump rather than wireshark, especially when you want to do a remote capture and do not want the network load associated with running Wireshark remotely (not to mention all the X traffic polluting your capture).

However, the default tcpdump parameters result in a capture file where each packet is truncated, because most versions of tcpdump, will, by default, only capture the first 68 or 96 bytes of each packet.

To ensure that you capture complete packets, use the following command:

```
tcpdump -A
tcpdump -i <interface> -s 65535 -w <xxx-file>
```

## Text2pcap

- Source:Integrated with Wireshark
- Command line tool (CLI)
- Converts raw Hex
- Generate a capture file from an ASCII hexdump of packets.

## Tshark

Examples: Page 826

- Tshark (Command Line Capture <CLI> and Remote Analysis)

```
-a      autostop condition
        filesize:x (KB) Files:x (Stop after x files)

-c x    Stop after x packets

-w      Output File

-r      Read an existing capture file
-e

-R      Display filter
-T      fields
-Y      Display filter

tshark -h       Display Tshark parameters
tshark -D       ➔ Shows the interfaces

tshark -b filesize:1000 -b files:2 -n -w traces-test.pcap
tshark -a duration:20 -n -w shorttace.pcap
tshark -c 100 -n -w 100kpts.pcap
tshark -c 100 -f arp
tshark -q -z io,stat,5,ip.addr==255.255.255.255
tshark -q -z conv,tcp
tshark -i 2 -c 2000 -w output.pcap
tshark -i2 -a files :6 duration:30 -w <outputfile.pcapng>

Captures on Local Area Connection 2
-----------------------------------
tshark -i 1 -a filesize:100000 -b files:10 -w D:\Users\...\Documents\Daten\_Buffer\Output.enc
tshark -i 1 -b filesize:1000 -b files:2 -w D:\Users\...\Documents\Daten\_Buffer\test.pcapng
tshark -i 1 -a duration:20 -w D:\Users\...\Documents\Daten\_Buffer\test.pcapng
tshark -i 1 -a filesize:1000 -b files:1 -w "output.pcapng"
tshark -i 1 -f "port 161" output pcapng


Capture hosts
-------------
tshark -i 1 -qz hosts > D:\Users\...\Documents\Daten\_Buffer\host.txt


Statistic on Local Area Connection 2
------------------------------------
tshark -i 1 -qz io,phs
tshark -i 1 -qz conv,eth -qz conv,tcp
tshark -i 1 -qz <protocol>,srt              Service Response Times

BAT-File
--------
Eine Batch-Datei mit folgendem for loop, filtert alle Dateien in %source% mit Filter %filter% und
schreibt die Resultate in %dest%

for /f %%f in ('dir /b %source%') do (
                %tshark% -r %source%\%%f -R %filter% -w %dest%\%%f
)
```

## WireEdit

- Source:Omnipacket.com
- Editing pcap files
- Free to use, but License Key required to use

## Kleine Spielerei mit Excel

- http://jeffsoh.blogspot.ch/2012/01/bpfs-and-bit-masking.html
- http://www.wireshark.org/tools/string-cf.html

capturing HTTP GET requests

```
port 80 and
(tcp[((tcp[12:1] & 0xf0) >> 2):4] = 0x47455420 or
 tcp[((tcp[12:1] & 0xf0) >> 2)+8:4] = 0x20323030)
```

# INSTALLATION

Download: https://www.wireshark.org/download.html

Configuration-Files: **Help →About Wireshark → Folders**

- ❑ "File" dialogs
- ❑ Temp
- ❑ **Personal configuration (Save This)**
  `e.g.: <Drive>:\Users\<user>\AppData\Roaming\Wireshark`
  - cfilters default capture filters
  - dfilters default display filters
    You may copy them to your Folder
  - dfilter_macros
  - colorfilters default coloring rules
    You may find them in the Frame Section.
  - manuf default Organizationally Unique Identifier (OUI) list (global)
  - services default MIB modules to load
- ❑ Global configuration
- ❑ System
- ❑ Program
- ❑ Personal Plugins
- ❑ Global Plugins
- ❑ GeoIP path (Save This)
  `e.g.: C:\Temp\GeoIP`

### *Edit → Preferences*

**Disable:**

- IP-, UDP- and TCP-Checksum
- TCP <Allow subdissector to reassemble TCP streams>
- Capture: Update list of packets in real time
- If you don't disable network name resolution Wireshark will send DNS PTR queries to your DNS-Server.

**Enable:**

- Edit → Preferences → Protocols → TCP <Calculate conversation timestamps>
- TCP <Track number of bytes in flight>

Edit →Preferences → User Interface

**Change <User Interface>:**

- ❑ Maximum recent filters ➔ 30
- ❑ Maximum recent files ➔ 30

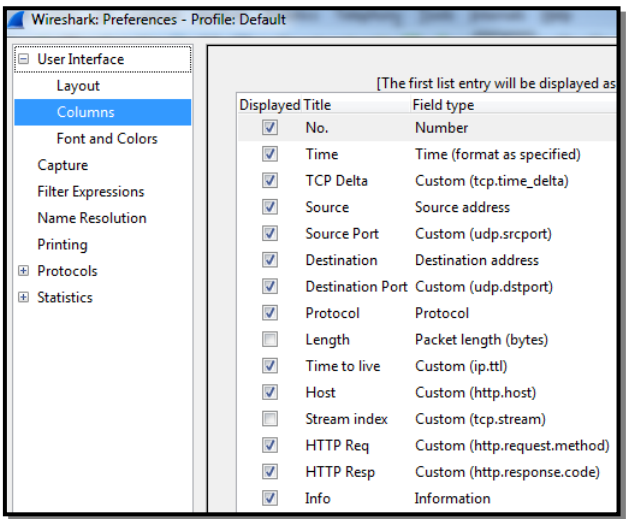



**Figure 1: Configure View**

# Supported File-Extensions

.enc Sniffer DOS file format

# Setup GeoIP

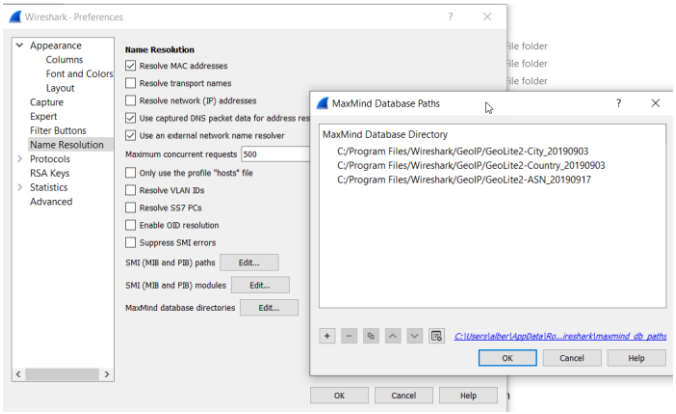
1. **Download:** [https://dev.maxmind.com/geoip/geoip2/geolite2/](https://dev.maxmind.com/geoip/geoip2/geolite2/)

| Database | MaxMind DB binary, gzipped | CSV format, zipped |
|---|---|---|
| GeoLite2 City | Download (md5 checksum) | Download (md5 checksum) |
| GeoLite2 Country | Download (md5 checksum) | Download (md5 checksum) |
| GeoLite2 ASN (Autonomous System Number) | Download (md5 checksum) | Download (md5 checksum) |

2. **Unzip them to your Wireshark directory and you should have now 3 folders.**
   Link: C:\Program Files\Wireshark\GeoIP

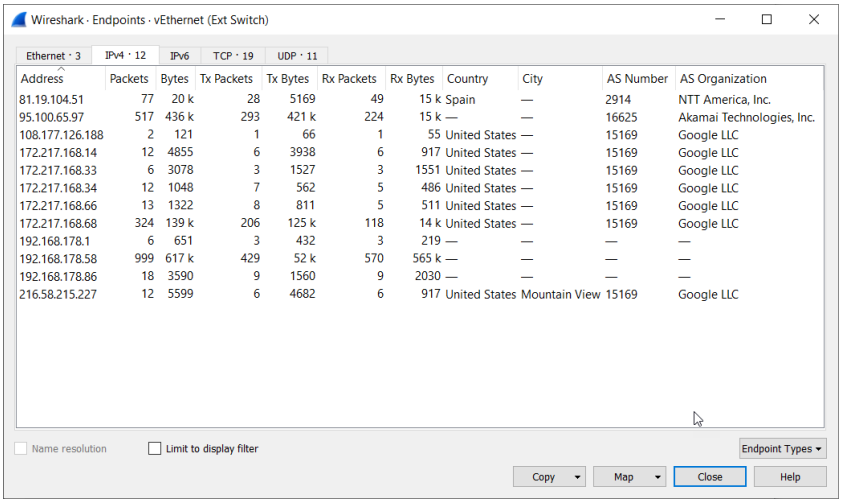
| | | |
|---|---|---|
| GeoLite2-ASN_20190917 | 17/09/2019 16:36 | File folder |
| GeoLite2-City_20190903 | 17/09/2019 16:37 | File folder |
| GeoLite2-Country_20190903 | 17/09/2019 16:37 | File folder |

3. **Open Wireshark and go to:** Edit → Preferences → Name Resolution → Max Mind …
   **Add the folders as shown below.**



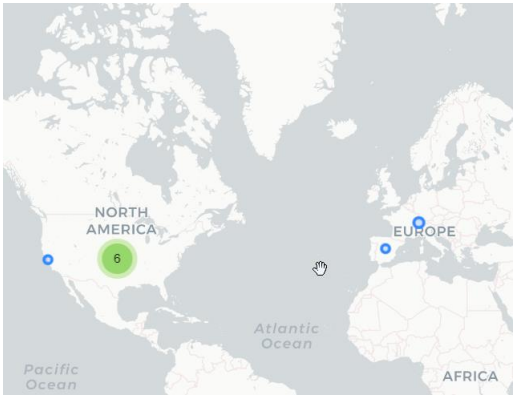

4. **Restart Wireshark**
   Create your capture and assure you have traffic to external sites.

5. **Open now:** **Statistics → Endpoints → Map → Open in Browser**




| Address | Packets | Bytes | Tx Packets | Tx Bytes | Rx Packets | Rx Bytes | Country | City | AS Number | AS Organization |
|---|---|---|---|---|---|---|---|---|---|---|
| 81.19.104.51 | 77 | 20 k | 28 | 5169 | 49 | 15 k | Spain | — | 2914 | NTT America, Inc. |
| 95.100.65.97 | 517 | 436 k | 293 | 421 k | 224 | 15 k | — | — | 16625 | Akamai Technologies, Inc. |
| 108.177.126.188 | 2 | 121 | 1 | 66 | 1 | 55 | United States | — | 15169 | Google LLC |
| 172.217.168.14 | 12 | 4855 | 6 | 3938 | 6 | 917 | United States | — | 15169 | Google LLC |
| 172.217.168.33 | 6 | 3078 | 3 | 1527 | 3 | 1551 | United States | — | 15169 | Google LLC |
| 172.217.168.34 | 12 | 1048 | 7 | 562 | 5 | 486 | United States | — | 15169 | Google LLC |
| 172.217.168.66 | 13 | 1322 | 8 | 811 | 5 | 511 | United States | — | 15169 | Google LLC |
| 172.217.168.68 | 324 | 139 k | 206 | 125 k | 118 | 14 k | United States | — | 15169 | Google LLC |
| 192.168.178.1 | 6 | 651 | 3 | 432 | 3 | 219 | — | — | — | — |
| 192.168.178.58 | 999 | 617 k | 429 | 52 k | 570 | 565 k | — | — | — | — |
| 192.168.178.86 | 18 | 3590 | 9 | 1560 | 9 | 2030 | — | — | — | — |
| 216.58.215.227 | 12 | 5599 | 6 | 4682 | 6 | 917 | United States | Mountain View | 15169 | Google LLC |

6. **You should be able to see the GeoMap for the connected IP's**


NORTH
AMERICA
6
EUROPE
Atlantic
Ocean
Pacific
Ocean
AFRICA

## Setup TCP Key Settings

- ❑ **Show TCP summary in protocol tree**
  Recommendation: ON
- ❑ Validate the TCP checksum if possible
  **If the system supports TCP Checksum Offloading**
  Recommendation: OFF
- ❑ **Allow subdissector to reassemble TCP streams**
  The most often changed setting! E.g.HTTP-Traffic
  On if you want to reassemble objects.
  Recommendation: OFF (On for HTTP)
- ❑ **Analyze TCP sequence numbers**
  Recommendation: ON
- ❑ **Relative sequence numbers**
  Can be used to match seq numbers in different captures
  Recommendation: ON
- ❑ **Scaling factor to use when not available from capture**
  Can be used if the SYN SYN/ACK packets are missing.
  Recommendation: OFF
- ❑ **Track number of bytes in flight**
  Recommendation: ON
- ❑ **Calculate conversion timestamps**
  Unacknowledged Bytes.
  See also tcp.time_delta
  Recommendation: ON
- ❑ **Try heuristic sub-dissectors first**
  Recommendation: OFF
- ❑ **Ignore TCP Timestamps in summary**
  Recommendation: OFF
- ❑ **Do not call subdissectors for error packets**
  Recommendation: OFF
- ❑ **TCP Experimental Options with a Magic Number**
  Options 253 & 254 (Experiments with TCP [ExID])
  Recommendation: ON
- ❑ **Display process information via IPFIX**
  ?

## Setup Ugly Colors

Colors: https://en.wikipedia.org/wiki/X11_color_names

***Proposed Colors for Errors***

orange (1-2) `#FFA500`
red
red1-4
salmon

***Proposed Colors for Normal Traffic***

steelblue
green
grey

# Wireshark Screen

- Wireshark Command-Line: See page 813

```
⊞ Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)
⊞ Ethernet II, Src: Cisco_8d:a0:a3 (7c:69:f6:8d:a0:a3), Dst: NexcomIn_3c:4a:91 (00:10:f3:3c:4a:91)
⊞ Internet Protocol Version 4, Src: 10.72.96.17 (10.72.96.17), Dst: 10.8.200.2 (10.8.200.2)
⊞ User Datagram Protocol, Src Port: 137 (137), Dst Port: 137 (137)
⊞ NetBIOS Name Service
```

## Protocol Data Unit (PDU)

PDUs are relevant in relation to each of the first layers of the OSI model as follows:

## SPFB

| | |
|---|---|
| Segments | Transport Layer |
| Packets | Network Layer |
| Frames | Data-Link Layer |
| Bits | Physical Layer |

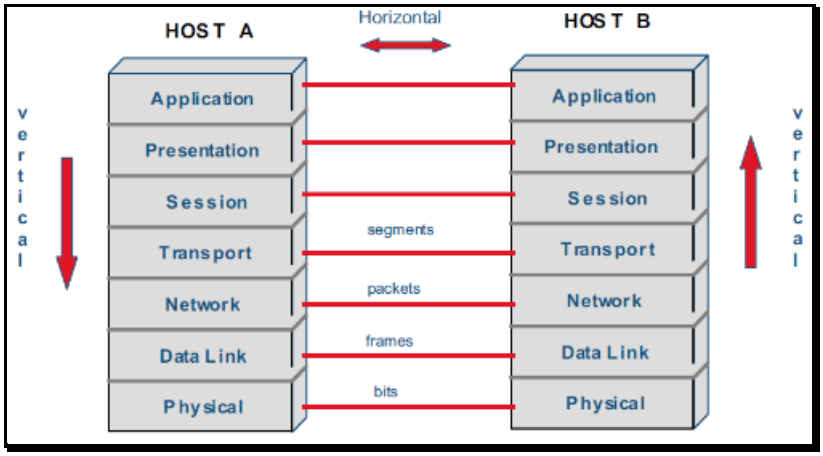



**Figure 2: SPFB**

## Data (OSI-Layer 5, 6, 7)

## Segment (OSI-Layer 4)

- Transport Layer – PDU
- Datagram is the PDU used at Layer 4 (Transport Layer).

## Packets (OSI-Layer 3)

- Network Layer - PDU
- Are the PDU used at Layer 3 (Network Layer).

Packet is the content of a MAC-Frame.
MAC-Sender/ MAC-Receiver (mostly the Default Router)
In WS it's called Ethernet II

***PCAPNG***

| | |
|---|---|
| Start - End Byte: | 76 – 89 (14 Bytes) |
| MAC Dst: | 76 – 81 (6 Bytes) |
| MAC Src: | 82 – 87 (6 Bytes) |

## Frames (OSI-Layer 2)

- Data Link Layer – PDU
- Are the “Protocol Data Unit (PDU) used at Layer 2 (Data Link Layer).
- To be able to go fast you need to have larger frame sizes.
- With larger frame size, and thus larger ***payload*** size, you can have less protocol overhead and are able to achieve higher protocol efficiency.
- In other words, your "***Goodput***" improves with larger frame sizes.

Frame contains the traffic on MAC-Level (MAC-Frame).
It starts with the row: **ETHERNET II**

Size of frameSize, time and time difference.

The typical size of an Ethernet Header is **18 Bytes**

## Bits (OSI-Layer 1)

- Physical Layer – PDU

# Menue Items

## File

## Edit

## View

## Go

## Capture

Capture as close as possible to the complaining user.

- ❑ Install Analyzer on the device in question
- ❑ Hubbing Out
  Works only in Half-Duplex environments
- ❑ Tap In

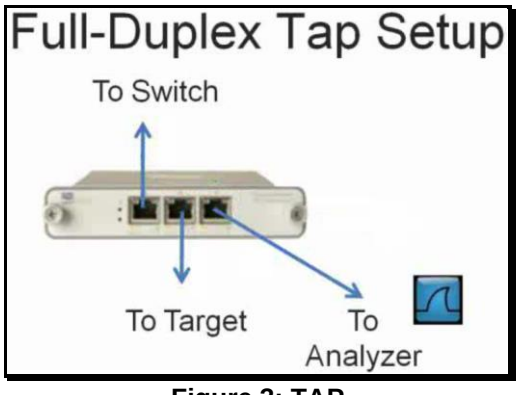



**Figure 3: TAP**

- ❑ Port Spanning
  Some switches discards MAX layer errors!

## Analyze

Analyze → Conversation Filter  See www.profibus.com (Process Automation)

## Statistics

## Statistics → Protocol Hierarchy

## Statistics → IO Graphs

E.g. 1400 Packets/s SAP 10.6.10.62

## Statistics → TCP Stream Graphs

## Statistics → WLAN Traffic

## Telephony

## Internals

# The top 10 reasons for a slow network

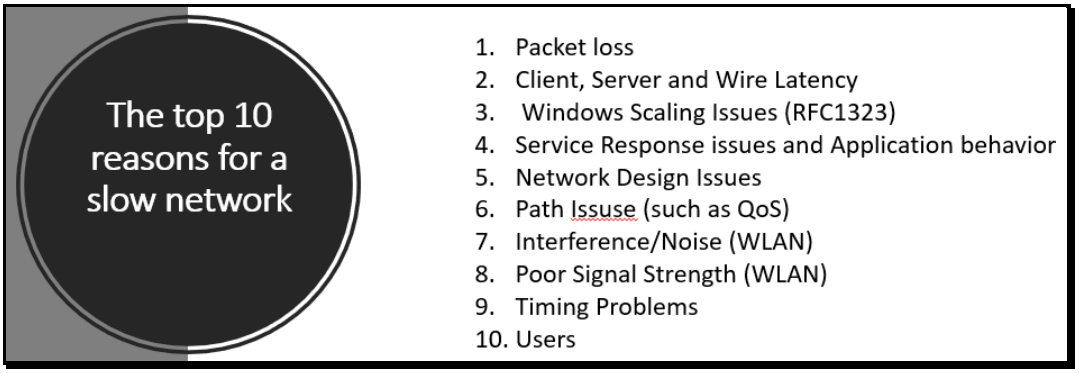



**Figure 4: The top 10 reasons for a slow network**

## No. 1 / Packet-Loss

**Packet loss is the number 1 reason for slow networks!**

Packet loss typically occurs at interconnecting devices such as switches and routers.
Causes:

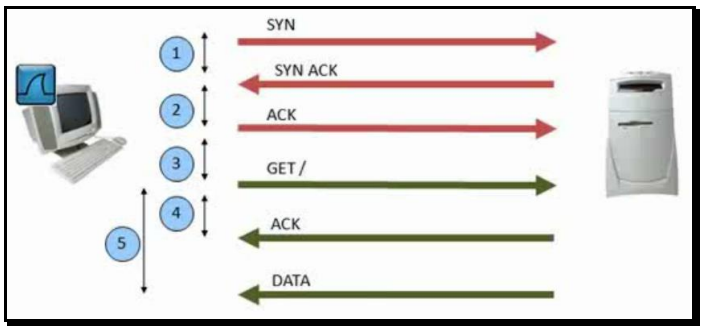
- Retransmissions
- Duplicate Acknowledgments
- **Previous Segment Lost**
- Cuts the congestion window (cwnd) in half
- Negatively impacts throughput

1. Isolate the TCP session you want to analyze
   Statistics → Conversations
2. Add filter “and tcp.analysis.lost_segment”

## No. 2 / Client, Server and Wire Latency

Tools: Latency Calculator [Network Calculator]

Check the first 6 packets to detect latency problems.



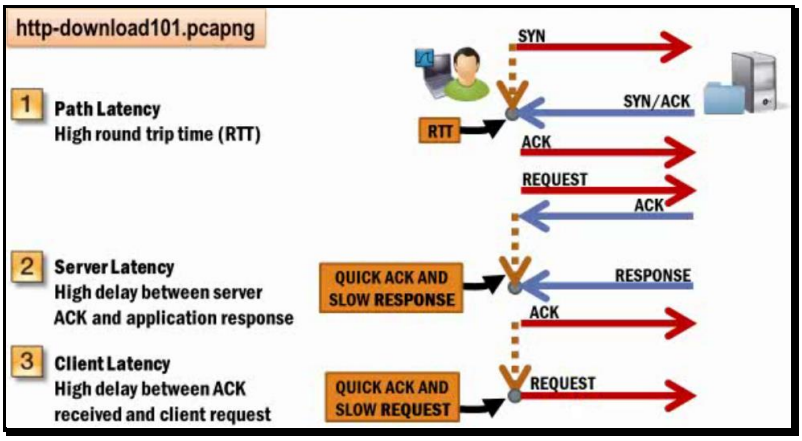




**Figure 5: Latency**

**RTT**
Use the [**Round Trip Time Graph**]
Statistics -> TCP Stream Graph -> Round Trip Time Graph.

**Add TCP-Delta Column**: <Time since pervious frame in this TCP stream>

First rule for analyzing Performance Problems is to watch the **Time Column** (tcp.time_delta)
* You must enable [Calculate Conversation Timestamps]

**Small Payload Size**
If a 500 MB file exchanged in 512 byte segments instead of 1'460 byte segments, the data exchange will require 664'898 more packets to complete the transfer.

Use: **tcp.len** to easily see payload size.

Name resolution faults, weather **DNS**, **LDAP**, **NetBIOS Name Service** or another name resolution process is used, can be significantly detrimental to network performance.

Columns:
- Time
- Info

- ❑ Select a "**Conversation**" first
- ❑ View → Time Display Format → **SECONDS SINCE PREVIOUS DISPLAYED PACKET**

- ❑ The first problem to watch for is delay
- ❑ Check QoS settings
- ❑ Check also the Windows Event Log

## Client-Latency/Service Response Time (SRT)

See also processor latency.

Time between SYN/ACK and ACK
Time between ACK and GET

User idle?

## Server-Latency/Service Response Time (SRT)

See also processor latency.

Time between ACK and DATA

Server-Performance?
Bad application?
Processor overload?

TCP Delta High for <HTTP/1.1 200 OK> !

Tcp.analysis.ack_rtt
Use Advanced IO Graph with calc: AVG(*) to check out.
Use ip.src==[responder of interest]

## Path-Latency/Wire-Latency (RTT)

Time between SYN and SYN/ACK
Time between GET and ACK

QoS?

Base: 3-Path TCP-Handshake / TCP-Connection
[SYN, ACK] Indicates RTT from the capture point, since no payload is included.

Normal Behavior:
Download of Icons (.ico).
Download of pictures (jpg, png …).

# No. 3 / Windows Scaling Issues

- See also: ***TCP Flow Control***
- RFC 1323, 2001

The purpose of ***flow control*** is to provide a way for the receiving device to control the amount of data sent by the sender. The types of flow control are ***buffering***, ***windowing*** and ***congestion avoidance***.

## Congestion Window (cwnd)

The throughput rate of TCP communications is based on the congestion window.
This is the amount of data segments, measured in bytes, that the transmitting machine is allowed to send without receiving an acknowledgment is called a window.

The congestion window shows the number of outstanding/unacknowledged packets.
Packet loss cuts the **cwnd** value in half.
See also the congestion avoidance algorithm.

- The receiver's TCP buffer space advertisement
- The senders transmit buffer capability
- The amount of traffic allowed on the network

## Receive Window / Window Scaling (rwin)

RFC 1323

Window scaling issues are the number 3 reason for slow networks!

The maximum number of bytes a client may receive without acknowledging the sender.

```
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\ interface-name
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Tcp1323Opts
       0 Disables RFC 1323 options
       1 Enables window scaling only
       2 Enables timestamps only
       3 Enables both options
```

**Window Scaling** is defined as a TCP option during the handshake process.
Both sides must support **Window Scaling!**

## No. 4 / Service Response issues and Application behavior

## No. 5 / Network Design Issues

## No.6 / Path Issuse (such as QoS)

## No. 7 / Interference/Noise (WLAN)

## No. 8 / Poor Signal Strength (WLAN)

## No. 9 / Timing Problems

## No. 10 / Users

# STARTING ANALYSIS

- Define your own ***profiles*** to analyze different problems:

HTTP
TCP
HW
…

- Slice big capture files into smaller pieces with EDITCAP.
- The sliced capture files can be marked in windows explorer, drag and drop them into Wireshark, this will merge the files.

***Use specific naming conventions:***

A- Analysis
BU- Bad Ugly
P- Performance
RX- Regex Filters
S- Security
T- Troubleshooting
E- Error

***Categorizing Buttons:***

gui.filter_expressions.label: ----- [P]erformance -----
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: frame

***Possible Causes:***

- ❑ Client Hardware
- ❑ Client Software
- ❑ Client Connection to Switch
- ❑ Switch Configuration
- ❑ Server Connection to Switch
- ❑ Server Hardware
- ❑ Server Software

## General Analysis Steps

- In general, try to keep the ***whole story*** together.
- So it's better to filter on conversations, based on TCP or UDP streams.
- Or even to filter on eth.addr.

❑ Place the analyzer as close as possible to the **user** who is facing the problem.

❑ Check out for **MALFORMED PACKAGES**
You cannot see them in the statistics!

Display Filter: <_ws.malformed>

❑ **Slice very big captures GB**
Slice them by Ethernet conversations first.
xxx – **MAC x.x.x.x.x.x**.pcapng ➔ eth.addr==00:10:db:ff:10:00
…

❑ **Slice big capture**
xxx - _**ORG**.pcapng ➔ The original file
xxx - _**PART**.pcapng ➔ Reduced by protocol-files (see below)
Until: xxx - _**Golden Cut**.pcapng

❑ **Extract the traffic neither UDP nor TCP based**
xxx - **ARP**.pcapng ➔ Filter <arp>
xxx - **ICMP**.pcapng ➔ Filter <icmp>

xxx - **SNMP**.pcapng ➔ Filter <snmp>
xxx - **LLC**.pcapng ➔ Filter <llc> Logical Link Control (STP etc.)
xxx - **VRRP**.pcapng ➔ Filter <vrrp>
xxx - **NDP**.pcapng ➔ Filter <ndp> Nortel Discovery Protocol
xxx - **IGMP**.pcapng ➔ Filter <igmp>

xxx - _**BgNoise**.pcapng
arp || icmp || snmp || llc || vrrp || ndp || igmp || ntp || cdp || ssdp
! (arp || icmp || snmp || llc || vrrp || ndp || igmp || ntp || cdp || ssdp)

- **Extract the STREAMS of interest UDP or TCP based**
  xxx – **TCP stream X.**pcapng ➔ Statistics / Conversation
  xxx – **UDP stream X.**pcapng ➔ Statistics / Conversation

- **Statistics ➔ Protocol Hierarchy**
  Search for unusual traffic

- **Extract the Protocols of interest**
  xxx - **DHCP**.pcapng ➔ Filter <udp.stream==x> or <bootp>
  xxx - **DNS**.pcapng ➔ Filter <dns>
  xxx – **IPv6**.pcapng ➔ Filter <ipv6>
  xxx - **PNMRP**.pcapng ➔ Filter <pn_mrp> Profinet Traffic
  xxx - **SSDP**.pcapng ➔ Filter <ssdp>
  xxx – **HTTP.**pcapng ➔ It's better to use port based filter like <tcp.port=80>
  xxx - **<protocol>**.pcapng ➔ Extract other disturbing traffic

- **Expert Information's**
  Check Errors, Warning an Notes in the expert info's
  See tcp.analysis.flags

  ECE ECN-Echo (Explicit Congestion Notification
  URG Urgent data

- **Check the TCP Flow**
  Statistics → TCP StreamGraphs → Time-Sequence (Stevens)
  Statistics → TCP StreamGraphs → Time-Sequence (tcptrace)

You may export data to excel: File → Export Packet Dissections

**Maximum Transmission Unit (MTU)**

- See: RFC 791 / IP
- The maximum sized datagram that can be transmitted through the next network is called the maximum transmission unit (MTU).

**Maximum Segment Size (MSS)**

- It is used in the first two packets of the three –way handshake process.
- The purpose is to define what segment size the host supports.
- TCP MSS refers to the amount of data in the segment (without the inclusion of any headers)
- It is the amount of data that can be contained in a single TCP segment

Typical size is **1460 Bytes**

Frame.Len is equal Packet Length (Bytes)

**Maximum Segment Lifetime (MSL)**

- See: RFC 793 / TCP
- MSL is less than 4.55 hours

**ISN – Initial Sequence Number**

**ISS - Initial Send Sequence number**

- The initial send sequence number (ISS) is chosen by the data sending TCP.

**IRS - Initial Receive Sequence Number**

- The initial receive sequence number (IRS) is learned during the connection establishing procedure.

**Senders Maximum Segment Size (SMSS)**

- ???

**Citrix**
Be aware Citrix is using much more BW if used for Full HD transmissions as usually expected!

Name resolution: You may use hosts file

## How to capture on the wire

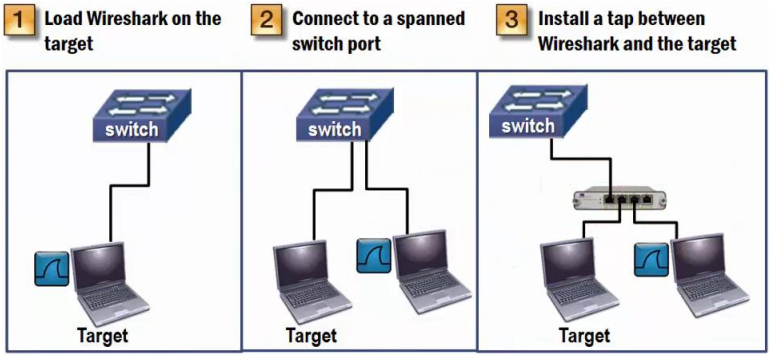



**Figure 6: How to capture**

How to capture on the Wireless Networks (802.11):

- Use **AirPcap** in monitor mode.

## Security Tasks for Network Analysts

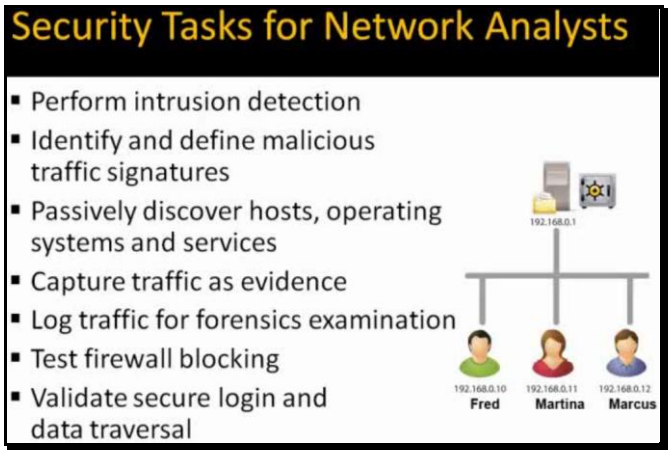



**Figure 7: Security Tasks for Network Analysts**

# Time Display Format

Wireshark's epoch time is based on the time since **January 1 00:00:00 of 1970** (UNIX Time).
Coordinated Universal Time (UTC)

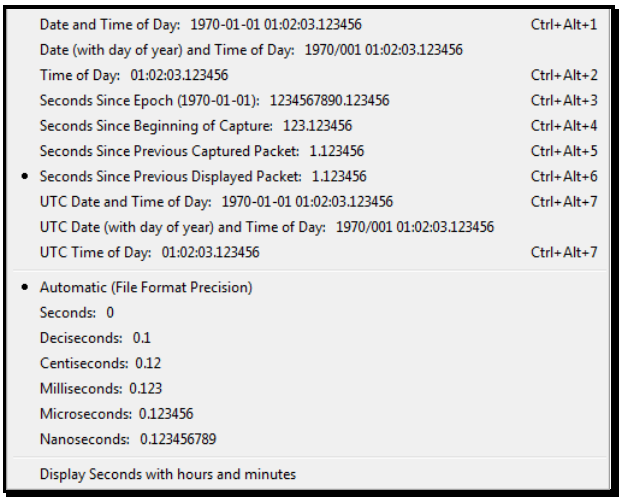



**Figure 8: Time Display Formats**

Seconds Since Previous Displayed Packet ➔
Seconds Since Previous Captured Packet ➔ Delta Time

Ctrl + Alt + 1 Date and Time of Day
Ctrl + Alt + 6 Seconds Since Previous Displayed Packet

# BASELINING

Create baseline measurements during normal state of the network.
You cannot identify *unusual traffic* unless you are aware of the *usual traffic* on your network.
The values can be used for analysis during abnormal behavior and for planning purposes (trends).

Baselining **Bot up Sequences** is important since this sequence sets up the client's general configuration and performance for the remaining up time.

Statistics → Protocol Hierarchy

## Create following baselines:

Baseline Broadcast and Multicast Type and Rates
Baseline Protocols and Applications
Baseline Boot up Sequences
Baseline Login Sequences
Baseline Traffic during Idle Times
Baseline Application Launch Sequences and Key Tasks
Baseline Web Browsing Sessions
Baseline Name Resolution Sessions
Baseline Throughput Tests
Baseline Wireless Connectivity
Baseline VoIP Communications
Baseline Connectivity tests through the network
Baseline Application shut-down
Baseline Logoff Sequences
Baseline All typical traffic

Die Aufzeichnung der folgenden Parameter hat sich bewährt:

| Parameter | verwendbar für |
|---|---|
| SysUpTime | Erkennung von Reboots |
| CPU- und Speicherauslastung | Trendanalyse, Erkennung von DoS-Attacken |
| Interface-Auslastung | Trendanalyse |
| Interface-Fehler | Erkennung von Hardwarefehlern (Leitung, NIC) |
| Round Trip Time | Erkennung von Leitungsproblemen und Routingänderungen |
| Protokollverteilung | Erkennung von DoS-Attacken |
| Größenverteilung | Analyse von Performanceproblemen |
| Top Talker | Trendanalyse, Erkennung von DoS-Attacken |

Broadcasting: Who is broadcasting
What Application is using broadcast
Typical broadcast rate in packets per second

Multicasting: Who is multicasting
What Application is using multicast
Typical multicast rate in packets per second

## Background Noise (_BgNoise)

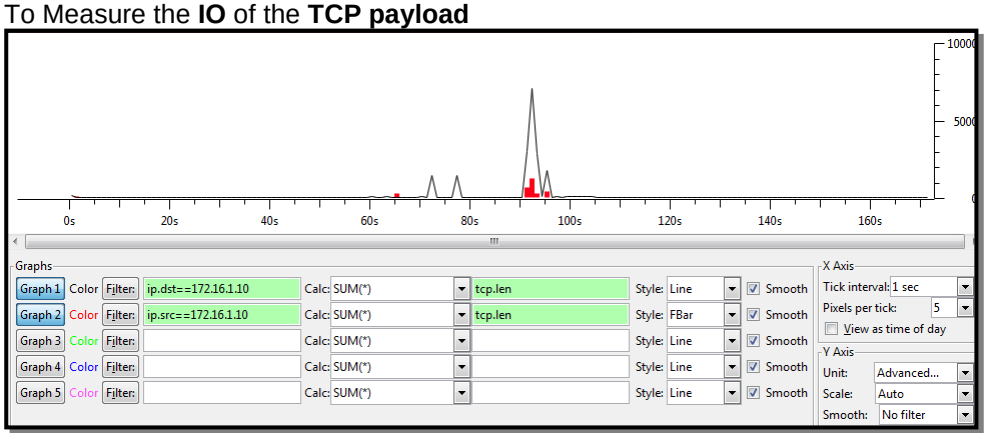
ARP ➔ Broadcast = To DG
LLMNR ➔ <host> = <FQDN> / <IP>
NBNS ➔ <host> = <FQDN> / <IP>

# IO Graphs

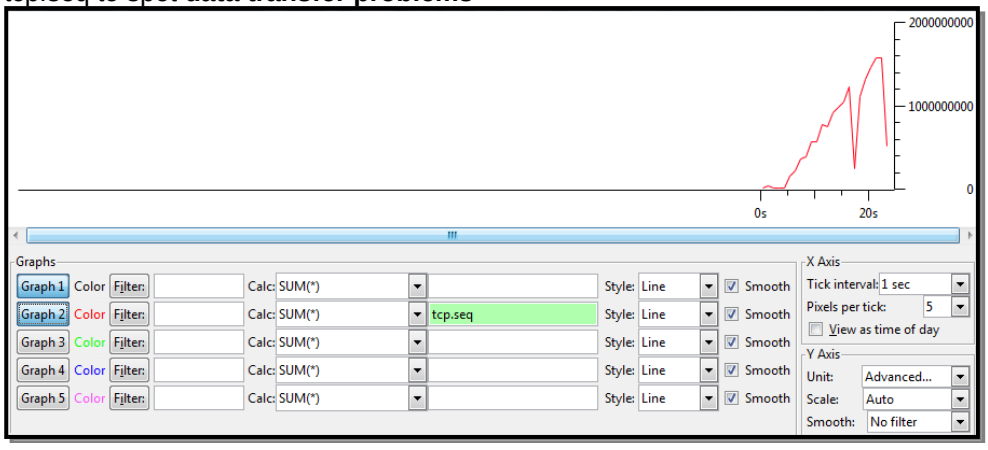
Statistics → IO Graphs

Use: Bits/Tick

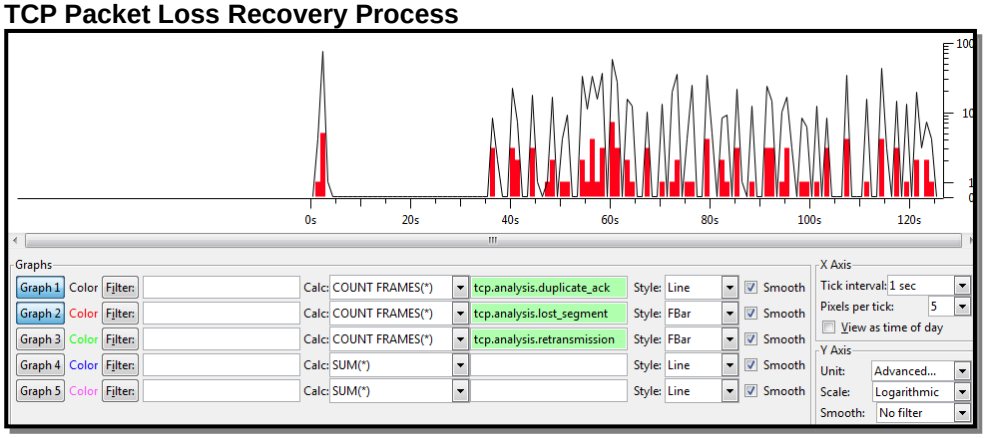
To Measure the **IO** of the **TCP payload**

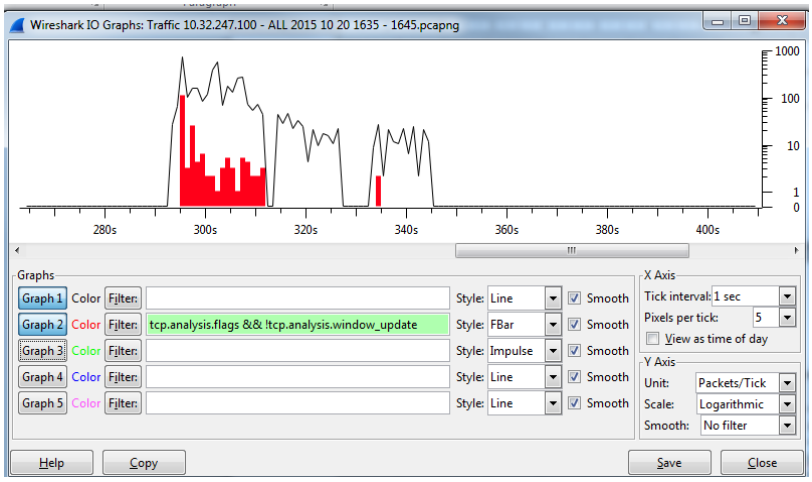


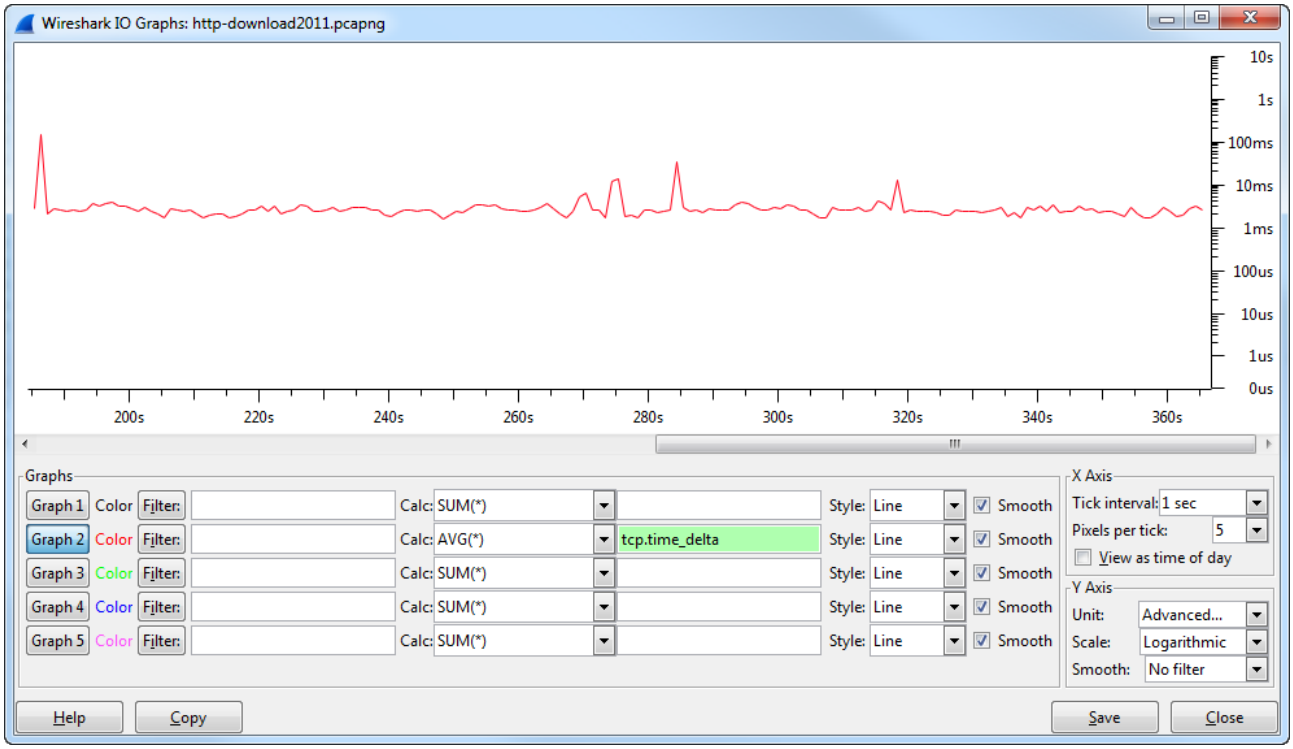

tcp.seq to spot **data transfer problems**



**TCP Packet Loss Recovery Process**




# Network Forensics

malware-traffic-analysis.net
emergingtrhreads.net

Malware patterns

*Useful Display Filter:*

| | |
|---|---|
| My Filter [**S-ThisPgm**]: | frame contains “This program cannot be run in DOS mode” |
| My Filter [**S-PWD**]: | frame contains “pass” |
| My Filter [**S-HOST**]: | http.host contains “.ru” |
| My Filter [**S-HOST!com..**]: | !http.host matches “(?i)\.(com\|org\|net)$” && http.host |
| My Filter [**S-TFTP**]: | tftp contains 00:01:02 |
| My Filter [**S-ICMPScan**]: | icmp.type==3 && icmp.code==3 |
| My Filter [**S-ICMPBlHole**]: | icmp.type==3 && icmp.code==4 |
| My Filter [**S-ICMP3-5**]: | icmp.type==3 && icmp.code==5 |
| My Filter [**S-SYNnoMSS**]: | ((tcp.flags==0x0002)) & !(tcp.option_kind==2) |
| My Filter [**S-IPScan**]: | has no INFO! |
| My Filter [**S-EtterCap**]: | IPv4 Identification: 0xe77e |

***REGEX:*** www.regular-expressions.info
www.regexbuddy.com

| | |
|---|---|
| **.** | Indicates any character |
| **\x** | Precedes hex value<br>e.g \xFC |
| **[]** | Creates a character class |
| **(?i)** | Case Insensitivity |
| **^** | Anchor, Start at the beginning<br>Inside a character class it means **not** |
| **{}** | Interval Quantifier |

**Consider case sensitivity**

```
ftp.request.arg matches "anonymous"
ftp.request.arg matches "(?i)anonymous"
```

**Consider variable characters**

```
frame matches "building[Aa]eng"
frame matches "building[AaBb]eng"
frame matches "(?i)(cat|dog)"
```

```
ftp.request.arg matches "ne.r"
```

“.” indicates any character except a carriage return or line feed

```
ftp.request.arg matches "ne..r"
```

Now we’re looking for any two characters between me and r

```
ftp.request.arg matches "ne.{1,3}r"
```

{#,#} indicates minimum and maximum number of repeating characters

```
ftp.request.arg matches "ne.r$"
```

“$” indicates end of line or field

```
ftp.request.arg matches "^ne.r"
```

“^” indicates end of line or field

# Security Problems

www.netresec.com/?page=PcapFiles
wiki/wireshark.org/Tools
HTCIA  Private Investigators!
National Vulnerabilities Database

- Unassigned MAC Addresses are referred as "**dark MAC addresses**".
    - Such traffic are indications of blind discovery processes to find hosts on the network.
- Unassigned IP Addresses are referred as "**dark IP addresses**".
    - Such traffic are indications of blind discovery processes to find hosts on the network.
- Validating that applications are using **encryption for password** setting and password input is an important step in analyzing network security.

**Denial of service Attacks (DOS)** are designed to make a resource unavailable to others.

- I you see SYN packets coming from different IP addresses, check the TTL.
- If the ***TTL value*** is all the time the same, it's obvious this comes from one source!

**Flooding** can be used to saturate a network link.
- Consider, that a configuration error could be the cause of the flood.
- Check the expert infos -> Warnings to see Illegal Source MAC warnings.

**ARP Poisoning** is typically used for man-in-the-middle attacks. The attacker generates a series of AARP packets with false information that alters the ARP tables of the victim hosts. (See Ettercap and "Cain and Abel")

**MAC Address Spoofed**
Source MAC address cannot have the Individual Group (IG) bit set to 1
Source Address: **1**9:19:19:19:19:19  (Cannot be odd)
Addresses starting with odd are either Broadcast or Multicast
eth.ig ==0  ➔ Only Multicast communication
Addresses starting with even addresses are Unicast
eth.ig ==1  ➔ Only Unicast communication

Check the first 3 Bytes of the MAC-Address (OUI)

**IP Address Spoofed**
Source MAC address cannot have the Individual Group (IG) bit set to 1

**Network Flood**

**Man-in-the-Middle Attack**

Spot **TCP Splicing**

- Detect ARP Scans
- Detect ICMP Ping Sweeps
- Detect Various Types of TCP Port Scans
- Detect TCP Half-Open Scan (Stealth Scan)
- Detect TCP Full Connect Scan
- Detect Null Scans
- Detect Xmas Scan
- Detect FIN Scan
- Detect ACK Scan
- Detect UDP Port Scans
- Detect IP Protocol Scans
- Detect Malformed packets
- Detect Invalid addresses
- Detect Privileged requests
- Detect Clear text passwords
- Detect Phone home behavior
- Detect Atypical protocols and applications

## Remote Procedure Call (RPC)

See: trace31.pcap (e.g. Blaster)

Be careful with this traffic!

- Blaster listens on **port 4444**

## Honeypot

- Define a server in the server-farm as Honeypot.
- Capture vital data there.

## Watch for reconnaissance processes

Reconnaissance processes appear before the breach.

### IP Scans

IP scans will change the protocol header.
Responses are usual icmp.type /-code 3/2.

### TCP Scans

- SYN followed immediately by an RST indicates a TCP-Scan.

My Filter [**S-TCPScan**]: ((tcp.flags.reset==1) && (tcp.seq==1)) && (tcp.ack==1)

### UDP Scans

- They are not easy to detect.
- If a Server answers with icmp.type / - code 3/3, this means either the port is not open or a FW is blocking the traffic.

### OS Fingerprinting

- TCP/IP probes.
- ***Passive OS Fingerprinting*** can detect ***botnet's***.

The attacker is preparing a OS specific attack.
Icmp-type 13, 15, 17

My Filter [**S-OSFinger**]: icmp.type==13 || icmp.type==15 || icmp.type==17

### Address Scans

- The attacker is scanning IP addresses or MAC addresses.
- The network should never see unknown IP addresses or unknown MAC addresses.

### Application Scans

- The attacker is sending unusual commands to the server.
- The answers are giving an idea of what application is running on the system.

### Macof Scans

The SYN packets have an unusual small window size!
Search for SYN packets with a window size less than 1000.

My Filter [**S-Macof**]: (tcp.flags==0x02) && (tcpwindow_size <1000)
gui.filter_expressions.label: S-Macof
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: (tcp.flags==0x02) && (tcp.window_size<1000)

## Packet Handcrafting

Tool: Packet Builder ➔ www.colasoft.com
NtScantools

# TCP-Problems

RFC 793, 2001, 2018, 2582
RFC 2581 TCP Congestion Control

***Rule of Thumb:*** ***No Packet Loss in DC!***

**PCAPNG**
Start - End Byte: 110 – 129 (20 Bytes)

**netstat –a** to display the current status of TCP connections (see page 461/WCNA)

- The ***TCP header*** is 20 to 60 bytes long.

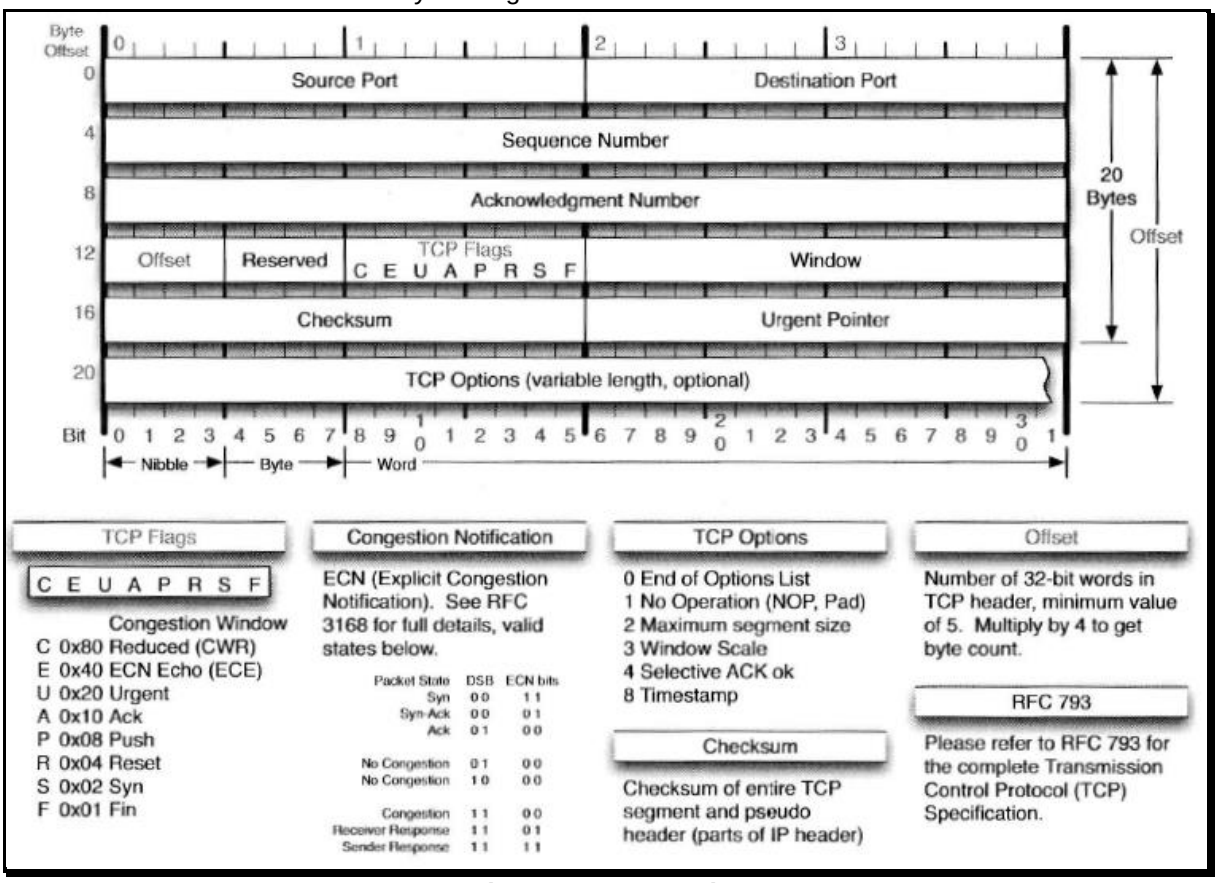



**Figure 9: TCP Header**

# ICMP

RFC 792

See: www.iana.org/assignments/icmp-parameters

- IP Header Protocol Field = 1
- The ICMP messages typically report errors in the processing of datagrams.

Constant Fields:

- ❑ Type
- ❑ Code
- ❑ Checksum

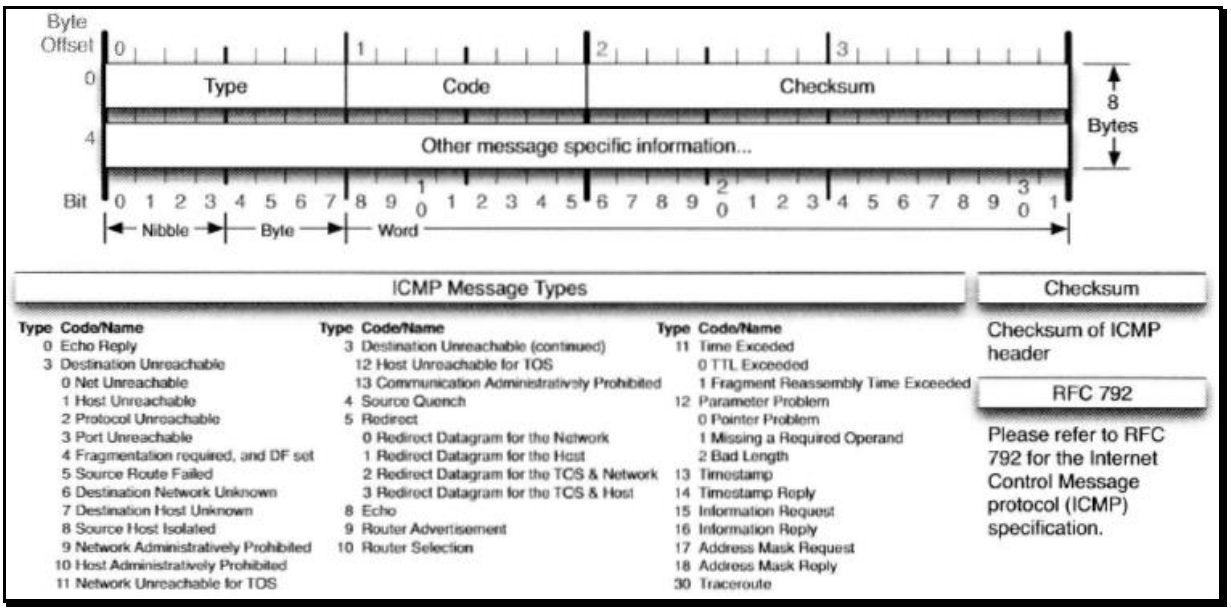



**Figure 10: ICMP Header**

## Exceeded Messages:

- Type = 11

## Echo Messages:

- Type = 0 for echo reply message
- Type = 8 for echo message
- Test end-to-end connectivity.
- The data received in the echo message must be returned in the echo reply message.

## Redirect messages:

- Type = 5
- Routers to measure better paths.
- If this packet is not sent by a router, it should be considered as suspect.
- The redirect message updates the router cache of the client!

## Destination Unreachable messages:

- Type = 3
- A large number of these packets could indicate an unsuccessful UDP port scan is underway or a service is not running properly.
- Maybe the port is firewalled. This message is being generated by the firewall.

## Parameter Problem messages:

- Type = 12

## Source Quench messages:

- Type = 4
- The source quench message is a request to the host to ***cut back the rate*** at which it is sending traffic to the internet destination.

## Timestamp Reply messages:

- Type = 13 for timestamp message
- Type = 14 for timestamp reply message

## Information Request messages:

- Type = 15 for information request message

- Type = 16 for information reply message

## TRACERT

There are three variations of tracert:

- **ICMP-Based**
  ICMP must be supported on the target system.
- **TCP-Based**
  The target system must answer to the SYN with either RST or SYN/ACK
- **UDP-Based**
  The target system will answer with an ICMP Type 3/Code3

## Nagle Algorithm

RFC 896

- The Nagle algorithm can slow down network communications for small data transfers and is disabled by many TCP implementations.

## Explicit Congestion Notification [ECN]

RFC 3540

## What Triggers TCP Retransmission (Note)

- Retransmissions are the result of packet loss and are triggered when the sender's "TCP retransmission timeout (RTO)+ timer expires or a receiver sends "Duplicate Acknowledgements" to request a missing segment.

After ***3 Duplicate Ack's*** the device is supposed to resend the packet.

If a TCP segment contains data and it uses the same sequence number as a previous packet, it must be a **TCP retransmission** or a **fast retransmission**.

Filter: tcp.analysis.retransmission

A fast retransmission is if the packet is resent within **20ms**.

***Possible Causes:***

- ❑ Cabling issues
- ❑ Switch/Client Duplex Mismatches
- ❑ Switch/Server Duplex Mismatches
- ❑ Server, Client and Switch Hardware issues

## What Triggers Previous Segment Not Captured (Warning)

- When an expected sequence number is skipped, Wireshark indicates a previous segment has been lost on the packet immediately following the missing packet in the stream.

## What Triggers ACKed Lost Packet (Warning)

- When Wireshark detects an acknowledgement, but it has not seen the packet that is being acknowledged, an ACKed Lost Packet warning is triggered.
- This often indicates, that the network supports multiple paths (**asymmetrical routing**) or the process is faulty.

## What Triggers Keep Alive (Warning)

Each side of a TCP connection maintains a keep alive timer. When the **keep alive timer expires**, a TCP host sends a keep alive probe to the remote host. If the remote host responds with a keep alive ACK (or any TCP packet, for that case), it is assumed, the connection is still valid.

## What Triggers Duplicate ACK (Note)

A TCP host that supports a feature called **Fast Recovery** will continue to generate Duplicate ACKs requesting the missing segment, When the host sending the TCP segments receives three identical ACKs (the original ACK and two Duplicate ACKs), it assumes there is **packet loss** and it resends the missing packet – regardless of whether the RTP expired or not. A high number of Duplicate ACKs may be an indication of **high latency** between TCP host as well as **packet loss**.

## What Triggers Zero Window (Warning)

- When a receiver has no receive buffer space available, it sends Zero Window packets indicating the TCP window size is zero.
- This, in effect, shuts down data transfer to the receiver.
- The data transfer will not resume until that receiver sends a packet with a window size sufficient to accept the next amount of queued data from the sender, which is usually **1 MSS**.
- Ultimately the cause of a Zero Window condition is an application that is not pulling data out of the receive buffer fast enough.
- This might be caused by an **underpowered system**, running too many CPU-intensive applications on the host or a dog-slow application.
- Alternatively, the **starting window size** may be too small.

WXP Check the registry Keys
W7 Disable Auto Tuning

## What Triggers Zero Window Probe (Note)

- A Zero Window Probe packet may be sent by a TCP host when the remote host advertises a window size of zero.

## What Triggers Zero Window Probe ACK (Note)

- Response to the Zero Window Probe packet.

## What Triggers Keep Alive ACK (Note)

- Keep Alive ACKs are sent in response to a Keep Alive.
- If **no Keep Alive ACKs** are sent in response to **TCP Keep Alive** packets, the connection is considered broken.

## What Triggers Out-of-Order (Warning)

- An out-of-order packet contains a lower sequence number than a previous packet.
- Causes: Multiple paths (asymmetrical routing) or queuing devices which are reordering the forwarded packets.

## What Triggers Fast Retransmission (Warning)

- A Fast Retransmission occurs within ***20ms of a Duplicate ACK***.

Filter: tcp.analysis.retransmission

The difference between Fast Retransmission and Retransmission is who noticed the packet loss first.
Find the location of the packet loss to fix the problem.

## What Triggers Window Update (Chat)

A Window Update packet contains no data, but indicates that the sender's TCP window size value has increased.
This are **good packets**. AQ client just advertised a **larger receive buffer space** indicating an application just picked up some data from the receive buffer. These packets are the only recovery for a Window Zero condition and do not require action.

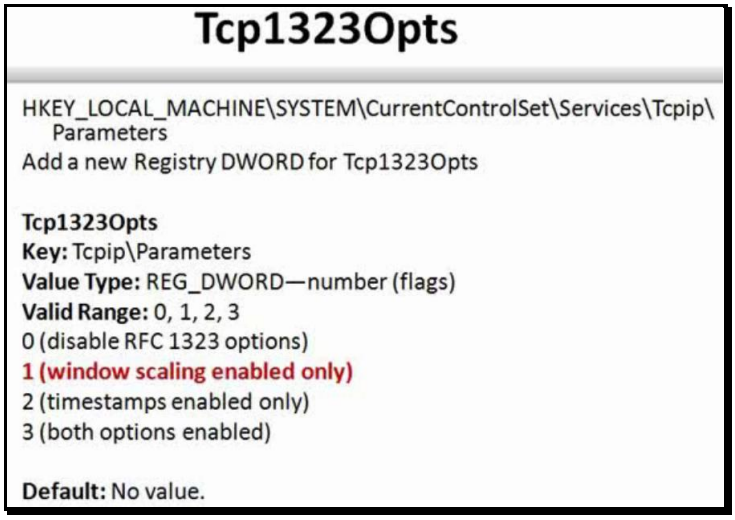
## What Triggers Window is Full (Note)

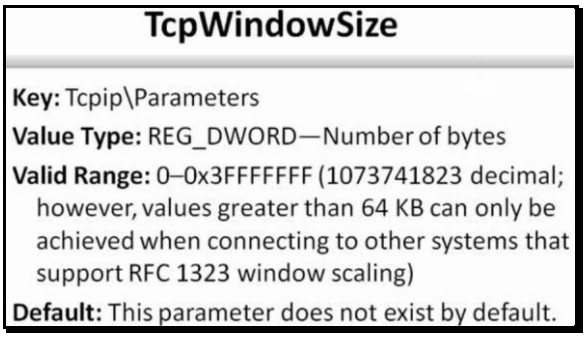
When a data packet is sent that will fill up the remaining buffer space.

This packet itself will not have the Window size value of 0. This packet is an indication that a window size value of 0 may come from the **other side** if their receive window size is not updated.
Focus on the **destination IP host** for Window is Full packets. That destination IP address indicates the device that is having issues with the application not picking up data fast enough from its receive buffer.

***Check:***

- ❑ Tcp1323Opts
- ❑ TcpWindowSize




## What Triggers TCP Ports Reused (Note)

This expert notification is triggered when a new TCP session begins using the same IP address and ort number combination as an earlier conversation in the trace file.
This is often seen during a vulnerability scan or reconnaissance process. These packets should be investigated to see if there is a security issue to address.

## What Triggers 4 NOP in a Row (Warning)

RFC 793

NOP = No Operation (0x90)

This warning is triggered when Wireshark sees an illogical pattern **01:01:01:01** in a TCP SYNor SYN/ACK packet. A NOP is used to pad a TCP option so it ends on a 4-byte boundary.
Typically, a router issue. router bug, poor router configuration or inability to support a particular TCP option.

Used as padding for other options in TCP communications.

**NOP Problem**

You should **never see 4 NOP's in a row!** This would indicate that a device ion the way stripped of a TCP Option.

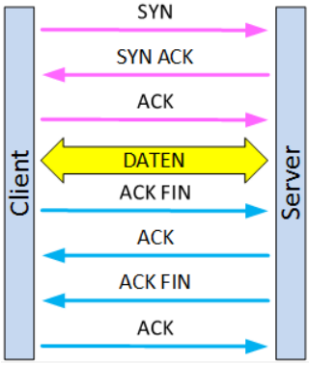
Filter: **tcp.options contains 01:01:01:01**

# TCP-Handshake / TCP-Connection

- RFC 793

Establishing: **SYN, SYN/ACK, ACK**
Terminating: **FIN, ACK/FIN, ACK**

Remember the Phantom Byte in the handshake process SYN and SYN/ACK.
The typical size of an IP Header is 20 Bytes




***Three-Way Handshake***

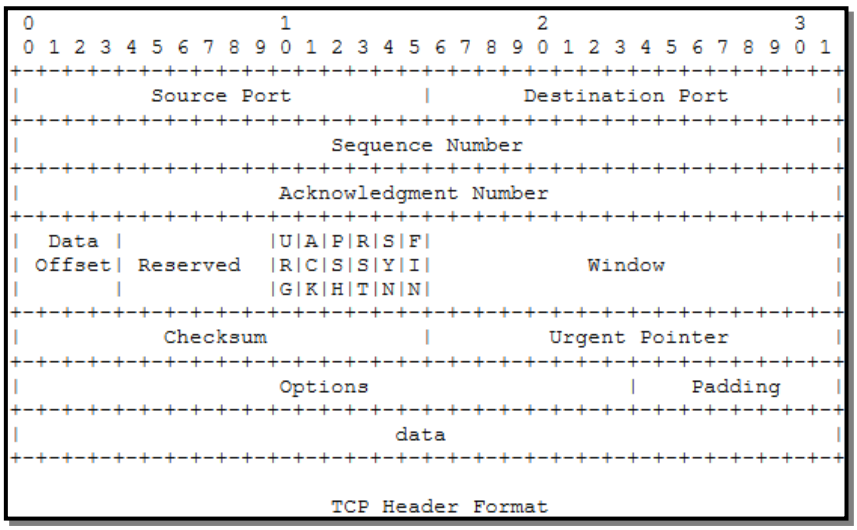
```
A --> B SYN my sequence number is    X
A <-- B SYN my sequence number is    Y
        ACK your sequence number is  X
A --> B ACK your sequence number is  Y
```

```
    TCP A                                                TCP B

1.  CLOSED                                               LISTEN

2.  SYN-SENT    --> <SEQ=100><CTL=SYN>               --> SYN-RECEIVED

3.  ESTABLISHED <-- <SEQ=300><ACK=101><CTL=SYN,ACK>  <-- SYN-RECEIVED

4.  ESTABLISHED --> <SEQ=101><ACK=301><CTL=ACK>       --> ESTABLISHED

5.  ESTABLISHED --> <SEQ=101><ACK=301><CTL=ACK><DATA> --> ESTABLISHED

        Basic 3-Way Handshake for Connection Synchronization
```

**SRC: [SYN]** Check:

| | |
|---|---|
| Seq=x | Initial Sequence Number (ISN) |
| **Win=x** | Window Size, 16 bit-field, max. 65535 Bytes |
| Len=x | |
| **MSS=x** | Maximum Segment Size |
| WS=x | Window Scaling, x-times Win |
| SACK_PERM=x | Selective ACK permitted RFC 2018 |

**DST: [SYN, ACK]**
**SRC: [ACK]**




## SYN (Client)

TCP Header
- Window size: 65535

Options:
- Maximum sement size: 1460 bytes
- Window scale: 2 (multiply by 4) ➔ Window size 65535 * 4 = 262'140
- SACK permitted ➔ This is good!

## SYN/ACK (Server)

TCP Header
- Window size: 5840

Options:
- Maximum sement size: 1460 bytes ➔ Fits with client!
- SACK permitted ➔ Fits with client!
- Window scale: 7 (multiply by 128) ➔ Window size 5840 * 128 = 747'520 (Receive buffer size)

## ACK

**SACK** support must be established by both hosts in the TCP options during the handshake process.
See SYN/ACK Tcp Frame → Options

If a TCP [SYN] does not receive any response:
a) Then SYN packet didn't make it to the target
b) The SYN/ACK packet didn't make it back

If a TCP [SYN] receives a Reset [RST, ACK] respnse
- The port is closed
- TCP scan is underway

**Filter for [SYN] packages without [SYN, ACK] response.**

Since [SYN] packages without answered [SYN, ACK] packages are followed by a retransmission, the following filter can be used.

**tcp.flags ==2 && tcp.time_relative>0**
(Enable Conversation Timestamps)

==========

**Important values:**

Sequence number → Acknowledgment number → Window size value [tcp.window_size]

```
  Sequence Number In          (tcp.seq)
+ Bytes of Data Received
-------------------------
= Acknowledgment Number Out
```

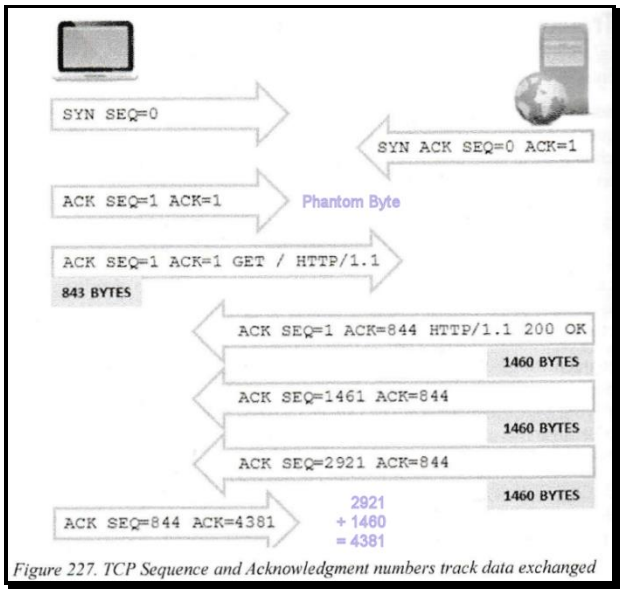


*Figure 227. TCP Sequence and Acknowledgment numbers track data exchanged*

Analyze the following values:
tcp.window_size
tcp.window_size_value

## TCP Options

See: www.iana.org/assignments/tcp-parameters/tcp-parameters.xhtml
IANA.org Protocol Assignments
Search for TCP Options

**Kind**
**34** **TCP Fast Open** (**TFO**) is an extension to speed up the opening of successive Transmission Control Protocol (TCP) connections between two endpoints.

| Kind | Length | Meaning | Reference |
|---|---|---|---|
| 0 | - | End of Option List | [RFC793] |
| 1 | - | No-Operation | [RFC793] |
| 2 | 4 | Maximum Segment Size | [RFC793] |
| 3 | 3 | WSOPT - Window Scale | [RFC1323] |
| 4 | 2 | SACK Permitted | [RFC2018] |
| 5 | N | SACK | [RFC2018] |
| 6 | 6 | Echo (obsoleted by option 8) | [RFC1072] |
| 7 | 6 | Echo Reply (obsoleted by option 8) | [RFC1072] |
| 8 | 10 | TSOPT - Time Stamp Option | [RFC1323] |
| 9 | 2 | Partial Order Connection Permitted | [RFC1693] |

**Figure 11: TCP Options**

==========

Investigate systems which are modifying the TCP-Header such as **F5-Loadbalancer**.

==========

**Keep an eye on Jumbo Frames**

==========

[ACK] The sender of ACK is acknowledging SYN packet was received.
netstat –a = ESTABLISHED

[FIN] Finish the connection gracefully with both sides shutting it down in an

ordered manner. No more data from sender,
This is the ***most common way to end a session***.
Each side of the communication will send a FIN flagged packet.
RFC 793

[FIN, ACK] netstat - = CLOSE_WAIT (Close) FIN-WAIT-1
= TIME_WAIT (Close) LAST-ACK

[SYN] Synchronize or begin the connection.
SYN packets are only used during the initial three way TCP handshake.
Synchronize sequence numbers
- Start of new TCP connection
Interesting value: **Connection Request Rate**
TCP-Header: ToS (type of service
13th Byte of TCP-Header

[SYN, ACK]
[SYN, ACK, ECN]
[SYN, ECN, CWR] CWR = Congestion Window Reduced

[PSH] The push bit is set by the programmer.
Request the data be immediately pushed to the application.
The push flag indicates to the other connection that it should process this data as soon as possible instead of queuing it in the TCP buffer.
Push Function indicates data received and forward directly to the application
TCP-Header: ToS (type of service
13th Byte of TCP-Header

[PSH, ACK] Indicates that a host is acknowledging having received some previous data and also transmitting some more data.

[**RST**] As a ***general rule***, reset (RST) must be sent whenever a segment arrives which apparently is not intended for the current connection.

Problem occurred and the sender wants to reset the connection.
Reset the connection – Improper End of TCP connection -
Reset tears down the connection with no further exchange of packets, usually used when a host denies an attempted connection, or the client is terminated abruptly, like closing a web browser.
RST always shows tcp.window_size==0
Causes an ***immediate*** and ***abrupt session termination***.

TCP-Header: ToS (type of service
13th Byte of TCP-Header
Kerberos: Doesn't follow RFC and creates RSTs

[RST, ACK] End of TCP connection
[SACK] Selective Acknowledgment

[TCP ACKed Lost Packet] Packet ACKed but original Paket missing.
[TCP ACKed unseen segement]

[TCP Dup ACK 999#99] See also fast retransmission

[TCP Retransmission] The sender didn't receive an ACK and assumes, the packet was lost.
The sender retransmits than the Packet.
[TCP Fast Retransmission] **Fast Retransmit** is an enhancement to TCP which reduces the time a sender waits before retransmitting a lost segment.
Resend of packet within 20 ms
[TCP Keep-Alive]
[TCP Out-Of-Order] Out-of-order delivery can be caused by packets following **multiple paths** through a network, or via parallel processing paths within network equipment that are not designed to ensure that packet ordering is preserved.
[TCP Previous segment not captured]

| | |
|---|---|
| | TCP-Sequenznumber inconsistent<br>Error caused usually by switches and routers. |
| [TCP Reused Ports] | To be investigated, maybe Network scanner |
| [TCP Retransmission] | |
| [TCP Spurious Retransmission] | |
| [TCP Window Full] | Before “Zero Window” state. |
| 1. "TCP Receive Buffer" erhöhen<br>2. Mehr Memory<br>3. CPU | |
| [TCP Zero Window] | |
| [TCP Zero Window Probe] | To check the receive window of the host. |
| [URG] | Identifies that priority data transfer is invoked.<br>The urgent flag is used to process data that must be processed immediately (such as a character in a telnet session) and uses a pointer to indicate the last byte of ***urgent data***. |
| GET | See <Info-Field> |

# TCP-Terminating

FIN, ACK/FIN, ACK

```
    TCP A                                                TCP B

1.  ESTABLISHED                                          ESTABLISHED

2.  (Close)
    FIN-WAIT-1  --> <SEQ=100><ACK=300><CTL=FIN,ACK>  --> CLOSE-WAIT

3.  FIN-WAIT-2  <-- <SEQ=300><ACK=101><CTL=ACK>      <-- CLOSE-WAIT

4.                                                       (Close)
    TIME-WAIT   <-- <SEQ=300><ACK=101><CTL=FIN,ACK>  <-- LAST-ACK

5.  TIME-WAIT   --> <SEQ=101><ACK=301><CTL=ACK>      --> CLOSED

6.  (2 MSL)
    CLOSED

                        Normal Close Sequence
```

# UDP-Problems

- Common applications that use UDP are DHCP/BOOTP, SIP, RTP, DNS, TFTP and various streaming video applications.
- One potential problem is blocked traffic based on the UDP port number value.

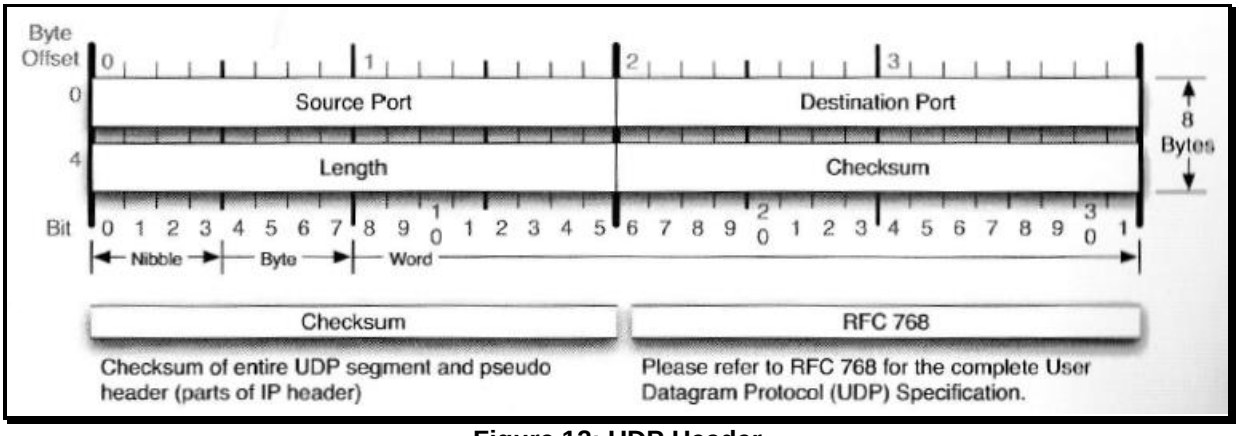



**Figure 12: UDP Header**

# IPv4-Problems

RFC 791 DoD Standard Internet Protocol
RFC 1191 Path MTU Discovery (Fragmentation)

Protocol Type: 0x0800

Wireshark uses by default the **DiffServ** (Differentiated Services) interpretation in the IP header instead of the **Type of Service** (TOS) interpretation. This can be changed in the settings, by disabling **Decode IPv4 TOS field as DiffServ field** in the IP preferences.

IP is connectionless and unreliable, providing best effort delivery of datagrams between IP hosts.
IP-Problems typically deal with fragmentation, unusual IP addresses and excessive broadcasts.

Duplicate addresses
Not allowed addresses

Typical IPVv4 Address Header = 20 bytes

Time to Live (TTL)

**Broadcast/Multicast Traffic**

| | |
|---|---|
| 255.255.255.255 | General Broadcast |
| 10.2.255.255 | Subnet Broadcast |
| 224.x.x.x – 239.x.x.x | Multicast |

| | |
|---|---|
| 224.0.0.1 | IGMPv3 |
| 224.0.0.22 | IGMPv3 |
| 224.0.0.251 | IGMPv2 |
| 224.0.0.253 | IPv6 |

Differentiated Services Field(DSCP) prioritization of traffic along a path (QoS)

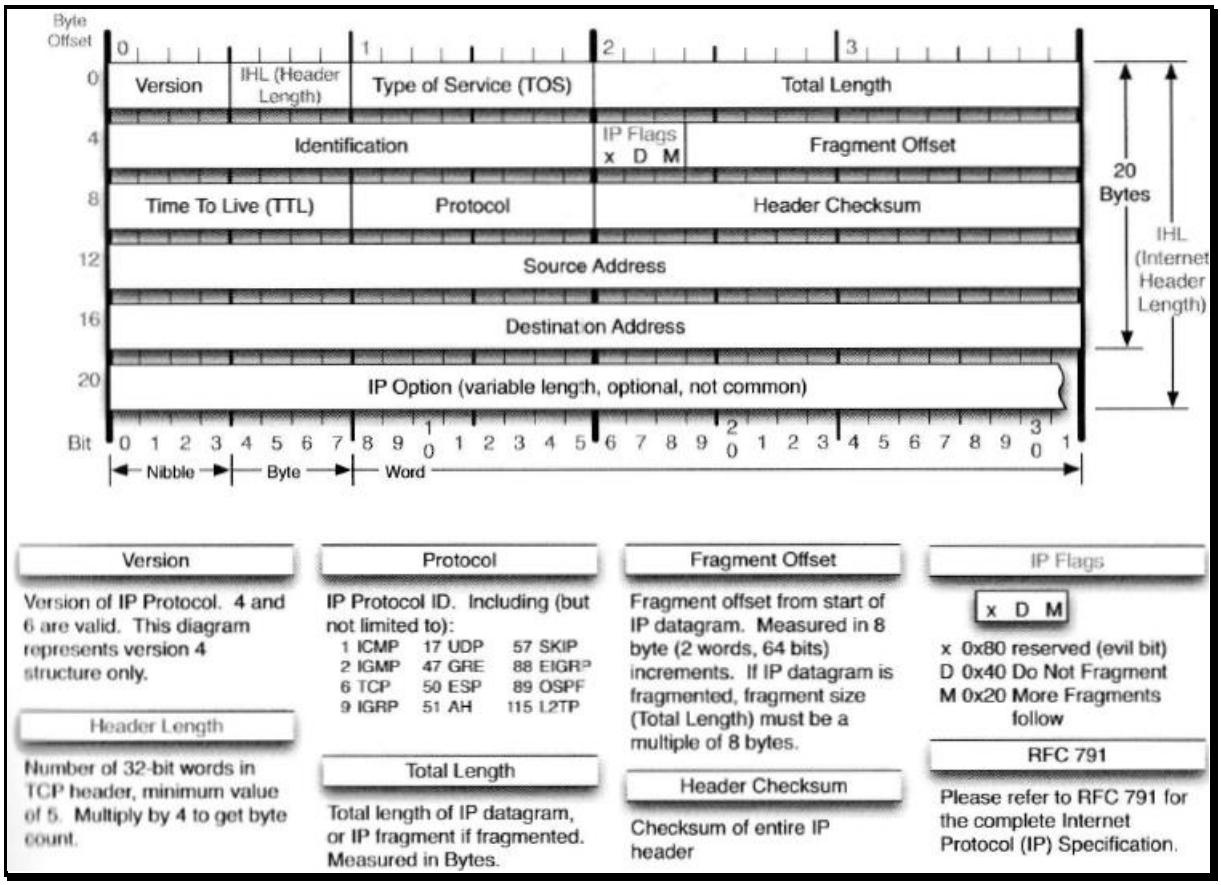



**Figure 13: IPv4 Header**

# IPv6-Problems

RFC 791, 2460, 4291, 4861
RFC 3697 IPv6 Flow Label Specification

Protocol Type: 0x86dd

- Unicast single interface address
- Multicast Group od interfaces
- Anycast Nearest of a group of interfaces

There is no broadcast address in IPv6 **multicasts** are used as a replacement for network broadcasts.
"ICMPv6 Neighbor Solicitation (NS)" this function replaces ARP.
IPv6 Stateless Address AutoConfiguration (**SLAAC**).

**ICMPv6** plays several very important roles in IPv6 communications including **duplicate address detection**, **neighbor discovery** and more.

# ARP/RARP Problems

RFC 826
RFC 1027 Proxy ARP
RFC 903 Reverse Address Resolution Protocol (RARP)

HW-Types: www.iana.org

ARP is used to associate a hardware address with an IP address on a local network and to test for duplicate IPv4 addresses.

**Address Resolution Protocol(ARP)** is used to to identify network addressing or configuration problems. Normal ARP communications consists of a simple request and a simple reply.

- ***ARP Request***
  Msg: If this is your IP-Address, please tell me your MAC-Address
- ***ARP Reply***
  Msg: Sends the MAC-Address and the IP-Address back to the sender of the ARP-Request.

ARP packets do not have an IP header.
ARP packets are non-routable packets.

**ARP poisoning** Wireshark has detected that duplicate address use has occurred. We can see host <mac> advertising 2 IP-Addresses. This is the classic signature of ARP-based **man-in-the-middle** traffic.

ARP/RARP Setting: **Detect ARP request storm**

## Gratuitous ARP (GARP)

- A gratuitous ARP reply is a reply to which no request has been made.
- Every time an IP interface or link goes up, the driver for that interface will typically send a gratuitous ARP to preload the ARP tables of all other local hosts.
- GARP to detect duplicate IP's
- Assist if an IP address is moved from one NIC to another (Failover)

| | |
|---|---|
| Windows XP | OK, are sending GARP's |
| Windows 2003 | NOK, do not generate GARP's. |
| Vista/2008 | NOK, do not generate GARP's. |
| Windows 7 | NOK, do not generate GARP's. |
| Windows 2008 R2 | NOK, do not generate GARP's. |
| Windows 8 | OK, are sending GARP's |
| Windows 2012 | OK, are sending GARP's |

# DHCP Problems

RFC 1534, 2131

ARX: bootps-udp

Statistics → BOOTP-DHCP

| | | | |
|---|---|---|---|
| 67 | TCP | bootps-tcp | Bootstrap Protocol Server |
| 67 | UDP | bootps-udp | Bootstrap Protocol Server |
| 68 | TCP | bootpc-tcp | Bootstrap Protocol Client |
| 68 | UDP | bootpc-udp | Bootstrap Protocol Client |

**DHCPDISCOVER → DHCPOFFER → DHCPREQUEST → DHCPACK → DHCPRELEASE**
One **"Transaction ID"** is issued.

**DHCP-Sequence (DORA)** if the client is **out of lease time**:
**SRC: D**iscover Port 68 (Client, bootpc-tcp, bootpc-udp)
**DST: O**ffer Port 67 (Server, bootps-tcp, bootps-udp)
**SRC: R**equest
**DST: A**ck

**DHCP-Sequence (RA)** if the client is **within lease time**:
**SRC: R**equest
**DST: A**ck

Check: bootp.options.type
bootp.option.type == 55 (Parameters)
bootp.option.type == 55 && bootp.option.value contains 2e (NetBios…Node Type)

- ❑ You should never get **DHCP Decline's** in the DHCP process. Decline means there is a IP conflict.

Don't use **capture filter** when troubleshooting the boot up process.

If hosts on the network have statically assigned addresses and the DHCP server is unaware of this, it may inadvertently offer an address that is already in use unless it performs a duplicate address test (typically using ICMP Echo request).

# DHCP-Relay Agent Problems

- DHCP Relay Agent must be on the same network segment as the DHCP-Client.

# DHCPv6 Problems

RFC 3315

DHCPv6 client port 546
DHCPv6 server port 547

ARX: dhcpv6-client-tcp, dhcpv6-client-udp, dhcpv6-server-tcp, dhcpv6-server-udp

Uses multicast address for all DHCP_Relay_Agents_and_Servers (ff02::1:2).

**DHCP = DORA, DHCPv6 = SARR**
DHCP **S**olicit
DHCP **A**dvertise
DHCP **R**equest
DHCP **R**eply

# DNS Problems

RFC 1034, 1035, 2671, 2535

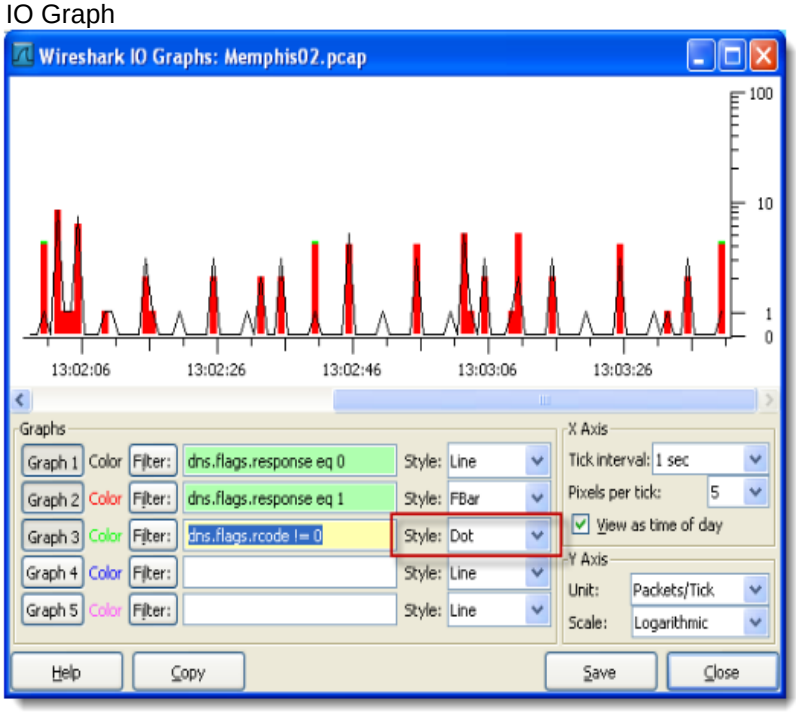
**DNS** can run over **UDP** or **TCP**.
**DNS Zone Transfers** and **large client queries** are using **TCP** over **port 53**.
**DNS queries** and **responses** are using **UDP port 53**.

Max. payload for DNS over UDP is **512 bytes**. If a response requires **more than 512 bytes** of space, however, a truncating flag bit is sent in the response. This triggers the resolver to send the DNS query again using TCP which allows for al larger packet size.

**RFC 2671** DNS (EDNS0) allows for **greater than 512 bytes over UDP**, this may cause some problems with certain devices.

**MulticastDNS (mDNS)** offers a name resolution process for smaller networks that do not have a DNS server installed. The top level **mDNS** names end with <**.local**>. The **mDNS** multicast address is **224.0.0.251** or **FF02::FB**.

Statistics → IO Graph




There above graph shows several errors.

- The green dot (on top some of the red bars) indicates the DNS server responded, at that point, with an error.
- The line without a red bar indicates a request without a response.

**dns.id** **Transaction ID e.g. 0x0000**
Associates DNS queries with responses.

**dns.time** **SRT**
X

**dns.flags** **Truncation**
Indicates the field was truncateded because of the length. If a client sees a truncated DNS response, it should retry the query over TCP.

**dns.flags** **Recursion**
Allows a DNS server to ask another server for an answer on the client's behalf.

**dns.flags.rcode** **Response Code**
- Everything greater than 0 is an error

| | |
|---|---|
| 0 | No Error |
| 1 | Format error – Query could not be interpreted. |
| 2 | Server failure |
| 3 | Name error – domain name does not exist |
| 4 | Not implemented |
| 5 | Refused – Name server refuses to perform function due to policy |

**dns.count.queries** **Question Count**
Typically, you will see only one question per query packet.

**Answer RRs** **Answer Resource Record**
- Bot-infected hosts may receive DNS responses with a high number of Answer RRs.:
  My Filter [**S-DnsRr**]: dns.count.answers >= 10

**Authority RRs**
**Additional RRs** **Additional RRs Count**

**dns.qry.type** **Type**

| | |
|---|---|
| A | Host address |
| NS | Authoritative name server |
| CNAME | Canonical name for an alias |
| SOA | Start of Zone Authority |
| PTR | Pointer Record |
| HINFO | Host information |
| MX | Mail exchange |
| AAAA | IPv6 Address |

**Root Hints**
Are stored in the ***cache.dns*** file at %Systemroot%\System32\Dns folder.

# FTP Problems

RFC 959
Ports: 20, 21

- **Active Mode**
  Uses the PORT command, the data transfer connection is established by the FTP server to the FTP client.
- **Passive Mode**
  Uses the PASV command, the data transfer connection is established by the FTP client to the FTP server.

To reassemble FTP traffic: Follow TCP Stream
Format raw
Save AS

Response codes in the range 400 to 500 indicates that a problem has occurred.

# HTTP Problems

HTTP v1.0
HTTP v1.1 RFC 2616

If you are using Application based display filter e.g. **http**. You will lose the TCP-Traffic such as the handshake and you lose the view on the whole picture.

It's better using:
**<tcp.stream == x>**
or
**<tcp.port == 80>**

***Tip:*** Turn **<OFF> TCP preferences for reassembly** when working HTTP.
Otherwise you can't see the **HTTP response codes** in the Info column.

***Tip:*** If you want to extract HTTP objects, turn <**ON**> **TCP preferences for reassembly.**
Then go to: File → Export Objects → HTTP

See: Edit → Preferences →Protocols → TCP → <**Allow subdissector to reassemble TCP streams>**
Edit → Preferences →Protocols → HTTP → Reassemble HTTP headers ….
Edit → Preferences →Protocols → HTTP → Reassemble HTTP bodies ….

HTTP uses a request/response model.

- Use <**Flow Graphs**> to spot Web Browsing issues
- During analysis of HTTP keep care of the "cache".
- Site name resolution pay attention of DNS-Errors
- non-existing pages or items
- Congestion
- Is the HTTP daemon running on the web server (RST, ACK)?

- ❑ Watch for "**If-Modified-Since**" request modifier.
  This indicates that the client has a page in cache. If the server responds with code "304 Not Modified", the client will load the page from cache instead of across the network.
- ❑ Cause: "Advertisers" on other webpages.
- ❑ Non–Optimized web sites

File → Export Objects → HTTP
Statistics → HTTP → Load Distribution
Statistics → HTTP → Packet Counter
Statistics → HTTP → Requests
Statistics → Flow Graph

**http.response.code**
Everything greater than 399 is an error

- http.response.code <= 399 No Error
- http.response.code > 399 error

Interesting Fields:

- host
- location
- referrer
- x-slogan → Hidden Messages

# HTTPS/SSL Problems

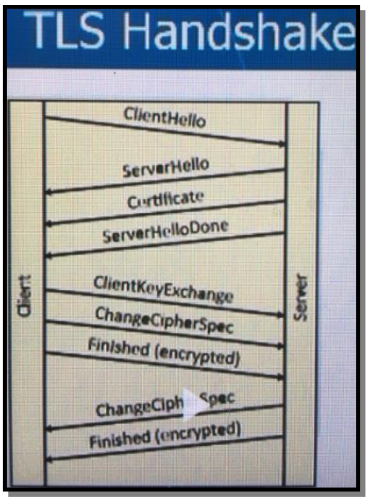
RFC 2246 Transport Layer Security version 1.0 (SSL v3.0)
RFC 2818 HTTP over Transport Layer Security (TLS)

***Tip:*** Turn on TCP pref for reassembly when working HTTPS.
There is no RFC specification for HTTP 2.0
SSL/TLS vulnerabilities: www.phonefactor.com/sslgap

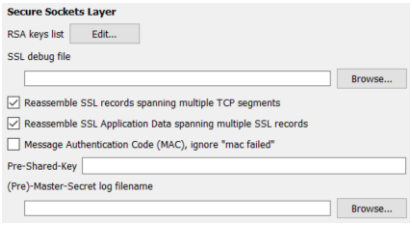
For **SSL-Traffic** the RSA-Key (Private Key) has to be properly loaded and configured.
.key
SSL → RSA keys list: 127.0.0.1,443, http,c:\xxxx\x.key

Use <**Flow Graphs**> to spot Web Browsing issues

File → Export → SSL Session Keys (.key)



## SSL Settings




IP Address SSL-Server, any, anyipv4, anyipv6
Port: start_tls or 0
Protocol: http or data
Key File: Path to the RSA private key
.pem, .pfx, .p12

RSA <EPMS> <PMS>
RSA Session-ID: <SSLID> Master-Key: <MS>
CLIENT_RANDOM <CRAND> <MS>
PMS_CLIENT_RANDOM <CRAND> <PMS>

***Where:***
<EPMS> = First 8 bytes of the Encrypted PMS
<PMS> = The Pre-Master-Secret (PMS) used to derive the MS
<SSLID> = The SSL Session ID
<MS> = The Master-Secret (MS)
<CRAND> = The Client's random number from the ClientHello message

***Some well-known TCP ports for SSL traffic are***
443 https
636 ldaps
989 ftps-data
990 ftps
992 telnets
993 imaps
994 ircs
995 pop3s
5061 sips

## X.509

Is a **ITU-T** standard for a "Public Key Infrastructur (PKI)" to create digital certificates.

# IMAP Problems

RFC 1730

# Lotus Notes Problems

Port: 1352

DRDA Problems check reassemble…

# POP Problems

RFC 1939
Port: 110 (POP3)

Used to retrieve Email.

- Capacity issues at the POP server

- TCP Connection problems
- High latency
- Packet loss

# SMB-Problems

Streaming Blocksize

NetAPP IOP's
Oracle TNS Listener
Java DB Connections (Fetch Size)
Tomcat Server (Socket Buffer default 9000 bytes)

*Check:*
QUERY_PATH_INFO
FIND_FIRST2
QUERY_FILE_INFO
QUERY_FS_INFO

**smb.time** **SRT**
X

## SMB 1

Investigate SMB Blocksize (see NetAPP)!!!
Default MaxMpx Cnt 50 smb.max_mpx_count

## SMB 2

Credit System smb2.credit
Pipelining smb2.file_pipe_info
SMB Branch Cachw

**smb2.time** **SRT**
X

## SMB 3

Multiple Pipelines
Encryption

# SMTP Problems

Simple Mail Transfer Protocol

RFC 5321, 1939
Port: 25

Used to send Email.

- Capacity issues at the SMTP server
- TCP Connection problems
- High latency
- Packet loss

SMTP communications are not secure.
SMTP messages are delivered in "Internet Message Format (RFC 2822)".

**Pipelinging** indicates that the client can send another request to the server without waiting for response to previous one(s).

# SSH Problems

RFC 4253
Secure Shell (SSH) is a replacement for older remote shell programs such as telnet. SSH uses encryption to protect the contents (most notably passwords) being sent over its connection. Simple Mail Transfer Protocol.

ssh.compression_algorithms_client_to_server
ssh.compression_algorithms_server_to_client
ssh.message_code

# VoIP Problems

Wireshark cannot playback encrypted VoIP conversations.

SIP: Protocol: 5060_udp, 5060_tcp
RFC 3261
It's more common, that SIP is running over UDP.

RTP: Carries the voice itself.
RFC 3550
Wireshark includes a RTP player

RTCP: Realtime Transport Control Protocol
Provides out-of-band statistics and control information for RTP flow.

DTMF: Dual-Tone Multi-Frequency

SDP: Session Description Protocol
RFC 4566

Session Initiation Protocol (SIP) is an example of a VoIP signaling protocol.

Codecs
G711
G723.1
G726
G728
G729

Statistics: Telephony → SIP
Playback: Telephony → VoIP Calls

Problems:
- Calls may not go through
- Quality may be degraded

Causes
- **Packet loss**
  VoIP doesn't deal well with packet loss!
  Analysis: Telephony → RTP → Show All Streams
- **Jitter**
  RFC 3550
  Jitter is a variance in the packet rate. High jitter rate > 20ms will negative affect the call.

# WLAN Problems

IEEE 802.11
CSMA/CA protocol

Tools: www.metageek.net

➔ In **promiscuous mode**, an 802.11 adapter only captures packets of the ***SSID*** the adapter has joined.

The network card will pass all traffic it receives to the central processing unit rather than just packets addressed to it.

CHKROOTKIT-Tool to check mode.

➔ **Monitor mode (rfmon-mode)** is not supported by Windows (WinPcap)

Consider using a “AirPcap Adapter (Device)”

Statistics → WLAN Traffic

- Signal Strength
- Packet LOSS Rate
- WLAN Retry Rate
- Round Trip Latency Time
- Location

- Normal **signal strength** values
  The signal strength indicator value defines the power, but not the quality of the signal.
  The value is defined in dBm (power ratio in decibels referenced to one milliwatt).
  0 to -65 dBm Excelent
  -80 dBm Problem
- Normal **radio frequency (RF)** signals
  Devices transmitting RF on the same frequency may interfere your signal.
- Normal association processes
- Normal data exchange over 802.11
- Normal disassociation
- Basic Service Set (BSS)

Spectrum analyzer (Must have tool)

For decrypted traffic you need to input the Key (see Protocol 802.11
Wireshark can decrypt **WEP**, **WPA** and **WPA2** traffic.

There are three types of 802.11 frames seen on WLANs

- **Data**
  See page 636
- **Management**
  See page 636-637
- **Control**

*Steps:*

- ❑ Check Interferences
- ❑ Connection Process
- ❑ Authentication
- ❑ WLAN Control and Management Processes
- ❑ Analysie Packets

# Application Problems

# Non-Standard Ports

# Suspicious Hosts/Traffic

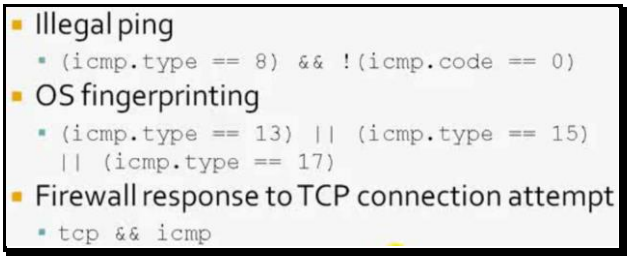
Statistics → Protocol Hierarchy

- Bot-Infected hosts often use Internet Relay Chat (IRC) see destination port **18067**.
  Search for the **JOIN** command.

# Network Scans

Search for unusual ICMP Echo packets.
**ICMP Request must be all the time type=8 and code=0.**

```
(icmp.type==8) && !(icmp.code==0x00)      ➔ Illegal ping request
(icmp.type==8) && (icmp.code==0x7b)       ➔ Indicates Xprobe2 scan 8/123
```




# Dissector

- https://www.wireshark.org/docs/wsdg_html_chunked/ChapterDissection.html

1. Frame Dissector
   See Field: Encapsulation type
   Ethernet (1)

2. Ethernet Dissector
   See Field: Type
   IPv4 (0x0800)

3. IPv4 Dissector
   See Field: Protocol
   TCP (6)

4. TCP Dissector
   See Field: Destination Port
   80 → HTTP

5. HTTP Dissector

- If Wireshark doesn't recognize the protocol, then the traffic will be shown as:
  **IP/Data**
  **TCP/Data** or **UDP/Data**.
  See **Statistics → Protocol Hierarchy**.

# Internet Protocol Version 4

OSI-Layer 3
IP-Flags, Time to live (TTL), Protocoll, sender/receiver.

The typical size of an IP Header is **20 Bytes**

**PCAPNG**

| | |
|---|---|
| Start - End Byte: | 90 – 109 (20 Bytes) |
| IP Src: | 102 - 105 (4 Bytes) |
| IP Dst: | 106 – 109 (4 Bytes) |

# Router

- Router are changing the IP-Header:
- Reducing "**T**ime **T**o **L**ive-Value (**TTL**)" by one.
- NAT-Routers are replacing the SRC-IP with their own IP.

Only routers are sending ***"Time-to-live exceeded"***.

# Switches & Hubs

- Pure switches and hubs don't change frames.

# DFILTERS - VIEW FILTERS / DISPLAY FILTERS

Links: https://www.wireshark.org/docs/dfref/

```
data contains nesus                              ➔ Look only in the data payload
frame contains "dns"
frame matches "(attachment|tar|exe|zip)"         ➔ Uses REGEX
frame matches "(?)(user|password)"               ➔ Uses REGEX
       Regex e.g.:    http matches "\. (?i) (zip|exe)"
http.host matches "(?i)\.(ru|cn)$"
frame.time_delta > 1

Adavanced Filter
frame[54:8] == "GET / get"                       Gets in the frame at pos 54 8Bytes
```

### *Operators*

```
--------------------------------------------------------------------------------------------------
&&
--------------------------------------------------------------------------------------------------
||
--------------------------------------------------------------------------------------------------
==
--------------------------------------------------------------------------------------------------
!=
--------------------------------------------------------------------------------------------------
>
--------------------------------------------------------------------------------------------------
<
--------------------------------------------------------------------------------------------------
>=
--------------------------------------------------------------------------------------------------
<=
--------------------------------------------------------------------------------------------------
```

### *Filters*

```
==================================================================================================
arp
--------------------------------------------------------------------------------------------------
arp.dst.drarp_error_status == 5                  1 = restricted
                                                 2 = no address
                                                 3 = serverdown
                                                 4 = moved
                                                 5 = failure [ARP_Failure5]
--------------------------------------------------------------------------------------------------
arp.duplicate-address-detected
--------------------------------------------------------------------------------------------------
arp.duplicate-address-frame
--------------------------------------------------------------------------------------------------
arp.packet-storm-detected
--------------------------------------------------------------------------------------------------
arp.isgratuitous
==================================================================================================
bgp
==================================================================================================
bittorrent                                       UNWANTED TRAFFIC!
                                                 Also IRC
                                                 - commands: USeR, NiCK and JOiN
==================================================================================================
bootp
--------------------------------------------------------------------------------------------------
bootp.ip.relay !=0.0.0.0                         Where Relay Agent is used
--------------------------------------------------------------------------------------------------
bootp.option.hostname
--------------------------------------------------------------------------------------------------
bootp.option.value==0                            0 = DHCPv4 Discover message
                                                 4 = DHCPv4 Decline message
--------------------------------------------------------------------------------------------------
See page 528
bootp.option.dhcp == x                           1 = DHCP Discover
```

| | |
|---|---|
| | 2 = DHCP Offer<br>3 = DHCP Request<br>4 = DHCP Decline<br>5 = DHCP ACK<br>6 = DHCP Negative ACK<br>7 = DHCP Release<br>8 = DHCP Informational |
| dns | |
| dns.count.answers > 5 | Contains more than 5 responses |
| dns.flags.rcode!=0 | Shows all DNS errors |
| dns.flags.rcode==3 | 0 = NoError<br>1 = Format Error<br>2 = Server Failure<br>3 = Non-Existent Domain<br>5 = Query Refused<br>9 = Not Authorized (for Zone) |
| dns.flags.recdesired==1 | DNS with recursion required |
| dns.flags.recavail==1 | DNS response stating recursion available |
| dns.time > 5 | DNS queries taking longer than 5 second |
| dns.id==0x0000 | Queries and responses of one DNS request |
| dns.qry.type == 1 | **1 = A-Records**<br>2 = NS (Authoritative Name Server)<br>3 = MD (Mail Destination)<br>4 = MF (Mail Forwarder)<br>**5 = CNAME**<br>**6 = SOA**<br>7 = MB (Mailbox domain name)<br>8 = MG (Mail Group Member)<br>11 = WKS (Well Known Service)<br>12 = PTR (domain name PoinTeR)<br>15 = MX (Mail exchange)<br>17 = TXT (Text strings)<br>28 = AAAA (IPv6 Address)<br>**33 = SRV (Server Selections)**<br>37 = CERT<br>39 = DNAME<br>101 = UID<br>102 = GID<br>252 = AXFR (transfer of an entire zone)<br>255 = * (A requests for all records server cache)<br>256 = URI<br>65281 = WINS<br>65282 = WINS-R |
| dns.qry.name==”www.abc.com” | DNS query for “www.abc.com” |
| dns.resp.type==0x0005 | 0x0005 = Response contains CNAME<br>0x0006 = Response cpntains SOA |
| dns and ip.addr==x.x.x.x | See only dns traffic sent /received by this ip |
| dns.flags.response==x | 0 = DNS queries<br>1 = DNS responses |
| dns && (dns.flags.response==0) &&<br>!dns.response_in | |
| **eth** | |
| eth.addr==12:34:56:78:90:12<br>!(eth.addr==12:34:56:78:90:12) | Filter out a MAC Address |
| eth.dst==ff:ff:ff:ff:ff:ff | Layer 2 Broadcast |
| eth.src[4:2]==22:1b | Offset 0x221b after 5 Byte |
| _ws.expert.severity==error | Error |
| _ws.expert.severity==warn | Warn |
| _ws.expert.severity==note | Note |
| _ws.expert.severity==chat | Chat |

**frame**

| | |
|---|---|
| frame.time_delta>1 | |
| frame.time_relative>0 | |

**ftp**

| | |
|---|---|
| ftp.request.arg contains “admin” | |
| ftp.request.arg matches “(?i)admin” | Case insensitive |
| ftp.request.arg matches “(?i)(a¦b¦c)” | Case insensitive plus several matches |
| *See page 583*<br>ftp.response.code==x | 110 =<br>257 = Successful directory creation response<br>400 = FTP-Problem<br>425 = Can’t open data connection |
| ftp.request.command==”x” | “USER”<br>“PASS”<br>“CWD”<br>“QUIT”<br>“PORT”<br>“PASV”<br>“TYPE”<br>“RETR”<br>“STOR”<br>“DELE”<br>“RMD”<br>“MKD”<br>“PWD”<br>“NSLT”<br>“HELP” |
| ftp.request.command == “USER” ||<br>ftp.request.command == “PASS” | |

**hsrp**

| | |
|---|---|
| hsrp.state !=8 && hsrp.state !=16 | |

**http**

| | |
|---|---|
| http contains “GET” | |
| http matches “(?i)get” | |
| http matches “(?i)(get\|rating\|google)” | |
| http && ip.addr ==<x.x.x.x> | |
| http.cookie | Client has a http cookie |
| http.host | |
| http.request | Displays all HTTP GET requests |
| http.request.uri==”/” | |
| http.request.uri==”.exe” | To see who is downloading exe files |
| http.request.uri contains “jpg”<br>http.request.uri contains “.exe” | |
| http.request or ssl.handshake.type == 1 | |
| http.host<br>http.host contains “host” | Displays all packets with a host in. |
| http.request.method==”x” | “GET”<br>“HEAD”<br>“POST”<br>“OPTIONS”<br>“PUT”<br>“DELETE”<br>“TRACE”<br>“CONNECT” |

| | |
|---|---|
| http.request.method==”GET” (POST) | Shows all GET(POST) HTTP messages |
| http.response.code >=300 &&<br>http.response.code < 400 | Shows REDIRECTIONS |
| *See page 548*<br>http.response.code == x | 100 = Continue<br>200 = OK |
| http.response.code==404 | |
| http.set_cookie | |
| http.time > 0.1 | |

**icmp**

| | |
|---|---|
| *See page 430*<br>icmp.type==x and | ***types:***<br>0 = Echo Reply (RFC 792)<br>**3 = Destination Unreachable**<br>4 = Source Qunch<br>5 = Redirect<br>6 = Alternate Host Address<br>**8 = Echo**<br>9 = Router Advertisement<br>10 = Router Solicitation<br>**11 = Destination Network Unreachable for Type of Srv**<br>12 =<br>**13 = Timestamp**<br>14 = Timestamp Reply<br>**15 =**<br>**17 = Address Mask Request**<br>18 = Address Mask Reply<br>..<br>30 = Traceroute<br><br>***Codes for type 3 (Destination unreachable)***<br>0 Network unreachable (from gateway)<br>1 Host unreachable (from gateway)<br>2 Protocol unreachable (from host)<br>3 Port unreachable (from host)<br>4 Fragmentation needed but don't fragment bit is set<br>5 Source route failed<br>6 Destination network unknown<br>7 Destination host unknown<br>8 Source host isolated<br>9 Destination network administratively prohibited<br>10 Destination host administratively prohibited<br>11 Network unreachable for type of service (TOS)<br>12 Host unreachable for TOS<br>13 Comm. administratively prohibited by filtering<br>14 Host prededence violation<br>15 Precedence cutoff in effect |
| icmp.type==x && icmp.code==y | 3/3 = Destination Unreachable/Port Unreachable<br>Indicates that a FW is blocking the traffic.<br>8/0 = Illegal Ping packets<br>5/1 = ICMP Redirect<br>11/0 = Time Exceeded |
| *See page 431*<br>icmp.code==y | 0 = Net Unreachable<br>1 = Host Unreachable<br>2 = Protocol Unreachable<br>..<br>3 = Port Unreachable<br>..<br>15 = Precedence Cutoff |

***icmpv6***

| | |
|---|---|
| *See page 434*<br>icmpv6.type == 128 | 128 = Echo Request<br>129 = Echo Reply<br>133 = Router Solicitation [RFC 4861]<br>134 = Router Advertisement [RFC 4861]<br>.. |
| *See page 436* | |

| | |
|---|---|
| icmpv6.code == 0 | 0 = No Route to destination |
| icmpv6.type==3 and<br>icmpv6.code == 4 | Fragmentation Needed, but no fragment bit set. |
| **igmp** | |
| ip | |
| ip.ttl | Time to Live |
| ip.host==www.wireshark.org | |
| **ip.addr==x.x.x.x** | |
| !(ip.addr==x.x.x.x) | Filter out a n IP completely |
| **ip.addr==x.x.x.x/24** | **Filter addresses of subnet** |
| ip.src==x.x.x.x | |
| ip.dst==x.x.x.x | |
| ip.proto==x | Protocol Field<br>1 = ICMP<br>2 = IGMP<br>6 = TCP<br>8 = EGP<br>9 = Any private interior gateway, e.g Cisco's IGRP<br>17 = UDP<br>45 = IDRP<br>88 = Cisco EIGRP<br>89 = OSPF |
| ip.hdr_len > 20 | Header longer than 20 bytes |
| **ipv6** | |
| ipv6.nxt==0x06 | Precede a TCP header |
| ipv6.hlim < 10 | IPv6 packet with a Hop lower than 10 |
| ip.proto==41 | IPv6 packet |
| media | |
| netbios | Show only the NetBIOS protocol-based traffic |
| nfs.status2>0 or nfs.status3>0 | NFS error filter |
| **ntp** | Show only NTP |
| **ospf** | Shows only OSPF traffic |
| ospf.msg !=1 | Routing state changes |
| **pkt_comment** | **SHOW PACKET COMMENTED** |
| **pop** | |
| pop.request.command == "x" | "USER"<br>"PASS"<br>"QUIT"<br>"STAT"<br>"LIST"<br>"RETR"<br>"DELE"<br>"PIPELINING"<br>"UIDL" |
| pop.response.indicator == "x" | "+OK"<br>"-ERR" |
| **port** | |
| dst port 135 and tcp port 135 and ip[2:2]==48 | Blaster Worm |
| **icmp** | |
| icmp[icmptype]==icmp-echo | Welchia worm |

| | |
|---|---|
| and ip[2:2]==92<br>and icmp[8:4]==0xAAAAAAAA | |
| icmp.type eq 3 \|\| icmp.type eq 4 \|\|<br>icmp.type eq 5 \|\| icmp.type eq 11 \|\|<br>icmpv6.type eq 1 \|\| icmpv6.type eq 2 \|\|<br>icmpv6.type eq 3 \|\| icmpv6.type eq 4 | ICMP Errors |
| **sip** | **Session Initiation protocol (SIP)**<br>**Application layer control protocol**<br>**Internet telephone calls, multimedia etc.** |
| *See page 668*<br>sip.Method=="x" | "INVITE"<br>"ACK"<br>"BYE"<br>"CANCEL"<br>"OPTIONS"<br>"SUBSCRIBE"<br>"REGISTER"<br>"INFO" |
| sip.Status-Code==x | 100 = Trying<br>180 = Ringing<br>..<br>606 = Not Acceptable |
| sip.Status-Code>300 | To detect SIP errors |
| sip.Status-Code>399 | To detect SIP errors |
| tcp port sip | SIP traffic to the standard SIP port 5060 |
| udp port sip | |
| port sip | |
| pppoes | CapturePPPoE traffic |
| **radiotap** | |
| radiotap.datarate | WLAN |
| radiotap.dbm_antsignal | WLAN |
| **smb** | |
| smb.nt_status > 0 \|\| smb2.nt_status > 0 | Shows all SMB/2 Errors |
| smb.nt_status!=0x0 | Zugriff auf nicht vorhandene Shares und Files |
| **smb2** | |
| smb2.cmd == x | 5 = Create<br>6 = Close<br>9 = Write<br>11 = Ioctl<br>17 = SetInfo |
| **smtp** | |
| *See page 615*<br>smtp.response.code == x | 211 = System status<br>399 |
| smtp.req.command == "x" | "HELO"<br>"EHLO"<br>"MAIL"<br>"RCPT"<br>"DATA"<br>"VRFY"<br>"RSET"<br>"NOOP"<br>"QUIT"<br>"EXPN"<br>"HELP" |
| **stp** | |
| stp.type==0x00 | Normal |

| | |
|---|---|
| stp.type==0x80 | Spanning Tree Topology Change Notification<br>TCN sollte in einem stabilen LAN nicht zu sehen sein |
| smb.nt_status>0 | SMB errors |
| smb.nt_status>0 ¦¦ smb2.nt_status>0 | SMB errors |
| smb.time>1 | SMB Delays |
| **smb2.buffer_code==0x09** | SMB2 TreeConnect |
| **ssl** | Filter Certificate flow |
| ssl.handshake.certificate | |
| ssl.record.content_type==22 | TLS handshake packets |
| **tcp** | |
| tcp contains traffic | Displays all TCP packets which<br>contains the word <traffic> |
| tcp.analysis.flags | Show packets that match the Expert Info Notes.<br>Bad TCP(TCP retransmissions, out-of-order, Dup etc.) |
| tcp.analysis.lost_segment | Packet Loss |
| **tcp.analysis.retransmission** | **Displays all retransmissions, Fast and …** |
| **tcp.analysis.ack_rtt** | **Displays all retransmissions** |
| **tcp.analysis.duplicate_ack** | **Doppelte Acknowledgements** |
| **tcp.analysis.window_update** | TCP Window Update<br>Senders TDP receive buffer space increased |
| tcp.analysis.bytes_in_flight | Unacknowledged bytes in flow |
| tcp.analysis.flags &&<br>!tcp.analysis.window_update | Shows TCP problems |
| tcp.analysis.zero_window | |
| tcp.flags.fin==1 | FIN bit used |
| tcp.flags.syn==1<br>tcp.flags.syn!=1 | SYN bit used |
| tcp.flags.reset==1 | Displays all TCP resets (conn refused, terminat.) |
| tcp.flags.urg==1 | URG bit used<br>The receiver must examine this field to see where<br>to lok/read first in the packet |
| tcp.flags==0x2 | SYN Packets Only! [TCP_Syn_Only] |
| tcp.flags==0x16 | ACK Packets |
| tcp.flags==0x18 | SYN/ACK Packets |
| tcp.flags==0x012 | SYN and ACK bit set |
| tcp.flags==0x020 | |
| tcp.len | Payload size |
| tcp.time_delta | |
| tcp.raessembled.length | |
| tcp.window_size > 1500 | |
| tcp.window_size < 65535 | |
| tcp.options | |
| **tcp.port==x** | **21 = FTP**<br>**22 = SSH**<br>**23 = Telnet** |

| | |
|---|---|
| | **25 = SMTP**<br>**80 = http**<br>**110 = POP**<br>**123 = NTP**<br>**6881 = Unwanted BitTorrent traffic!**<br>**6881-6889 = Unwanted BitTorrent traffic!** |
| **tcp.port==25 or icmp** | **Shows only SMTP and ICMP Traffic** |
| tcp.stream==2 | |
| tcp.urgent_pointer | |
| tftp | |
| **udp** | |
| udp.dstport==x | 1900 = UPnP (SSDP) |
| udp.srcport==x | 1900 = UPnP (SSDP) |
| vlan | |
| vlan.id == 996 | |
| vrrp | |
| **wlan** | |
| wlan.fc.retry==1 | To locate WLAN retries |
| *See page 644*<br>wlan.fc.type_subtype==x | 0x05 = Probe Respnse packets<br>0x08 = AP Beacons |

## Filter for Conversations

- FILTER [TCP¦UDP] STREAM
- CONVERSATION FILTER [TCP¦UDP]
- FOLLOW [TCP¦UDP] STREAM

# Filter Expressions

To be imported into **<preferences>** with Notepad++ or Wordpad.

**####### Filter Expressions ########**
**gui.filter_expressions.label: MyMAC**
**gui.filter_expressions.enabled: TRUE**
**gui.filter_expressions.expr: eth.addr==D4-85-64-A7-BF-A3**

gui.filter_expressions.label: NotMyMAC
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: !eth.addr==D4-85-64-A7-BF-A3

**gui.filter_expressions.label: SYNs**
**gui.filter_expressions.enabled: TRUE**
**gui.filter_expressions.expr: tcp.flags.syn==1**

**gui.filter_expressions.label: ----T----**
**gui.filter_expressions.enabled: TRUE**
**gui.filter_expressions.expr: frame**

gui.filter_expressions.label: T-smallwin
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: tcp.window_size < 1320 && tcp.window_size > 0

gui.filter_expressions.label: T->2secs
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: tcp.time_delta > 2

gui.filter_expressions.label: T-DNSerrs
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: !dns.flags.rcode==0 && dns.flags.response==1

gui.filter_expressions.label: T-HTTPerrs
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: http.response.code > 399

**gui.filter_expressions.label: -----S-----**
**gui.filter_expressions.enabled: TRUE**
**gui.filter_expressions.expr: frame**

gui.filter_expressions.label: MZ/This...
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: frame contains "MZ" and frame contains "This program cannot be run in DOS mode"

gui.filter_expressions.label: URI-ExE/ZiP/JaR
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: http.request.uri matches "\.(?i)(exe|zip|jar)" && !frame matches "(?i)gzip"

gui.filter_expressions.label: 5Rdr
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: icmp.type==5

gui.filter_expressions.label: NSTPro
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: icmp.type==8 && icmp.code==1

gui.filter_expressions.label: NMap
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: icmp.type==8 && icmp.code==9

gui.filter_expressions.label: LNSS
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: icmp.type==8 && icmp.code==19

gui.filter_expressions.label: Nessus
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: frame matches "(?i)nessus"

gui.filter_expressions.label: Xprobe
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: icmp.type==8 && icmp.code==123

gui.filter_expressions.label: Macof
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: tcp.window_size==512 && tcp.flags == 0x0002"

gui.filter_expressions.label: Ettercap
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: ip.id==0xe77e

gui.filter_expressions.label: .Ru/.Cn/.Nu
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: http.host matches "\.(?i)(ru$|cn$|nu$)"

gui.filter_expressions.label: SMTP:zip/exe/jar
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: smtp matches "(?i)attachment" && (smtp matches "(?i)(zip|exe)"

gui.filter_expressions.label: SMTPattach
gui.filter_expressions.enabled: TRUE
gui.filter_expressions.expr: smtp && smtp contains "attachment"

# CFILTERS - CAPTURE FILTERS

Limiting the packets if you are in a busy network.

Syntax: Berkeley Packet Filtering (BPF) / tcpdump syntax.

See: www.wiki.wireshark.org/CaptureFilters

Operators:

- not or !
- and or &
- or or |

In case of problems to capture, increase **CAPTURE BUFFER** size.

| | |
|---|---|
| **arp** | |
| arp.opcode==0x0001 | 0x0001 = ARP requests<br>0x0002 = ARP Replys |
| **Broadcast** | |
| not broadcast and not multicast | |
| **dst** | |
| dst host ff02::1 | Traffic of all hosts to IPv6 Multicast |
| dst host ff02::2 | Traffic of all router to IPv6 Multicast |
| **ether** | |
| ether proto 0x888e | Capture only Ethernet type EAPOL |
| ether host 00:00:00:00:00:00<br>not ether host 00:00:00:00:00:00 | |
| ether src 00:00:00:00:00:00 | Filter all traffic for this MAC address |
| ether dst 00:00:00:00:00:00 | |
| **host** | |
| host x.x.x.x<br>host x.x.x.x and port 80<br>host x.x.x.x and not port 80 | |
| host xxxx:yyyy:xxxx:yyyy:xxxx | |
| not host x.x.x.x | |
| src host x.x.x.x | |
| dst host x.x.x.x | |
| host x.x.x.x or y.y.y.y | |
| host <FQDN> | |
| host www.example.com<br>and not (port 80 or port 25) | Capture non-HTTP and non –SMTP traffic |
| host 10.x.x.x<br>and tcp dst 53 | Capture all traffic sent to port 53 DNS queries and responses |
| **icmp** | |
| icmp<br>icmp[0]=8 | Capture all ICMP Type 8 (Echo Request) packets |
| **igmp** | |
| **ip** | **Capture only IP traffic** |
| ip proto 0x59 | Filter OSPF |

| | |
|---|---|
| **ip proto ospf** | Capture only OSPF traffic |
| **ip proto 0x59** | Capture only OSPF traffic |
| **ip6** | Only IPv6 Traffic |
| ip6 net fe80::/8 | Filter Subnet |
| **net** | |
| net x.x.x.x/x<br>or<br>not net 10.12.8.0/24 | |
| net x.x.x.x mask 255.255.255.0 | |
| ip6 net xxxx:yyyy:xxxx:yyyy:xxxx/64 | |
| dst net x.x.x.x/16 | |
| not dst net x.x.x.x/16 | |
| **netbeui** | Capture NetBIOS based traffic only: |
| icmp | |
| icmp[0]=8 | Ping |
| icmp[0]=17 | Address Mask Request |
| icmp[0]=3 and not icmp[0]=4 | Destination Unreachable |
| ip broadcast | |
| ip multicast | |
| **mDNS** | |
| port 5353 | Filter mDNS |
| **port** | |
| port 53 | Capture only DNS Traffic |
| not port 53 | |
| port 67 or port 68 | DHCP/BOOTP Traffic |
| **udp** | |
| udp port 53 and (udp[10] & 1 == 1) and src net not <10.0.0.0/8> | |
| udp port 67 | Usually DHCP (Server) |
| udp port 123 | Captures NTP only |
| udp port 137 | NetBIOS Name Service |
| udp port 161 | SNMP response |
| portrange 1-80 | |
| tcp portrange 1-80 | |
| port 20 or port 21 | Usually FTP control- and data-channel |
| **src** | |
| src net x.x.x.x/16 | |
| src net x.x.x.x mask 255.255.255.0 | |
| **stp** | |
| **tcp** | **Capture TCP Only** |
| tcp[13] & 0x12 = 18 | Filter tcp.flags: SYN, ACK and ECN |

```
0000 0000 0000
|||| |||| |||+-> FIN   <tcp[13] & 0x12 = 1> Filters all where the FIN-Flag is set
```

```
|||| |||| ||+--> SYN   <tcp[13] & 0x12 = 2> Filters all where the SYN-Flag is set
|||| |||| |+---> RST   <tcp[13] & 0x12 = 6> Filters all where the RST-Flag is set
|||| |||| +----> PSH   <tcp[13] & 0x12 = 12> Filters all where the PSH-Flag is set
|||| |||+------> ACK   <tcp[13] & 0x12 = 24> Filters all where the ACK-Flag is set
|||| ||+-------> URG   <tcp[13] & 0x12 = 48> Filters all where the RST-Flag is set
|||| |+--------> ECN   <tcp[13] & 0x12 = 56> Filters all where the RST-Flag is set
|||| +---------> CWR   <tcp[13] & 0x12 = 112> Filters all where the RST-Flag is set
|||+----------> Nonce
||+-----------> Reserved
|+------------> Reserved
+-------------> Reserved
```

| | |
|---|---|
| tcp portrange 1501-1549 | |
| tcp udp src port 67 and udp dst port 68 | Usually DHCP traffic |
| tcp port x | 21 = Usually FTP control channel<br>25 = SMTP Traffic<br>110 = POP Traffic<br>179 = Captures BGP Traffic |
| tcp port http | Captures HTTP only |
| tcp port https | Captures HTTPS only |
| tcp.srcport==21 | Captures FTP only |
| tcp[tcpflags] & (tcp-syn) !=0 | TCP Connection attempts (OK or NOK) |
| **vlan** | **Capture vlan traffic** |
| vlan 17 | Capture VLAN 17 only |
| vlan and (host x.x.x.x and port 80) | |
| **vrrp** | **Capture vrrp traffic** |
| **wlan** | |
| wlan addr1=xx.xx.xx.xx.xx.xx | Receiver address |
| wlan addr2=xx.xx.xx.xx.xx.xx | Transmitter address |
| wlan dst=x.xx.xx.xx.xx.xx | Destination address |
| wlan src=x.xx.xx.xx.xx.xx | Source address |

# TECHNIQUES

## BitTorrent

- BitTorrent is a protocol designed for transferring files. It is peer-to-peer in nature, as users connect to each other directly to send and receive portions of the file.
- However, there is a ***central server (called a tracker) which coordinates the action of all such peers***.
- The tracker only manages connections, it does not have any knowledge of the contents of the files being distributed, and therefore a large number of users can be supported with relatively limited tracker bandwidth.
- A recent extension to BitTorrent is the **DHT** ("distributed sloppy hash table" or simply called UDP tracker) protocol. A UDP based peer to peer tracker protocol. And the uTorrent imports another UDP based Micro Transport Protocol, called uTP.

TCP: Typically, BitTorrent uses TCP as its transport protocol. The well-known TCP port for BitTorrent traffic is **6881-6889** (and **6969** for the tracker port). The DHT extension (peer2peer tracker) uses various UDP ports negotiated by the peers.

## DIG

To find DNS records

dig <dns-server> <target> any

## IPv4 Multicast Addresses

- IPv4 multicast addresses are defined by the leading address bits of 1110, originating from the classful network design of the early Internet when this group of addresses was designated as Class D.
- The Classless Inter-Domain Routing (CIDR) prefix of this group is **224.0.0.0/4**.
- The group includes the addresses from 224.0.0.0 to 239.255.255.255. Address assignments from within this range are specified in **RFC 5771.**

| IP multicast address | Description |
|---|---|
| 224.0.0.0 | Base address (reserved) |
| 224.0.0.1 | The *All Hosts* multicast group addresses all hosts on the same network segment. |
| 224.0.0.2 | The *All Routers* multicast group addresses all routers on the same network segment. |
| 224.0.0.4 | This address is used in the Distance Vector Multicast Routing Protocol (DVMRP) to address multicast routers. |
| 224.0.0.5 | The Open Shortest Path First (OSPF) *All OSPF Routers* address is used to send Hello packets to all OSPF routers on a network segment. |
| 224.0.0.6 | The OSPF *All Designated Routers* ""(DR)"" address is used to send OSPF routing information to designated routers on a network segment. |
| 224.0.0.9 | The Routing Information Protocol (RIP) version 2 group address is used to send routing information to all RIP2-aware routers on a network segment. |
| 224.0.0.10 | The Enhanced Interior Gateway Routing Protocol (EIGRP) group address is used to send routing information to all EIGRP routers on a network segment. |
| 224.0.0.13 | Protocol Independent Multicast (PIM) Version 2 |
| **224.0.0.18** | Virtual Router Redundancy Protocol (VRRP) |
| 224.0.0.19 - 21 | IS-IS over IP |
| 224.0.0.22 | Internet Group Management Protocol (IGMP) version 3[2] |

| | |
|---|---|
| 224.0.0.102 | Hot Standby Router Protocol version 2 (HSRPv2) / Gateway Load Balancing Protocol (GLBP) |
| 224.0.0.107 | Precision Time Protocol version 2 peer delay measurement messaging |
| 224.0.0.251 | Multicast DNS (mDNS) address |
| 224.0.0.252 | Link-local Multicast Name Resolution (LLMNR) address |
| 224.0.0.253 | Teredo tunneling client discovery address[3] |
| 224.0.1.1 | Network Time Protocol clients listen on this address for protocol messages when operating in multicast mode. |
| 224.0.1.39 | The Cisco multicast router *AUTO-RP-ANNOUNCE* address is used by RP mapping agents to listen for candidate announcements. |
| 224.0.1.40 | The Cisco multicast router *AUTO-RP-DISCOVERY* address is the destination address for messages from the RP mapping agent to discover candidates. |
| 224.0.1.41 | H.323 Gatekeeper discovery address |
| 224.0.1.129 - 132 | Precision Time Protocol version 1 time announcements |
| 224.0.1.129 | Precision Time Protocol version 2 time announcements |
| **239.255.255.250** | Simple Service Discovery Protocol address (On port 1900) |

## WHOIS

- The Whois service (http://rs.Internic.net/whois.html) is a TCP port 43 transaction-based query/response daemon, running on a few specific central machines.
- It provides networkwide directory services to local and/or Internet users. Many sites maintain local Whois directory servers with information about individuals, departments, and services at that specific domain.
- This service is an element in one the core steps of the discovery phase of a security analysis, and is performed by hackers, crackers, phreaks, and cyberpunks, as well as tiger teams. The most popular Whois databases can be queried from the InterNIC.

Overview Domains: WHOIS-Server(whois root-dom) **rs.internic.net**

## Troubleshooting: Registrar

To find the present Registrar: http://www.whois.com/whois/
http://www.whois.com/whois/<IP-Address>

```
whois -h whois.arin.net n <target>
```

# HTTP STATUS CODES

Link: http://en.wikipedia.org/wiki/List_of_HTTP_status_codes

**1xx Information for later use**

**100 Continue**

This means that the server has received the request headers, and that the client should proceed to send the request body (in the case of a request for which a body needs to be sent; for example, a POST request). If the request body is large, sending it to a server when a request has already been rejected based upon inappropriate headers is inefficient. To have a server check if the request could be accepted based on the request's headers alone, a client must send `Expect: 100-continue` as a header in its initial request and check if a `100 Continue` status code is received in response before continuing (or receive `417 Expectation Failed` and not continue).

**2xx Successful received, understood and accepted**

**200 OK**

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action.

**3xx Redirection, more activities required to fulfill request**

**301 Moved Permanently**

This and all future requests should be directed to the given URI.

**302 Found**

This is an example of industry practice contradicting the standard. The HTTP/1.0 specification (RFC 1945) required the client to perform a temporary redirect (the original describing phrase was "Moved Temporarily"), but popular browsers implemented 302 with the functionality of a 303 See Other. Therefore, HTTP/1.1 added status codes 303 and 307 to distinguish between the two behaviours. However, some Web applications and frameworks use the 302 status code as if it were the 303.

**303 See Other (since HTTP/1.1)**

The response to the request can be found under another URI using a GET method. When received in response to a POST (or PUT/DELETE), it should be assumed that the server has received the data and the redirect should be issued with a separate GET message.

**304 Not Modified**

Indicates that the resource has not been modified since the version specified by the request headers If-Modified-Since or If-Match. This means that there is no need to retransmit the resource, since the client still has a previously-downloaded copy.

## 4xx Client Error

**400 Bad Request**

The request cannot be fulfilled due to bad syntax.

**401 Unauthorized**

Similar to *403 Forbidden*, but specifically for use when authentication is required and has failed or has not yet been provided. The response must include a WWW-Authenticate header field containing a challenge applicable to the requested resource. See Basic access authentication and Digest access authentication.

**403 Forbidden**

The request was a valid request, but the server is refusing to respond to it. Unlike a 401 Unauthorized response, authenticating will make no difference.

**404 Not Found**

The requested resource could not be found but may be available again in the future. Subsequent requests by the client are permissible.

## 5xx Server-Error

**503 Service Unavailable**
The server is currently unavailable (because it is overloaded or down for maintenance). Generally, this is a temporary state.

# PROTOCOLS

http://wiki.wireshark.org/ProtocolReference
http://wiki.wireshark.org/SampleCaptures

104apci
Example: 10.28.28.5 <-> 10.138.42.200
Port: 2404

104asdu
Application Service Data Unit (ASDU)
Port: 2404

BJNP Canon BJNP

BROWSER
Local Master Announcement (MS Computersuchdienst)

BJNP

CLDAP Connectionless Lightweight Directory Access Protocol
Uses UDP as transport protocol.
Netlogon to <FQDN>

**DATA** Wireshark does not find a dissector for this!
SUSPICIOS

DB-LSO-DISC

DB-LSP-DISC Dropbox LAN sync Discovery Protocol

DCERPC Distributed Computing Environment / Remote Procedure Call

DHCP Dynamic Host Configuration Protocol
RFC 2131
Port: 67 bootp & 68 bootp
DHCP is a client/server protocol used to dynamically assign IP-address parameters (and other things) to a DHCP client. It is implemented as an option of ***BOOTP***.

DHCPv6

DNS
RFC 1035, 1034

DRSUAPI Microsoft Directory Replication Service

DTP Dynamic Trunk Protocol

ESP Encapsulating Security Payload

EPM DCE/RPC

FTP-DATA

GVSP GigE Vision Streaming Protocol

Webcam streaming protocol

ICMP Internet Control Message Protocol

ICMPv6
Router Advertisement of IPv6 Routers

IMAP Internet Message Access Protocol

IGMP Internet Group Management Protocol

IGMPv2
RFC 2236

IGMPv3

IPX SAP Internet Packet Exchange / Service Advertisement Protocol
KRB5 Kerberos
kerberos.error_code == 25 (0x18) Error: eRR-PREAUTH-REQUIRED
Usually means bad password
LDAP

LLC Logical Link Contol such as:
STP (Spanning Tree Protocol), NDP (Nortel Discovery Protocol)

LLDP Link Layer Discovery Protocol

LLMNR Link-local Multicast Name Resolution

LOOP

LPD Line Printer Daemon Protocol
RFC 1179

LPR Windows Line Printer
RFC 1179

MDNS Port: 5353

MS NLB Microsoft Network Load Balancing

**NetBIOS**
To completely filter NetBIOS traffic, use:
My Filter [**NB_All**]: udp.port==137 || udp.port==138 || tcp.port==139

NBNS NetBIOS Name Server/Service (WINS)
Port: 137 udp
RFC 1001, RFC 1002
NBNS uses UDP as its transport protocol.
The well-known UDP port for NBNS traffic is 137.

NBDS NetBIOS Datagram Service
Port: 138 udp

NBSS NetBIOS Session Service
Port: 139 tcp

NDP Nortel-Discovery Protocol
Data Link Layer (OSI Layer 2) network protocol.
To auto-discover Nortel networking devices.

NFS Network File System

NTP Network Time Protocol
Port: 123_udp

OSPF Version 2
IP Protocol type: 0x59
RFC 2328

***OSPF Packet Types:***

| Type | Packet name | Protocol function |
|---|---|---|
| 1 | Hello | Discover/maintain neighbors |
| 2 | Database Description | Summarize database contents |
| 3 | Link State Request | Database download |
| **4** | **Link State Update** | **Database update** |
| **5** | **Link State Ack** | **Flooding acknowledgment** |

OCSP Online Certificate Status Protocol

PN-MRP PROFINET MRP Test (SIEMENS)

Portmap Part of the ONC-RPC protocol family

QUIC Quick UDP Internet Connections
HTTP/2 over UDP
Experimental by GOOGLE
Fortinet: http://kb.fortinet.com/kb/documentLink.do?externalID=FD36680

RDP Remote Desktop Protocol
Port: 3389_tcp_udp

RTMP Real Time Messaging Protocol

SAMETIME

SAMR

SCCP Skinny Call Control Protocol
Cisco proprietary protocol for Cisco VoIP phones and Cisco Call Manager.

SMB

SMB2 Server Message Block version 2

SMTP
220 Server Ready
HELO Client nennt seinen Namen (EHLO = Extended HELO)
250 Server bestätigt

SNMP Simple Network Management Protocol
Ports: 161_udp, 162_udp (Trap)

SNMPv3

SPOOLSS

SSDP Simple Service Discovery Protocol
Searches for UPnP-Devices in the network.
Usually port 1900
It has to advertise itself to:
- **Multicast-Address 239.255.255.250:1900**

SSL Secure Socket Layer

RSA-Key must be implemented as *PEM* format private key or a PKCS#12 keystore.

STP Spanning Tree Protocol (STP)

TDS Tabular Data Stream
Application Layer Protocol to transfer data between a database server and a client.
Initially for Sybase today also for MS SQL Server.

TLSV Transport Layer Security
More known as SSL

TLSv1 Bevor closing TCP connection [RST]

TLSv1.2 Bevor closing TCP connection [RST]

TPKT ISO transport services on top of the TCP (TPKT)
RFC 1006,
Port: 102
Typically, RDP uses **TPKT** as its transport protocol.
As TCP becomes more and more popular (around 1995?), a mechanism was needed to encapsulate ISO services on top of TCP transport, as both protocols have similar tasks and **COTP** was becoming obsolete these days.

VLAN Virtual Bridged LAN (VLAN, IEEE 802.1Q)

VRRP Virtual Router Redundancy Protocol
IP Protocol type: 0x70
RFC 2338
VRRP uses IP as its transport protocol.

# STANDARD PORTs

Link: http://de.wikipedia.org/wiki/Liste_der_standardisierten_Ports
Link: Port-Numbers IANA www.iana.org/assignments/port-numbers

- Since port numbers are ***16-digits*** binary numbers, the total number of ports is 2^16, or ***65'536***.
- A port is also called a ***socket***.
- Ports allow a ***single IP address*** to be able to support ***multiple communications***, each using a different port number.

There are 3 categories of port numbers:

- Well Known Ports 0 – 1023
- Registered Software Ports 1024 – 49151 (see ***iana.org***)
- Dynamic and Private Ports 49152 - 65535

| Value (Dec) | Protocol | Description |
|---|---|---|
| **Standardised Ports 0 - 1023** | | |
| 0 | | Reserviert |
| 1 | TCPMUX | TCP Port Service Multiplexer |
| 2 | IGMP | Internet Group Management Protocol (Multicast) |
| 3 | | Nicht festgelegt |
| 4 | | Nicht festgelegt |
| 5 | RJE | Remote Job Entry |
| 6 | | |
| **7** | **ECHO** | **ping (Echo Service)** |
| **8** | **ICMP** | **ICMP (Ping)** |
| 9 | DISCARD | Discard |
| 11 | USERS | Active Users |
| **13** | **DAYTIME** | **NIST Daytime** |
| **15** | **NETSTAT** | **Used by netstat** |
| 17 | QUOTE | Quote of the Day |
| 19 | CHARGEN | Character Generator |
| **20** | **FTP-DATA(Datenport)** | **File Transfer [Default Data]** |
| **21** | **FTP (Kontrollport)** | **File Transfer [Control]** |
| **22** | **SSH** | **Secure Shell / SFTP (e.g. PuTTY)** |
| **23** | **TELNET** | **Telnet** |
| **25** | **SMTP** | **Simple Mail Transfer Protocol** |
| 27 | NSW-FE | NSW User System FE |
| 29 | MSG-ICP | MSG ICP |
| 31 | MSG-AUTH | MSG Authentication |
| 33 | DSP | Display Support Protocol |
| 35 | | any private printer server |
| **37** | **TIME** | **Time(tcp)/Time Service → Better use NTP !** |
| 39 | RLP | Resource Location Protocol |
| 41 | GRAPHICS | Graphics |
| **42** | **NAMESERVER** | **Host Name Server (IEN-116)** |
| 43 | NICNAME | Who Is |
| 44 | MPM-FLAGS | MPM FLAGS Protocol |
| 45 | MPM | Message Processing Module [recv] |
| 46 | MPM-SND | MPM [default send] |
| 47 | NI-FTP | NI FTP |
| **49** | **LOGIN** | **Login Host Protocol (TACACS)** |
| 51 | LA-MAINT | IMP Logical Address Maintenance |

| | | |
|---|---|---|
| **53** | **DNS** | **DNS - Domain Name Server**<br>- DNS zone transfer |
| 55 | ISI-GL | ISI Graphics Language |
| 57 | | any private terminal access |
| 58 | | XNS (Xerox Network Systems) Mail |
| 59 | | any private file service |
| 61 | NI-MAIL | NI-MAIL |
| 63 | VIA-FTP | VIA Systems - FTP |
| 65 | TACACS-DS | TACACS-Database Service |
| **67** | **BOOTPS** | **Bootstrap Protocol Server** |
| **68** | **BOOTPC** | **Bootstrap Protocol Client** |
| **69_udp** | **TFTP** | **TFTP (Trivial File Transfer Protocol)** |
| 70 | GOPHER | Gopher (Informationsdienst) |
| 71 | NETRJS-1 | Remote Job Service |
| 72 | NETRJS-2 | Remote Job Service |
| 73 | NETRJS-3 | Remote Job Service |
| 74 | NETRJS-4 | Remote Job Service |
| 75 | | any private dial our service |
| 77 | | any private RJE service |
| **79** | **FINGER** | **Finger** |
| **80** | **WWW** | **Worl-Wide-Web-Publikationsdienst(HTTP)** |
| **81** | **HOSTS2-NS** | **HOSTS2 Name Server** |
| 83 | MIT-ML-DEV | MIT ML Device |
| 85 | MIT-ML-DEV | MIT ML Device |
| 87 | | any private terminal link |
| **88_tcp** | **TCP** | **Kerberos** |
| **88_udp** | **UDP** | **Kerberos** |
| 89 | SU-MIT-TG | SU/MIT Telnet Gateway |
| 90 | DNSIX | DNSIX (DoD Network Security for IE) |
| 91 | MIT-DOV | MIT Dover Spooler |
| 93 | DCP | Device Control Protocol |
| 95 | SUPDUP | SUPDUP |
| 97 | SWIFT-RVF | Swift Remote Vitural File Protocol |
| 98 | TACNEWS | TAC News |
| 99 | METAGRAM | Metagram Relay |
| 101 | HOSTNAME | NIC Host Name Server |
| **102** | **ISO-TSAP** | **ISO-TSAP (DE)** |
| 103 | X400 | X400 |
| 104 | X400-SND | X400-SND |
| 105 | CSNET-NS | Mailbox Name Nameserver |
| 107 | RTELNET | Remote Telnet Service |
| **109** | **POP2** | **Post Office Protocol - Version 2** |
| **110** | **POP3** | **POP3 – Post Office Protocol - Version 3** |
| **111** | **SUNRPC** | **NFS / SUN Remote Procedure Call** |
| 113 | AUTH | Authentication Service |
| **115** | **SFTP** | **Simple File Transfer Protocol** |
| 117 | UUCP-PATH | UUCP Path Service |
| **119** | **NNTP** | **NNTP - Network News Transfer Protocol** |
| 121 | ERPC | Encore Expedited Remote Proc.edure Call |
| **123** | **NTP** | **Network Time Protocol (UDP)** |
| 125 | LOCUS-MAP | Locus PC-Interface Net Map Server |
| 127 | LOCUS-CON | Locus PC-Interface Conn Server |

| | | |
|---|---|---|
| 129 | PWDGEN | Password Generator Protocol |
| 130 | CISCO-FNA | CISCO FNATIVE |
| 131 | CISCO-TNA | CISCO TNATIVE |
| 132 | CISCO-SYS | CISCO SYSMAINT |
| 133 | STATSRV | Statistics Service |
| 134 | INGRES-NET | INGRES-NET Service |
| **135** | **NETBIOS LOC-SRV** | **Location Service** |
| 136 | PROFILE | PROFILE Naming System |
| **NT-Configuration Port 137-139** | | |
| **137** | **NETBIOS-NS** | **NETBIOS Name Service (nbname) / NBTSTAT** |
| **138** | **NETBIOS-DGM** | **NETBIOS Datagram Service (nbdatagram)** |
| **139** | **NETBIOS-SSN** | **NETBIOS Session Service (nbsession)** |
| 140 | EMFIS-DATA | EMFIS Data Service |
| 141 | EMFIS-CNTL | EMFIS Control Service |
| 142 | BL-IDM | Britton-Lee IDM |
| **143** | **IMAP2/IMAP4** | **Interim Mail Access Protocol v2** |
| 144 | NEWS | NewS |
| 145 | UAAC | UAAC Protocol |
| 146 | ISO-TP0 | ISO-IP0 |
| 147 | ISO-IP | ISO-IP |
| 148 | CRONUS | CRONUS-SUPPORT |
| 149 | AED-512 | AED 512 Emulation Service |
| 150 | SQL-NET | SQL-NET |
| 151 | HEMS | HEMS |
| 152 | BFTP | Background File Transfer Program |
| **153** | **SGMP** | **SGMP** |
| 154 | NETSC-PROD | NETSC |
| 155 | NETSC-DEV | NETSC |
| 156 | SQLSRV | SQL Service |
| 157 | KNET-CMP | KNET/VM Command/Message Protocol |
| 158 | PCMail-SRV | PCMail Server |
| 159 | NSS-Routing | NSS-Routing |
| 160 | SGMP-TRAPS | SGMP-TRAPS |
| **161_udp** | **SNMP** | **SNMP-Simple Network Management Protocol** |
| **162_udp** | **SNMPTRAP** | **SNMPTRAP** |
| 163 | CMIP-Manage | CMIP/TCP Manager |
| 164 | CMIP-Agent | CMIP/TCP Agent |
| 165 | XNS-Courier | Xerox |
| 166 | S-Net | Sirius Systems |
| 167 | NAMP | NAMP |
| 168 | RSVD | RSVD |
| 169 | SEND | SEND |
| 170 | Print-SRV | Network PostScript |
| 171 | Multiplex | Network Innovations Multiplex |
| 172 | CL/1 | Network Innovations CL/1 |
| 173 | Xyplex-MUX | Xyplex |
| 174 | MAILQ | MAILQ |
| 175 | VMNET | VMNET |
| 176 | GENRAD-MUX | GENRAD-MUX |
| 177 | XDMCP | X Display Manager Control Protocol |
| 178 | NextStep | NextStep Window Server |

| | | |
|---|---|---|
| **179** | **BGP** | **Border Gateway Protocol** |
| 180 | RIS | Intergraph |
| 181 | Unify | Unify |
| 182 | Unisys-Cam | Unisys-Cam |
| 183 | OCBinder | OCBinder |
| 184 | OCServer | OCServer |
| 185 | Remote-KIS | Remote-KIS |
| 186 | KIS | KIS Protocol |
| 187 | ACI | Application Communication Interface |
| 188 | MUMPS | MUMPS |
| 189 | QFT | Queued File Transport |
| 190 | GACP | Gateway Access Control Protocol |
| 191 | Prospero | Prospero |
| 192 | OSU-NMS | OSU Network Monitoring System |
| 193 | SRMP | Spider Remote Monitoring Protocol |
| **194** | **IRC** | **Internel Relay Chat Protocol** |
| 195 | DN6-NLM-AUD | DNSIX Network Level Module Audit |
| 196 | DN6-SMM-RED | DNSIX Session Mgt Module Audit Redirect |
| 197 | DLS | Directory Location Service |
| 198 | DLS-Mon | Directory Location Service Monitor |
| 198-200 | | Nicht festgelegt |
| 201 | AT-RMTP | AppleTalk Routing Maintenance |
| 202 | AT-NBP | AppleTalk Name Binding |
| 203 | AT-3 | AppleTalk Unused |
| 204 | AT-ECHO | AppleTalk Echo |
| 205 | AT-5 | AppleTalk Unused |
| 206 | AT-ZIS | AppleTalk Zone Information |
| 207 | AT-7 | AppleTalk Unused |
| 208 | AT-8 | AppleTalk Unused |
| 209-219 | | Nicht festgelegt |
| **220** | **IMAP3** | **IMAP3 (UDP)** |
| 221-223 | | Nicht festgelegt |
| 224-241 | | Reserviert |
| 243 | SUR-MEAS | Survey Measurement |
| 245 | LINK LINK | |
| 246 | DSP3270 | Display Systems Protocol |
| 247-255 | | Reserviert |
| 256 | TCP/UDP | 2DEV "2SP" Port |
| **389** | **LDAP** | **Lightweight Directory Access Protocol** |
| **427** | **SLP** | |
| **443** | **HTTPS** | **SSL Hypertext Transfer Protocol** |
| **445** | **SMB/CIFS** | **Microsoft-DS SMB-Shares (SAMBA)** |
| **464** | **Kerberos** | **Kerberos change/set password** |
| **465** | **SMTPS** | **SMTPS (TCP)** |
| **500** | **UDP** | **Internet Security Association and Key Management Protocol (ISAKMP)** |
| **500** | **TCP** | **DE (Checking for IPSec)** |
| **513** | **TCP** | **rlogin** |
| **514** | **RSH** | **Remote Shell (e.g. snort)** |
| **514_udp** | **Syslog** | **Syslog** |
| **515** | **Printer** | **Line Printer Daemon** |
| **520** | **RIP** | **Routing Information Protocol** |

| | | |
|---|---|---|
| **546** | **DHCPv6-Client** | **DHCPv6-Client** |
| **547** | **DHCPv6-Server** | **DHCPv6-Server** |
| **554** | **RTSP** | **Real Time Streaming Protocol (CCTV Sec Cameras)** |
| **587** | **SMTP** | **e-mail message submission (SMTP)** |
| **636** | **Active Directory(AD)** | **SSL SecureAuth** |
| **647_udp** | **DHCP Failover** | **10.12.1.11 / 10.12.1.10** |
| 953 | RNDC Service | Domain Name System (DNS) RNDC Service |
| 989 | FTPS-DATA | SSL FTP |
| **990** | **FTPS** | **FTPS Protocol (control): FTP over TLS/SSL** |
| 992 | TELNETS | SSL Telenet |
| 993 | IMAPS | SSL IMAP |
| 994 | IRCS | SSL IRCS |
| **995** | **POP3S** | **SSL POP3S (TCP)** |
| **Registered Ports 1024 - 49151** | | |
| **1025** | **NFS, IIS or Teradata** | |
| **1352** | **LN RPC** | **IBM Lotus Notes/Domino** |
| **1433** | **?** | **NOA (US)** |
| **1434** | **MSSQL** | **Microsoft SQL Server** |
| **1494** | **Citrix / Thin Client** | **Citrix XenApp Independent Computing Architecture (ICA) thin client protocol** |
| **1521** | **Oracle & nCube** | **Oracle Datenbank, zukünftig jedoch 2483** |
| **1533** | **IBM Sametime** | **IBM Sametime IM – Virtual Places Chat Microsoft SQL Server** |
| **1701_tcp** | **L2TP VPN** | **Virtual Private Networking** |
| **1701_udp** | **L2TP** | **MAC OS X Server VPN Service<br>Layer-2 Tunneling Protocol** |
| **1812** | **RADIUS** | **Radius Authentication** |
| **1853** | **GoToMeeting** | **GoToMeeting (Cytrix)** |
| 1900 udp | SSDP | UPnP Advertisement |
| **1985 udp** | **Cisco HSRP** | **Hot Standby Router Protocol** |
| **2049_udp** | **NFS** | **File Sharing Protocol** |
| **2080** | **US** | **US (see rule: 166662)** |
| **2111** | **DE** | **DE** |
| **2112** | **DE** | **DE** |
| **2218** | **?** | **Bounzza IRC Proxy** |
| **2404** | **?** | **Router Prorocol ?** |
| **2468** | **QIP** | **QIP (or Virus)** |
| **2502** | | **?** |
| 2712 | aocp | Axapta Object Comm Protocol |
| **3050** | **?** | **?** |
| 3128 | BR | See Ticket I1511-1201 / 762621 |
| **3130** | | |
| **3200** | **SAP** | **? SAPRouter** |
| **3201** | **SAP** | **DE Warehouse (EWx/WMx)** |
| **3221** | **SAP** | **SAPRouter Monitoring** |
| **3230** | **SAP** | **EWx/WMx** |
| **3237** | **SAP** | **SOD** |
| **3241** | **SAP** | **sapasp** |
| **3299** | **SAP-Router (SAP R/3)** | **SAP-Router (SAP R/3)** |
| **3301** | **SAP** | **DE Warehouse (SAP)** |
| **3302** | **?** | **?** |
| **3321** | **SAP** | **Standard port SAP (Monitoring)** |
| **3306** | **MySQL** | **MySQL database system** |
| **3330** | **SAP** | **RETARUS** |

| | | |
|---|---|---|
| **3353** | **SAP** | **RETARUS** |
| **3389** | **RDC/RDP** | **RDC/MSTSC** |
| 3544 | Teredo | To be closed by security! |
| **3601** | **SAP** | **DE Warehouse (SAP)** |
| **3651** | **SAP** | **RETARUS** |
| **3653** | **SAP** | **RETARUS** |
| **3655** | **SAP** | **? SAPRouter** |
| **3689** | **daap** | **Apple iTunes DAAP 11.1b37** |
| **3790** | **Metasploit** | **Metasploit** |
| **3982** | **?** | **?** |
| 4308 | | |
| **4500** | | **IPSec NAT Traversal (RFC 3947)** |
| **5000** | rtsp | AirTunes rtspd 220.68 |
| **5060** | **SIP** | **SIP (Telefonie)** |
| 5061 | SIPS | SSL SIP |
| **5080** | **US** | **US (see rule: 166662)** |
| 5190 | ICQ & AOL | ICQ & AOL Instant messenger |
| **5353** | **mDNS** | **Mulicast DNS** |
| **5354** | | |
| **5355_udp** | **LLMNR** | **10.12.8.19** |
| **5404** | **?** | **? HPOMS-DPS-LSTN** |
| **5405** | **?** | **? NetSupport** |
| **5444** | **?** | **?** |
| **5551** | **OSAG** | **OSAG** |
| **5555** | **MS Dynamics CRM** | **MS Dynamics CRM** |
| **5800** | **DE** | **DE** |
| **5666** | **NRPE** | **Nagios** |
| **5900** | **DE** | **DE** |
| **5938** | | **Teamviewer (RDP)** |
| **5939** | | |
| **5985** | **http** | **PowerShell / WinRM** |
| **5986** | **https** | **PowerShell / WinRM** |
| 6001 | | |
| 6068 | ANCP | Access Node Control Protocol |
| **6619** | **OFTP2** | **<X..>.<sik..>.com** |
| **7000** | **rtspd** | **AirTunes rtspd 220.68** |
| **7100** | **http** | **Apple AirPlay httpd** |
| **8001** | **SAP** | **EWx/WMx** |
| **8021** | **SAP** | **AED** |
| **8025** | **SAP** | **AED** |
| **8027** | **SAP** | **<sapaei>.<sap>.<sik.>.com** |
| **8037** | **SAP** | **SOD** |
| **8039** | **SAP** | **SOP Solman 7.1** |
| **8049** | **SAP** | **AEH** |
| **8050** | **SAP** | **AES** |
| **8080** | **Webcache** | **Webcache** |
| **8081** | | |
| **8088** | | |
| **8443** | **<Sik.> MobileIron** | **MobileIron** |
| **8530** | **WSUS** | **WSUS** |
| **8554** | **RTSP** | **Real Time Streaming Protocol (CCTV Sec Cameras)** |
| **8779** | **SMB2** | **SMB2** |

| | | |
|---|---|---|
| **8834** | **Nessus** | **Nessus** |
| 9020 | | |
| **9022** | | **I1505-1162** |
| 9021 | | |
| **9043** | **hpisisms05** | **SAP - hpisisms05** |
| **9080** | **PDL** | **PDL Data Stream** |
| **9100** | **PDL** | **PDL Data Stream** |
| **9184** | | |
| 9300 | IBM Cognos | IBM Cognos |
| **9443** | **hpisisms05** | **SAP - hpisisms05** |
| **9791** | | |
| 9929 | Nping echo | Nping echo |
| **9996** | **ACE/Netflow** | **ACE/Netflow** |
| **17500** | **Dropbox** | **Dropbox** |
| **25000** | **Arcserv** | **India** |
| **31337** | **Back Orifice** | **Back Orifice Activity** |
| **36609** | **SAP** | **BWx** |
| **37777** | **VideoObserve?** | **CN** |
| **50000** | **SAP** | **sapserv2 to US 10.32.x.x** |
| **50003** | **SAP** | **SOP Solman 7.1 (sapsop)** |
| **52200** | **SAP** | **SOP Solman 7.1** |
| **53500** | **SAP** | **SOP Solman 7.1** |
| **53700** | **SAP** | **SOP Solman 7.1** |
| **53900** | **SAP** | **SOP Solman 7.1 (sapsop)** |

# Abbreviations

| | |
|---|---|
| ANCP | Access Node Control Protocol (Port: 6068) |
| BACnet | Building Automation and Control Network |
| BRAS | Broadband Remote Access Servers |
| EPAN | Ethernet Protocol Analyzer |
| MIB | Management Information Base (ASN 1) |
| MSDU | MAC Service Data Unit (max. 2304 bytes) |
| NSE | Nmap Scripting Engine |
| OCTET | An eight-bit Byte |
| OID | Object IDs |
| ONC | Open Network Computing (var of RPC) |
| PDU | Protocol Data Unit (Segment) |
| PPI | Per-Packet Information (WLAN) |
| RTP | Real-time Transport Protocol |
| RTO | TCP Retransmission Timeout |
| SIP | Session Initiation Protocol |
| SOC | Security Operation Center |
| SRT | Service Response Time |
| TSO | TCP Segmentation Offload |

# Table of Figures

Figure 1: Configure View ..... 11
Figure 2: SPFB ..... 15
Figure 3: TAP ..... 16
Figure 4: The top 10 reasons for a slow network ..... 17
Figure 5: Latency ..... 18
Figure 6: How to capture ..... 23
Figure 7: Security Tasks for Network Analysts ..... 23
Figure 8: Time Display Formats ..... 24
Figure 9: TCP Header ..... 30
Figure 10: ICMP Header ..... 31
Figure 11: TCP Options ..... 37
Figure 12: UDP Header ..... 39
Figure 13: IPv4 Header ..... 41

# Tables

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

# Index

.pcapng .....5
**ARP** .....41
ASDU .....68
BPF .....5, 62
**DiffServ** .....40
DSCP .....40
GARP .....42
lipcap .....5
**MTU** .....22
NFS .....69
NOP .....34
PDU .....15
Phantom Byte .....35
PORT .....67, 72
RARP .....41
SLAAC .....41
**TACACS** .....72
TAP .....5
**TTL** .....51
WinPcap .....5