

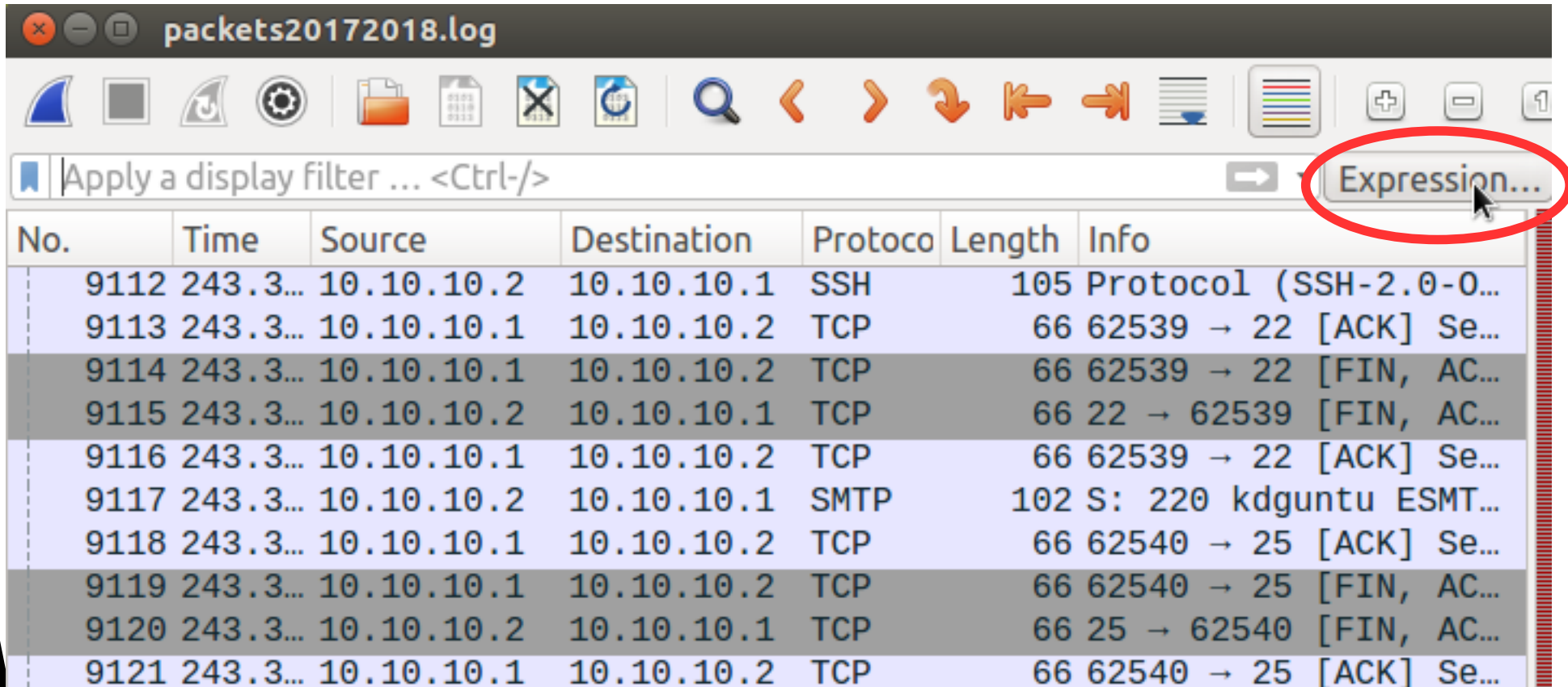
Portscan logs  
wireshark, nmap, p0f, ettercap, snort

# Scans

- **nmap 192.168.56.100-254**
- **TCP ACK scan met source port**
  - **nmap --source-port 8080 -sA 192.168.56.103**
- **XMAS scan**
  - **nmap -sX 192.168.56.103**
- **TCP connect opeenvolgende poorten**
  - **nmap -r -sT 192.168.56.103**
- **Decoy scan**
  - **nmap -D 192.168.1.5,192.86.32.91,209.33.28.4 192.168.56.100 192.168.56.103**

# Wie is het slachtoffer?

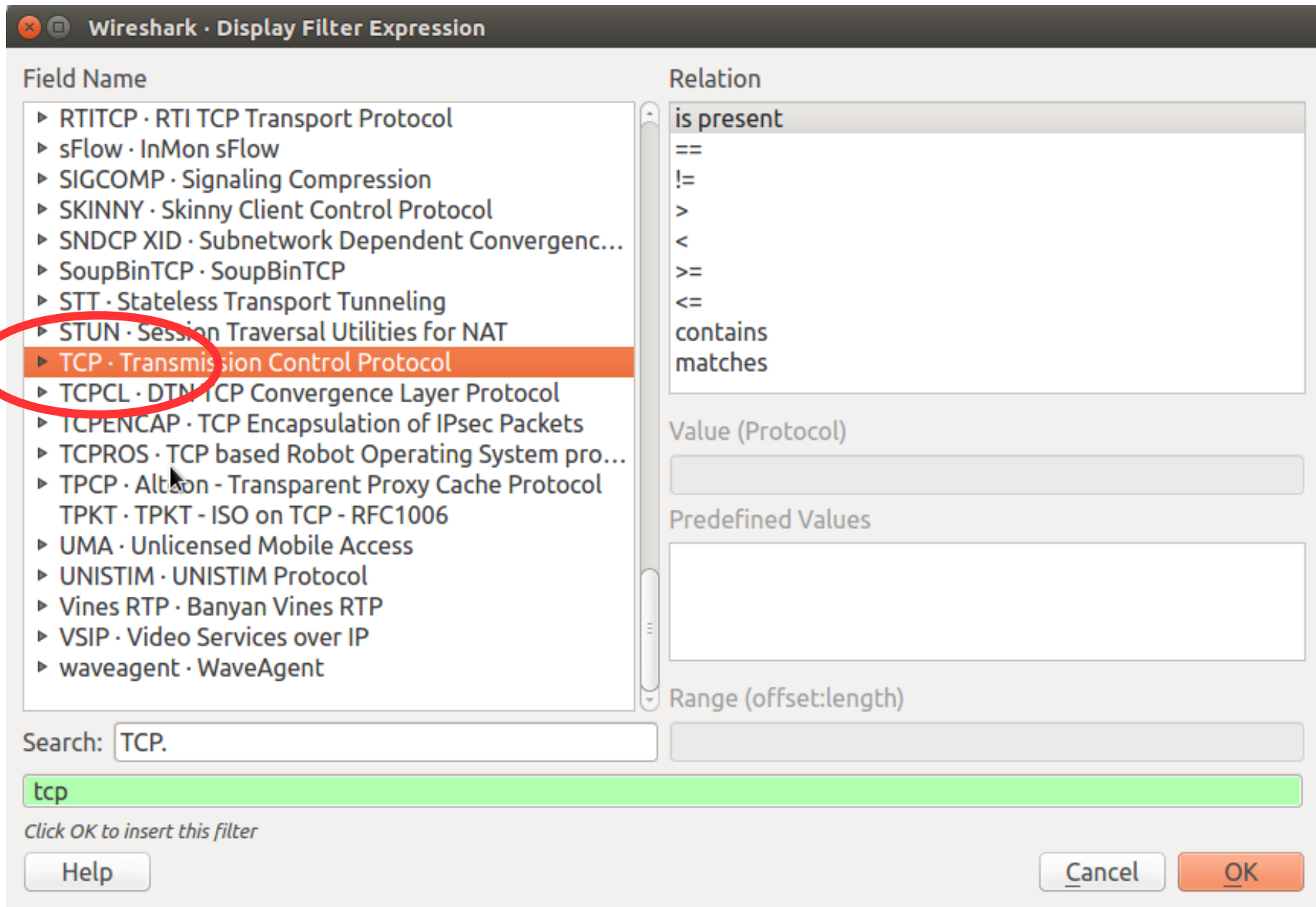
- Diegene die met een SYN ACK antwoord
- Wireshark klik expression



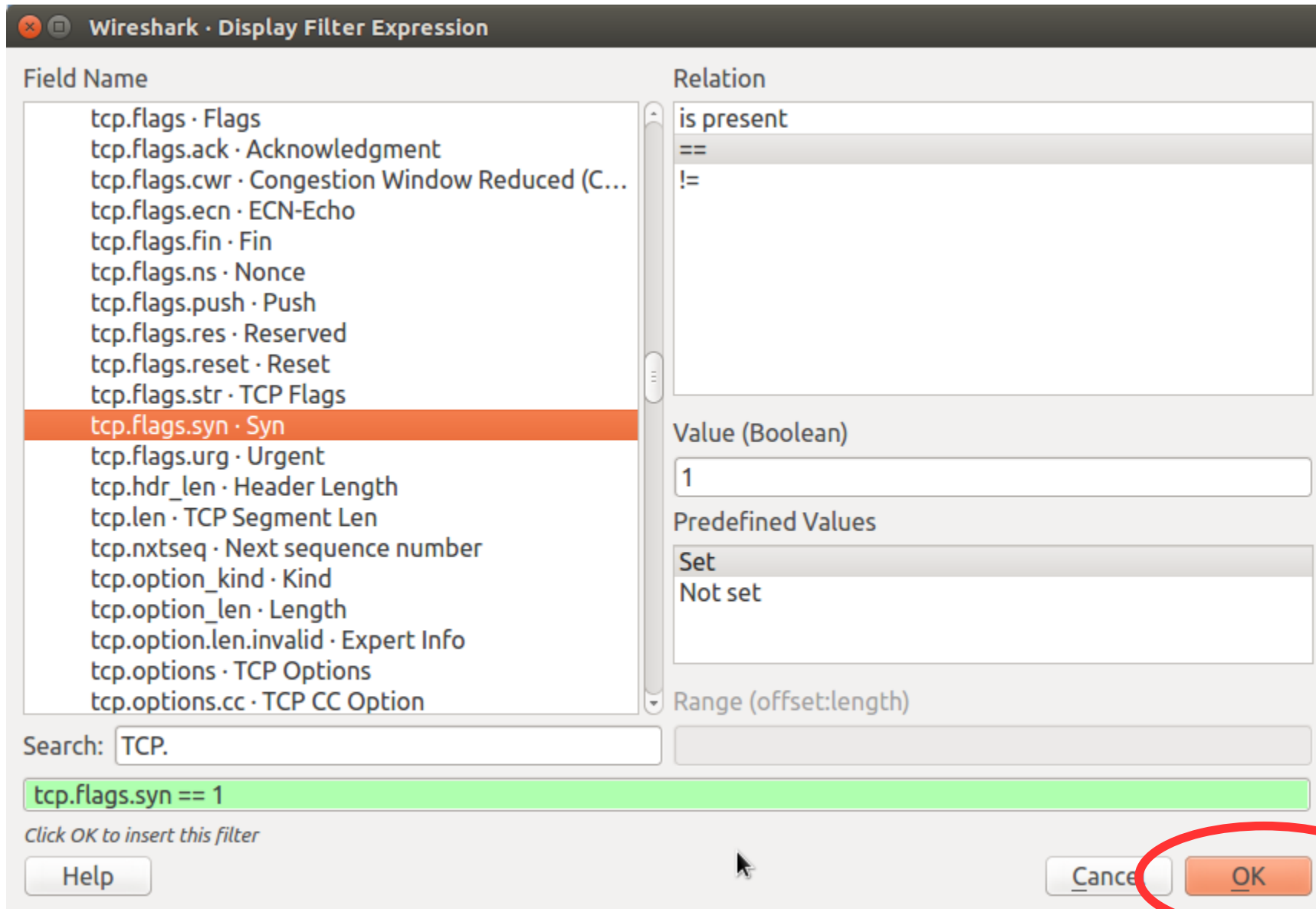
The screenshot shows the Wireshark interface with a packet capture named 'packets20172018.log'. The display filter bar contains the text 'Apply a display filter ... <Ctrl-/>' and a red circle highlights the 'Expression...' button. Below the filter bar is a table of captured packets.

No.	Time	Source	Destination	Protocol	Length	Info
9112	243.3...	10.10.10.2	10.10.10.1	SSH	105	Protocol (SSH-2.0-0...
9113	243.3...	10.10.10.1	10.10.10.2	TCP	66	62539 → 22 [ACK] Se...
9114	243.3...	10.10.10.1	10.10.10.2	TCP	66	62539 → 22 [FIN, AC...
9115	243.3...	10.10.10.2	10.10.10.1	TCP	66	22 → 62539 [FIN, AC...
9116	243.3...	10.10.10.1	10.10.10.2	TCP	66	62539 → 22 [ACK] Se...
9117	243.3...	10.10.10.2	10.10.10.1	SMTP	102	S: 220 kdguntu ESMT...
9118	243.3...	10.10.10.1	10.10.10.2	TCP	66	62540 → 25 [ACK] Se...
9119	243.3...	10.10.10.1	10.10.10.2	TCP	66	62540 → 25 [FIN, AC...
9120	243.3...	10.10.10.2	10.10.10.1	TCP	66	25 → 62540 [FIN, AC...
9121	243.3...	10.10.10.1	10.10.10.2	TCP	66	62540 → 25 [ACK] Se...

# Wireshark SYN-ACK



# Wireshark SYN-ACK



# Wireshark SYN-ACK

packets20172018.log

tcp.flags.syn == 1

No.	Time	Source	Destination	Protocol	Length	Info
9051	121.9...	10.10.10.1	10.10.10.2	TCP	78	62537 → 65389 [SYN] Seq=0 Win=...
9082	243.2...	10.10.10.1	10.10.10.2	TCP	60	47457 → 80 [SYN] Seq=0 Win=307...
9083	243.2...	10.10.10.2	10.10.10.1	TCP	58	80 → 47457 [SYN, ACK] Seq=0 Ac...
9084	243.2...	10.10.10.1	10.10.10.2	TCP	60	47457 → 445 [SYN] Seq=0 Win=40...
9086	243.2...	10.10.10.1	10.10.10.2	TCP	60	47457 → 25 [SYN] Seq=0 Win=204...
9087	243.2...	10.10.10.2	10.10.10.1	TCP	58	25 → 47457 [SYN, ACK] Seq=0 Ac...
9088	243.2...	10.10.10.1	10.10.10.2	TCP	60	47457 → 110 [SYN] Seq=0 Win=40...
9090	243.2...	10.10.10.1	10.10.10.2	TCP	60	47457 → 139 [SYN] Seq=0 Win=30...
9092	243.2...	10.10.10.1	10.10.10.2	TCP	60	47457 → 22 [SYN] Seq=0 Win=307...
9093	243.2...	10.10.10.2	10.10.10.1	TCP	58	22 → 47457 [SYN, ACK] Seq=0 Ac...
9094	243.2...	10.10.10.1	10.10.10.2	TCP	60	47457 → 23 [SYN] Seq=0 Win=102...

▶ Frame 9094: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)

▶ Ethernet II, Src: Apple\_53:2e:a6 (00:26:bb:53:2e:a6), Dst: HewlettP\_83:2d:70 (00:1a:4b:83:2d:70)

▶ Internet Protocol Version 4, Src: 10.10.10.1, Dst: 10.10.10.2

▶ Transmission Control Protocol, Src Port: 47457, Dst Port: 23, Seq: 0, Len: 0

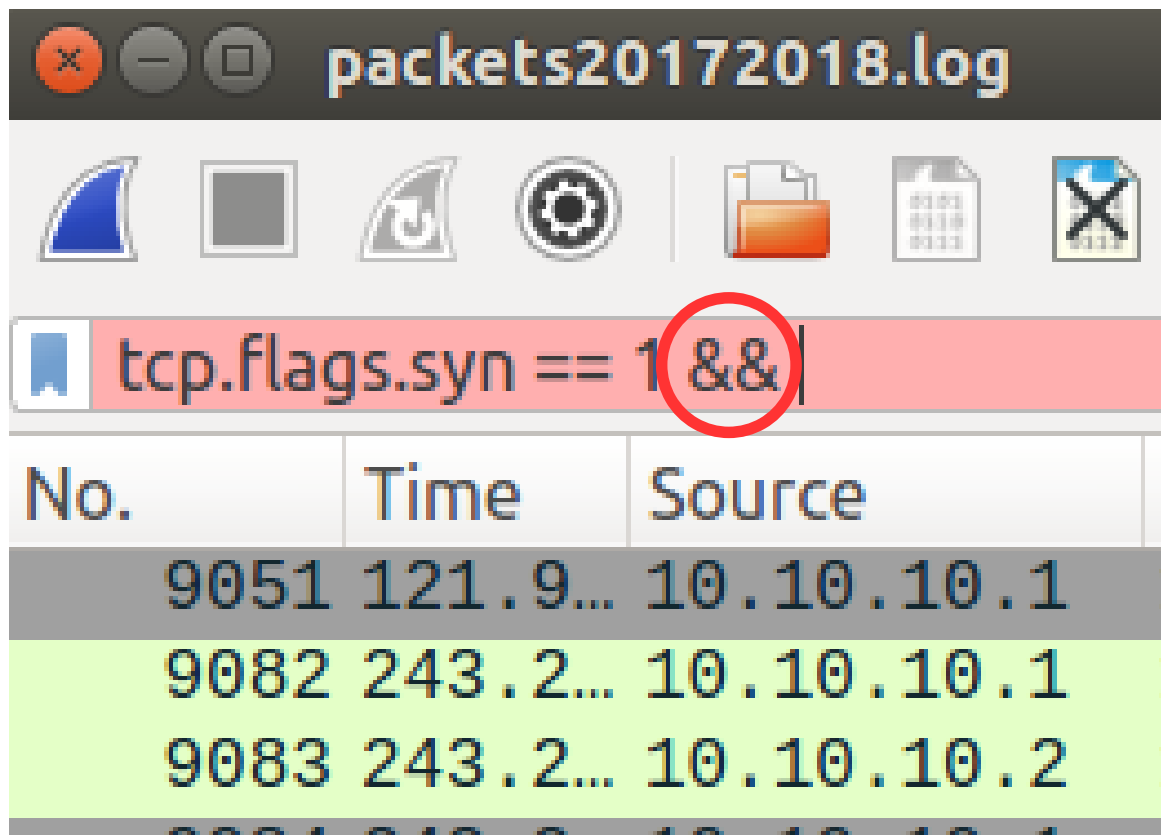
0000 00 1a 4b 83 2d 70 00 26 bb 53 2e a6 08 00 45 00 ..K.-p.& .S....E.  
0010 00 2c 7f 80 00 00 28 06 eb 35 0a 0a 0a 01 0a 0a .,....(. .5.....  
0020 0a 02 b9 61 00 17 1d cc 75 37 00 00 00 00 60 02 ...a.... u7....`.  
0030 04 00 1f 94 00 00 02 04 05 b4 00 00 ..... .....

packets20172018 Packets: 9504 · Displayed: 5094 (53.6%) · Load time: 0:0.74 Profile: Default

# Wireshark SYN ACK

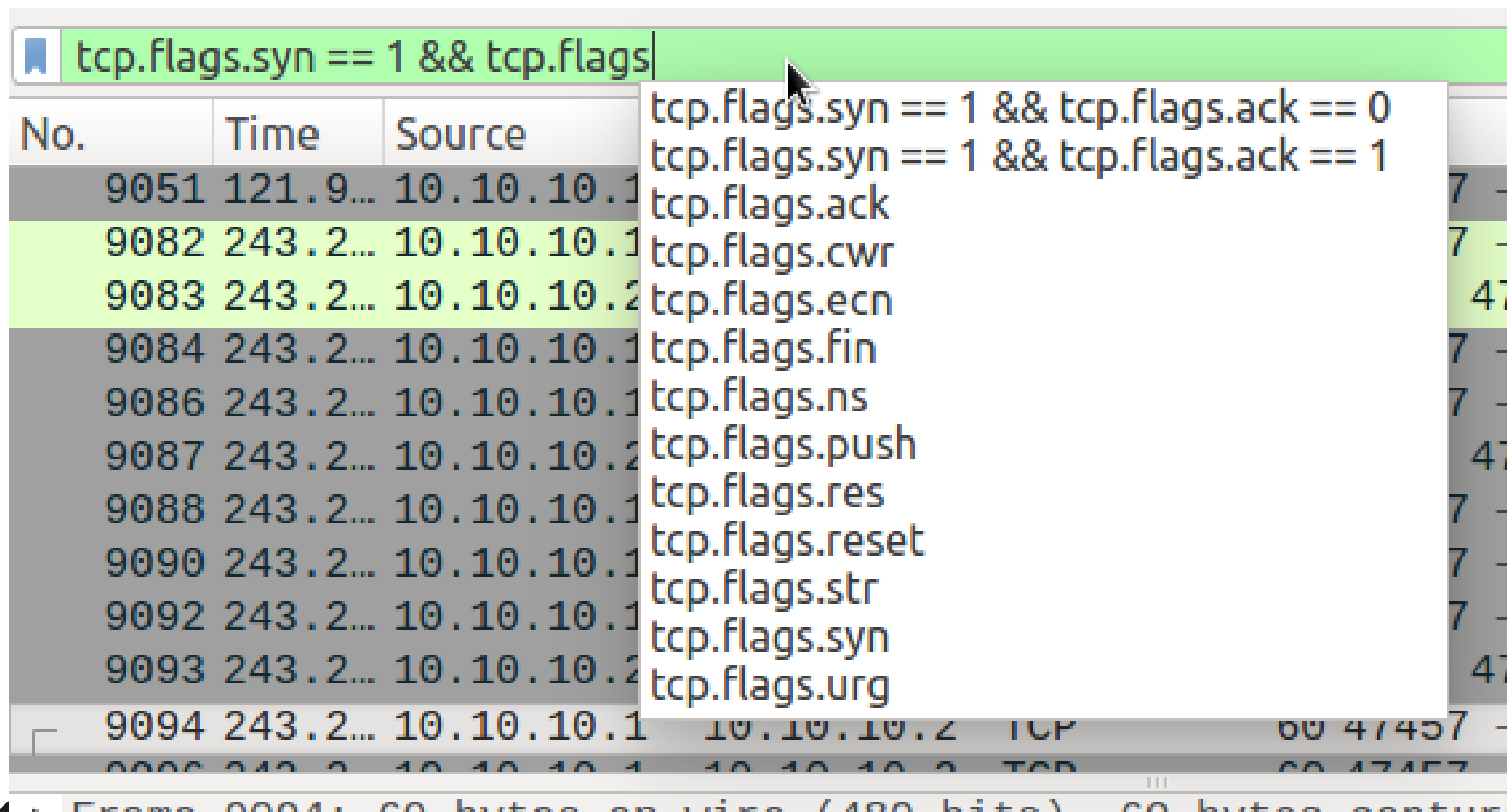
## ■ Logische AND &&

### ■ of OR ||



# Wireshark SYN ACK

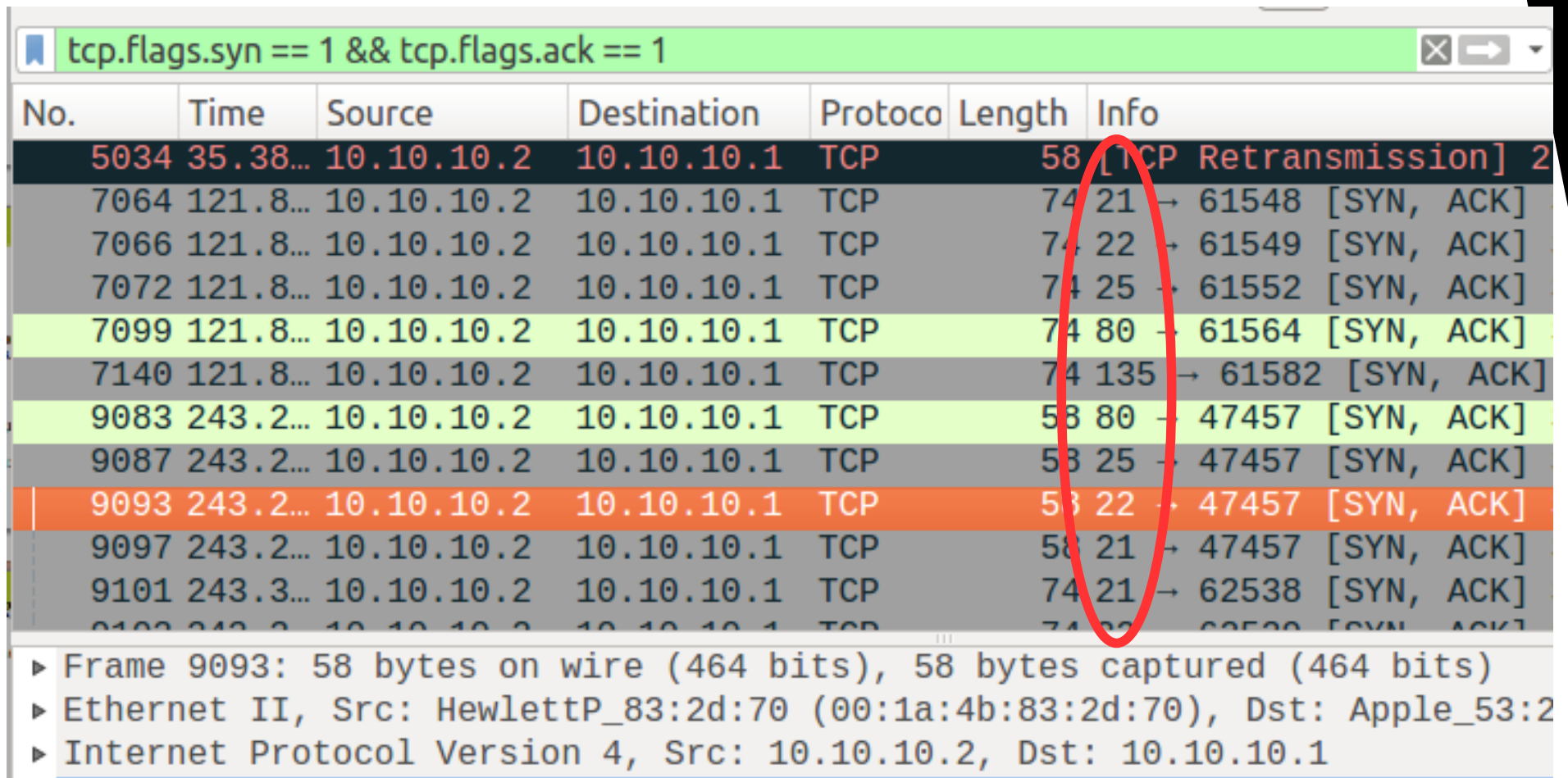
- Autocomplete
- Alternatief is voor SYN-ACK binair `tcp.flags == 0x0012`





# Wireshark SYN ACK

- Poorten die open staan, want ze antwoorden met SYN ACK



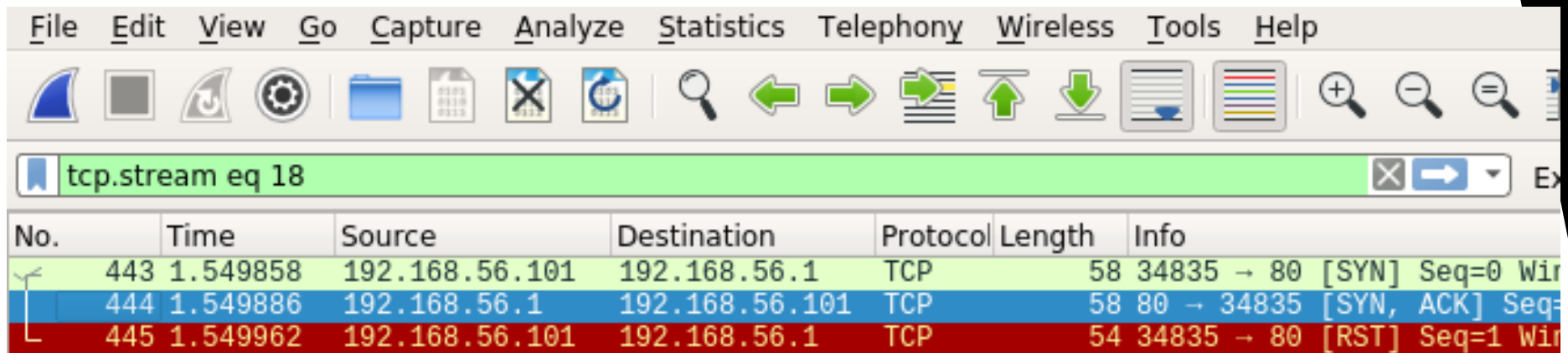
tcp.flags.syn == 1 && tcp.flags.ack == 1

No.	Time	Source	Destination	Protocol	Length	Info
5034	35.38...	10.10.10.2	10.10.10.1	TCP	58	[TCP Retransmission] 2
7064	121.8...	10.10.10.2	10.10.10.1	TCP	74	21 → 61548 [SYN, ACK]
7066	121.8...	10.10.10.2	10.10.10.1	TCP	74	22 → 61549 [SYN, ACK]
7072	121.8...	10.10.10.2	10.10.10.1	TCP	74	25 → 61552 [SYN, ACK]
7099	121.8...	10.10.10.2	10.10.10.1	TCP	74	80 → 61564 [SYN, ACK]
7140	121.8...	10.10.10.2	10.10.10.1	TCP	74	135 → 61582 [SYN, ACK]
9083	243.2...	10.10.10.2	10.10.10.1	TCP	58	80 → 47457 [SYN, ACK]
9087	243.2...	10.10.10.2	10.10.10.1	TCP	58	25 → 47457 [SYN, ACK]
9093	243.2...	10.10.10.2	10.10.10.1	TCP	58	22 → 47457 [SYN, ACK]
9097	243.2...	10.10.10.2	10.10.10.1	TCP	58	21 → 47457 [SYN, ACK]
9101	243.3...	10.10.10.2	10.10.10.1	TCP	74	21 → 62538 [SYN, ACK]
9103	243.3...	10.10.10.2	10.10.10.1	TCP	74	22 → 62538 [SYN, ACK]

▶ Frame 9093: 58 bytes on wire (464 bits), 58 bytes captured (464 bits)  
▶ Ethernet II, Src: HewlettP\_83:2d:70 (00:1a:4b:83:2d:70), Dst: Apple\_53:2  
▶ Internet Protocol Version 4, Src: 10.10.10.2, Dst: 10.10.10.1

# Voorbeeldfilters

- Op elke lijn in wireshark kan je rechtermuistoets, Follow, TCP stream uitvoeren



- TCP connect

- `tcp.flags.reset == 1 && tcp.flags.ack == 1`

- NULL scan

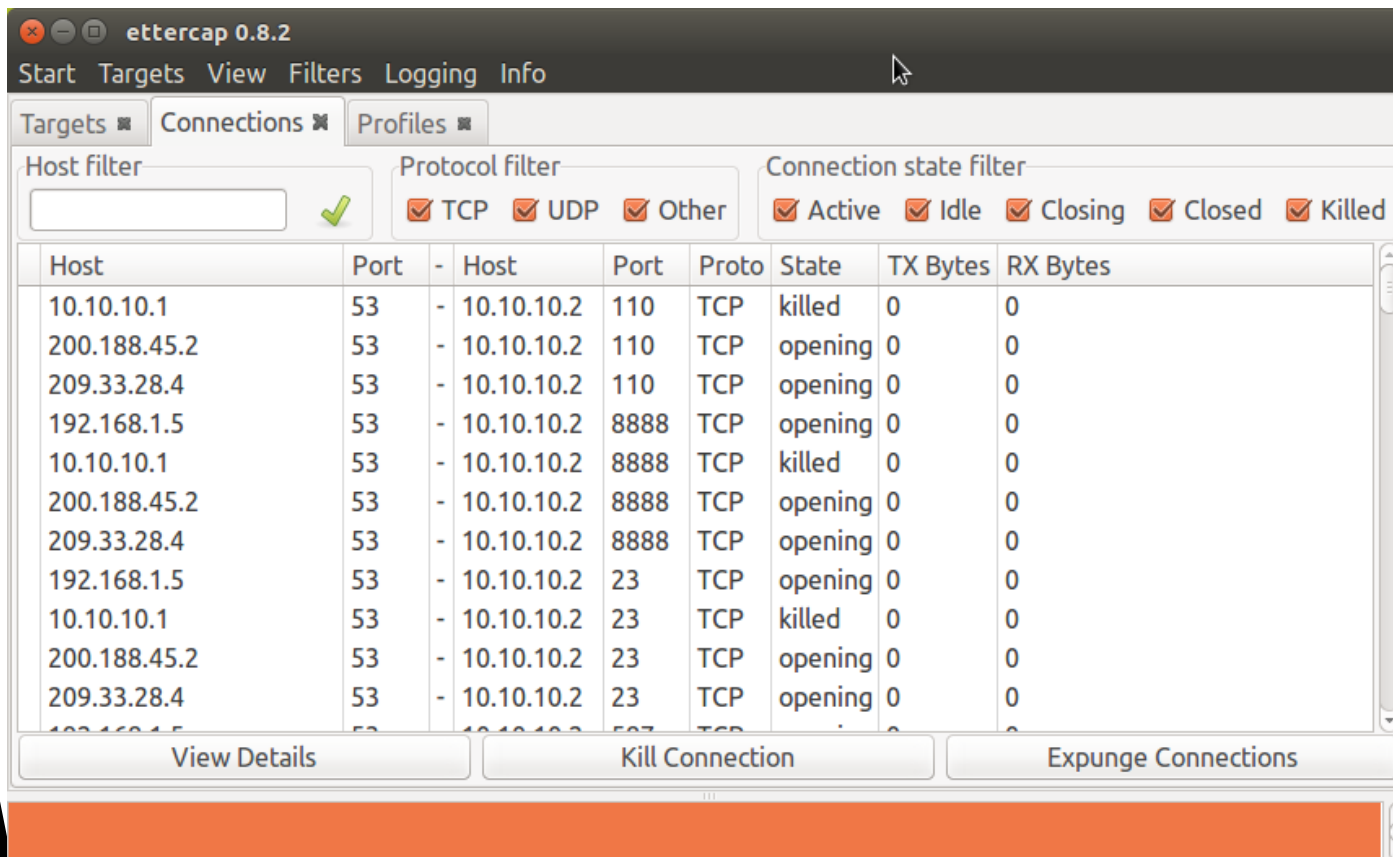
- `tcp.flags.reset == 0 && tcp.flags.ack == 0 && tcp.flags.syn == 0 && tcp.flags.fin == 0`

# Waarom een fixed source port?

- **Attacker hoopt dat in de firewall er misschien regels staan die ALLE verkeer komend van die poort doorlaten. Denk aan:**
  - 53
  - 80
  - 443

# Ettercap

- `sudo apt-get install ettercap-graphical`
  - of `sudo apt-get install ettercap-text-only`
- `ettercap -G, File, Open -> pcap log`



# Ettercap profiles

The screenshot shows the Ettercap 0.8.2 application window. The 'Targets' tab is active, displaying a list of IP addresses and hostnames. The 'Profile Details' window is open, showing information for the target 10.10.10.2.

**Ettercap 0.8.2 Targets:**

IP Address	Hostname
10.10.10.1	
10.10.10.2	
192.168.1.5	
200.188.45.2	
209.33.28.4	209-33-28-4.sym
fe80::226:bbff:fe53:2ea6	

**Profile Details for 10.10.10.2:**

**Host Information:**

- IP address: 10.10.10.2
- Hostname:
- MAC address: 00:1A:4B:83:2D:70
- Manufacturer: Hewlett-Packard Company

**Connectivity Information:**

- Distance: 1
- Type: unknown

**OS and Service Information:**

- Fingerprint: 16A0:05B4:40:07:1:1:1:1:A:3C
- Operating System: unknown fingerprint (please submit it)  
Neares one is: Linux.2.4.20-web100
- Fingerprint: TCP 21 ftp []
- Fingerprint: TCP 22 ssh []
- Fingerprint: TCP 25 smtp []
- Fingerprint: TCP 80 http []
- Fingerprint: TCP 135 loc-srv []
- Fingerprint: UDP 5353 []

# Fingerprinting

- Hoe wordt bepaald welk OS gedraaid werd?
  - Window Size
  - TCP Options
  - Soorten FLAGS
  - Actieve OS fingerprinting:
    - Een pakket opsturen
    - Zien wat de reactie is

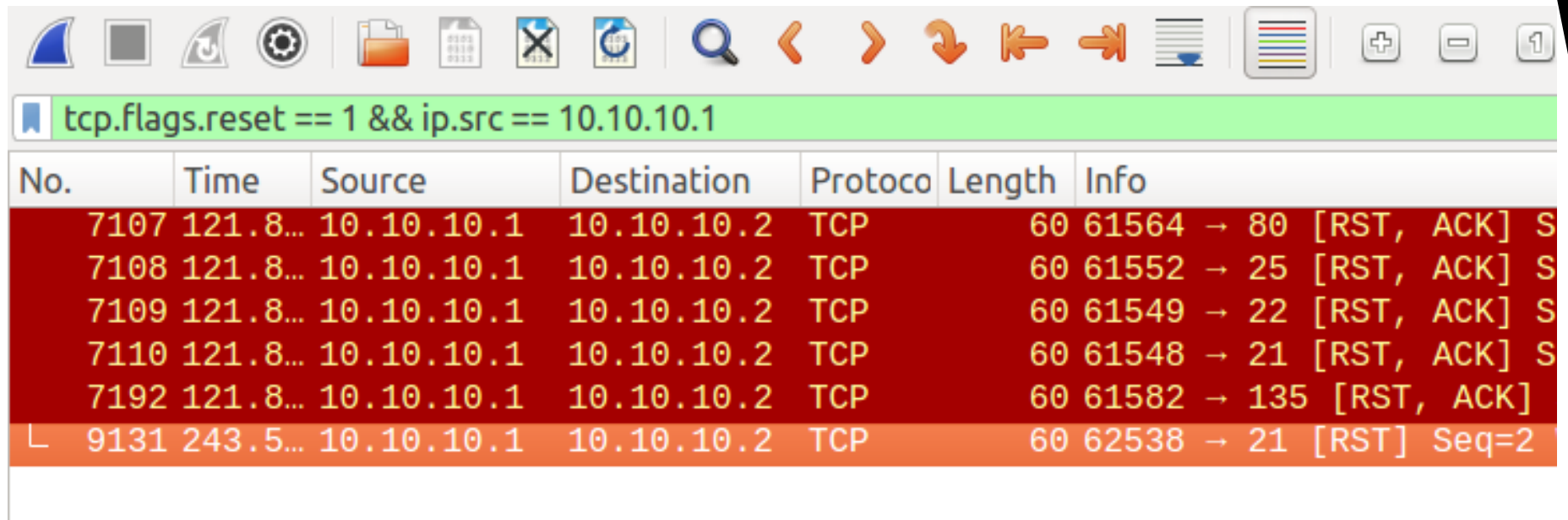
# Fingerprinting p0f

- `sudo apt-get install p0f`
- Leest fingerprints uit pcap log
- Opgelet, dus ook die van het slachtoffer
- `p0f -r portscan.pcap`

```
.-[ 192.168.56.101/39650 -> 192.168.56.1/22 (syn) ]-  
|  
| client      = 192.168.56.101/39650  
| os          = Linux 2.2.x-3.x  
| dist        = 0  
| params      = generic  
| raw_sig     = 4:64+0:0:1460:mss*3,6:mss,sok,ts,nop,ws:df,id+:0  
|  
`-----
```

# Welke pakketten zijn echt?

- Deze bij een TCP scan. Aanvaller stuurt een RST,ACK
- `tcp.flags.reset == 1 && tcp.flags.ack == 1`



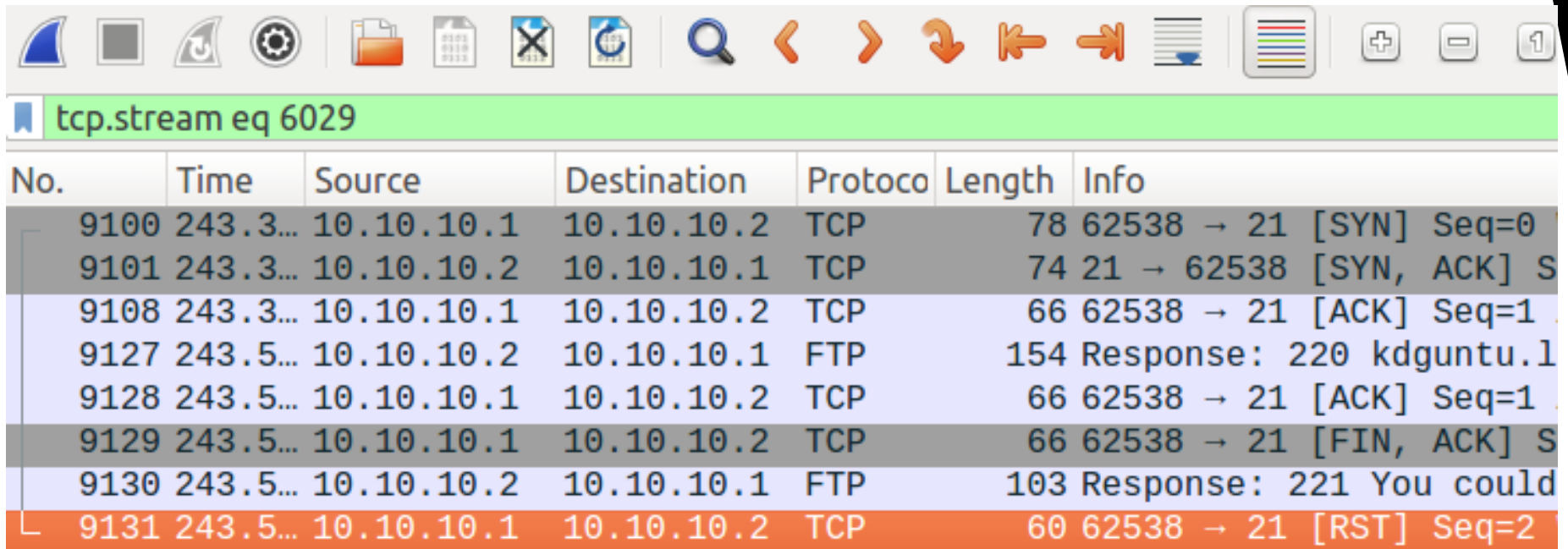
The image shows a Wireshark packet capture window. The filter bar at the top contains the expression `tcp.flags.reset == 1 && ip.src == 10.10.10.1`. Below the filter, a list of packets is displayed. The first five packets (7107-7110) and the last packet (9131) are highlighted in red, indicating they match the filter. These packets are all TCP segments from source 10.10.10.1 to destination 10.10.10.2, with the RST flag set. The sixth packet (7192) is highlighted in orange and does not match the filter as its RST flag is not set.

No.	Time	Source	Destination	Protocol	Length	Info
7107	121.8...	10.10.10.1	10.10.10.2	TCP	60	61564 → 80 [RST, ACK] S
7108	121.8...	10.10.10.1	10.10.10.2	TCP	60	61552 → 25 [RST, ACK] S
7109	121.8...	10.10.10.1	10.10.10.2	TCP	60	61549 → 22 [RST, ACK] S
7110	121.8...	10.10.10.1	10.10.10.2	TCP	60	61548 → 21 [RST, ACK] S
7192	121.8...	10.10.10.1	10.10.10.2	TCP	60	61582 → 135 [RST, ACK]
L 9131	243.5...	10.10.10.1	10.10.10.2	TCP	60	62538 → 21 [RST] Seq=2



# Welke pakketten zijn echt? TCP stream

- Op connectie, Rechtermuis, Follow, Stream
  - FTP klaagt dat RST FIN niet vriendelijk is
  - Aanvaller stuurt vriendelijk een RST



No.	Time	Source	Destination	Protocol	Length	Info
9100	243.3...	10.10.10.1	10.10.10.2	TCP	78	62538 → 21 [SYN] Seq=0
9101	243.3...	10.10.10.2	10.10.10.1	TCP	74	21 → 62538 [SYN, ACK] S
9108	243.3...	10.10.10.1	10.10.10.2	TCP	66	62538 → 21 [ACK] Seq=1
9127	243.5...	10.10.10.2	10.10.10.1	FTP	154	Response: 220 kdguntu.1
9128	243.5...	10.10.10.1	10.10.10.2	TCP	66	62538 → 21 [ACK] Seq=1
9129	243.5...	10.10.10.1	10.10.10.2	TCP	66	62538 → 21 [FIN, ACK] S
9130	243.5...	10.10.10.2	10.10.10.1	FTP	103	Response: 221 You could
9131	243.5...	10.10.10.1	10.10.10.2	TCP	60	62538 → 21 [RST] Seq=2

# SNORT

- **sudo apt-get install snort**
  - **snort vraagt de naam van je netwerkpoort. Dat is niet eth0!  
De echte naam kan je bekijken met het commando ip address**
- **1. Editeer /etc/snort/snort.conf, zoek op “Portscan”**
- **2. Zet de preprocessor sfportscan uit commentaar en voeg wat velden toe:**
  - **preprocessor sfportscan: proto { all } scan\_type { all } memcap { 1000000 } logfile { portscans.log } sense\_level { high }**
- **3. Draaien van snort met pcap log**  
**snort -r portscan.pcap -c /etc/snort/snort.conf -A full**
- **4. Bekijk de log in /var/log/snort/alert**

# Snort output voorbeeld

■ **[\*\*] [1:469:3] ICMP PING NMAP [\*\*]**  
**[Classification: Attempted Information Leak] [Priority: 2]**  
**08/27-00:59:50.250759 10.10.10.1 -> 10.10.10.2**  
**ICMP TTL:40 TOS:0x0 ID:34827 IpLen:20 DgmLen:28**  
**Type:8 Code:0 ID:31465 Seq:0 ECHO**

# Andere filters

- In de wireshark Filter:

- `(ip.addr eq 10.10.10.1) and (tcp.port eq 12345)`
- `ip.addr eq 10.10.10.1 && tcp.port eq 12345`

- Voorbeeld om logs te filteren:

- `tcpdump -n -r portscan.log | cut -d ' ' -f3  
| cut -d '.' -f5 > allesourcepoorten.txt`

# Referenties

## ■ Transport Control Protocol

- Starten/stoppen van een TCP verbinding
- <http://www.ietf.org/rfc/rfc793.txt>

## ■ Port scanners

<https://resources.infosecinstitute.com/port-scanning-using-scapy/>

## ■ Snort portscan preprocessor

<https://www.snort.org/faq/readme-sfportscan>

## ■ Requirements for Internet Hosts

- Hfst 4 Verklaringen TCP flags
- <http://www.ietf.org/rfc/rfc1122.txt>