

Thema

Laborprotokoll TCP-UPD Header Verleich



Figure 1: memes klauen ist nicht ethisch

Unterrichtsgegenstand: NWT1|ZIVK
Jahrgang: 2BHIT
Name: Stefan Fürst, Marcel Raichle
Betreuer: ZIVK
Übungsdaten: 24.5.2024, 31.5.2024
Abgabedatum: Datum

Contents

1	Aufgabenstellung	3
2	Zusammenfassung	3
3	Vollständige Netzwerktopologie der gesamten Übung	4
4	Übungsdurchführung	5
4.1	Aufsetzen der Server	5
4.1.1	Verbinden mit dem Sever (TCP)	5
4.1.2	TCP-Verbindungsaufbau	5
4.1.3	Nachrichten verschicken und empfangen	5
4.1.4	TCP-Flags	6
4.1.5	TCP-Header	7
4.1.6	Verbindungsabbruch	7
4.1.7	Verbinden mit dem Sever (UDP)	8
4.1.8	UDP-Verbindungsaufbau	8
4.1.9	Nachrichten verschicken und empfangen	8
5	Vollständige Konfigurationsdateien	9
6	Abbildungsverzeichnis	10

1 Aufgabenstellung

TCP-UDP Header vergleich.

2 Zusammenfassung

Netcad um einen TCP/UPD Server starten, mit Netcad verbinden und mit Wireshark die Verbindungen analysieren. test

3 Vollständige Netzwerktopologie der gesamten Übung

4 Übungsdurchführung

4.1 Aufsetzen der Server

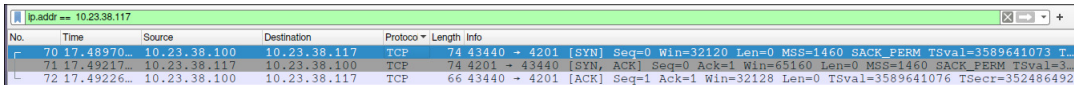
TCP nc -l -p 5000

UDP nc -l -u -p 5000

4.1.1 Verbinden mit dem Sever (TCP)

nc 10.23.38.117 4201

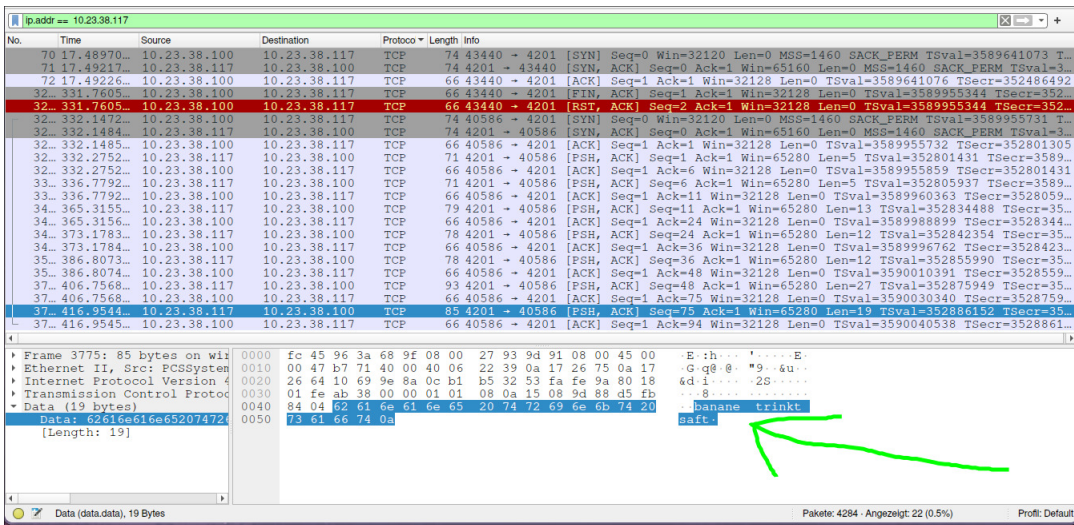
4.1.2 TCP-Verbindungsaufbau



No.	Time	Source	Destination	Protocol	Length	Info
70	17.48970...	10.23.38.100	10.23.38.117	TCP	74	43440 → 4201 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=3589641073 T...
71	17.49217...	10.23.38.117	10.23.38.100	TCP	74	4201 → 43440 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3...
72	17.49226...	10.23.38.100	10.23.38.117	TCP	66	43440 → 4201 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=3589641076 TSecr=352486492

Figure 2: TCP 3 Way Handshake

4.1.3 Nachrichten verschicken und empfangen



No.	Time	Source	Destination	Protocol	Length	Info
70	17.48970...	10.23.38.100	10.23.38.117	TCP	74	43440 → 4201 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=3589641073 T...
71	17.49217...	10.23.38.117	10.23.38.100	TCP	74	4201 → 43440 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3...
72	17.49226...	10.23.38.100	10.23.38.117	TCP	66	43440 → 4201 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=3589641076 TSecr=352486492
32	331.7605...	10.23.38.100	10.23.38.117	TCP	66	43440 → 4201 [RST, ACK] Seq=2 Ack=1 Win=32128 Len=0 TSval=3589955344 TSecr=352...
32	332.1472...	10.23.38.100	10.23.38.117	TCP	74	40586 → 4201 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=3589955731 T...
32	332.1484...	10.23.38.117	10.23.38.100	TCP	74	4201 → 40586 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3...
32	332.1485...	10.23.38.100	10.23.38.117	TCP	66	40586 → 4201 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=3589955732 TSecr=352801305
32	332.2752...	10.23.38.117	10.23.38.100	TCP	71	4201 → 40586 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=5 TSval=352801431 TSecr=3589...
32	332.2752...	10.23.38.100	10.23.38.117	TCP	66	40586 → 4201 [ACK] Seq=1 Ack=6 Win=32128 Len=0 TSval=3589955859 TSecr=352801431
33	336.7792...	10.23.38.117	10.23.38.100	TCP	71	4201 → 40586 [PSH, ACK] Seq=6 Ack=1 Win=65280 Len=5 TSval=352805937 TSecr=3589...
33	336.7792...	10.23.38.100	10.23.38.117	TCP	66	40586 → 4201 [ACK] Seq=1 Ack=11 Win=32128 Len=0 TSval=3589960363 TSecr=3528059...
34	365.3155...	10.23.38.117	10.23.38.100	TCP	79	4201 → 40586 [PSH, ACK] Seq=11 Ack=1 Win=65280 Len=13 TSval=352834488 TSecr=35...
34	365.3156...	10.23.38.100	10.23.38.117	TCP	66	40586 → 4201 [ACK] Seq=1 Ack=24 Win=32128 Len=0 TSval=3589988899 TSecr=3528344...
34	373.1783...	10.23.38.117	10.23.38.100	TCP	78	4201 → 40586 [PSH, ACK] Seq=24 Ack=1 Win=65280 Len=12 TSval=352842354 TSecr=35...
34	373.1784...	10.23.38.100	10.23.38.117	TCP	66	40586 → 4201 [ACK] Seq=1 Ack=36 Win=32128 Len=0 TSval=358996762 TSecr=3528423...
35	386.8073...	10.23.38.117	10.23.38.100	TCP	78	4201 → 40586 [PSH, ACK] Seq=36 Ack=1 Win=65280 Len=12 TSval=352855990 TSecr=35...
35	386.8074...	10.23.38.100	10.23.38.117	TCP	66	40586 → 4201 [ACK] Seq=1 Ack=48 Win=32128 Len=0 TSval=3590010391 TSecr=3528559...
37	406.7568...	10.23.38.117	10.23.38.100	TCP	93	4201 → 40586 [PSH, ACK] Seq=48 Ack=1 Win=65280 Len=27 TSval=352875949 TSecr=35...
37	406.7568...	10.23.38.100	10.23.38.117	TCP	66	40586 → 4201 [ACK] Seq=1 Ack=75 Win=32128 Len=0 TSval=3590030340 TSecr=3528759...
37	416.9544...	10.23.38.117	10.23.38.100	TCP	85	4201 → 40586 [PSH, ACK] Seq=75 Ack=1 Win=65280 Len=19 TSval=352886152 TSecr=35...
37	416.9545...	10.23.38.100	10.23.38.117	TCP	66	40586 → 4201 [ACK] Seq=1 Ack=94 Win=32128 Len=0 TSval=3590040538 TSecr=3528861...

Frame 3775: 85 bytes on wire (0x55)	Ethernet II, Src: PCSysSystem (08:00:00:00:00:00), Dst: 10.23.38.117 (08:00:00:00:00:00)	Internet Protocol Version 4, Src: 10.23.38.100, Dst: 10.23.38.117	Transmission Control Protocol, Src Port: 4201, Dst Port: 40586	Data (19 bytes)
				0000 fc 45 96 3a 68 9f 08 00 27 93 9d 91 08 00 45 00 0010 00 47 b7 71 40 00 04 06 22 39 0a 17 26 75 0a 17 0020 26 64 10 69 9e 8a 0c b1 b5 32 53 fa fe 9a 80 18 0030 01 fe ab 38 00 00 01 01 08 0a 15 08 9d 88 d5 fb 0040 84 04 02 61 6e 61 6e 65 20 74 72 69 6e 6b 74 20 0050 73 61 66 74 0a

Figure 3: Nachricht

4.1.4 TCP-Flags

TCP-Flags dienen dazu um den Zustand, oder andere zusätzliche Informationen der Verbindung anzuzeigen. Diese dienen zum Troubleshooten. In der Übung ist die Push flag gesetzt, was bedeutet, dass die Nachricht sofort übertragen wird, ohne darauf zu warten, dass zusätzliche Informationen auf der Senderseite gebuffert werden.

Wird oft in Echtzeitanwendung benutzt.

```
▼ Transmission Control Protocol, Src Port: 4201, Dst Port: 40586, Seq: 75, Ack: 1, Len: 19
  Source Port: 4201
  Destination Port: 40586
  [Stream index: 32]
  ▶ [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 19]
  Sequence Number: 75      (relative sequence number)
  Sequence Number (raw): 212972850
  [Next Sequence Number: 94      (relative sequence number)]
  Acknowledgment Number: 1      (relative ack number)
  Acknowledgment number (raw): 1408958106
  1000 .... = Header Length: 32 bytes (8)
  ▼ Flags: 0x018 (PSH, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Accurate ECN: Not set
    .... 0... = Congestion Window Reduced: Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 = Acknowledgment: Set
    .... .... 1... = Push: Set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
    [TCP Flags: .....AP....]
  Window: 510
  [Calculated window size: 65280]
  [Window size scaling factor: 128]
  Checksum: 0xab38 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▶ [Timestamps]
  ▶ [SEQ/ACK analysis]
  TCP payload (19 bytes)
```

Figure 4: TCP-Flags

4.1.5 TCP-Header

Vorhandenen Felder

- Source Port
- Destination Port
- Sequence Number
 - Zeigt an wie viele Daten in der TCP Session übertragen werden.
- Acknowledgment Number
 - Vom Empfänger benutzt um das Nächste TCP Segment anzufordern

Nicht Vorhandene Felder

- Window
 - Gibt an wie viele bytes die der Empfänger empfangen will. Wird genutzt damit der Empfänger sagen kann, dass er mehr Daten empfangen will.
- RSV
 - 3 Reservierte Bits, die immer 0 sind.
- Urgent Pointer
- DO
 - Länge des Headers.
- Flags
 - Vorher bereits erklärt.
- Checksum
 - Benutzt für eine Prüfsumme um sicherzugehen, dass der TCP header korrekt ist.

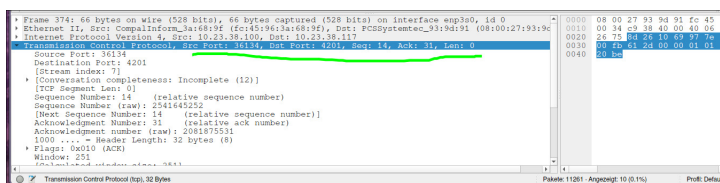


Figure 5: TCP-Header

4.1.6 Verbindungsabbruch

Die TCP-Flag Reset wird im Packet gesetzt und dann ist die Verbindung mit diesem Packet beendet.

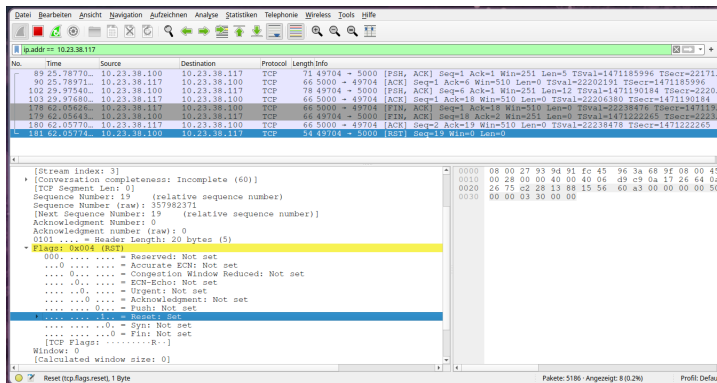


Figure 6: Verbindungsabbruch

4.1.7 Verbinden mit dem Sever (UDP)

nc -u 10.23.38.117 4201

4.1.8 UDP-Verbindungsaufbau

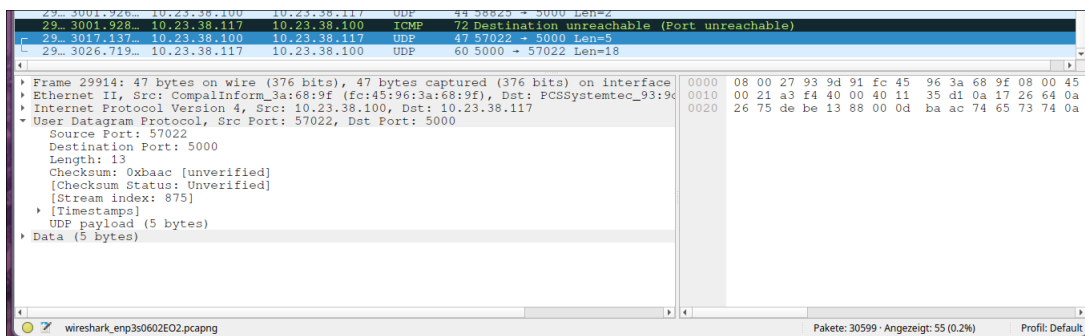


Figure 7: UDP-Verbindungsaufbau

4.1.9 Nachrichten verschicken und empfangen

idk ob das dahingehört tbh

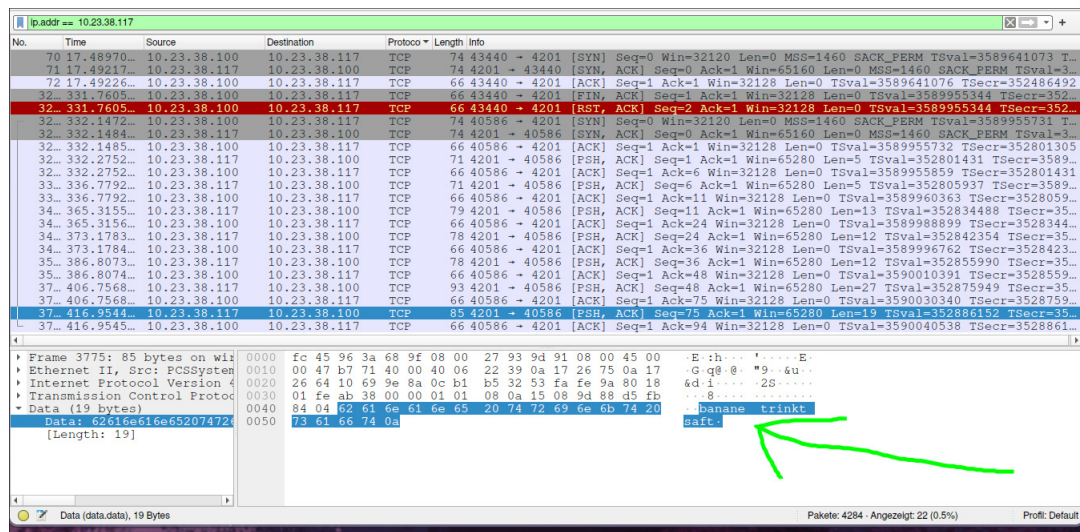


Figure 8: Nachrichten

5 Vollständige Konfigurationsdateien

6 **Abbildungsverzeichnis**

List of Figures

1	memes klauen ist nicht ethisch	1
2	TCP 3 Way Handshake	5
3	Nachricht	5
4	TCP-Flags	6
5	TCP-Header	7
6	Verbindungsabbruch	8
7	UDP-Verbindungsaufbau	8
8	Nachrichten	9