VCS TCP/IP Bulletin Board

Protokollanalyse

26. November 2018

Lara Kammerer

Valentin Platzgummer

1 Protokoll Client → Server

1.1 Art der ausgetauschten Daten

Welchen Typ haben die vom Client zum Server gesendeten Daten eines Requests wenn die Option **`-i`/`--image` nicht** verwendet wird?

☑Text

☐ Binär

☐ Gemischt

Pasten Sie hier einen Teil des **`tcpdump`** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
16:40:05.352645 IP (tos 0x0, ttl 64, id 23371, offset 0, flags [DF], proto TCP
(6), length 10
1)
    127.0.0.1.42528 > 127.0.0.1.7329: Flags [P.], cksum 0xfe59 (incorrect ->
0x7b60), seq 1:5
0, ack 1, win 342, options [nop,nop,TS val 3649383389 ecr 3649383389], length 49
        0x0000:  4500 0065 5b4b 4000 4006 e145 7f00 0001  E..e[K@.@..E....
        0x0010:  7f00 0001 a620 1ca1 065a f5b0 7cb8 b15b  .........Z..|..[
        0x0020:  8018 0156 fe59 0000 0101 080a d985 2bdd  ...V.Y........+.
        0x0030:  d985 2bdd 7573 6572 3d69 6331 3762 3039  ..+.user=ic17b09
        0x0040:  360a 4e6f 7720 7765 2061 7265 2069 6e20  6.Now.we.are.in.
        0x0050:  7365 2074 6563 686e 696b 756d 202d 2074  se.technikum.-.t
        0x0060:  6573 7420 32                             est.2
```

Erklären Sie kurz (Stichworte), was in diesem Teil des **`tcpdump`** Outputs zu sehen ist.

- Push & Ack von Client an Server
- Client an Server – erkennbar an Ports (42528 ist Port von Client, 7329 ist Port von Server)
- **Plaintext wird übermittelt** (ASCII, siehe unterstrichen, lesbar)
- win size wurde in Frame davor auf 342 zurückgesetzt
- Länge 49 → sendet somit Sequenz 1-49 (50 ist nächste zu schickende)

Welchen Typ haben die vom Client zum Server gesendeten Daten eines Requests wenn die Option **`-i`**/**`--image`** verwendet wird?

☑Text

☐ Binär

☐ Gemischt

Pasten Sie hier einen Teil des **`tcpdump`** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
16:43:04.512208 IP (tos 0x0, ttl 64, id 51370, offset 0, flags [DF], proto TCP
(6), length 17
9)
    127.0.0.1.42542 > 127.0.0.1.7329: Flags [P.], cksum 0xfea7 (incorrect ->
0x7091), seq 1:1
28, ack 1, win 342, options [nop,nop,TS val 3649562440 ecr 3649562440], length
127
        0x0000:  4500 00b3 c8aa 4000 4006 7398 7f00 0001  E.....@.@.s.....
        0x0010:  7f00 0001 a62e 1ca1 f689 9575 b8b7 1f1d  ..........u....
        0x0020:  8018 0156 fea7 0000 0101 080a d987 e748  ...V...........H
        0x0030:  d987 e748 7573 6572 3d69 6331 3762 3039  ...Huser=ic17b09
        0x0040:  360a 696d 673d 6874 7470 733a 2f2f 7570  6.img=https://up
        0x0050:  6c6f 6164 2e77 696b 696d 6564 6961 2e6f  load.wikimedia.o
        0x0060:  7267 2f77 696b 6970 6564 6961 2f63 6f6d  rg/wikipedia/com
        0x0070:  6d6f 6e73 2f62 2f62 302f 4f72 616e 6765  mons/b/b0/Orange
        0x0080:  426c 6f73 735f 7762 2e6a 7067 0a4e 6f77  Bloss_wb.jpg.Now
        0x0090:  2077 6520 6172 6520 696e 2073 6520 7465  .we.are.in.se.te
        0x00a0:  6368 6e69 6b75 6d20 2d20 6d69 7420 696d  chnikum.-.mit.im
        0x00b0:  6167 65                                  age
```

Erklären Sie kurz (Stichworte), was in diesem Teil des **`tcpdump`** Outputs zu sehen ist.

- Push & Ack von Client an Server
- Client an Server – erkennbar an Ports (42542 ist Port von Client, 7329 ist Port von Server)
- **Text wird übermittelt** (ASCII, siehe unterstrichen), Bild wird in Form eines Links in Plaintext übertragen
- win size wurde in Frame davor auf 342 zurückgesetzt
- Länge 127 → sendet somit Sequenz 1-127 (128 ist nächste zu schickende)
- Bild wird als Plaintext-Link verschickt

1.2 Markierung der Requestgrenzen

Durch welche Technik werden die Requestgrenzen markiert?

☐ Verwenden einer fixen Requestlänge

☐ Requestlänge als definierter Teil des Requests

☐ Expliziter Teminator

☑End-of-file (EOF)

Pasten Sie hier einen Teil des **tcpdump** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
16:40:05.352645 IP (tos 0x0, ttl 64, id 23371, offset 0, flags [DF], proto TCP
(6), length 10
1)
    127.0.0.1.42528 > 127.0.0.1.7329: Flags [P.], cksum 0xfe59 (incorrect ->
0x7b60), seq 1:5
0, ack 1, win 342, options [nop,nop,TS val 3649383389 ecr 3649383389], length 49
        0x0000:  4500 0065 5b4b 4000 4006 e145 7f00 0001  E..e[K@.@..E....
        0x0010:  7f00 0001 a620 1ca1 065a f5b0 7cb8 b15b  .........Z..|..[
        0x0020:  8018 0156 fe59 0000 0101 080a d985 2bdd  ...V.Y........+.
        0x0030:  d985 2bdd 7573 6572 3d69 6331 3762 3039  ..+.user=ic17b09
        0x0040:  360a 4e6f 7720 7765 2061 7265 2069 6e20  6.Now.we.are.in.
        0x0050:  7365 2074 6563 686e 696b 756d 202d 2074  se.technikum.-.t
        0x0060:  6573 7420 32                             est.2
16:40:05.352654 IP (tos 0x0, ttl 64, id 8125, offset 0, flags [DF], proto TCP
(6), length 52)
    127.0.0.1.7329 > 127.0.0.1.42528: Flags [.], cksum 0xfe28 (incorrect ->
0x7f8d), seq 1, ack 50, win 342, options [nop,nop,TS val 3649383389 ecr
3649383389], length 0
        0x0000:  4500 0034 1fbd 4000 4006 1d05 7f00 0001  E..4..@.@.......
        0x0010:  7f00 0001 1ca1 a620 7cb8 b15b 065a f5e1  ........|..[.Z..
        0x0020:  8010 0156 fe28 0000 0101 080a d985 2bdd  ...V.(........+.
        0x0030:  d985 2bdd                                ..+.
16:40:05.352672 IP (tos 0x0, ttl 64, id 23372, offset 0, flags [DF], proto TCP
(6), length 52)
    127.0.0.1.42528 > 127.0.0.1.7329: Flags [F.], cksum 0xfe28 (incorrect ->
0x7f8c), seq 50, ack 1, win 342, options [nop,nop,TS val 3649383389 ecr
3649383389], length 0
        0x0000:  4500 0034 5b4c 4000 4006 e175 7f00 0001  E..4[L@.@..u....
        0x0010:  7f00 0001 a620 1ca1 065a f5e1 7cb8 b15b  .........Z..|..[
        0x0020:  8011 0156 fe28 0000 0101 080a d985 2bdd  ...V.(........+.
        0x0030:  d985 2bdd                                ..+.
```

Erklären Sie kurz (Stichworte), was in diesem Teil des **tcpdump** Outputs zu sehen ist.

- **Anfang: Push & Ack von Client an Server** (1. Push nach 3-Way-Handshake)
- **Ende: Finish & Ack von Client an Server**
- Client an Server – erkennbar an Ports (42528 ist Port von Client, 7329 ist Port von Server)
- win size auf 342
- Länge 0 (da FIN keine Daten mitschickt)
- Seq 50 (da Nachricht zuvor Seq 1:50 & Ack von Server =50)
- Ack 1 → Server kann anfangen 1. Sequenz (Nummer 1) zu schicken

FIN gesetzt → Client beendet Request & sagt „Habe fertig“

Request: Datenstrom mit allen Daten des Clienten pro Anfrage.

Auch wenn Request mit 2 messages abgesetzt wird, bleibt ein Record pro Request:

```
simple_message_client -s'127.0.0.1' -u'17b096' -p7329 -m  -m "Dieser Post hat
den Zeitstempel: $(date) und sogar ein Image" -i
'https://upload.wikimedia.org/wikipedia/en/7/78/Small_scream.png' -m"Zeiter Teil
der Nachtricht."
```

Zuerst wird erste Nachricht übertragen, Server returniert Reply, dann der zweite Teil. Server retourniert Reply.

Output:**ic17b096 sagt:**

ic17b096 sagt:

Dieser Post hat den Zeitstempel: So Nov 25 07:38:20 UTC 2018 und sogar ein Image

ic17b096 sagt:

Zeiter Teil der Nachtricht.

1.3 Markierung der Recordgrenzen

Durch welche Technik werden die Recordgrenzen markiert?

☐ Verwenden einer fixen Recordlänge/-struktur

☐ Recordlänge als definierter Teil des Records

☐ Expliziter Teminator

☑End-of-file (EOF)

Pasten Sie hier einen Teil des **tcpdump** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
16:43:04.512208 IP (tos 0x0, ttl 64, id 51370, offset 0, flags [DF], proto TCP
(6), length 179)
    127.0.0.1.42542 > 127.0.0.1.7329: Flags [P.], cksum 0xfea7 (incorrect ->
0x7091), seq 1:128, ack 1, win 342, options [nop,nop,TS val 3649562440 ecr
3649562440], length 127
        0x0000:  4500 00b3 c8aa 4000 4006 7398 7f00 0001  E.....@.@.s.....
        0x0010:  7f00 0001 a62e 1ca1 f689 9575 b8b7 1f1d  ..........u....
        0x0020:  8018 0156 fea7 0000 0101 080a d987 e748  ...V...........H
        0x0030:  d987 e748 7573 6572 3d69 6331 3762 3039  ...Huser=ic17b09
        0x0040:  360a 696d 673d 6874 7470 733a 2f2f 7570  6.img=https://up
        0x0050:  6c6f 6164 2e77 696b 696d 6564 6961 2e6f  load.wikimedia.o
        0x0060:  7267 2f77 696b 6970 6564 6961 2f63 6f6d  rg/wikipedia/com
        0x0070:  6d6f 6e73 2f62 2f62 302f 4f72 616e 6765  mons/b/b0/Orange
        0x0080:  426c 6f73 735f 7762 2e6a 7067 0a4e 6f77  Bloss_wb.jpg.Now
        0x0090:  2077 6520 6172 6520 696e 2073 6520 7465  .we.are.in.se.te
        0x00a0:  6368 6e69 6b75 6d20 2d20 6d69 7420 696d  chnikum.-.mit.im
        0x00b0:  6167 65                                  age
16:43:04.512218 IP (tos 0x0, ttl 64, id 17089, offset 0, flags [DF], proto TCP
(6), length 52)
    127.0.0.1.7329 > 127.0.0.1.42542: Flags [.], cksum 0xfe28 (incorrect ->
0xcea0), seq 1, ack 128, win 342, options [nop,nop,TS val 3649562440 ecr
3649562440], length 0
        0x0000:  4500 0034 42c1 4000 4006 fa00 7f00 0001  E..4B.@.@.......
        0x0010:  7f00 0001 1ca1 a62e b8b7 1f1d f689 95f4  ................
        0x0020:  8010 0156 fe28 0000 0101 080a d987 e748  ...V.(.........H
        0x0030:  d987 e748                                ...H
16:43:04.512237 IP (tos 0x0, ttl 64, id 51371, offset 0, flags [DF], proto TCP
(6), length 52)
    127.0.0.1.42542 > 127.0.0.1.7329: Flags [F.], cksum 0xfe28 (incorrect ->
0xce9f), seq 128, ack 1, win 342, options [nop,nop,TS val 3649562440 ecr
3649562440], length 0
        0x0000:  4500 0034 c8ab 4000 4006 7416 7f00 0001  E..4..@.@.t.....
        0x0010:  7f00 0001 a62e 1ca1 f689 95f4 b8b7 1f1d  ................
        0x0020:  8011 0156 fe28 0000 0101 080a d987 e748  ...V.(.........H
        0x0030:  d987 e748                                ...H
```

Erklären Sie kurz (Stichworte), was in diesem Teil des **tcpdump** Outputs zu sehen ist.

Recordgrenzen entsprechen Requestgrenzen, da Client **pro Request einen Record** sendet. Data Sektor beginnt bei "User…" und endet mit „…image". Nächster Client-Frame ist bereits FIN -Flag ohne Payload.

Record: senden aller spezifizierten Daten (User und Administration) pro Sitzung.

1.4 Markierung der Feldgrenzen

Durch welche Technik werden die Feldgrenzen markiert?

☐ Verwenden einer fixen Feldlänge

☐ Feldlänge als definierter Teil des Feldes

☑Expliziter Teminator

☐ End-of-file (EOF)

Pasten Sie hier einen Teil des **tcpdump** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
16:43:04.512208 IP (tos 0x0, ttl 64, id 51370, offset 0, flags [DF], proto TCP
(6), length 17
9)
    127.0.0.1.42542 > 127.0.0.1.7329: Flags [P.], cksum 0xfea7 (incorrect ->
0x7091), seq 1:1
28, ack 1, win 342, options [nop,nop,TS val 3649562440 ecr 3649562440], length
127
        0x0000:  4500 00b3 c8aa 4000 4006 7398 7f00 0001  E.....@.@.s.....
        0x0010:  7f00 0001 a62e 1ca1 f689 9575 b8b7 1f1d  ...........u....
        0x0020:  8018 0156 fea7 0000 0101 080a d987 e748  ...V...........H
        0x0030:  d987 e748 7573 6572 3d69 6331 3762 3039  ...Huser=ic17b09
        0x0040:  360a 696d 673d 6874 7470 733a 2f2f 7570  6.img=https://up
        0x0050:  6c6f 6164 2e77 696b 696d 6564 6961 2e6f  load.wikimedia.o
        0x0060:  7267 2f77 696b 6970 6564 6961 2f63 6f6d  rg/wikipedia/com
        0x0070:  6d6f 6e73 2f62 2f62 302f 4f72 616e 6765  mons/b/b0/Orange
        0x0080:  426c 6f73 735f 7762 2e6a 7067 0a4e 6f77  Bloss_wb.jpg.Now
        0x0090:  2077 6520 6172 6520 696e 2073 6520 7465  .we.are.in.se.te
        0x00a0:  6368 6e69 6b75 6d20 2d20 6d69 7420 696d  chnikum.-.mit.im
        0x00b0:  6167 65                                  age
```

Erklären Sie kurz (Stichworte), was in diesem Teil des **tcpdump** Outputs zu sehen ist.

- Expliziter Terminator: 0a = ASCII Line Feed (siehe unterstrichen)
- Wir nehmen an: **1 Record mit 3 Feldern**:

1. User, hat ein Key- und einen Value
2. Image: hat einen Key und eine Value
3. Message: keinen Key/Value, unmittelbar auf User/ Image
4. terminiert werden Felder mit 0a (Line Feed)

1.5 Beschaffenheit des Requests

Skizzieren Sie hier, wie die Struktur des Requests der vom Client zum Server geschickt wird aussieht, wenn die Option **`-i/--image` nicht** verwendet wird?

```
erste Stelle Hexa – Ipv4/IPv6

struct message {
       // endet mit 0A
       char user[] = „user=<username>“;
       // endet nicht mit 0A, sondern sofort danach FIN
       char message[] = „<message-text“;
}
```

Pasten Sie hier einen Teil des **`tcpdump`** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
Frame 6: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits)
10:17:27.925407 IP (tos 0x0, ttl 64, id 36791, offset 0, flags [DF], proto TCP
(6), length 148)
    localhost.43526 > localhost.7329: Flags [P.], cksum 0xfe88 (incorrect ->
0x0d07), seq 1:97, ack 1, win 342, options [nop,nop,TS val 3795544664 ecr
3795544662], length 96
        0x0000:  4500 0094 8fb7 4000 4006 acaa 7f00 0001  E.....@.@.......
        0x0010:  7f00 0001 aa06 1ca1 eea1 025d 33a6 5f25  ...........]3._%
        0x0020:  8018 0156 fe88 0000 0101 080a e23b 6a58  ...V.........;jX
        0x0030:  e23b 6a56 7573 6572 3d69 6331 3762 3039  .;jVuser=ic17b09
        0x0040:  360a 4865 6c6c 6f20 746f 2074 6865 2053  6.Hello.to.the.S
        0x0050:  6572 7665 7220 6f6e 2074 6865 206f 7468  erver.on.the.oth
        0x0060:  6572 2073 6964 650a 2020 2020 5469 6d65  er.side.....Time
        0x0070:  2069 7320 6e6f 773a 2046 7220 4e6f 7620  .is.now:.Fr.Nov.
        0x0080:  3233 2030 393a 3137 3a32 3720 5554 4320  23.09:17:27.UTC.
        0x0090:  3230 3138                                2018

Frame 7 ist Ack vom Server, kein Payload.
Frame 8: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
10:17:27.925534 IP (tos 0x0, ttl 64, id 36792, offset 0, flags [DF], proto TCP
(6), length 52)
    localhost.43526 > localhost.7329: Flags [F.], cksum 0xfe28 (incorrect ->
0x9369), seq 97, ack 1, win 342, options [nop,nop,TS val 3795544664 ecr
3795544664], length 0
        0x0000:  4500 0034 8fb8 4000 4006 ad09 7f00 0001  E..4..@.@.......
        0x0010:  7f00 0001 aa06 1ca1 eea1 02bd 33a6 5f25  ............3._%
        0x0020:  8011 0156 fe28 0000 0101 080a e23b 6a58  ...V.(.......;jX
        0x0030:  e23b 6a58                                .;jX
```

Skizzieren Sie hier, wie die Struktur des Requests der vom Client zum Server geschickt wird aussieht, wenn die Option **`-i`**/**`--image`** verwendet wird?

```
struct message_image {
       // endet mit 0A
       char user[] = „user=<username>“;
       // endet mit 0A
       char imageURL[] = „img=<userUrl>
       // endet nicht mit 0A, sondern sofort danach FIN
       char message[] = „<message-text“;
}
```

Pasten Sie hier einen Teil des **`tcpdump`** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
Frame 60
10:17:27.935082 IP (tos 0x0, ttl 64, id 41903, offset 0, flags [DF], proto TCP
(6), length 219)
    localhost.43530 > localhost.7329: Flags [P.], cksum 0xfecf (incorrect ->
0x3bd2), seq 1:168, ack 1, win 342, options [nop,nop,TS val 3795544674 ecr
3795544674], length 167
        0x0000:  4500 00db a3af 4000 4006 986b 7f00 0001  E.....@.@..k....
        0x0010:  7f00 0001 aa0a 1ca1 e376 2e42 6e3c d9c1  .........v.Bn<..
        0x0020:  8018 0156 fecf 0000 0101 080a e23b 6a62  ...V.........;jb
        0x0030:  e23b 6a62 7573 6572 3d69 6331 3762 3039  .;jbuser=ic17b09
        0x0040:  360a 696d 673d 6874 7470 733a 2f2f 7570  6.img=https://up
        0x0050:  6c6f 6164 2e77 696b 696d 6564 6961 2e6f  load.wikimedia.o
        0x0060:  7267 2f77 696b 6970 6564 6961 2f65 6e2f  rg/wikipedia/en/
        0x0070:  372f 3738 2f53 6d61 6c6c 5f73 6372 6561  7/78/Small_screa
        0x0080:  6d2e 706e 670a 4469 6573 6572 2050 6f73  m.png.Dieser.Pos
        0x0090:  7420 6861 7420 6465 6e20 5a65 6974 7374  t.hat.den.Zeitst
        0x00a0:  656d 7065 6c3a 2046 7220 4e6f 7620 3233  empel:.Fr.Nov.23
        0x00b0:  2030 393a 3137 3a32 3720 5554 4320 3230  .09:17:27.UTC.20
        0x00c0:  3138 0a20 2020 2020 756e 6420 736f 6761  18......und.soga
        0x00d0:  7220 6569 6e20 496d 6167 65              r.ein.Image

Frame 61 Ack vom Server, kein Payload
Transmission Control Protocol, Src Port: 7329, Dst Port: 43530, Seq: 1, Ack:
168, Len: 0

Frame 62
10:17:27.935102 IP (tos 0x0, ttl 64, id 41904, offset 0, flags [DF], proto TCP
(6), length 52)
    localhost.43530 > localhost.7329: Flags [F.], cksum 0xfe28 (incorrect ->
0xbd1d), seq 168, ack 1, win 342, options [nop,nop,TS val 3795544674 ecr
3795544674], length 0
        0x0000:  4500 0034 a3b0 4000 4006 9911 7f00 0001  E..4..@.@.......
        0x0010:  7f00 0001 aa0a 1ca1 e376 2ee9 6e3c d9c1  .........v..n<..
        0x0020:  8011 0156 fe28 0000 0101 080a e23b 6a62  ...V.(.......;jb
        0x0030:  e23b 6a62                                .;jb
```

1.6 Protokollschwäche

Leider hat der Designer des Simple Message Protokolls einen Designfehler begangen, sodaß der konkrete Request des Clients nicht immer eindeutig auf Serverseite interpretiert werden kann.

Geben Sie ein Beispiel eines Kommandozeilenaufrufes auf Clientseite an, welches zu einer Fehlinterpretation eines Requests auf der Serverseite führen würde.

```
1. Schwäche: simple_message_client -p7329 -s'localhost' -u$'valentin Patzgummer
\nLara Kammerer' -m$'In dem Text sind\n jede menge lustiger Dinge' -i'gag.png'

2. Error: /usr/local/bin/simple_message_client  -u'ic17b096' -p7329 -
s'127.0.0.1' -m"" -i"Dieser Post: $(date) steht im image Tag"

1. Output: valentin Patzgummer sagt:
Lara Kammerer img=gag.png In dem Text sind jede menge lustiger Dinge

2.Output:
"ic17b096 sagt:
-iDieser Post: So Nov 25 05:25:45 UTC 2018 steht im image Tag "
```

user ist eigentlich „Valentin Platzgummer Lara Kammerer“, da dazwischen aber 0a (LF) als Feldgrenze ist, wird der zweite Teil als Part der Message angesehen. LF ist Feldgrenze, Server geht nach Finden von LF einfach zum Nächsten Feld.

2.Fall: Dieses Problem betrifft den Argument Parser, der nicht mehr unterscheiden kann zwischen message und user

Skizzieren Sie hier, wie die Struktur des Requests, der vom Client zum Server geschickt wird, zu diesem Kommandozeilenaufruf aussieht.

```
erste Stelle Hexa – Ipv4/IPv6

struct message {
       // endet mit 0A
       char user[] = „user=Valentin Platzgummer 0a
       char image[] = „
       char message[] = „Lara KamererIn dem Text sind jede menge lustiger
Dinge"
}
```

Skizzieren Sie hier, wie ein alternativer Kommandozeilenaufrufes auf Clientseite aussehen könnte, der zu dem selben Request führt.

```
simple_message_client -p7329 -s'localhost' -u$'valentin Patzgummer"' -m' Lara
Kammerer In dem Text sind jede menge lustiger Dinge' -i'gag.png'

Ouput: ic17b096 sagt:
Dieser Post: So Nov 25 05:26:32 UTC 2018 steht im message Tag
```

Geben Sie ein Beispiel, wie die Struktur des Requests geändert werden müsste, um eine solche Fehlinterpretation zu verhindern.

1. Möglichkeint: Compound Terminator verwenden, der sich nicht mit $() Befehlen konstruieren läßt.

2. Möglichkeit: Feld spezifizieren wie unteres Struct zeigt. Damit wird zumindest Misverständnis zwischen den einzelnen Feldern vermieden.

```
struct message {
       // endet mit 0A
       char user[] = „user=<username>“;
       char image[] = „image=<image-url>;
       char message[] = „message=<message-text>“;
}
```

Zugehöriger Hex Dump

```
0x0030:  ee09 cc7b 7573 6572 3d76 616c 656e 7469  ...{user=valenti
       0x0040:  6e20 5061 747a 6775 6d6d 6572 200a 4c61  n.Patzgummer..La
       0x0050:  7261 204b 616d 6d65 7265 720a 696d 673d  ra.Kammerer.img=
       0x0060:  6761 672e 706e 670a 496e 2064 656d 2054  gag.png.In.dem.T
       0x0070:  6578 7420 7369 6e64 0a20 6a65 6465 206d  ext.sind..jede.m
       0x0080:  656e 6765 206c 7573 7469 6765 7220 4469  enge.lustiger.Di
       0x0090:  6e67 65                                  nge
```

2 Protokoll Server → Client

2.1 Art der ausgetauschten Daten

Welchen Typ haben die vom Server zum Client gesendeten Daten eines Replies.

☐ Text

☐ Binär

☑ Gemischt

Pasten Sie hier einen Teil des **`tcpdump`** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

Erklären Sie kurz (Stichworte), was in diesem Teil des **`tcpdump`** Outputs zu sehen ist.

- Reply vom Server Port 7329 → Port 42528
- **Plain Text, lesbar** (Frame 21) status, file.html, length.html, message.html, file.png, length.png)
- **Binary des PNG**. (Frame 41) Erkennbar am Header von PNG: 8950 4e47 0d0a 1a0a
- **Gemischte Übertragung**

```
Frame 21: 16:40:05.355536 IP (tos 0x0, ttl 64, id 8133, offset 0, flags [DF],
proto TCP (6), length 427)
    127.0.0.1.7329 > 127.0.0.1.42528: Flags [P.], cksum 0xff9f (incorrect ->
0x0c0b), seq 169:544, ack 51, win 342, options [nop,nop,TS val 3649383392 ecr
3649383392], length 375
       0x0000:  4500 01ab 1fc5 4000 4006 1b86 7f00 0001  E.....@.@.......
       0x0010:  7f00 0001 1ca1 a620 7cb8 b203 065a f5e2  ........|....Z..
       0x0020:  8018 0156 ff9f 0000 0101 080a d985 2be0  ...V..........+.
       0x0030:  d985 2be0 220a 2268 7474 703a 2f2f 7777  ..+."."http://ww
       0x0040:  772e 7733 2e6f 7267 2f54 522f 7868 746d  w.w3.org/TR/xhtm
       0x0050:  6c31 2f44 5444 2f78 6874 6d6c 312d 7472  l1/DTD/xhtml1-tr
       0x0060:  616e 7369 7469 6f6e 616c 2e64 7464 223e  ansitional.dtd">
       0x0070:  0a3c 6874 6d6c 3e0a 2020 3c68 6561 643e  .<html>...<head>
       0x0080:  0a20 2020 203c 6d65 7461 2068 7474 702d  .....<meta.http-
       0x0090:  6571 7569 763d 2243 6f6e 7465 6e74 2d54  equiv="Content-T
       0x00a0:  7970 6522 2063 6f6e 7465 6e74 3d22 7465  ype".content="te
       0x00b0:  7874 2f68 746d 6c3b 0a63 6861 7273 6574  xt/html;.charset
       0x00c0:  3d49 534f 2d38 3835 392d 3135 2220 2f3e  =ISO-8859-15"./>
       0x00d0:  0a20 2020 203c 6c69 6e6b 2072 656c 3d22  .....<link.rel="
       0x00e0:  7374 796c 6573 6865 6574 2220 7479 7065  stylesheet".type
       0x00f0:  3d22 7465 7874 2f63 7373 2220 6872 6566  ="text/css".href
       0x0100:  3d22 7463 7069 702e 6373 7322 202f 3e0a  ="tcpip.css"./>.
       0x0110:  2020 2020 3c74 6974 6c65 3e56 6572 7465  ....<title>Verte
       0x0120:  696c 7465 2043 6f6d 7075 7465 7273 7973  ilte.Computersys
       0x0130:  7465 6d65 202d 2054 4350 2f49 5020 2d0a  teme.-.TCP/IP.-.
       0x0140:  5265 7370 6f6e 7365 3c2f 7469 746c 653e  Response</title>
       0x0150:  0a20 203c 2f68 6561 643e 0a20 203c 626f  ...</head>...<bo
       0x0160:  6479 3e0a 2020 2020 3c68 722f 3e0a 2020  dy>.....<hr/>...
       0x0170:  2020 3c63 656e 7465 723e 0a20 2020 2020  ..<center>......
       0x0180:  2020 3c68 313e 5665 7274 6569 6c74 6520  ..<h1>Verteilte.
       0x0190:  436f 6d70 7574 6572 7379 7374 656d 6520  Computersysteme.
       0x01a0:  2d20 5443 502f 4950 202d 20              -.TCP/IP.-.

Frame 41: 16:40:05.356140 IP (tos 0x0, ttl 64, id 8144, offset 0, flags [DF],
proto TCP (6), length 1697)
    127.0.0.1.7329 > 127.0.0.1.42528: Flags [P.], cksum 0x0496 (incorrect ->
0x93a4), seq 1345:2990, ack 51, win 342, options [nop,nop,TS val 3649383392 ecr
3649383392], length 1645
       0x0000:  4500 06a1 1fd0 4000 4006 1685 7f00 0001  E.....@.@.......
       0x0010:  7f00 0001 1ca1 a620 7cb8 b69b 065a f5e2  ........|....Z..
       0x0020:  8018 0156 0496 0000 0101 080a d985 2be0  ...V..........+.
       0x0030:  d985 2be0 8950 4e47 0d0a 1a0a 0000 000d  ..+..PNG........
       0x0040:  4948 4452 0000 0040 0000 0040 0806 0000  IHDR...@...@....
       0x0050:  00aa 6971 de00 0000 0473 4249 5408 0808  ..iq.....sBIT...
       0x0060:  087c 0864 8800 0000 0970 4859 7300 0006  .|.d.....pHYs...
       0x0070:  ec00 0006 ec01 1e75 3835 0000 0019 7445  .......u85....tE
       0x0080:  5874 536f 6674 7761 7265 0077 7777 2e69  XtSoftware.www.i
       0x0090:  6e6b 7363 6170 652e 6f72 679b ee3c 1a00  nkscape.org..<..
       0x00a0:  000b 0249 4441 5478 daed 9a09 7014 551a  ...IDATx....p.U.
```

2.2 Markierung der Replygrenzen

Durch welche Technik werden die Replygrenzen markiert?

☐ Verwenden einer fixen Replylänge

☐ Replylänge als definierter Teil des Replies

☐ Expliziter Teminator

☑ End-of-file (EOF)

Pasten Sie hier einen Teil des **tcpdump** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
Frame 8: 17:22:47.417429 IP (tos 0x0, ttl 64, id 10802, offset 0, flags [DF],
proto TCP (6), length 52)
    127.0.0.1.44370 > 127.0.0.1.7329: Flags [F.], cksum 0xfe28 (incorrect ->
0x6a1e), seq 98, ack 1, win 342, options [nop,nop,TS val 3824647965 ecr
3824647964], length 0
        0x0000:  4500 0034 2a32 4000 4006 1290 7f00 0001  E..4*2@.@.......
        0x0010:  7f00 0001 ad52 1ca1 d4eb 8928 fdd9 213a  .....R.....(..!:
        0x0020:  8011 0156 fe28 0000 0101 080a e3f7 7f1d  ...V.(..........
        0x0030:  e3f7 7f1c                                ....

Frame 9: 17:22:47.421802 IP (tos 0x0, ttl 64, id 33748, offset 0, flags [DF],
proto TCP (6), length 57
)
    127.0.0.1.7329 > 127.0.0.1.44370: Flags [P.], cksum 0xfe2d (incorrect ->
0x2023), seq 1:6
, ack 99, win 342, options [nop,nop,TS val 3824647969 ecr 3824647965], length 5
        0x0000:  4500 0039 83d4 4000 4006 b8e8 7f00 0001  E..9..@.@.......
        0x0010:  7f00 0001 1ca1 ad52 fdd9 213a d4eb 8929  .......R..!:...)
        0x0020:  8018 0156 fe2d 0000 0101 080a e3f7 7f21  ...V.-.........!
        0x0030:  e3f7 7f1d 7374 6174 75                   ....statu

Frame 57: 17:22:47.426650 IP (tos 0x0, ttl 64, id 33772, offset 0, flags [DF],
proto TCP (6), length 52)
    127.0.0.1.7329 > 127.0.0.1.44370: Flags [F.], cksum 0xfe28 (incorrect ->
0x5945), seq 4294, ack 99, win 342, options [nop,nop,TS val 3824647974 ecr
3824647974], length 0
        0x0000:  4500 0034 83ec 4000 4006 b8d5 7f00 0001  E..4..@.@.......
        0x0010:  7f00 0001 1ca1 ad52 fdd9 31ff d4eb 8929  .......R..1....)
        0x0020:  8011 0156 fe28 0000 0101 080a e3f7 7f26  ...V.(.........&
        0x0030:  e3f7 7f26                                ...&
```

Erklären Sie kurz (Stichworte), was in diesem Teil des **tcpdump** Outputs zu sehen ist.

- Begrenzt durch FIN von Client und eigenes FIN (Frame 8 & 57) – Beginn daher bei 1. Push (Frame 9)
- Request gesamter Datenstrom, den Server an Client proSitzung (von SYN bis FIN) schickt.
- Data beginnt hier bei „status…“ (im Dump sichtbar) und endet bei „.“ letzter Frame vor Server -FIN (Frame 57).

2.3 Markierung der Recordgrenzen

Durch welche Technik werden die Recordgrenzen markiert?

☐ Verwenden einer fixen Recordlänge/-struktur

☑ Recordlänge als definierter Teil des Records

☐ Expliziter Teminator

☐ End-of-file (EOF)

Pasten Sie hier einen Teil des **tcpdump** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
Frame 15: 17:22:47.422285 IP (tos 0x0, ttl 64, id 33751, offset 0, flags [DF],
proto TCP (6), length 101)
    127.0.0.1.7329 > 127.0.0.1.44370: Flags [P.], cksum 0xfe59 (incorrect ->
0x9a59), seq 10:59, ack 99, win 342, options [nop,nop,TS val 3824647970 ecr
3824647969], length 49
        0x0000:  4500 0065 83d7 4000 4006 b8b9 7f00 0001  E..e..@.@.......
        0x0010:  7f00 0001 1ca1 ad52 fdd9 2143 d4eb 8929  .......R..!C...)
        0x0020:  8018 0156 fe59 0000 0101 080a e3f7 7f22  ...V.Y........."
        0x0030:  e3f7 7f21 6669 6c65 3d76 6373 5f74 6370  ...!file=vcs_tcp
        0x0040:  6970 5f62 756c 6c65 7469 6e5f 626f 6172  ip_bulletin_boar
        0x0050:  645f 7265 7370 6f6e 7365 2e68 746d 6c0a  d_response.html.
        0x0060:  6c65 6e3d 31                             len=1

Frame 17: 17:22:47.422603 IP (tos 0x0, ttl 64, id 33752, offset 0, flags [DF],
proto TCP (6), length 56)
    127.0.0.1.7329 > 127.0.0.1.44370: Flags [P.], cksum 0xfe2c (incorrect ->
0x068d), seq 59:63, ack 99, win 342, options [nop,nop,TS val 3824647970 ecr
3824647970], length 4
        0x0000:  4500 0038 83d8 4000 4006 b8e5 7f00 0001  E..8..@.@.......
        0x0010:  7f00 0001 1ca1 ad52 fdd9 2174 d4eb 8929  .......R..!t...)
        0x0020:  8018 0156 fe2c 0000 0101 080a e3f7 7f22  ...V.,........."
        0x0030:  e3f7 7f22 3236 310a                      ..."261.

Frame 37: 17:22:47.423728 IP (tos 0x0, ttl 64, id 33762, offset 0, flags [DF],
proto TCP (6), length 72)
    127.0.0.1.7329 > 127.0.0.1.44370: Flags [P.], cksum 0xfe3c (incorrect ->
0xd0cf), seq 1324:1344, ack 99, win 342, options [nop,nop,TS val 3824647971 ecr
3824647971], length 20
        0x0000:  4500 0048 83e2 4000 4006 b8cb 7f00 0001  E..H..@.@.......
        0x0010:  7f00 0001 1ca1 ad52 fdd9 2665 d4eb 8929  .......R..&e...)
        0x0020:  8018 0156 fe3c 0000 0101 080a e3f7 7f23  ...V.<........#
        0x0030:  e3f7 7f23 6669 6c65 3d6f 6b2e 706e 670a  ...#file=ok.png.
        0x0040:  6c65 6e3d 3239 3439                      len=2949
```

Erklären Sie kurz (Stichworte), was in diesem Teil des **tcpdump** Outputs zu sehen ist.

- Server → Client
- Frame 15 + Frame 17 + Frame 37 übertragen **len** → Recordgrenze definiert

  textlänge (len= 1261), Bildlänge (len=2949)

2.4 Markierung der Feldgrenzen

Durch welche Technik werden die Feldgrenzen markiert?

☐ Verwenden einer fixen Feldlänge

☐ Feldlänge als definierter Teil des Feldes

☑ Expliziter Teminator

☐ End-of-file (EOF)

Pasten Sie hier einen Teil des **tcpdump** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

```
Frame 15
10:17:27.927951 IP (tos 0x0, ttl 64, id 53968, offset 0, flags [DF], proto TCP
(6), length 100)
    localhost.7329 > localhost.43526: Flags [P.], cksum 0xfe58 (incorrect ->
0x51ec), seq 13:61, ack 98, win 342, options [nop,nop,TS val 3795544667 ecr
3795544667], length 48
       0x0000:  4500 0064 d2d0 4000 4006 69c1 7f00 0001  E..d..@.@.i.....
       0x0010:  7f00 0001 1ca1 aa06 33a6 5f31 eea1 02be  ........3._1....
       0x0020:  8018 0156 fe58 0000 0101 080a e23b 6a5b  ...V.X.......;j[
       0x0030:  e23b 6a5b 653d 7663 735f 7463 7069 705f  .;j[e=vcs_tcpip_
       0x0040:  6275 6c6c 6574 696e 5f62 6f61 7264 5f72  bulletin_board_r
       0x0050:  6573 706f 6e73 652e 6874 6d6c 0a6c 656e  esponse.html.len
       0x0060:  3d31 3236                                =126
```

Erklären Sie kurz (Stichworte), was in diesem Teil des **tcpdump** Outputs zu sehen ist.

Zwischen Feldern File und Len ist Terminator 0A (LF)

2.5 Beschaffenheit des Replies

Skizzieren Sie hier, wie die Struktur des Replies der vom Server zum Client geschickt wird aussieht:

```
struct reply {
       char status [9]
       char file.html[] = "vcs_tcpip_bulletin_board.response.html";
       char length.html[] = 1261;
       char message.html[length.html];
       char file.png [];
       char length.png[] = 2949;
       binary image.png[length.png];
}
```

Pasten Sie hier einen Teil des **tcpdump** Outputs, und markieren Sie die entsprechenden Regionen, um Ihre Antwort zu untermauern.

Frame 15 – 27 sendet HTML Reply an Client. Frame 29 schließt unmittelbar darauf an, nur mit 0A terminiert.

Daher Teil des Replies und Teil der Struktur.

Macht Sinn da als nächstes Bild kommt, so kann Client nötigen Buffergröße für Bild bereitstellen.

Frame 9 +11 liefert „status“, Frame 13 + Frame 15 + Frame 17 „files ...“

Frame 21 folgt dann die „html message“. Frame 37 „file“ vom gesendeten Bild, Frame 43 dann das Binary Bild.

```
Frame 9: 17:22:47.421802 IP (tos 0x0, ttl 64, id 33748, offset 0, flags [DF],
proto TCP (6), length 57
)
    127.0.0.1.7329 > 127.0.0.1.44370: Flags [P.], cksum 0xfe2d (incorrect ->
0x2023), seq 1:6
, ack 99, win 342, options [nop,nop,TS val 3824647969 ecr 3824647965], length 5
        0x0000:  4500 0039 83d4 4000 4006 b8e8 7f00 0001  E..9..@.@.......
        0x0010:  7f00 0001 1ca1 ad52 fdd9 213a d4eb 8929  .......R..!:...)
        0x0020:  8018 0156 fe2d 0000 0101 080a e3f7 7f21  ...V.-.........!
        0x0030:  e3f7 7f1d 7374 6174 75                   ....statu

Frame 15: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
10:17:27.927951 IP (tos 0x0, ttl 64, id 53968, offset 0, flags [DF], proto TCP
(6), length 100)
    localhost.7329 > localhost.43526: Flags [P.], cksum 0xfe58 (incorrect ->
0x51ec), seq 13:61, ack 98, win 342, options [nop,nop,TS val 3795544667 ecr
3795544667], length 48
       0x0000:  4500 0064 d2d0 4000 4006 69c1 7f00 0001  E..d..@.@.i.....
       0x0010:  7f00 0001 1ca1 aa06 33a6 5f31 eea1 02be  ........3._1....
       0x0020:  8018 0156 fe58 0000 0101 080a e23b 6a5b  ...V.X.......;j[
       0x0030:  e23b 6a5b 653d 7663 735f 7463 7069 705f  .;j[e=vcs_tcpip_
       0x0040:  6275 6c6c 6574 696e 5f62 6f61 7264 5f72  bulletin_board_r
       0x0050:  6573 706f 6e73 652e 6874 6d6c 0a6c 656e  esponse.html.len
       0x0060:  3d31 3236                                =126

Frame 21: 1289 bytes on wire (10312 bits), 1289 bytes captured (10312 bits)
10:17:27.928454 IP (tos 0x0, ttl 64, id 53971, offset 0, flags [DF], proto TCP
(6), length 1275)
    localhost.7329 > localhost.43526: Flags [P.], cksum 0x02f0 (incorrect ->
0xe246), seq 63:1286, ack 98, win 342, options [nop,nop,TS val 3795544667 ecr
3795544667], length 1223
       0x0000:  4500 04fb d2d3 4000 4006 6527 7f00 0001  E.....@.@.e'....
       0x0010:  7f00 0001 1ca1 aa06 33a6 5f63 eea1 02be  ........3._c....
       0x0020:  8018 0156 02f0 0000 0101 080a e23b 6a5b  ...V.........;j[
       0x0030:  e23b 6a5b 3c3f 786d 6c20 7665 7273 696f  .;j[<?xml.versio
       0x0040:  6e3d 2231 2e30 2220 656e 636f 6469 6e67  n="1.0".encoding
       0x0050:  3d22 4953 4f2d 3838 3539 2d31 3522 3f3e  ="ISO-8859-15"?>
       0x0060:  0a3c 2144 4f43 5459 5045 2068 746d 6c20  .<!DOCTYPE.html.
       0x0070:  5055 424c 4943 2022 2d2f 2f57 3343 2f2f  PUBLIC."-//W3C//
       0x0080:  4454 4420 5848 544d 4c20 312e 3020 5472  DTD.XHTML.1.0.Tr
       0x0090:  616e 7369 7469 6f6e 616c 2f2f 454e 220a  ansitional//EN".
       0x00a0:  2268 7474 703a 2f2f 7777 772e 7733 2e6f  "http://www.w3.o
       0x00b0:  7267 2f54 522f 7868 746d 6c31 2f44 5444  rg/TR/xhtml1/DTD
       0x00c0:  2f78 6874 6d6c 312d 7472 616e 7369 7469  /xhtml1-transiti
       0x00d0:  6f6e 616c 2e64 7464 223e 0a3c 6874 6d6c  onal.dtd">.<html
       0x00e0:  3e0a 2020 3c68 6561 643e 0a20 2020 203c  >...<head>.....<

Frame 27: 79 bytes on wire (632 bits), 79 bytes captured (632 bits)
10:17:27.928708 IP (tos 0x0, ttl 64, id 53974, offset 0, flags [DF], proto TCP
(6), length 65)
    localhost.7329 > localhost.43526: Flags [P.], cksum 0xfe35 (incorrect ->
0x813b), seq 1311:1324, ack 98, win 342, options [nop,nop,TS val 3795544668 ecr
3795544668], length 13
       0x0000:  4500 0041 d2d6 4000 4006 69de 7f00 0001  E..A..@.@.i.....
       0x0010:  7f00 0001 1ca1 aa06 33a6 6443 eea1 02be  ........3.dC....
       0x0020:  8018 0156 fe35 0000 0101 080a e23b 6a5c  ...V.5.......;j\
       0x0030:  e23b 6a5c 6f64 793e 0a3c 2f68 746d 6c3e  .;j\ody>.</html>

Frame 29
10:17:27.929397 IP (tos 0x0, ttl 64, id 53975, offset 0, flags [DF], proto TCP
(6), length 58)
```

```
    localhost.7329 > localhost.43526: Flags [P.], cksum 0xfe2e (incorrect ->
0x7dea), seq 1324:1330, ack 98, win 342, options [nop,nop,TS val 3795544668 ecr
3795544668], length 6
       0x0000:  4500 003a d2d7 4000 4006 69e4 7f00 0001  E..:..@.@.i.....
       0x0010:  7f00 0001 1ca1 aa06 33a6 6450 eea1 02be  ........3.dP....
       0x0020:  8018 0156 fe2e 0000 0101 080a e23b 6a5c  ...V.........;j\
       0x0030:  e23b 6a5c 6669 6c65 3d6f                 .;j\file=o
```