SMTP Protocol Packet Crafting for Testing Network Forensics Solution

Introduction:

Packet crafting is a process of manually creating of editing an existing packet to test network behaviors and to test network forensic products. In packet crafting, one creates a completely new packet or edits the existing packet to change the information packet contains. Sometimes we do not get our requirement packets on the internet or capture those packets from simulated systems can be difficult.

High level Problem definition:

Our main aim was to test encrypted-email (PGP and S/MIME) feature tests for our product. But the problem was that we were not able to get any such packets available on the internet. After reading rfc4880 about those encrypted emails we can get the basic packet structure and working of encryption and decryption for emails. Then we tried to set up an email server with PGP and S/MIME encryption, which required a valid certificate for public key and private key. Getting a valid certificate from a Certificate Authority (CA) is a long formal process. Then we got a great documentary on "Protected Headers for Cryptographic E-mail", where we got a clear idea of encrypted email header with body with example. Then we thought of editing unencrypted email packets with those encrypted email headers and body.

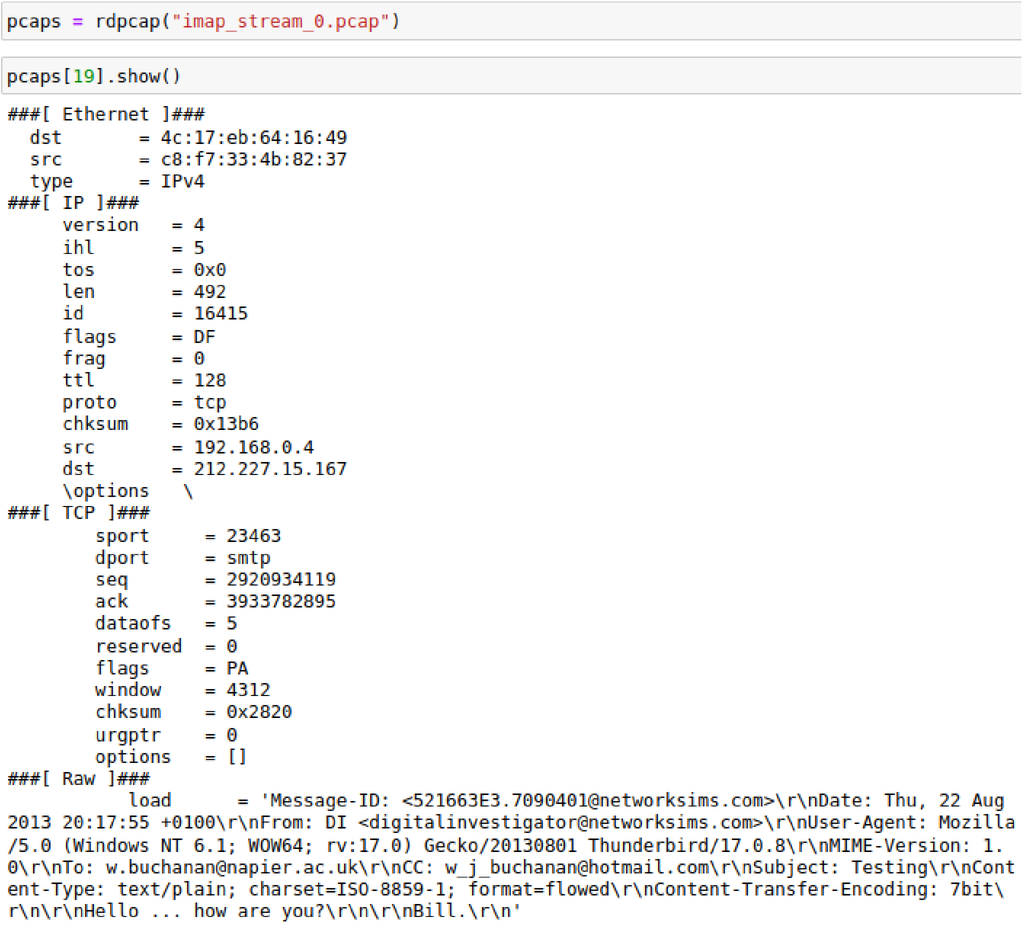
Our main aim was to capture/create packets for encrypted email (PGP and S/MIME). To send encrypted emails, we need to have an email server that has a valid certificate (CA certified) and support for sending PGP and S/MIME encrypted email. Setting up this type of server is non-viable as we need to buy a valid CA certificate. After reading some documentation about the difference between simple SMTP and encrypted email packet payload structures, we decided to craft simple SMTP packets to get encrypted packets.

High Level Sketch of the Solution:

For editing we choose to work with SMTP non-fragmented email packet. Which contain multiple packets of client and server communication. In those packets to replace unencrypted email with encrypted email, we needed to change 3-4 packets with payloads which are "MAIL FROM:", "RCPT TO:" and actual mail with email-header.

Detailed Description and Analysis of the Solution:

For editing payload of existing packet, we have used Scapy library of python, as python is a very high-level language and extremely easy to write code. In scapy, packets in one pcap file are represented as a list of packets which can be accessed by index. We can read pcap file using **rdpcap** function and write pcap file using **wrpcap** function. Scapy has **show()** function to view formatted packet.



```
pcaps = rdpcap("imap_stream_0.pcap")
```

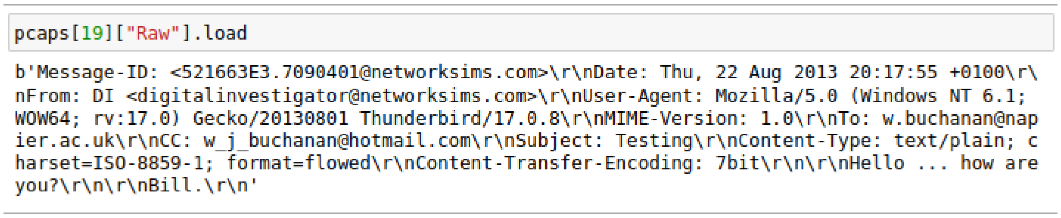
```
pcaps[19].show()
```

```
###[ Ethernet ]###
  dst       = 4c:17:eb:64:16:49
  src       = c8:f7:33:4b:82:37
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 492
     id        = 16415
     flags     = DF
     frag      = 0
     ttl       = 128
     proto     = tcp
     chksum    = 0x13b6
     src       = 192.168.0.4
     dst       = 212.227.15.167
     \options   \
###[ TCP ]###
        sport     = 23463
        dport     = smtp
        seq       = 2920934119
        ack       = 3933782895
        dataofs   = 5
        reserved  = 0
        flags     = PA
        window    = 4312
        chksum    = 0x2820
        urgptr    = 0
        options   = []
###[ Raw ]###
           load      = 'Message-ID: <521663E3.7090401@networksims.com>\r\nDate: Thu, 22 Aug
2013 20:17:55 +0100\r\nFrom: DI <digitalinvestigator@networksims.com>\r\nUser-Agent: Mozilla
/5.0 (Windows NT 6.1; WOW64; rv:17.0) Gecko/20130801 Thunderbird/17.0.8\r\nMIME-Version: 1.
0\r\nTo: w.buchanan@napier.ac.uk\r\nCC: w_j_buchanan@hotmail.com\r\nSubject: Testing\r\nCont
ent-Type: text/plain; charset=ISO-8859-1; format=flowed\r\nContent-Transfer-Encoding: 7bit\
r\n\r\nHello ... how are you?\r\n\r\nBill.\r\n'
```

In this output we can view and edit all layer's information except data-link layer.

1. **Payloads replace:**

In Scapy we can access payload using **pcaps[index][“Raw”].load,** which takes string and bytes type of values. Before replacing payload, we need to replace every newline with “\r\n”.

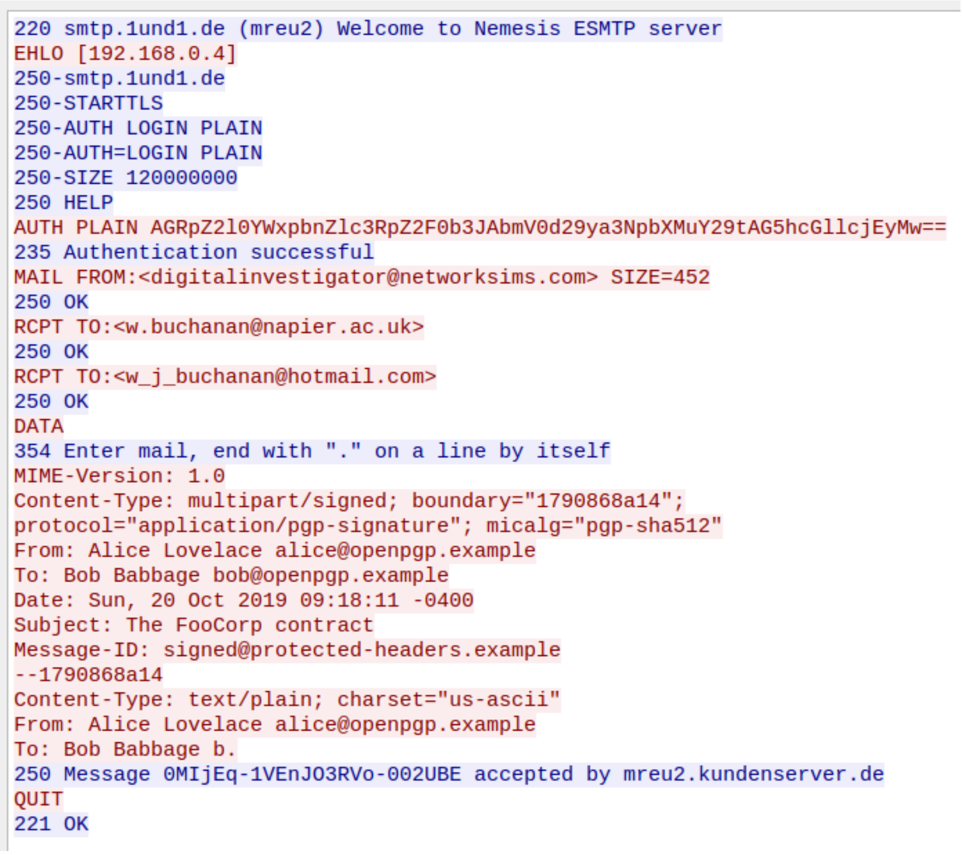


```
pcaps[19]["Raw"].load
```

```
b'Message-ID: <521663E3.7090401@networksims.com>\r\nDate: Thu, 22 Aug 2013 20:17:55 +0100\r\
nFrom: DI <digitalinvestigator@networksims.com>\r\nUser-Agent: Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:17.0) Gecko/20130801 Thunderbird/17.0.8\r\nMIME-Version: 1.0\r\nTo: w.buchanan@nap
ier.ac.uk\r\nCC: w_j_buchanan@hotmail.com\r\nSubject: Testing\r\nContent-Type: text/plain; c
harset=ISO-8859-1; format=flowed\r\nContent-Transfer-Encoding: 7bit\r\n\r\nHello ... how are
you?\r\n\r\nBill.\r\n'
```

2. **IP Header Length:**

After editing/replacing payload. Now if we follow TCP Stream, we will see cropped out message that is because we need to change IP Header length.

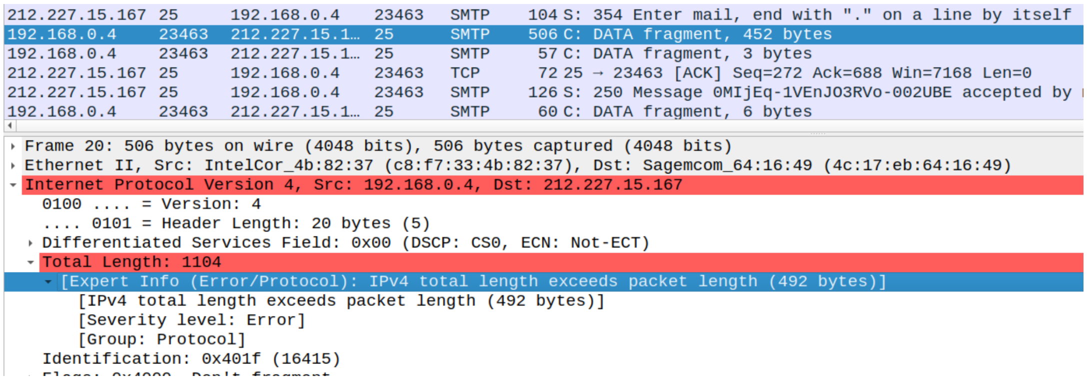


```
220 smtp.1und1.de (mreu2) Welcome to Nemesis ESMTP server
EHLO [192.168.0.4]
250-smtp.1und1.de
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250-SIZE 120000000
250 HELP
AUTH PLAIN AGRpZ2l0YWxpbnZlc3RpZ2F0b3JAbmV0d29ya3NpbXMuY29tAG5hcGllcjEyMw==
235 Authentication successful
MAIL FROM:<digitalinvestigator@networksims.com> SIZE=452
250 OK
RCPT TO:<w.buchanan@napier.ac.uk>
250 OK
RCPT TO:<w_j_buchanan@hotmail.com>
250 OK
DATA
354 Enter mail, end with "." on a line by itself
MIME-Version: 1.0
Content-Type: multipart/signed; boundary="1790868a14";
protocol="application/pgp-signature"; micalg="pgp-sha512"
From: Alice Lovelace alice@openpgp.example
To: Bob Babbage bob@openpgp.example
Date: Sun, 20 Oct 2019 09:18:11 -0400
Subject: The FooCorp contract
Message-ID: signed@protected-headers.example
--1790868a14
Content-Type: text/plain; charset="us-ascii"
From: Alice Lovelace alice@openpgp.example
To: Bob Babbage b.
250 Message 0MIjEq-1VEnJO3RVo-002UBE accepted by mreu2.kundenserver.de
QUIT
221 OK
```

We need to rectify IP Header length, which we will get from **len(pcaps[index][“IP”]).** We can directly replace IP header length value with new length of all the changed packets using **pcaps[index][“IP”].len = len(pcaps[index][“IP”]).**

3. Frame length:

Now if we view the pcap file using Wireshark we will encounter a “total length exceeds packet length”.



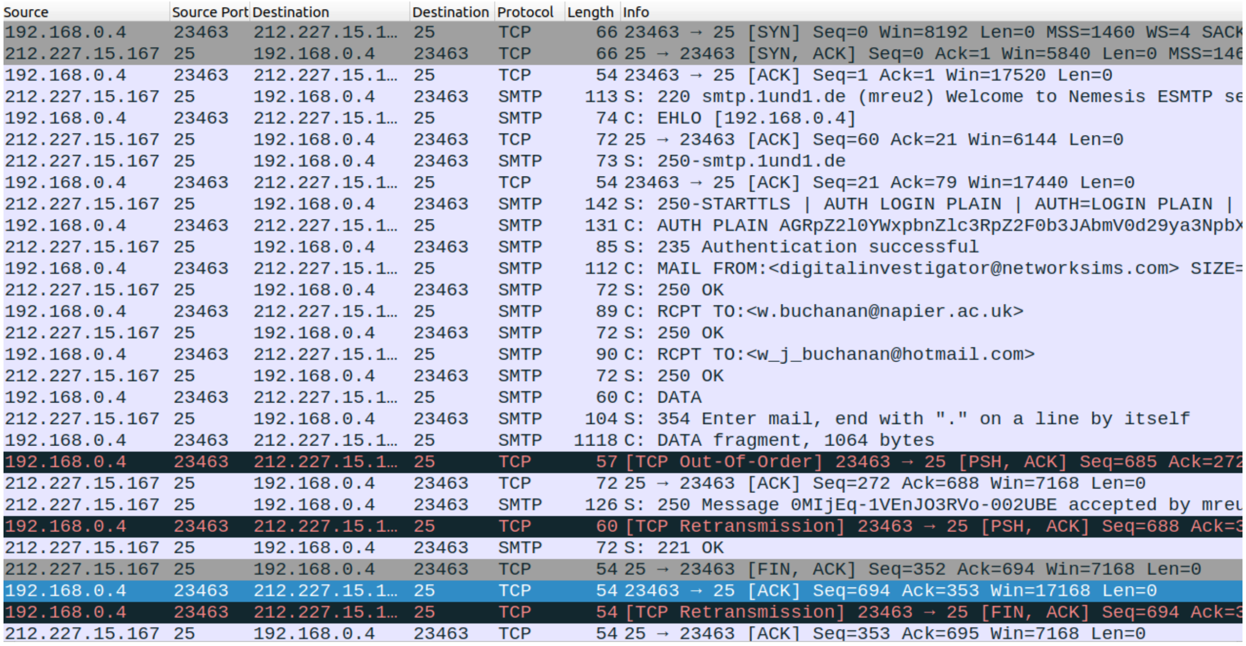
```
212.227.15.167  25      192.168.0.4     23463   SMTP     104 S: 354 Enter mail, end with "." on a line by itself
192.168.0.4     23463   212.227.15.1…   25      SMTP     506 C: DATA fragment, 452 bytes
192.168.0.4     23463   212.227.15.1…   25      SMTP      57 C: DATA fragment, 3 bytes
212.227.15.167  25      192.168.0.4     23463   TCP       72 25 → 23463 [ACK] Seq=272 Ack=688 Win=7168 Len=0
212.227.15.167  25      192.168.0.4     23463   SMTP     126 S: 250 Message 0MIjEq-1VEnJO3RVo-002UBE accepted by
192.168.0.4     23463   212.227.15.1…   25      SMTP      60 C: DATA fragment, 6 bytes

› Frame 20: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits)
› Ethernet II, Src: IntelCor_4b:82:37 (c8:f7:33:4b:82:37), Dst: Sagemcom_64:16:49 (4c:17:eb:64:16:49)
⌄ Internet Protocol Version 4, Src: 192.168.0.4, Dst: 212.227.15.167
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  › Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  ⌄ Total Length: 1104
    ⌄ [Expert Info (Error/Protocol): IPv4 total length exceeds packet length (492 bytes)]
        [IPv4 total length exceeds packet length (492 bytes)]
        [Severity level: Error]
        [Group: Protocol]
    Identification: 0x401f (16415)
  › Flags: 0x4000, Don't fragment
```

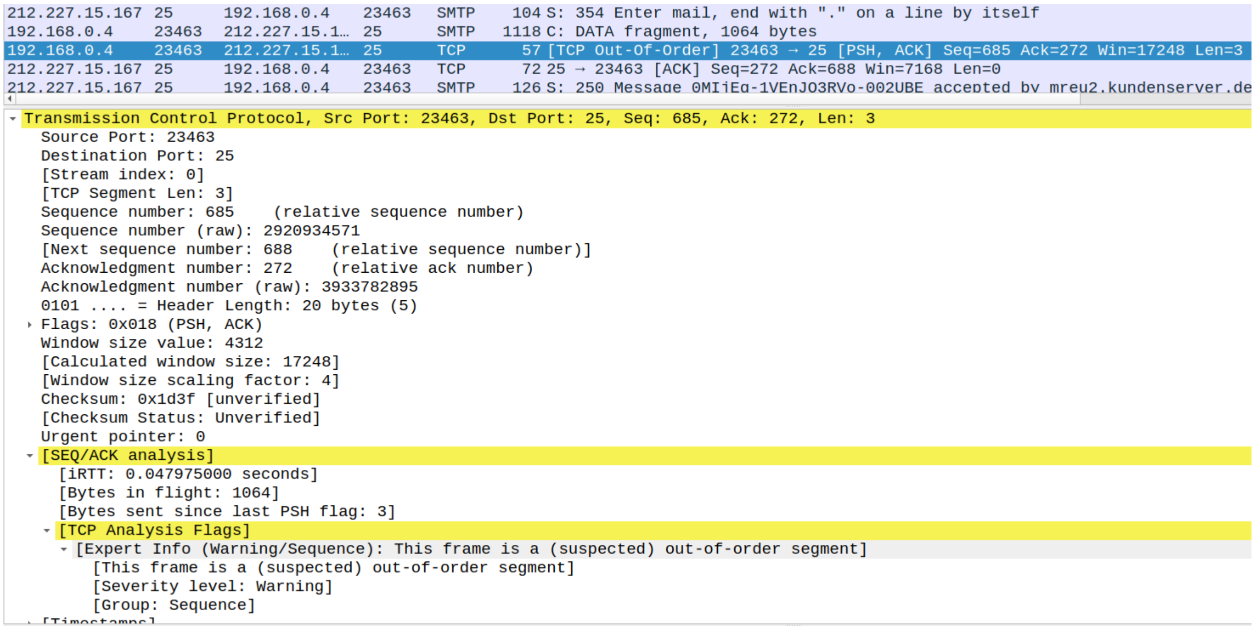
This means that the specified packet length in **Frame length** is less than total packet length. So, now we need to change **Frame length.** Scapy does not provide the option to change **Frame Length.** From Scapy we can get total length of the packet using **len(pcaps[index])** and convert that in hexadecimal. The last four bytes of data-link layer are for Frame length after that Ethernet layer starts and can view those hexadecimal values start bytes in Wireshark. Using Ghex application we can edit Frame length value to hexadecimal value of packet length.

4. Sequence and Acknowledgement numbers:

Now if we view the packet with Wireshark, we will encounter with “TCP Out-Of-Order" error and all the packets with same sequence series will show error of “TCP Retransmission”.



```
Source          Source Port Destination     Destination Protocol Length Info
192.168.0.4     23463  212.227.15.1…  25     TCP       66 23463 → 25 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK
212.227.15.167  25     192.168.0.4    23463  TCP       66 25 → 23463 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
192.168.0.4     23463  212.227.15.1…  25     TCP       54 23463 → 25 [ACK] Seq=1 Ack=1 Win=17520 Len=0
212.227.15.167  25     192.168.0.4    23463  SMTP     113 S: 220 smtp.1und1.de (mreu2) Welcome to Nemesis ESMTP se
192.168.0.4     23463  212.227.15.1…  25     SMTP      74 C: EHLO [192.168.0.4]
212.227.15.167  25     192.168.0.4    23463  TCP       72 25 → 23463 [ACK] Seq=60 Ack=21 Win=6144 Len=0
212.227.15.167  25     192.168.0.4    23463  SMTP      73 S: 250-smtp.1und1.de
192.168.0.4     23463  212.227.15.1…  25     TCP       54 23463 → 25 [ACK] Seq=21 Ack=79 Win=17440 Len=0
212.227.15.167  25     192.168.0.4    23463  SMTP     142 S: 250-STARTTLS | AUTH LOGIN PLAIN | AUTH=LOGIN PLAIN |
192.168.0.4     23463  212.227.15.1…  25     SMTP     131 C: AUTH PLAIN AGRpZ2l0YWxpbnZlc3RpZ2F0b3JAbmV0d29ya3NpbX
212.227.15.167  25     192.168.0.4    23463  SMTP      85 S: 235 Authentication successful
192.168.0.4     23463  212.227.15.1…  25     SMTP     112 C: MAIL FROM:<digitalinvestigator@networksims.com> SIZE=
212.227.15.167  25     192.168.0.4    23463  SMTP      72 S: 250 OK
192.168.0.4     23463  212.227.15.1…  25     SMTP      89 C: RCPT TO:<w.buchanan@napier.ac.uk>
212.227.15.167  25     192.168.0.4    23463  SMTP      72 S: 250 OK
192.168.0.4     23463  212.227.15.1…  25     SMTP      90 C: RCPT TO:<w_j_buchanan@hotmail.com>
212.227.15.167  25     192.168.0.4    23463  SMTP      72 S: 250 OK
192.168.0.4     23463  212.227.15.1…  25     SMTP      60 C: DATA
212.227.15.167  25     192.168.0.4    23463  SMTP     104 S: 354 Enter mail, end with "." on a line by itself
192.168.0.4     23463  212.227.15.1…  25     SMTP    1118 C: DATA fragment, 1064 bytes
192.168.0.4     23463  212.227.15.1…  25     TCP       57 [TCP Out-Of-Order] 23463 → 25 [PSH, ACK] Seq=685 Ack=272
212.227.15.167  25     192.168.0.4    23463  TCP       72 25 → 23463 [ACK] Seq=272 Ack=688 Win=7168 Len=0
212.227.15.167  25     192.168.0.4    23463  SMTP     126 S: 250 Message 0MIjEq-1VEnJO3RVo-002UBE accepted by mreu
192.168.0.4     23463  212.227.15.1…  25     TCP       60 [TCP Retransmission] 23463 → 25 [PSH, ACK] Seq=688 Ack=3
212.227.15.167  25     192.168.0.4    23463  SMTP      72 S: 221 OK
212.227.15.167  25     192.168.0.4    23463  TCP       54 25 → 23463 [FIN, ACK] Seq=352 Ack=694 Win=7168 Len=0
192.168.0.4     23463  212.227.15.1…  25     TCP       54 23463 → 25 [ACK] Seq=694 Ack=353 Win=17168 Len=0
192.168.0.4     23463  212.227.15.1…  25     TCP       54 [TCP Retransmission] 23463 → 25 [FIN, ACK] Seq=694 Ack=3
212.227.15.167  25     192.168.0.4    23463  TCP       54 25 → 23463 [ACK] Seq=353 Ack=695 Win=7168 Len=0
```

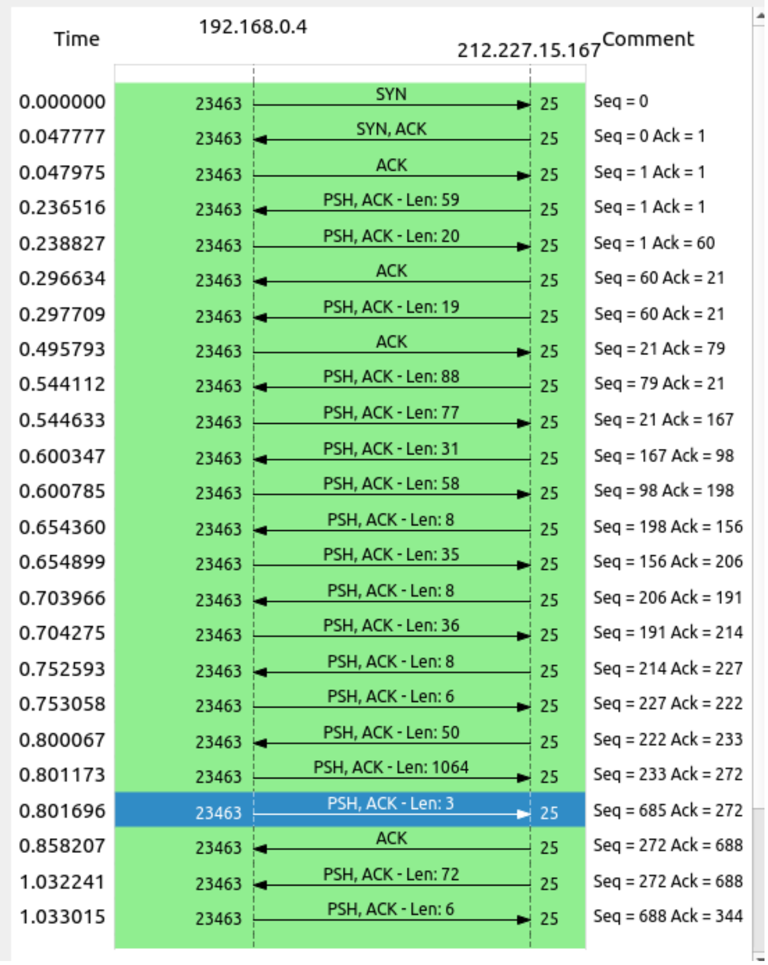


```
212.227.15.167  25     192.168.0.4    23463  SMTP     104 S: 354 Enter mail, end with "." on a line by itself
192.168.0.4     23463  212.227.15.1…  25     SMTP    1118 C: DATA fragment, 1064 bytes
192.168.0.4     23463  212.227.15.1…  25     TCP       57 [TCP Out-Of-Order] 23463 → 25 [PSH, ACK] Seq=685 Ack=272 Win=17248 Len=3
212.227.15.167  25     192.168.0.4    23463  TCP       72 25 → 23463 [ACK] Seq=272 Ack=688 Win=7168 Len=0
212.227.15.167  25     192.168.0.4    23463  SMTP     126 S: 250 Message 0MIjEq-1VEnJO3RVo-002UBE accepted by mreu2.kundenserver.de

▾ Transmission Control Protocol, Src Port: 23463, Dst Port: 25, Seq: 685, Ack: 272, Len: 3
    Source Port: 23463
    Destination Port: 25
    [Stream index: 0]
    [TCP Segment Len: 3]
    Sequence number: 685    (relative sequence number)
    Sequence number (raw): 2920934571
    [Next sequence number: 688    (relative sequence number)]
    Acknowledgment number: 272    (relative ack number)
    Acknowledgment number (raw): 3933782895
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 4312
    [Calculated window size: 17248]
    [Window size scaling factor: 4]
    Checksum: 0x1d3f [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▾ [SEQ/ACK analysis]
      [iRTT: 0.047975000 seconds]
      [Bytes in flight: 1064]
      [Bytes sent since last PSH flag: 3]
    ▾ [TCP Analysis Flags]
      ▾ [Expert Info (Warning/Sequence): This frame is a (suspected) out-of-order segment]
          [This frame is a (suspected) out-of-order segment]
          [Severity level: Warning]
          [Group: Sequence]
  ▸ [Timestamps]
```

This is coming as TCP sequence number not matching the expected data length.

The client of TCP keeps track of the amount of data sent (payload size) on each packet. This sequence number is included on each transmitted packet and acknowledged by the opposite host as an acknowledgement number to inform the sending host that the transmitted data was received successfully. Verified sequence and acknowledgement number from Wireshark by navigating **Statistics -> Flow Graph** and then select **TCP flow** and click **Ok**. Wireshark shows graphical view of the TCP sequence and acknowledgement number with zero initialization.



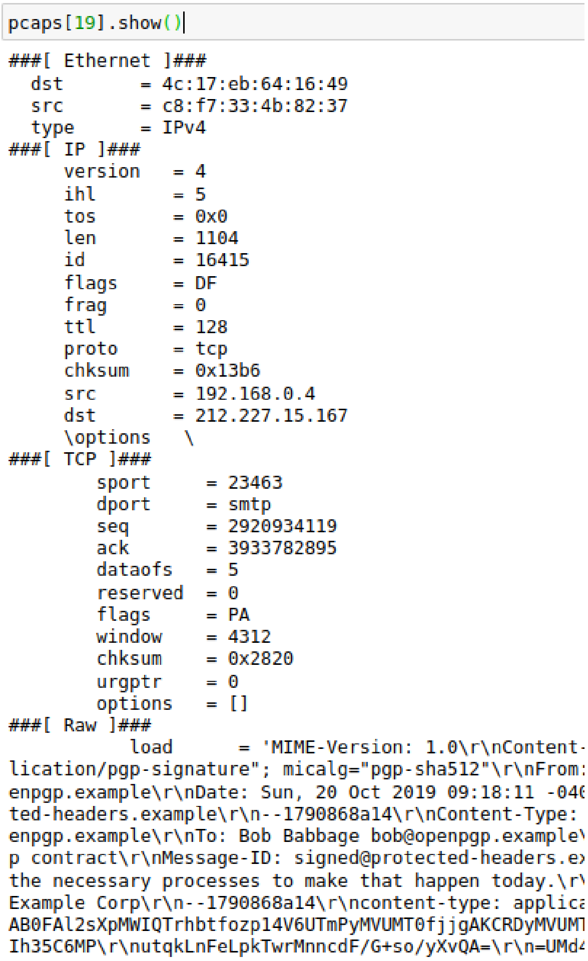
Time	192.168.0.4		212.227.15.167	Comment
0.000000	23463	SYN →	25	Seq = 0
0.047777	23463	← SYN, ACK	25	Seq = 0 Ack = 1
0.047975	23463	ACK →	25	Seq = 1 Ack = 1
0.236516	23463	← PSH, ACK - Len: 59	25	Seq = 1 Ack = 1
0.238827	23463	PSH, ACK - Len: 20 →	25	Seq = 1 Ack = 60
0.296634	23463	← ACK	25	Seq = 60 Ack = 21
0.297709	23463	← PSH, ACK - Len: 19	25	Seq = 60 Ack = 21
0.495793	23463	ACK →	25	Seq = 21 Ack = 79
0.544112	23463	← PSH, ACK - Len: 88	25	Seq = 79 Ack = 21
0.544633	23463	PSH, ACK - Len: 77 →	25	Seq = 21 Ack = 167
0.600347	23463	← PSH, ACK - Len: 31	25	Seq = 167 Ack = 98
0.600785	23463	PSH, ACK - Len: 58 →	25	Seq = 98 Ack = 198
0.654360	23463	← PSH, ACK - Len: 8	25	Seq = 198 Ack = 156
0.654899	23463	PSH, ACK - Len: 35 →	25	Seq = 156 Ack = 206
0.703966	23463	← PSH, ACK - Len: 8	25	Seq = 206 Ack = 191
0.704275	23463	PSH, ACK - Len: 36 →	25	Seq = 191 Ack = 214
0.752593	23463	← PSH, ACK - Len: 8	25	Seq = 214 Ack = 227
0.753058	23463	PSH, ACK - Len: 6 →	25	Seq = 227 Ack = 222
0.800067	23463	← PSH, ACK - Len: 50	25	Seq = 222 Ack = 233
0.801173	23463	PSH, ACK - Len: 1064 →	25	Seq = 233 Ack = 272
0.801696	23463	PSH, ACK - Len: 3 →	25	Seq = 685 Ack = 272
0.858207	23463	← ACK	25	Seq = 272 Ack = 688
1.032241	23463	← PSH, ACK - Len: 72	25	Seq = 272 Ack = 688
1.033015	23463	PSH, ACK - Len: 6 →	25	Seq = 688 Ack = 344

The increase of the sequence series in the next packet will be same as current packet's payload length. Here, in this example showing in blue the sequence number from the previous packet from 233 and payload length of 1064 should be (233+1064) = 1297, but here it is 685. So, we need to increase that sequence for all the next packets.

Using Scapy we have done that increment programmatically, and then viewed that capture by Wireshark.

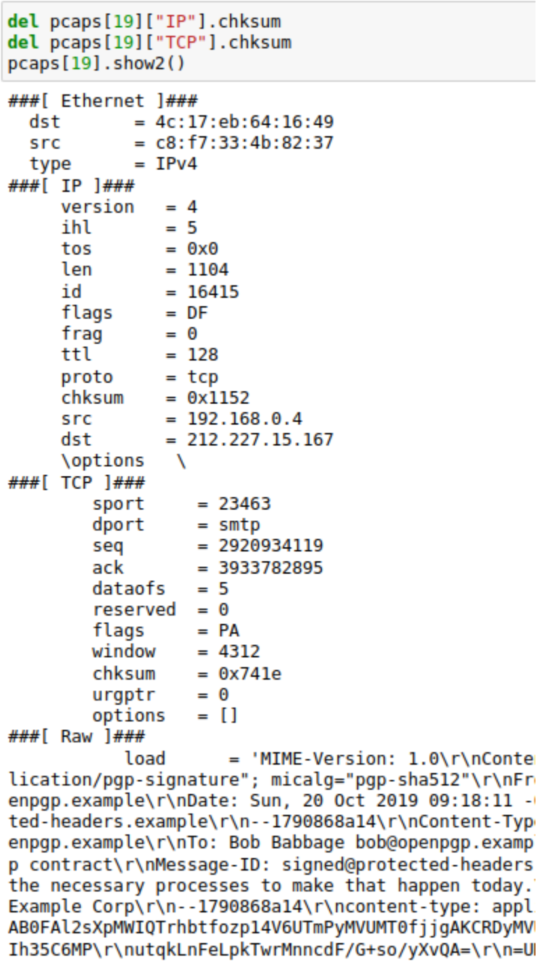
5. Rectifying IP and TCP checksum:

As we are changing headers data the IP and TCP checksum will become wrong. Scapy provides the option to recalculate those checksums. If we delete IP and TCP checksum of every packet, then scapy will automatically recalculate checksum while saving and we can view calculated checksum using **show2()** function. Below figures are shown before correcting checksum and after correcting checksum.



```
pcaps[19].show()
```

```
###[ Ethernet ]###
  dst       = 4c:17:eb:64:16:49
  src       = c8:f7:33:4b:82:37
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 1104
     id        = 16415
     flags     = DF
     frag      = 0
     ttl       = 128
     proto     = tcp
     chksum    = 0x13b6
     src       = 192.168.0.4
     dst       = 212.227.15.167
     \options   \
###[ TCP ]###
        sport     = 23463
        dport     = smtp
        seq       = 2920934119
        ack       = 3933782895
        dataofs   = 5
        reserved  = 0
        flags     = PA
        window    = 4312
        chksum    = 0x2820
        urgptr    = 0
        options   = []
###[ Raw ]###
           load      = 'MIME-Version: 1.0\r\nContent-
lication/pgp-signature"; micalg="pgp-sha512"\r\nFrom:
enpgp.example\r\nDate: Sun, 20 Oct 2019 09:18:11 -04
ted-headers.example\r\n--1790868a14\r\nContent-Type:
enpgp.example\r\nTo: Bob Babbage bob@openpgp.example
p contract\r\nMessage-ID: signed@protected-headers.e
the necessary processes to make that happen today.\r
Example Corp\r\n--1790868a14\r\ncontent-type: applic
AB0FAl2sXpMWIQTrhbtfozp14V6UTmPyMVUMT0fjjgAKCRDyMVUM
Ih35C6MP\r\nutqkLnFeLpkTwrMnncdF/G+so/yXvQA=\r\n=UMd
```

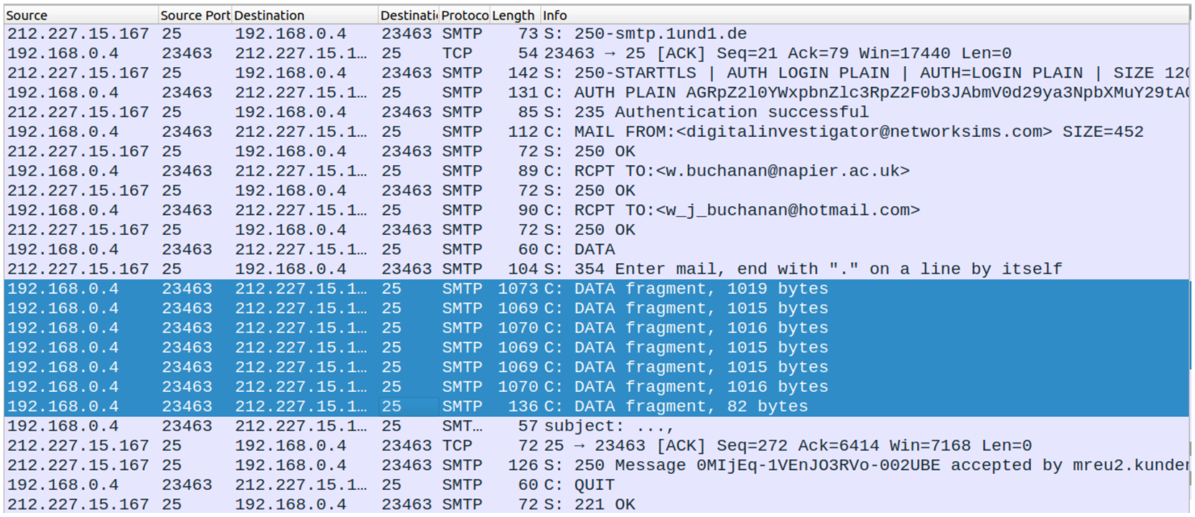


```
del pcaps[19]["IP"].chksum
del pcaps[19]["TCP"].chksum
pcaps[19].show2()
```

```
###[ Ethernet ]###
  dst       = 4c:17:eb:64:16:49
  src       = c8:f7:33:4b:82:37
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 1104
     id        = 16415
     flags     = DF
     frag      = 0
     ttl       = 128
     proto     = tcp
     chksum    = 0x1152
     src       = 192.168.0.4
     dst       = 212.227.15.167
     \options   \
###[ TCP ]###
        sport     = 23463
        dport     = smtp
        seq       = 2920934119
        ack       = 3933782895
        dataofs   = 5
        reserved  = 0
        flags     = PA
        window    = 4312
        chksum    = 0x741e
        urgptr    = 0
        options   = []
###[ Raw ]###
           load      = 'MIME-Version: 1.0\r\nConte
lication/pgp-signature"; micalg="pgp-sha512"\r\nFr
enpgp.example\r\nDate: Sun, 20 Oct 2019 09:18:11 -
ted-headers.example\r\n--1790868a14\r\nContent-Typ
enpgp.example\r\nTo: Bob Babbage bob@openpgp.examp
p contract\r\nMessage-ID: signed@protected-headers
the necessary processes to make that happen today.
Example Corp\r\n--1790868a14\r\ncontent-type: appl
AB0FAl2sXpMWIQTrhbtfozp14V6UTmPyMV
Ih35C6MP\r\nutqkLnFeLpkTwrMnncdF/G+so/yXvQA=\r\n=U
```

6. Fragmented payload:

Every SMTP server may have a fixed upper limit on message size. Any attempt by a client to transfer a message which is larger than that fixed upper limit will fail. For the message size limit constrain we have made fragmented packets if the message size is more than approximately 1000 bytes. For fragmented payload we need to add multiple packets as per message size and we need to rectify the above stated parameters and Identification number of IP layer.



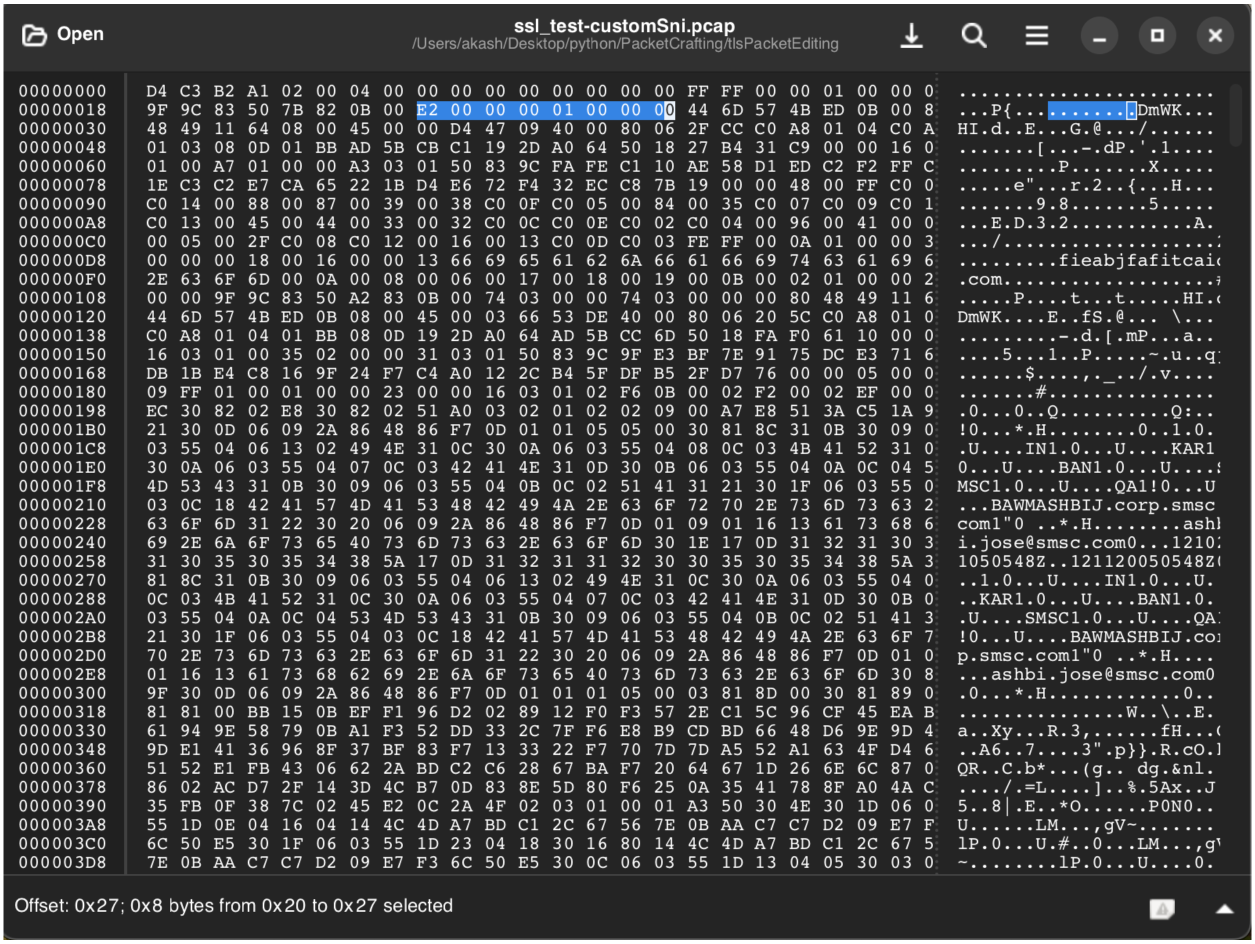
```
Source          Source Port Destination      Destinati Protoco Length Info
212.227.15.167  25          192.168.0.4      23463 SMTP     73 S: 250-smtp.1und1.de
192.168.0.4     23463       212.227.15.1…    25    TCP      54 23463 → 25 [ACK] Seq=21 Ack=79 Win=17440 Len=0
212.227.15.167  25          192.168.0.4      23463 SMTP    142 S: 250-STARTTLS | AUTH LOGIN PLAIN | AUTH=LOGIN PLAIN | SIZE 12
192.168.0.4     23463       212.227.15.1…    25    SMTP    131 C: AUTH PLAIN AGRpZ2l0YWxpbnZlc3RpZ2F0b3JAbmV0d29ya3NpbXMuY29tA
212.227.15.167  25          192.168.0.4      23463 SMTP     85 S: 235 Authentication successful
192.168.0.4     23463       212.227.15.1…    25    SMTP    112 C: MAIL FROM:<digitalinvestigator@networksims.com> SIZE=452
212.227.15.167  25          192.168.0.4      23463 SMTP     72 S: 250 OK
192.168.0.4     23463       212.227.15.1…    25    SMTP     89 C: RCPT TO:<w.buchanan@napier.ac.uk>
212.227.15.167  25          192.168.0.4      23463 SMTP     72 S: 250 OK
192.168.0.4     23463       212.227.15.1…    25    SMTP     90 C: RCPT TO:<w_j_buchanan@hotmail.com>
212.227.15.167  25          192.168.0.4      23463 SMTP     72 S: 250 OK
192.168.0.4     23463       212.227.15.1…    25    SMTP     60 C: DATA
212.227.15.167  25          192.168.0.4      23463 SMTP    104 S: 354 Enter mail, end with "." on a line by itself
192.168.0.4     23463       212.227.15.1…    25    SMTP   1073 C: DATA fragment, 1019 bytes
192.168.0.4     23463       212.227.15.1…    25    SMTP   1069 C: DATA fragment, 1015 bytes
192.168.0.4     23463       212.227.15.1…    25    SMTP   1070 C: DATA fragment, 1016 bytes
192.168.0.4     23463       212.227.15.1…    25    SMTP   1069 C: DATA fragment, 1015 bytes
192.168.0.4     23463       212.227.15.1…    25    SMTP   1069 C: DATA fragment, 1015 bytes
192.168.0.4     23463       212.227.15.1…    25    SMTP   1070 C: DATA fragment, 1016 bytes
192.168.0.4     23463       212.227.15.1…    25    SMTP    136 C: DATA fragment, 82 bytes
192.168.0.4     23463       212.227.15.1…    25    SMT…     57 subject: ...,
212.227.15.167  25          192.168.0.4      23463 TCP      72 25 → 23463 [ACK] Seq=272 Ack=6414 Win=7168 Len=0
212.227.15.167  25          192.168.0.4      23463 SMTP    126 S: 250 Message 0MIjEq-1VEnJO3RVo-002UBE accepted by mreu2.kunder
192.168.0.4     23463       212.227.15.1…    25    SMTP     60 C: QUIT
212.227.15.167  25          192.168.0.4      23463 SMTP     72 S: 221 OK
```

7. IP Identification number:

IP identification numbers are used for packet reassembly and should be unique within the pcap. In this case, we have two series of IP Identification numbers. In the fragmented packets, we are incrementing the series by one.

8. Frame layer correction:

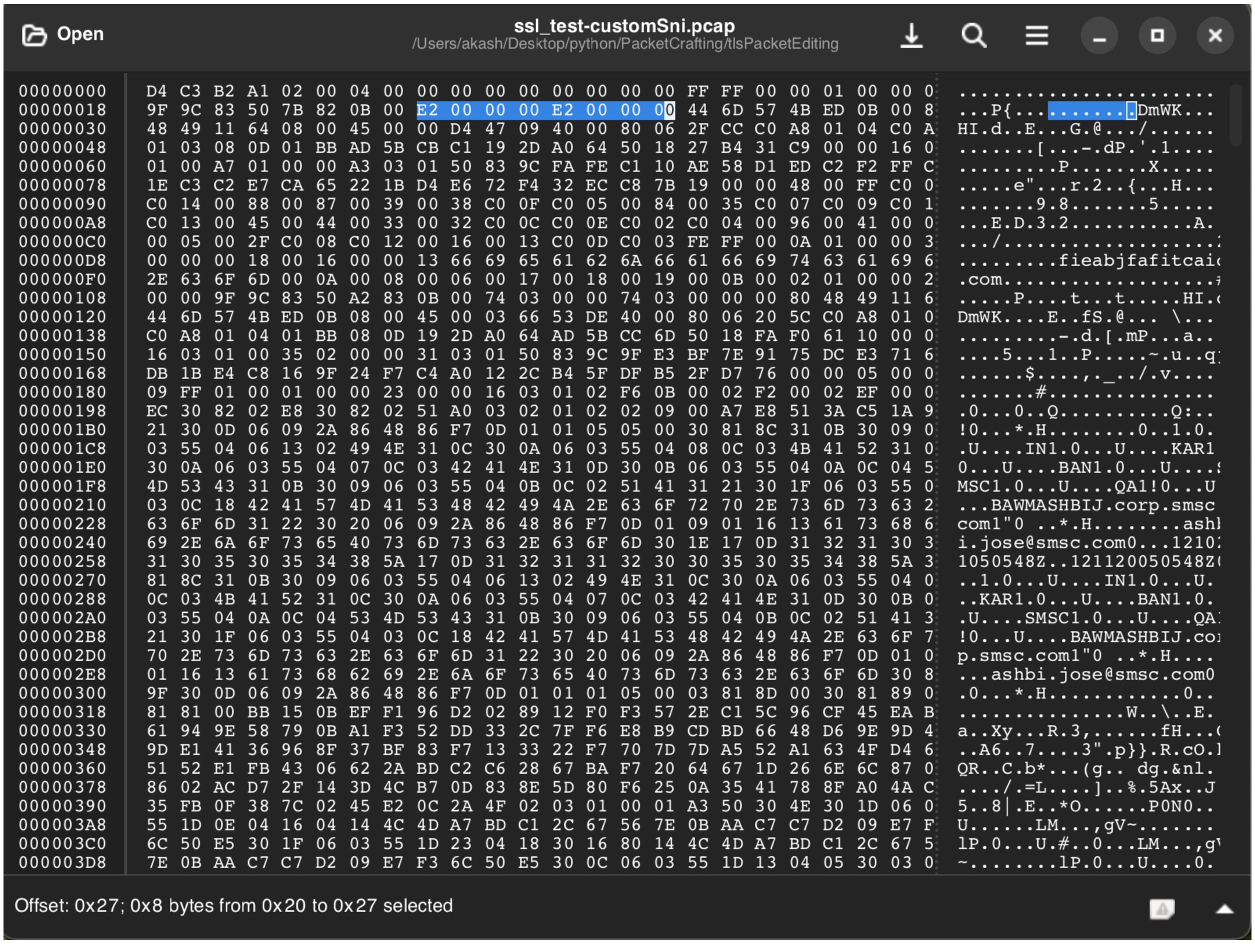
Scapy does not have the option to correct the frame layer for the edited packets, for this we will get error for the Frame layer while viewing .pcap file on the Wireshark. We can edit the frame layer using “GHex” editor.



ssl_test-customSni.pcap
/Users/akash/Desktop/python/PacketCrafting/tlsPacketEditing

```
00000000  D4 C3 B2 A1 02 00 04 00 00 00 00 00 00 00 00 00 FF FF 00 00 01 00 00 0  ........................
00000018  9F 9C 83 50 7B 82 0B 00 E2 00 00 00 01 00 00 00 44 6D 57 4B ED 0B 00 8  ...P{...........DmWK...
00000030  48 49 11 64 08 00 45 00 00 D4 47 09 40 00 80 06 2F CC C0 A8 01 04 C0 A  HI.d..E...G.@.../......
00000048  01 03 08 0D 01 BB AD 5B CB C1 19 2D A0 64 50 18 27 B4 31 C9 00 00 16 0  .......[...-.dP.'.1....
00000060  01 00 A7 01 00 00 A3 03 01 50 83 9C FA FE C1 10 AE 58 D1 ED C2 F2 FF C  .........P.......X.....
00000078  1E C3 C2 E7 CA 65 22 1B D4 E6 72 F4 32 EC C8 7B 19 00 00 48 00 FF C0 0  .....e"...r.2..{...H...
00000090  C0 14 00 88 00 87 00 39 00 38 C0 0F C0 05 00 84 00 35 C0 07 C0 09 C0 1  ........9.8........5.....
000000A8  C0 13 00 45 00 44 00 33 00 32 C0 0C C0 0E C0 02 C0 04 00 96 00 41 00 0  ...E.D.3.2...........A.
000000C0  00 05 00 2F C0 08 C0 12 00 16 00 13 C0 0D C0 03 FE FF 00 0A 01 00 00 3  .../.....................
000000D8  00 00 00 18 00 16 00 00 13 66 69 65 61 62 6A 66 61 66 69 74 63 61 69 6  .........fieabjfafitcai
000000F0  2E 63 6F 6D 00 0A 00 08 00 06 00 17 00 18 00 19 00 0B 00 02 01 00 00 2  .com....................
00000108  00 00 9F 9C 83 50 A2 83 0B 00 74 03 00 00 74 03 00 00 00 80 48 49 11 6  .....P....t...t.....HI.
00000120  44 6D 57 4B ED 0B 08 00 45 00 03 66 53 DE 40 00 80 06 20 5C C0 A8 01 0  DmWK....E..fS.@... \...
00000138  C0 A8 01 04 01 BB 08 0D 19 2D A0 64 AD 5B CC 6D 50 18 FA F0 61 10 00 0  ..........-.d.[.mP...a..
00000150  16 03 01 00 35 02 00 00 31 03 01 50 83 9C 9F E3 BF 7E 91 75 DC E3 71 6  ....5...1..P.....~.u..q
00000168  DB 1B E4 C8 16 9F 24 F7 C4 A0 12 2C B4 5F DF B5 2F D7 76 00 00 05 00 0  ......$....,._../.v....
00000180  09 FF 01 00 01 00 00 23 00 00 16 03 01 02 F6 0B 00 02 F2 00 02 EF 00 0  .......#...............
00000198  EC 30 82 02 E8 30 82 02 51 A0 03 02 01 02 02 09 00 A7 E8 51 3A C5 1A 9  .0...0..Q..........Q:..
000001B0  21 30 0D 06 09 2A 86 48 86 F7 0D 01 01 05 05 00 30 81 8C 31 0B 30 09 0  !0...*.H.........0..1.0.
000001C8  03 55 04 06 13 02 49 4E 31 0C 30 0A 06 03 55 04 08 0C 03 4B 41 52 31 0  .U....IN1.0...U....KAR1
000001E0  30 0A 06 03 55 04 07 0C 03 42 41 4E 31 0D 30 0B 06 03 55 04 0A 0C 04 5  0...U....BAN1.0...U....
000001F8  4D 53 43 31 0B 30 09 06 03 55 04 0B 0C 02 51 41 31 21 30 1F 06 03 55 0  MSC1.0...U....QA1!0...U
00000210  03 0C 18 42 41 57 4D 41 53 48 42 49 4A 2E 63 6F 72 70 2E 73 6D 73 63 2  ...BAWMASHBIJ.corp.smsc
00000228  63 6F 6D 31 22 30 20 06 09 2A 86 48 86 F7 0D 01 09 01 16 13 61 73 68 6  com1"0 ..*.H.........ash
00000240  69 2E 6A 6F 73 65 40 73 6D 73 63 2E 63 6F 6D 30 1E 17 0D 31 32 31 30 3  i.jose@smsc.com0...1210
00000258  31 30 35 30 35 34 38 5A 17 0D 31 32 31 31 32 30 30 35 30 35 34 38 5A 3  1050548Z..121120050548Z
00000270  81 8C 31 0B 30 09 06 03 55 04 06 13 02 49 4E 31 0C 30 0A 06 03 55 04 0  ..1.0...U....IN1.0...U.
00000288  0C 03 4B 41 52 31 0C 30 0A 06 03 55 04 07 0C 03 42 41 4E 31 0D 30 0B 0  ..KAR1.0...U....BAN1.0.
000002A0  03 55 04 0A 0C 04 53 4D 53 43 31 0B 30 09 06 03 55 04 0B 0C 02 51 41 3  .U....SMSC1.0...U....QA
000002B8  21 30 1F 06 03 55 04 03 0C 18 42 41 57 4D 41 53 48 42 49 4A 2E 63 6F 7  !0...U....BAWMASHBIJ.co
000002D0  70 2E 73 6D 73 63 2E 63 6F 6D 31 22 30 20 06 09 2A 86 48 86 F7 0D 01 0  p.smsc.com1"0 ..*.H....
000002E8  01 16 13 61 73 68 62 69 2E 6A 6F 73 65 40 73 6D 73 63 2E 63 6F 6D 30 8  ...ashbi.jose@smsc.com0
00000300  9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01 05 00 03 81 8D 00 30 81 89 0  .0...*.H............0..
00000318  81 81 00 BB 15 0B EF F1 96 D2 02 89 12 F0 F3 57 2E C1 5C 96 CF 45 EA B  ..................W..\..E.
00000330  61 94 9E 58 79 0B A1 F3 52 DD 33 2C 7F F6 E8 B9 CD BD 66 48 D6 9E 9D 4  a..Xy...R.3,......fH...
00000348  9D E1 41 36 96 8F 37 BF 83 F7 13 33 22 F7 70 7D 7D A5 52 A1 63 4F D4 6  ..A6..7....3".p}}.R.cO.
00000360  51 52 E1 FB 43 06 62 2A BD C2 C6 28 67 BA F7 20 64 67 1D 26 6E 6C 87 0  QR..C.b*...(g.. dg.&nl.
00000378  86 02 AC D7 2F 14 3D 4C B7 0D 83 8E 5D 80 F6 25 0A 35 41 78 8F A0 4A C  ..../.=L....]..%.5Ax..J
00000390  35 FB 0F 38 7C 02 45 E2 0C 2A 4F 02 03 01 00 01 A3 50 30 4E 30 1D 06 0  5..8|.E..*O......P0N0..
000003A8  55 1D 0E 04 16 04 14 4C 4D A7 BD C1 2C 67 56 7E 0B AA C7 C7 D2 09 E7 F  U......LM...,gV~.......
000003C0  6C 50 E5 30 1F 06 03 55 1D 23 04 18 30 16 80 14 4C 4D A7 BD C1 2C 67 5  lP.0...U.#..0...LM...,g
000003D8  7E 0B AA C7 C7 D2 09 E7 F3 6C 50 E5 30 0C 06 03 55 1D 13 04 05 30 03 0  ~.........lP.0...U....0.
```

Offset: 0x27; 0x8 bytes from 0x20 to 0x27 selected



ssl_test-customSni.pcap
/Users/akash/Desktop/python/PacketCrafting/tlsPacketEditing

```
00000000   D4 C3 B2 A1 02 00 04 00 00 00 00 00 00 00 00 00 FF FF 00 00 01 00 00 0   .......................
00000018   9F 9C 83 50 7B 82 0B 00 E2 00 00 00 E2 00 00 00 44 6D 57 4B ED 0B 00 8   ...P{...........DmWK...
00000030   48 49 11 64 08 00 45 00 00 D4 47 09 40 00 80 06 2F CC C0 A8 01 04 C0 A   HI.d..E...G.@.../......
00000048   01 03 08 0D 01 BB AD 5B CB C1 19 2D A0 64 50 18 27 B4 31 C9 00 00 16 0   .......[...-.dP.'.1....
00000060   01 00 A7 01 00 00 A3 03 01 50 83 9C FA FE C1 10 AE 58 D1 ED C2 F2 FF C   .........P.......X.....
00000078   1E C3 C2 E7 CA 65 22 1B D4 E6 72 F4 32 EC C8 7B 19 00 00 48 00 FF C0 0   .....e"...r.2..{...H...
00000090   C0 14 00 88 00 87 00 39 00 38 C0 0F C0 05 00 84 00 35 C0 07 C0 09 C0 1   .......9.8.......5.....
000000A8   C0 13 00 45 00 44 00 33 00 32 C0 0C C0 0E C0 02 C0 04 00 96 00 41 00 0   ...E.D.3.2...........A.
000000C0   00 05 00 2F C0 08 C0 12 00 16 00 13 C0 0D C0 03 FE FF 00 0A 01 00 00 3   .../...................
000000D8   00 00 00 18 00 16 00 00 13 66 69 65 61 62 6A 66 61 66 69 74 63 61 69 6   .........fieabjfafitcai
000000F0   2E 63 6F 6D 00 0A 00 08 00 06 00 17 00 18 00 19 00 0B 00 02 01 00 00 2   .com...................
00000108   00 00 9F 9C 83 50 A2 83 0B 00 74 03 00 00 74 03 00 00 00 80 48 49 11 6   .....P....t...t.....HI.
00000120   44 6D 57 4B ED 0B 08 00 45 00 03 66 53 DE 40 00 80 06 20 5C C0 A8 01 0   DmWK....E..fS.@... \...
00000138   C0 A8 01 04 01 BB 08 0D 19 2D A0 64 AD 5B CC 6D 50 18 FA F0 61 10 00 0   .........-.d.[.mP...a..
00000150   16 03 01 00 35 02 00 00 31 03 01 50 83 9C 9F E3 BF 7E 91 75 DC E3 71 6   ....5...1..P.....~.u..q
00000168   DB 1B E4 C8 16 9F 24 F7 C4 A0 12 2C B4 5F DF B5 2F D7 76 00 00 05 00 0   ......$....,._../.v....
00000180   09 FF 01 00 01 00 00 23 00 00 16 03 01 02 F6 0B 00 02 F2 00 02 EF 00 0   .......#...............
00000198   EC 30 82 02 E8 30 82 02 51 A0 03 02 01 02 02 09 00 A7 E8 51 3A C5 1A 9   .0...0..Q..........Q:..
000001B0   21 30 0D 06 09 2A 86 48 86 F7 0D 01 01 05 05 00 30 81 8C 31 0B 30 09 0   !0...*.H........0..1.0.
000001C8   03 55 04 06 13 02 49 4E 31 0C 30 0A 06 03 55 04 08 0C 03 4B 41 52 31 0   .U....IN1.0...U....KAR1
000001E0   30 0A 06 03 55 04 07 0C 03 42 41 4E 31 0D 30 0B 06 03 55 04 0A 0C 04 5   0...U....BAN1.0...U....
000001F8   4D 53 43 31 0B 30 09 06 03 55 04 0B 0C 02 51 41 31 21 30 1F 06 03 55 0   MSC1.0...U....QA1!0...U
00000210   03 0C 18 42 41 57 4D 41 53 48 42 49 4A 2E 63 6F 72 70 2E 73 6D 73 63 2   ...BAWMASHBIJ.corp.smsc
00000228   63 6F 6D 31 22 30 20 06 09 2A 86 48 86 F7 0D 01 09 01 16 13 61 73 68 6   com1"0 ..*.H........ash
00000240   69 2E 6A 6F 73 65 40 73 6D 73 63 2E 63 6F 6D 30 1E 17 0D 31 32 31 30 3   i.jose@smsc.com0...1210
00000258   31 30 35 30 35 34 38 5A 17 0D 31 32 31 31 32 30 30 35 30 35 34 38 5A 3   1050548Z..121120050548Z
00000270   81 8C 31 0B 30 09 06 03 55 04 06 13 02 49 4E 31 0C 30 0A 06 03 55 04 0   ..1.0...U....IN1.0...U.
00000288   0C 03 4B 41 52 31 0C 30 0A 06 03 55 04 07 0C 03 42 41 4E 31 0D 30 0B 0   ..KAR1.0...U....BAN1.0.
000002A0   03 55 04 0A 0C 04 53 4D 53 43 31 0B 30 09 06 03 55 04 0B 0C 02 51 41 3   .U....SMSC1.0...U....QA
000002B8   21 30 1F 06 03 55 04 03 0C 18 42 41 57 4D 41 53 48 42 49 4A 2E 63 6F 7   !0...U....BAWMASHBIJ.co
000002D0   70 2E 73 6D 73 63 2E 63 6F 6D 31 22 30 20 06 09 2A 86 48 86 F7 0D 01 0   p.smsc.com1"0 ..*.H....
000002E8   01 16 13 61 73 68 62 69 2E 6A 6F 73 65 40 73 6D 73 63 2E 63 6F 6D 30 8   ...ashbi.jose@smsc.com0
00000300   9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01 05 00 03 81 8D 00 30 81 89 0   .0...*.H............0..
00000318   81 81 00 BB 15 0B EF F1 96 D2 02 89 12 F0 F3 57 2E C1 5C 96 CF 45 EA B   ...............W..\..E.
00000330   61 94 9E 58 79 0B A1 F3 52 DD 33 2C 7F F6 E8 B9 CD BD 66 48 D6 9E 9D 4   a..Xy...R.3,......fH...
00000348   9D E1 41 36 96 8F 37 BF 83 F7 13 33 22 F7 70 7D 7D A5 52 A1 63 4F D4 6   ..A6..7....3".p}}.R.cO.
00000360   51 52 E1 FB 43 06 62 2A BD C2 C6 28 67 BA F7 20 64 67 1D 26 6E 6C 87 0   QR..C.b*...(g.. dg.&nl.
00000378   86 02 AC D7 2F 14 3D 4C B7 0D 83 8E 5D 80 F6 25 0A 35 41 78 8F A0 4A C   ..../.=L....]..%.5Ax..J
00000390   35 FB 0F 38 7C 02 45 E2 0C 2A 4F 02 03 01 00 01 A3 50 30 4E 30 1D 06 0   5..8|.E..*O......P0N0..
000003A8   55 1D 0E 04 16 04 14 4C 4D A7 BD C1 2C 67 56 7E 0B AA C7 C7 D2 09 E7 F   U......LM...,gV~.......
000003C0   6C 50 E5 30 1F 06 03 55 1D 23 04 18 30 16 80 14 4C 4D A7 BD C1 2C 67 5   lP.0...U.#..0...LM...,g
000003D8   7E 0B AA C7 C7 D2 09 E7 F3 6C 50 E5 30 0C 06 03 55 1D 13 04 05 30 03 0   ~........lP.0...U....0.
```

Offset: 0x27; 0x8 bytes from 0x20 to 0x27 selected

Conclusion:

Packet crafting is a good way to audit network security and exploit vulnerably. This document gives a detailed description of Packet editing from an existing packet by using one of powerful Packet crafting tool Scapy. Here we have discussed editing SMTP protocol, but we can do this for email protocols (POP3, IMAP), HTTP and other protocols.

References:

1. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification (rfc8551).
2. OpenPGP Message Format (rfc4880).
3. Protected Headers for Cryptographic E-mail.Bjarni Rúnar Einarsson, Daniel Kahn Gillmor.
4. **https://github.com/autocrypt/protected-headers**.
5. Test Vectors for E-mail Header Protection.
6. TCP Sequence and Acknowledgement Numbers Explained.
7. Understanding TCP Sequence and Acknowledgment Numbers.
8. How to Inject Code into HTTP Responses in the Network in Python (Scapy).