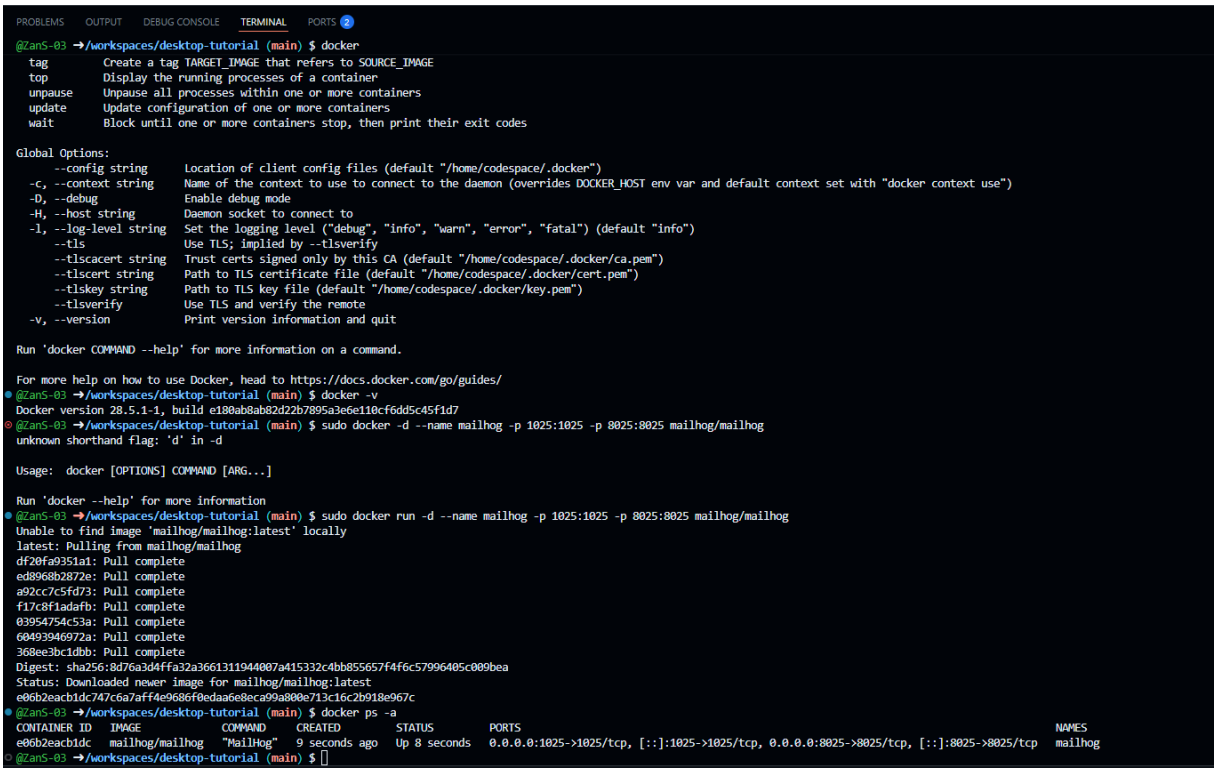
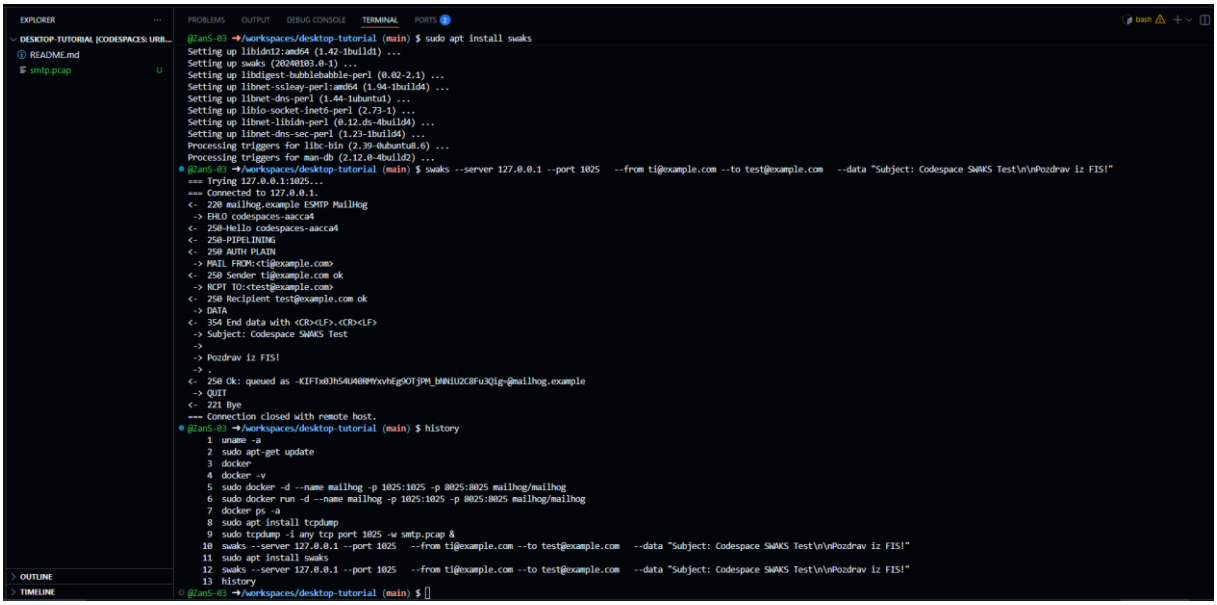
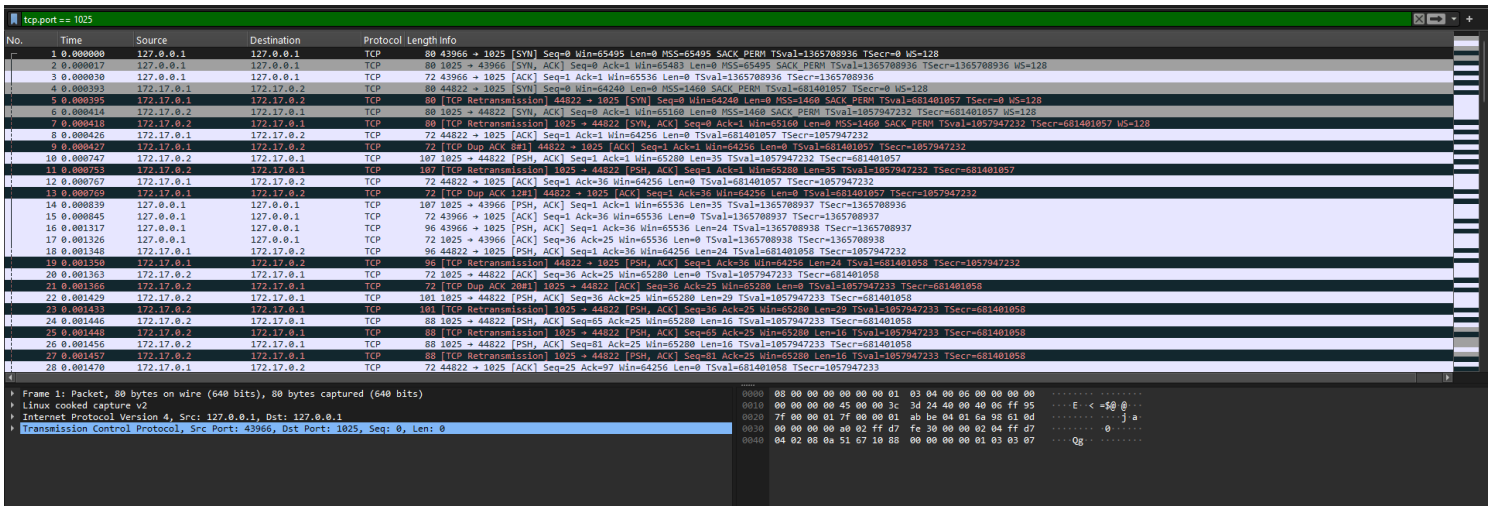
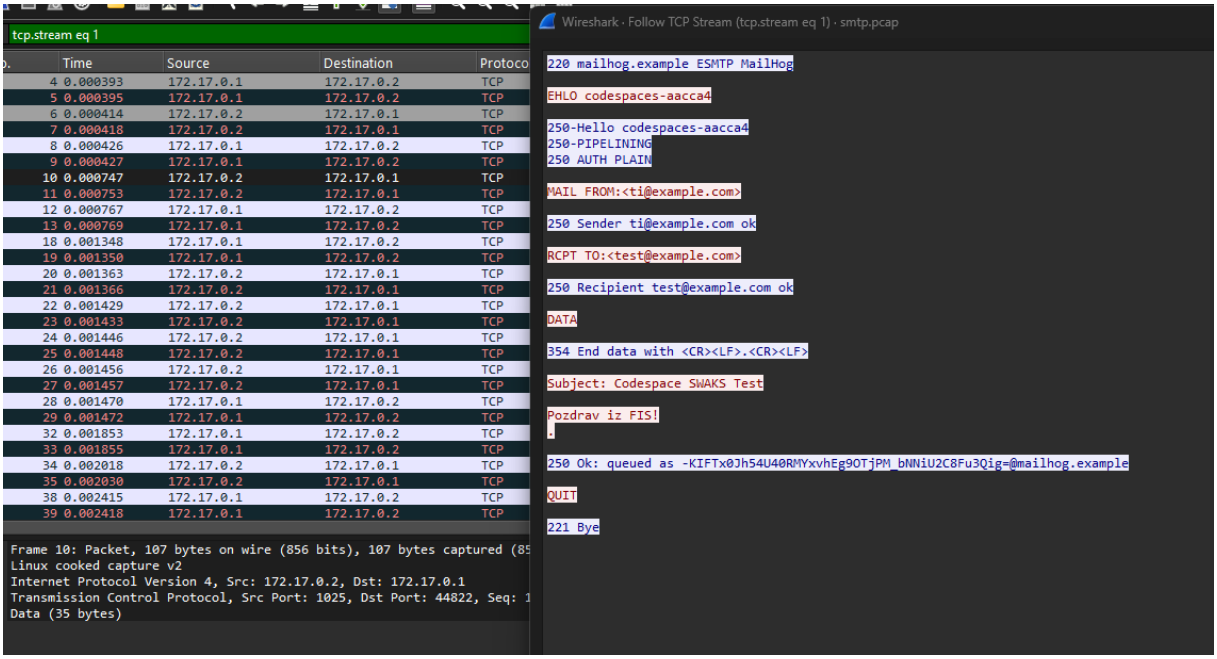


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 2

@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ docker
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  wait        Block until one or more containers stop, then print their exit codes

Global Options:
      --config string      Location of client config files (default "/home/codespace/.docker")
  -c, --context string     Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
  -D, --debug              Enable debug mode
  -H, --host string        Daemon socket to connect to
  -l, --log-level string   Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default "/home/codespace/.docker/ca.pem")
      --tlscert string     Path to TLS certificate file (default "/home/codespace/.docker/cert.pem")
      --tlskey string      Path to TLS key file (default "/home/codespace/.docker/key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ docker -v
Docker version 28.5.1-1, build e180ab8ab82d22b7895a3e6e110cf6dd5c45f1d7
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ sudo docker -d --name mailhog -p 1025:1025 -p 8025:8025 mailhog/mailhog
unknown shorthand flag: 'd' in -d

Usage:  docker [OPTIONS] COMMAND [ARG...]

Run 'docker --help' for more information
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ sudo docker run -d --name mailhog -p 1025:1025 -p 8025:8025 mailhog/mailhog
Unable to find image 'mailhog/mailhog:latest' locally
latest: Pulling from mailhog/mailhog
df20fa9351a1: Pull complete
ed8968b2872e: Pull complete
a92cc7c5fd73: Pull complete
f17c8f1adafb: Pull complete
03954754c53a: Pull complete
60493946972a: Pull complete
368ee3bc1dbb: Pull complete
Digest: sha256:8d76a3d4ffa32a3661311944007a415332c4bb855657f4f6c57996405c009bea
Status: Downloaded newer image for mailhog/mailhog:latest
e06b2eacb1dc747c6a7aff4e9686f0edaa6e8eca99a800e713c16c2b918e967c
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ docker ps -a
CONTAINER ID   IMAGE           COMMAND     CREATED         STATUS         PORTS                                                                                                                NAMES
e06b2eacb1dc   mailhog/mailhog   "MailHog"   9 seconds ago   Up 8 seconds   0.0.0.0:1025->1025/tcp, [::]:1025->1025/tcp, 0.0.0.0:8025->8025/tcp, [::]:8025->8025/tcp   mailhog
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $
```



```
EXPLORER
DESKTOP-TUTORIAL [CODESPACES: URB...
README.md
smtp.pcap  U
OUTLINE
TIMELINE

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 2                                                                                                  bash

@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ sudo apt install swaks
Setting up libidn12:amd64 (1.42-1build1) ...
Setting up swaks (20240103.0-1) ...
Setting up libdigest-bubblebabble-perl (0.02-2.1) ...
Setting up libnet-ssleay-perl:amd64 (1.94-1build4) ...
Setting up libnet-dns-perl (1.44-1ubuntu1) ...
Setting up libio-socket-inet6-perl (2.73-1) ...
Setting up libnet-libidn-perl (0.12.ds-4build4) ...
Setting up libnet-dns-sec-perl (1.23-1build4) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Processing triggers for man-db (2.12.0-4build2) ...
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ swaks --server 127.0.0.1 --port 1025   --from ti@example.com --to test@example.com   --data "Subject: Codespace SWAKS Test\n\nPozdrav iz FIS!"
=== Trying 127.0.0.1:1025...
=== Connected to 127.0.0.1.
<-  220 mailhog.example ESMTP MailHog
 -> EHLO codespaces-aacca4
<-  250-Hello codespaces-aacca4
<-  250-PIPELINING
<-  250 AUTH PLAIN
 -> MAIL FROM:<ti@example.com>
<-  250 Sender ti@example.com ok
 -> RCPT TO:<test@example.com>
<-  250 Recipient test@example.com ok
 -> DATA
<-  354 End data with <CR><LF>.<CR><LF>
 -> Subject: Codespace SWAKS Test
 ->
 -> Pozdrav iz FIS!
 ->
<-  250 Ok: queued as -KIFTx0Jh54U40RMYxvhEg9OTjPM_bNNiU2C8Fu3QIg-@mailhog.example
 -> QUIT
<-  221 Bye
=== Connection closed with remote host.
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ history
    1  uname -a
    2  sudo apt-get update
    3  docker
    4  docker -v
    5  sudo docker -d --name mailhog -p 1025:1025 -p 8025:8025 mailhog/mailhog
    6  sudo docker run -d --name mailhog -p 1025:1025 -p 8025:8025 mailhog/mailhog
    7  docker ps -a
    8  sudo apt install tcpdump
    9  sudo tcpdump -i any tcp port 1025 -w smtp.pcap &
   10  swaks --server 127.0.0.1 --port 1025   --from ti@example.com --to test@example.com   --data "Subject: Codespace SWAKS Test\n\nPozdrav iz FIS!"
   11  sudo apt install swaks
   12  swaks --server 127.0.0.1 --port 1025   --from ti@example.com --to test@example.com   --data "Subject: Codespace SWAKS Test\n\nPozdrav iz FIS!"
   13  history
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $
```

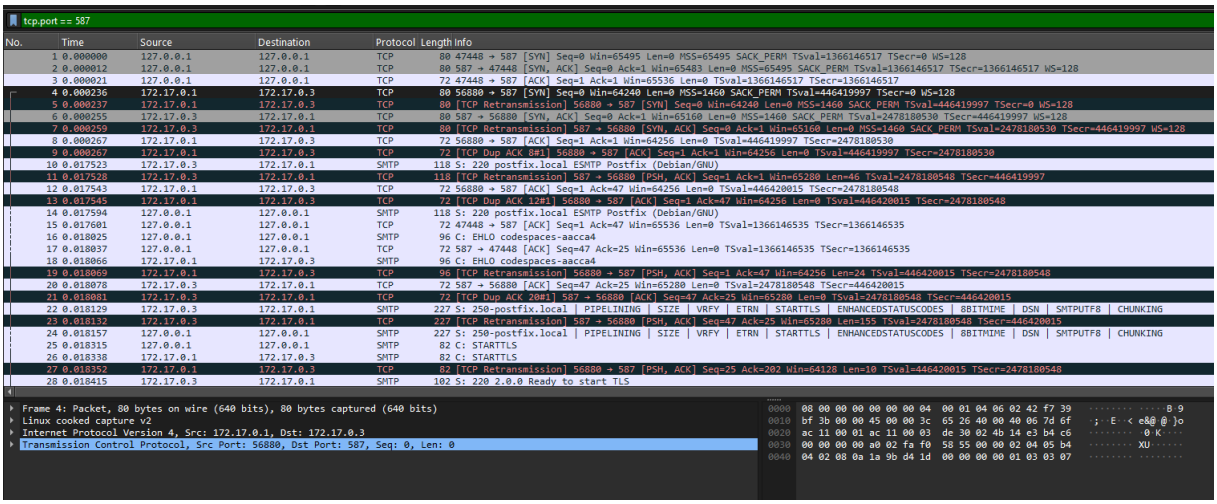

tcp.port == 1025
No. Time Source Destination Protocol Length Info
1 0.000000 127.0.0.1 127.0.0.1 TCP 80 43966 → 1025 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1365708936 TSecr=0 WS=128
2 0.000017 127.0.0.1 127.0.0.1 TCP 80 1025 → 43966 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=1365708936 TSecr=1365708936 WS=128
3 0.000030 127.0.0.1 127.0.0.1 TCP 72 43966 → 1025 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1365708936 TSecr=1365708936
4 0.000393 172.17.0.1 172.17.0.2 TCP 80 44822 → 1025 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=681401057 TSecr=0 WS=128
5 0.000395 172.17.0.1 172.17.0.2 TCP 80 [TCP Retransmission] 44822 → 1025 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=681401057 TSecr=0 WS=128
6 0.000414 172.17.0.2 172.17.0.1 TCP 80 1025 → 44822 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1057947232 TSecr=681401057 WS=128
7 0.000418 172.17.0.2 172.17.0.1 TCP 80 [TCP Retransmission] 1025 → 44822 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1057947232 TSecr=681401057 WS=128
8 0.000426 172.17.0.1 172.17.0.2 TCP 72 44822 → 1025 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=681401057 TSecr=1057947232
9 0.000427 172.17.0.1 172.17.0.2 TCP 72 [TCP Dup ACK 8#1] 44822 → 1025 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=681401057 TSecr=1057947232
10 0.000747 172.17.0.2 172.17.0.1 TCP 107 1025 → 44822 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=35 TSval=1057947232 TSecr=681401057
11 0.000753 172.17.0.2 172.17.0.1 TCP 107 [TCP Retransmission] 1025 → 44822 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=35 TSval=1057947232 TSecr=681401057
12 0.000767 172.17.0.1 172.17.0.2 TCP 72 44822 → 1025 [ACK] Seq=1 Ack=36 Win=64256 Len=0 TSval=681401057 TSecr=1057947232
13 0.000769 172.17.0.1 172.17.0.2 TCP 72 [TCP Dup ACK 12#1] 44822 → 1025 [ACK] Seq=1 Ack=36 Win=64256 Len=0 TSval=681401057 TSecr=1057947232
14 0.000839 127.0.0.1 127.0.0.1 TCP 107 1025 → 43966 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=35 TSval=1365708937 TSecr=1365708936
15 0.000845 127.0.0.1 127.0.0.1 TCP 72 43966 → 1025 [ACK] Seq=1 Ack=36 Win=65536 Len=0 TSval=1365708937 TSecr=1365708937
16 0.001317 127.0.0.1 127.0.0.1 TCP 96 43966 → 1025 [PSH, ACK] Seq=1 Ack=36 Win=65536 Len=24 TSval=1365708938 TSecr=1365708937
17 0.001326 127.0.0.1 127.0.0.1 TCP 72 1025 → 43966 [ACK] Seq=36 Ack=25 Win=65536 Len=0 TSval=1365708938 TSecr=1365708938
18 0.001348 172.17.0.1 172.17.0.2 TCP 96 44822 → 1025 [PSH, ACK] Seq=1 Ack=36 Win=64256 Len=24 TSval=681401058 TSecr=1057947232
19 0.001350 172.17.0.1 172.17.0.2 TCP 96 [TCP Retransmission] 44822 → 1025 [PSH, ACK] Seq=1 Ack=36 Win=64256 Len=24 TSval=681401058 TSecr=1057947232
20 0.001363 172.17.0.2 172.17.0.1 TCP 72 1025 → 44822 [ACK] Seq=36 Ack=25 Win=65280 Len=0 TSval=1057947233 TSecr=681401058
21 0.001366 172.17.0.2 172.17.0.1 TCP 72 [TCP Dup ACK 20#1] 1025 → 44822 [ACK] Seq=36 Ack=25 Win=65280 Len=0 TSval=1057947233 TSecr=681401058
22 0.001429 172.17.0.2 172.17.0.1 TCP 101 1025 → 44822 [PSH, ACK] Seq=36 Ack=25 Win=65280 Len=29 TSval=1057947233 TSecr=681401058
23 0.001433 172.17.0.2 172.17.0.1 TCP 101 [TCP Retransmission] 1025 → 44822 [PSH, ACK] Seq=36 Ack=25 Win=65280 Len=29 TSval=1057947233 TSecr=681401058
24 0.001446 172.17.0.2 172.17.0.1 TCP 88 1025 → 44822 [PSH, ACK] Seq=65 Ack=25 Win=65280 Len=16 TSval=1057947233 TSecr=681401058
25 0.001448 172.17.0.2 172.17.0.1 TCP 88 [TCP Retransmission] 1025 → 44822 [PSH, ACK] Seq=65 Ack=25 Win=65280 Len=16 TSval=1057947233 TSecr=681401058
26 0.001456 172.17.0.2 172.17.0.1 TCP 88 1025 → 44822 [PSH, ACK] Seq=81 Ack=25 Win=65280 Len=16 TSval=1057947233 TSecr=681401058
27 0.001457 172.17.0.2 172.17.0.1 TCP 88 [TCP Retransmission] 1025 → 44822 [PSH, ACK] Seq=81 Ack=25 Win=65280 Len=16 TSval=1057947233 TSecr=681401058
28 0.001470 172.17.0.1 172.17.0.2 TCP 72 44822 → 1025 [ACK] Seq=25 Ack=97 Win=64256 Len=0 TSval=681401058 TSecr=1057947233
Frame 1: Packet, 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
Linux cooked capture v2
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 43966, Dst Port: 1025, Seq: 0, Len: 0
0000 08 00 00 00 00 00 00 01 03 04 00 06 00 00 00 00
0010 00 00 00 00 45 00 00 3c 3d 24 40 00 40 06 ff 95
0020 7f 00 00 01 7f 00 00 01 ab be 04 01 6a 98 61 0d
0030 00 00 00 00 a0 02 ff d7 fe 30 00 00 02 04 ff d7
0040 04 02 08 0a 51 67 10 88 00 00 00 00 01 03 03 07

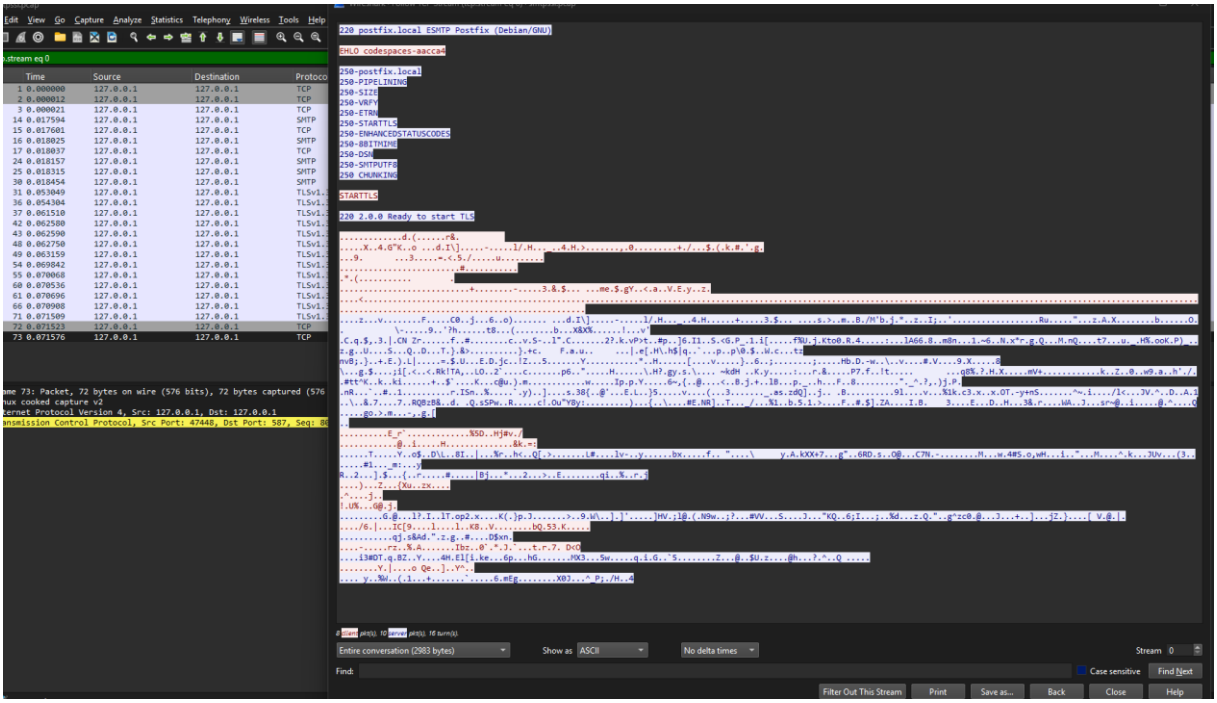

tcp.stream eq 1
Time Source Destination Protocol
4 0.000393 172.17.0.1 172.17.0.2 TCP
5 0.000395 172.17.0.1 172.17.0.2 TCP
6 0.000414 172.17.0.2 172.17.0.1 TCP
7 0.000418 172.17.0.2 172.17.0.1 TCP
8 0.000426 172.17.0.1 172.17.0.2 TCP
9 0.000427 172.17.0.1 172.17.0.2 TCP
10 0.000747 172.17.0.2 172.17.0.1 TCP
11 0.000753 172.17.0.2 172.17.0.1 TCP
12 0.000767 172.17.0.1 172.17.0.2 TCP
13 0.000769 172.17.0.1 172.17.0.2 TCP
18 0.001348 172.17.0.1 172.17.0.2 TCP
19 0.001350 172.17.0.1 172.17.0.2 TCP
20 0.001363 172.17.0.2 172.17.0.1 TCP
21 0.001366 172.17.0.2 172.17.0.1 TCP
22 0.001429 172.17.0.2 172.17.0.1 TCP
23 0.001433 172.17.0.2 172.17.0.1 TCP
24 0.001446 172.17.0.2 172.17.0.1 TCP
25 0.001448 172.17.0.2 172.17.0.1 TCP
26 0.001456 172.17.0.2 172.17.0.1 TCP
27 0.001457 172.17.0.2 172.17.0.1 TCP
28 0.001470 172.17.0.1 172.17.0.2 TCP
29 0.001472 172.17.0.1 172.17.0.2 TCP
32 0.001853 172.17.0.1 172.17.0.2 TCP
33 0.001855 172.17.0.1 172.17.0.2 TCP
34 0.002018 172.17.0.2 172.17.0.1 TCP
35 0.002030 172.17.0.2 172.17.0.1 TCP
38 0.002415 172.17.0.1 172.17.0.2 TCP
39 0.002418 172.17.0.1 172.17.0.2 TCP
Frame 10: Packet, 107 bytes on wire (856 bits), 107 bytes captured (85
Linux cooked capture v2
Internet Protocol Version 4, Src: 172.17.0.2, Dst: 172.17.0.1
Transmission Control Protocol, Src Port: 1025, Dst Port: 44822, Seq: 1
Data (35 bytes)
Wireshark · Follow TCP Stream (tcp.stream eq 1) · smtp.pcap
220 mailhog.example ESMTP MailHog
EHLO codespaces-aacca4
250-Hello codespaces-aacca4
250-PIPELINING
250 AUTH PLAIN
MAIL FROM:<ti@example.com>
250 Sender ti@example.com ok
RCPT TO:<test@example.com>
250 Recipient test@example.com ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Codespace SWAKS Test
Pozdrav iz FIS!
.
250 Ok: queued as -KIFTx0Jh54U40RMYxvhEg9OTjPM_bNNiU2C8Fu3Qig=@mailhog.example
QUIT
221 Bye

```
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ docker run -d --name postfix587   -p 587:587   -e ALLOWED_SENDER_DOMAINS="localdomain"   -e POSTFIX_myhostname=postfix.local   boky/postfix
Unable to find image 'boky/postfix:latest' locally
latest: Pulling from boky/postfix
af302e5c37e9: Pull complete
fd88f39fc50e: Pull complete
d03bb0060dde: Pull complete
0ba5e86a0cd4: Pull complete
f33184ad3f08: Pull complete
f518acd98a42: Pull complete
d0d48f97e6e0: Pull complete
a43fc3bfe47d: Pull complete
78ee785d01db: Pull complete
eaa7f414a812: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:f3f247fd42528b969e2603ac120d5a5b5db7dfe61f4505c49d438b9ba1822999
Status: Downloaded newer image for boky/postfix:latest
23416b2ce698a265a647ff1ff5324fa6e87534389253b7f30b190eb8e0274a87
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ docker ps -a
CONTAINER ID   IMAGE            COMMAND                  CREATED         STATUS                            PORTS                                                                                                NAMES
23416b2ce698   boky/postfix     "/bin/sh -c /scripts…"   4 seconds ago   Up 3 seconds (health: starting)   0.0.0.0:587->587/tcp, [::]:587->587/tcp                                                              postfix587
e06b2eacb1dc   mailhog/mailhog  "MailHog"                9 minutes ago   Up 8 minutes                      0.0.0.0:1025->1025/tcp, [::]:1025->1025/tcp, 0.0.0.0:8025->8025/tcp, [::]:8025->8025/tcp   mailhog
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $
```

```
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ sudo tcpdump -i any tcp port 587 -w smtpssl.pcap &
libibverbs: Warning: couldn't open config directory '/etc/libibverbs.d'.
tcpdump: data link type LINUX_SLL2
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
swaks --server 127.0.0.1 --port 587 --tls   --frswaks --server 127.0.0.1 --port 587 --tls   --from test@localdomain --to demc@localdomain   --header 'Subject: STARTTLS test'   --body 'Pozdrav prek TLS!'
=== Trying 127.0.0.1:587...
=== Connected to 127.0.0.1.
<-  220 postfix.local ESMTP Postfix (Debian/GNU)
 -> EHLO codespaces-aacca4
<-  250-postfix.local
<-  250-PIPELINING
<-  250-SIZE
<-  250-VRFY
<-  250-ETRN
<-  250-STARTTLS
<-  250-ENHANCEDSTATUSCODES
<-  250-8BITMIME
<-  250-DSN
<-  250-SMTPUTF8
<-  250 CHUNKING
 -> STARTTLS
<-  220 2.0.0 Ready to start TLS
=== TLS started with cipher TLSv1.3:TLS_AES_256_GCM_SHA384:256
=== TLS client certificate not requested and not sent
=== TLS no client certificate set
=== TLS peer[0]   subject=[/CN=localhost]
===               commonName=[localhost], subjectAltName=[DNS:localhost] notAfter=[2035-01-14T23:01:22Z]
=== TLS peer certificate failed CA verification (self-signed certificate), failed host verification (using host 127.0.0.1 to verify)
 ~> EHLO codespaces-aacca4
<~  250-postfix.local
<~  250-PIPELINING
<~  250-SIZE
<~  250-VRFY
<~  250-ETRN
<~  250-ENHANCEDSTATUSCODES
<~  250-8BITMIME
<~  250-DSN
<~  250-SMTPUTF8
<~  250 CHUNKING
 ~> MAIL FROM:<test@localdomain>
<~  250 2.1.0 Ok
 ~> RCPT TO:<demc@localdomain>
<~* 504 5.5.2 <demc@localdomain>: Recipient address rejected: need fully-qualified address
 ~> QUIT
<~  221 2.0.0 Bye
=== Connection closed with remote host.
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ sudo pkill tcpdumpž
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $
```

```
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ sudo pkill tcpdumpž
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $ history
    1  uname -a
    2  sudo apt-get update
    3  docker
    4  docker -v
    5  sudo docker -d --name mailhog -p 1025:1025 -p 8025:8025 mailhog/mailhog
    6  sudo docker run -d --name mailhog -p 1025:1025 -p 8025:8025 mailhog/mailhog
    7  docker ps -a
    8  sudo apt install tcpdump
    9  sudo tcpdump -i any tcp port 1025 -w smtp.pcap &
   10  swaks --server 127.0.0.1 --port 1025   --from ti@example.com --to test@example.com   --data "Subject: Codespace SWAKS Test\n\nPozdrav iz FIS!"
   11  sudo apt install swaks
   12  swaks --server 127.0.0.1 --port 1025   --from ti@example.com --to test@example.com   --data "Subject: Codespace SWAKS Test\n\nPozdrav iz FIS!"
   13  history
   14  sudo pkill tcpdump
   15  docker run -d --name postfix587   -p 587:587   -e ALLOWED_SENDER_DOMAINS="localdomain"   -e POSTFIX_myhostname=postfix.local   boky/postfix
   16  docker ps -a
   17  sudo tcpdump -i any tcp port 587 -w smtpssl.pcap &
   18  swaks --server 127.0.0.1 --port 587 --tls   --from test@localdomain --to demc@localdomain   --header 'Subject: STARTTLS test'   --body 'Pozdrav prek TLS!'
   19  sudo pkill tcpdumpž
   20  history
@ZanS-03 ➜/workspaces/desktop-tutorial (main) $
```





Refleksija:

Pri nešifriranem lahko napadalec vidi celotno vsebino eposte, med tem ko pri sifriranem tega ne more (kot je vidno v praksi zgoraj)

S preverjanjem vemo, da je javni ključ res pravilen in pripada pravi osebi.

PGP bi uporabili za eposto, signal pa za pogovor v realtime

Menim da bi moralo biti privzeto, da ne pride do napadov, vohunjena sporocil, ipd.