Solutions: Sheet 10

1. Packet Filters

(a) Which is the highest layer of the hybrid reference model on which a stateless packet filter firewall works? Which data are evaluated here?

- The highest layer where stateless packet filter firewall works is Layer 4 as it works on **Layer 3 (Network Layer)** and **Layer 4 (Network Layer)**
- **Data evaluted:** All the header data
  - packet size
  - sender/receiver IP address
  - sender/receiver port (service)
  - protocol ID, ICMP type and code

(b) Describe briefly how stateless and stateful packet filters work. What are the advantages and disadvantages of the two types of firewalls compared to each other?

Stateless packet filters:

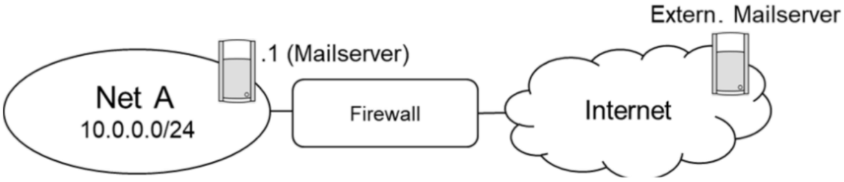
- Describes filter rules over an access list. It separates the inbound and outbound lists. Filter rules are applied to the headers of each IP packet without considering relationships between packets. Processing of the rules according to the first-match principle.
- **Advantages:**
  - simple configuration
  - higher speed than other approaches (e.g proxys)
  - no changes on client side
- **Disadvantages:**
  - problems with udp
  - only limited traffic control
  - layer 3 and 4 data only filtered
  - tunneling of data via permitted port (e.g P2P via port 80)

Stateful packet filters:

- Packets are filtered (drop or passed) based on rules which analyse the packets header ???
- Fixed static rules and the previously observed packet traffic of the connection. The states of the connections are stored in the state table.
- **Advantages:**
  - easy configuration
  - no changes on client side
- **Disadvantages:**
  - Possible attacks on state table (DOS)
  - layer 3 and 4 data only filtered
  - only very few ports ???

2. Linux Firewall netfilter (`iptables`)

(a) An organization operates an email server with the IP address `10.0.0.1` in its internal network A (`10.0.0.0/24`). The internal network is connected to the Internet through a firewall that works as a state-based packet filter.



Please create the necessary rules for the firewall in the following table in order to only allow SMTP/email communication (SMTP port: `25`):

- Allow access to an external email server on the Internet.
- Allow access from the Internet to the internal email server.

note

`>1023` means any port above `1023` as the first `1023` are predefined standard ports

note

- row 1 and 2 from device in network A to external mail server in internet
- row 3 and 4 from external device in internet to mail server of network A

Direction	Source IP	Destination IP	Protocol	Source Port	Destination Port	State	Action
Out	`10.0.0.0/24`	External	TCP	`>1023`	`25`	New	PERMIT
In	External	`10.0.0.0/24`	TCP	`25`	`>1023`	Established (ACK)	PERMIT
In	External	`10.0.0.1`	TCP	`>1023`	`25`	New	PERMIT
Out	`10.0.0.1`	External	TCP	`25`	`>1023`	Established (ACK)	PERMIT
Either	ANY	ANY	TCP	ANY	ANY	Established (ACK)	PERMIT
Either	ANY	ANY	ANY	ANY	ANY	ANY	DENY

- Rule of row 5 combines row 2 and 4

(b) Netfilter is the Linux kernel firewall. iptables can be used to configure the tables and the contained rules chains.

i. Briefly describe the tasks of the three tables filter, nat and mangle.

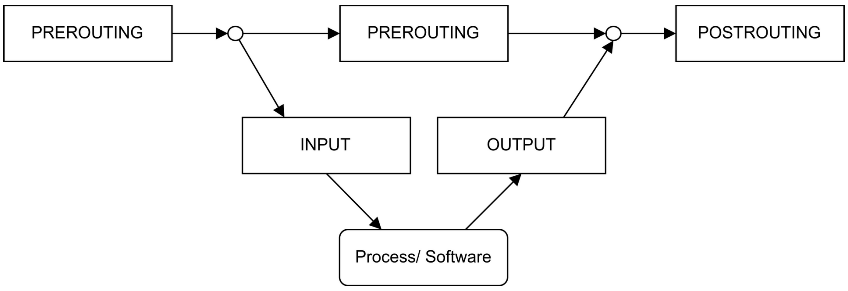
https://www.thegeekstuff.com/2011/01/iptables-fundamentals/

- **filter:** contains filter rules for filtering packet (drop or accept)
- **nat:** used to translate IP addresses and ports (network address translation). ???
- **mangle:** Used for packet manipulation.

ii. Briefly describe the standard chains INPUT, OUTPUT, FORWARD, PREROUTING and POSTROUTING and their tasks.

note

Packets coming in either handled by `INPUT` or `FORWARD`



- `INPUT`:The packet is delivered locally, i.e. the firewall is the destination.
- `OUTPUT`: The packet is created by the firewall
- `FORWARD`: The packet is routed (and not delivered locally)
- `PREROUTING`: The modification (NAT) of packets before a routing decision is made.
- `POSTROUTING`: The modification (NAT) of packets after the routing decision

iii. Briefly describe the rules ACCEPT, REJECT, DROP, LOG, REDIRECT and MASQUERADE.

> **note**
>
> two ways to drop files `REJECT` and `DROP`

> **note**
>
> `LOG` used to write information in system logs about accepting or dropping a packet

> **note**
>
> `REDIRECT` and `MASQUERADE` for Network Address Translation (NAT)

- `ACCEPT`: the packet can pass
- `REJECT`: the packet is rejected and an error message is sent
- `DROP`: the packet is ignored and no response is sent
- `LOG`: writes an entry in the syslog
- `REDIRECT`: the destination address of the packet is changed such that it is sent to the local computer
- `MASQUERADE`: the source address of the packet is replaced by the IP address of the interface on which it leaves the computer

iv. What is the principle behind the sequence in which the rules of a chain are processed?

frist match principle - The rules are checked one by one, and if one of them is true, the processing of the corresponding chain is terminated. $rightarrow$ why rule order is fixed

3. Transport Layer Security (TLS)

(a) What is end-to-end (E2E) security?

- Continously security between two endpoints (sender and receipient)
- General seucity of sender and receipient, e.g secure transmission of data → confidentiality when sending from sender to receiver

(b) In which layer of the hybrid reference model can end-to-end (E2E) security be implemented between applications? In which layer between devices (IT systems)?

- **Between Applications:** Application Layer (Layer 5) (security within the application)
- **Between Devices:** Transport Layer (secutity via TLS) or Network Layer (seucity via IPSec)

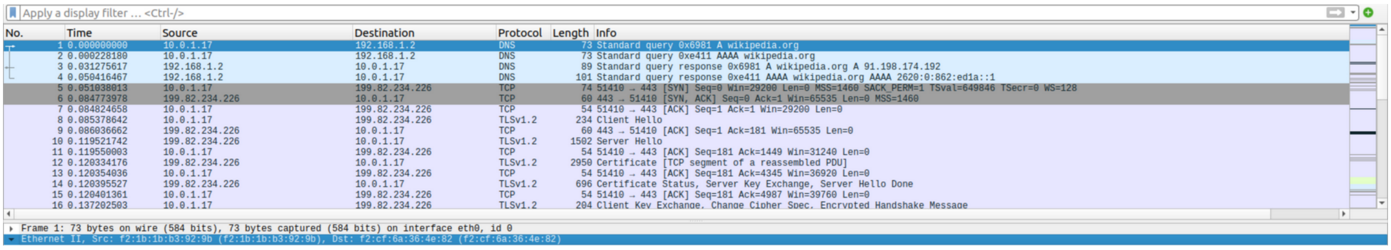
(c) Name the layer of the hybrid reference model to which TLS provides its services. Also name two protocol that can be secured by TLS.

- **Layer:** implemented in Transport Layer and provides service to Application Layer
- **Protocols:** `HTTP`, `SMTP`, `FTP`

(d) You can find the file `capture.pcapng` in Moodle which contains network traffic data between a client and a server. Analyze this traffic data using Wireshark and answer the following questions:

i. Determine the IP address and the MAC address of the host on which the recording was created. In the following this host is considered as the client.

- **IP address:** `10.0.1.17`
- **MAC address:** `f2:1b:1b:b3:92:9b`

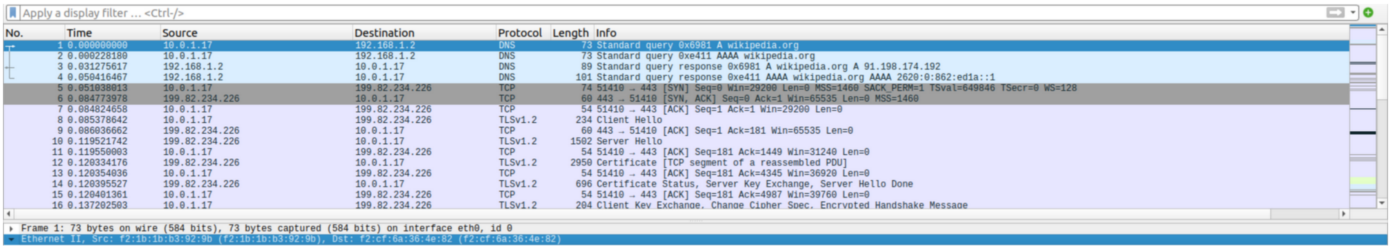


```
No.  Time         Source          Destination     Protocol  Length Info
   1 0.000000000  10.0.1.17       192.168.1.2     DNS         73 Standard query 0x6981 A wikipedia.org
   2 0.000228180  10.0.1.17       192.168.1.2     DNS         73 Standard query 0xe411 AAAA wikipedia.org
   3 0.031275617  192.168.1.2     10.0.1.17       DNS         89 Standard query response 0x6981 A wikipedia.org A 91.198.174.192
   4 0.050416467  192.168.1.2     10.0.1.17       DNS        101 Standard query response 0xe411 AAAA wikipedia.org AAAA 2620:0:862:ed1a::1
   5 0.051038013  10.0.1.17       199.82.234.226  TCP         74 51410 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=649846 TSecr=0 WS=128
   6 0.084773978  199.82.234.226  10.0.1.17       TCP         60 443 → 51410 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
   7 0.084824658  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=1 Ack=1 Win=29200 Len=0
   8 0.085378642  10.0.1.17       199.82.234.226  TLSv1.2    234 Client Hello
   9 0.086036662  199.82.234.226  10.0.1.17       TCP         60 443 → 51410 [ACK] Seq=1 Ack=181 Win=65535 Len=0
  10 0.119521742  199.82.234.226  10.0.1.17       TLSv1.2   1502 Server Hello
  11 0.119550003  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=181 Ack=1449 Win=31240 Len=0
  12 0.120334176  199.82.234.226  10.0.1.17       TLSv1.2   2950 Certificate [TCP segment of a reassembled PDU]
  13 0.120354036  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=181 Ack=4345 Win=36920 Len=0
  14 0.120395527  199.82.234.226  10.0.1.17       TLSv1.2    696 Certificate Status, Server Key Exchange, Server Hello Done
  15 0.120401361  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=181 Ack=4987 Win=39760 Len=0
  16 0.137202503  10.0.1.17       199.82.234.226  TLSv1.2    204 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

▸ Frame 1: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface eth0, id 0
▾ Ethernet II, Src: f2:1b:1b:b3:92:9b (f2:1b:1b:b3:92:9b), Dst: f2:cf:6a:36:4e:82 (f2:cf:6a:36:4e:82)
```

ii. Enter the IP address of the DNS server(s) that performed the name resolution for this connection.

- **IP address:** `192.168.1.2`

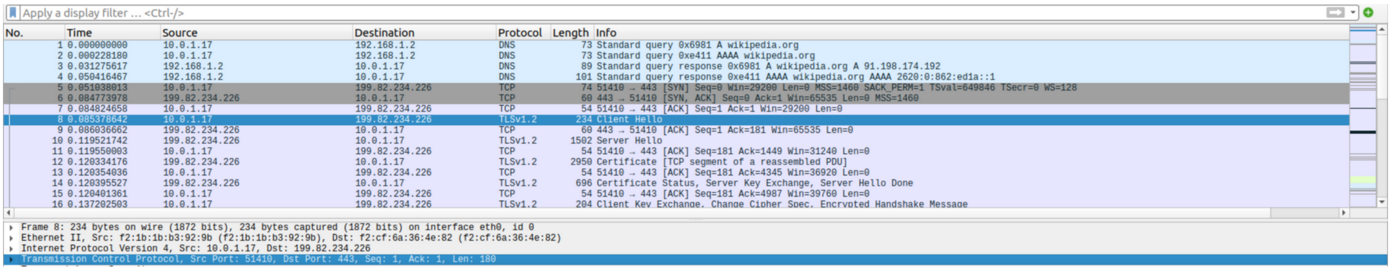


```
No.  Time         Source          Destination     Protocol  Length Info
   1 0.000000000  10.0.1.17       192.168.1.2     DNS         73 Standard query 0x6981 A wikipedia.org
   2 0.000228180  10.0.1.17       192.168.1.2     DNS         73 Standard query 0xe411 AAAA wikipedia.org
   3 0.031275617  192.168.1.2     10.0.1.17       DNS         89 Standard query response 0x6981 A wikipedia.org A 91.198.174.192
   4 0.050416467  192.168.1.2     10.0.1.17       DNS        101 Standard query response 0xe411 AAAA wikipedia.org AAAA 2620:0:862:ed1a::1
   5 0.051038013  10.0.1.17       199.82.234.226  TCP         74 51410 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=649846 TSecr=0 WS=128
   6 0.084773978  199.82.234.226  10.0.1.17       TCP         60 443 → 51410 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
   7 0.084824658  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=1 Ack=1 Win=29200 Len=0
   8 0.085378642  10.0.1.17       199.82.234.226  TLSv1.2    234 Client Hello
   9 0.086036662  199.82.234.226  10.0.1.17       TCP         60 443 → 51410 [ACK] Seq=1 Ack=181 Win=65535 Len=0
  10 0.119521742  199.82.234.226  10.0.1.17       TLSv1.2   1502 Server Hello
  11 0.119550003  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=181 Ack=1449 Win=31240 Len=0
  12 0.120334176  199.82.234.226  10.0.1.17       TLSv1.2   2950 Certificate [TCP segment of a reassembled PDU]
  13 0.120354036  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=181 Ack=4345 Win=36920 Len=0
  14 0.120395527  199.82.234.226  10.0.1.17       TLSv1.2    696 Certificate Status, Server Key Exchange, Server Hello Done
  15 0.120401361  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=181 Ack=4987 Win=39760 Len=0
  16 0.137202503  10.0.1.17       199.82.234.226  TLSv1.2    204 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

▸ Frame 1: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface eth0, id 0
▾ Ethernet II, Src: f2:1b:1b:b3:92:9b (f2:1b:1b:b3:92:9b), Dst: f2:cf:6a:36:4e:82 (f2:cf:6a:36:4e:82)
```

iii. Determine the IP address of the web server with which the client establishes the first TLS connection. Also determine the client and server-side ports of this TLS connection.

- **IP address:** `199.82.234.226`
- **Client port:** `51410`
- **Server-Side port:** `443` (http port)

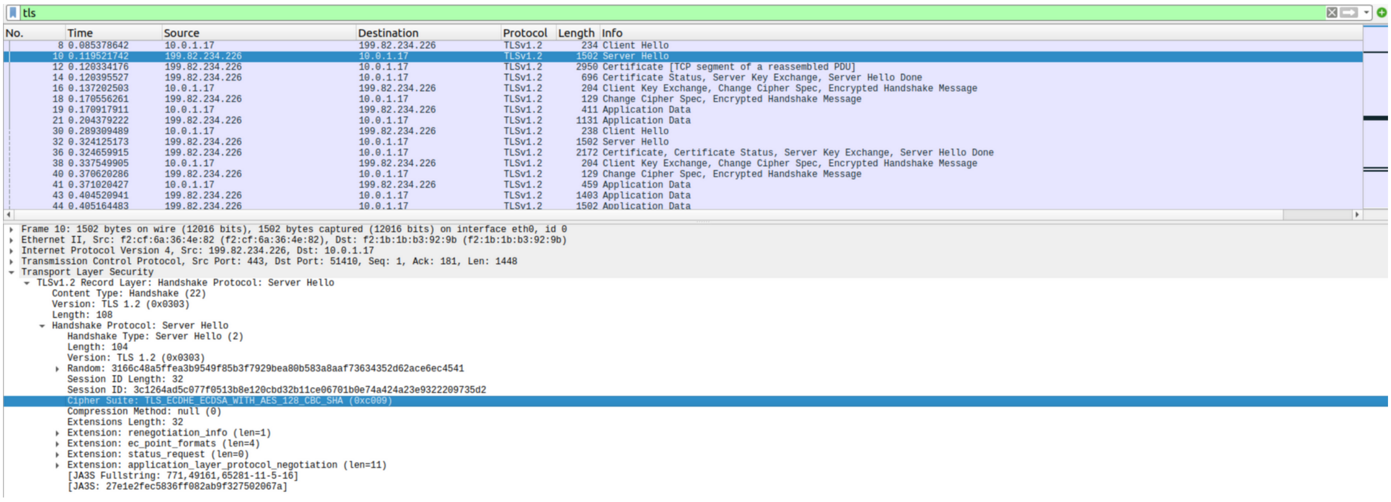


```
No.  Time         Source          Destination     Protocol  Length Info
   1 0.000000000  10.0.1.17       192.168.1.2     DNS         73 Standard query 0x6981 A wikipedia.org
   2 0.000228180  10.0.1.17       192.168.1.2     DNS         73 Standard query 0xe411 AAAA wikipedia.org
   3 0.031275617  192.168.1.2     10.0.1.17       DNS         89 Standard query response 0x6981 A wikipedia.org A 91.198.174.192
   4 0.050416467  192.168.1.2     10.0.1.17       DNS        101 Standard query response 0xe411 AAAA wikipedia.org AAAA 2620:0:862:ed1a::1
   5 0.051038013  10.0.1.17       199.82.234.226  TCP         74 51410 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=649846 TSecr=0 WS=128
   6 0.084773978  199.82.234.226  10.0.1.17       TCP         60 443 → 51410 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
   7 0.084824658  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=1 Ack=1 Win=29200 Len=0
   8 0.085378642  10.0.1.17       199.82.234.226  TLSv1.2    234 Client Hello
   9 0.086036662  199.82.234.226  10.0.1.17       TCP         60 443 → 51410 [ACK] Seq=1 Ack=181 Win=65535 Len=0
  10 0.119521742  199.82.234.226  10.0.1.17       TLSv1.2   1502 Server Hello
  11 0.119550003  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=181 Ack=1449 Win=31240 Len=0
  12 0.120334176  199.82.234.226  10.0.1.17       TLSv1.2   2950 Certificate [TCP segment of a reassembled PDU]
  13 0.120354036  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=181 Ack=4345 Win=36920 Len=0
  14 0.120395527  199.82.234.226  10.0.1.17       TLSv1.2    696 Certificate Status, Server Key Exchange, Server Hello Done
  15 0.120401361  10.0.1.17       199.82.234.226  TCP         54 51410 → 443 [ACK] Seq=181 Ack=4987 Win=39760 Len=0
  16 0.137202503  10.0.1.17       199.82.234.226  TLSv1.2    204 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

▸ Frame 8: 234 bytes on wire (1872 bits), 234 bytes captured (1872 bits) on interface eth0, id 0
▸ Ethernet II, Src: f2:1b:1b:b3:92:9b (f2:1b:1b:b3:92:9b), Dst: f2:cf:6a:36:4e:82 (f2:cf:6a:36:4e:82)
▸ Internet Protocol Version 4, Src: 10.0.1.17, Dst: 199.82.234.226
▸ Transmission Control Protocol, Src Port: 51410, Dst Port: 443, Seq: 1, Ack: 1, Len: 180
```

iv. Determine the cipher suite that is used for the connection and evaluate its security.

- **Cipher Suite:** `TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA` (prefered cipher suited used by server, but client offers more)
- **Evalutation:** https://ciphersuite.info/cs/TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA/
  - **Protocol:** Transport Layer Security (TLS)
  - **Key Exchange:** Elliptic Curve Diffie-Hellman Ephemeral (ECDHE)
  - **Authentication:** Elliptic Curve Digital Signature Algorithm (ECDSA)
  - **Encryption:** Advanced Encryption Standard with 128 bit key in Cipher Block Chaining mode (AES 128 CBC) → timing attack against several TLS implementations using the CBC possible
  - **Hash:** SHA 1 → proven insecure in 2017



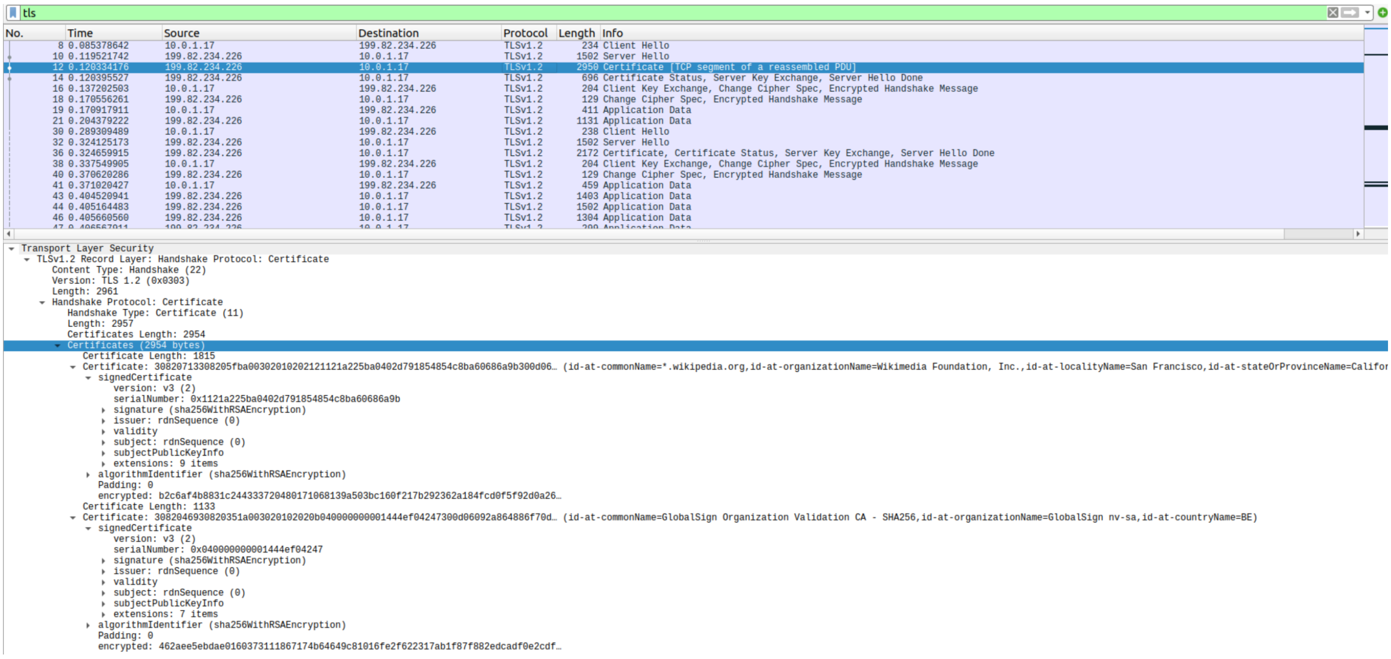
```
tls
No.     Time         Source           Destination      Protocol  Length Info
      8 0.085378642  10.0.1.17        199.82.234.226   TLSv1.2    234 Client Hello
     10 0.119521742  199.82.234.226   10.0.1.17        TLSv1.2   1502 Server Hello
     12 0.120334176  199.82.234.226   10.0.1.17        TLSv1.2   2950 Certificate [TCP segment of a reassembled PDU]
     14 0.120395527  199.82.234.226   10.0.1.17        TLSv1.2    696 Certificate Status, Server Key Exchange, Server Hello Done
     16 0.137202503  10.0.1.17        199.82.234.226   TLSv1.2    204 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
     18 0.170556261  199.82.234.226   10.0.1.17        TLSv1.2    129 Change Cipher Spec, Encrypted Handshake Message
     19 0.170917911  10.0.1.17        199.82.234.226   TLSv1.2    411 Application Data
     21 0.204379222  199.82.234.226   10.0.1.17        TLSv1.2   1131 Application Data
     30 0.289309489  10.0.1.17        199.82.234.226   TLSv1.2    238 Client Hello
     32 0.324125173  199.82.234.226   10.0.1.17        TLSv1.2   1502 Server Hello
     36 0.324659915  199.82.234.226   10.0.1.17        TLSv1.2   2172 Certificate, Certificate Status, Server Key Exchange, Server Hello Done
     38 0.337549905  10.0.1.17        199.82.234.226   TLSv1.2    204 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
     40 0.370620286  199.82.234.226   10.0.1.17        TLSv1.2    129 Change Cipher Spec, Encrypted Handshake Message
     41 0.371020427  10.0.1.17        199.82.234.226   TLSv1.2    459 Application Data
     43 0.404520941  199.82.234.226   10.0.1.17        TLSv1.2   1403 Application Data
     44 0.405164483  199.82.234.226   10.0.1.17        TLSv1.2   1502 Application Data
```

```
▸ Frame 10: 1502 bytes on wire (12016 bits), 1502 bytes captured (12016 bits) on interface eth0, id 0
▸ Ethernet II, Src: f2:cf:6a:36:4e:82 (f2:cf:6a:36:4e:82), Dst: f2:1b:1b:b3:92:9b (f2:1b:1b:b3:92:9b)
▸ Internet Protocol Version 4, Src: 199.82.234.226, Dst: 10.0.1.17
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 51410, Seq: 1, Ack: 181, Len: 1448
▾ Transport Layer Security
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 108
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 104
        Version: TLS 1.2 (0x0303)
      ▸ Random: 3166c48a5ffea3b9549f85b3f7929bea80b583a8aaf73634352d62ace6ec4541
        Session ID Length: 32
        Session ID: 3c1264ad5c077f0513b8e120cbd32b11ce06701b0e74a424a23e9322209735d2
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
        Compression Method: null (0)
        Extensions Length: 32
      ▸ Extension: renegotiation_info (len=1)
      ▸ Extension: ec_point_formats (len=4)
      ▸ Extension: status_request (len=0)
      ▸ Extension: application_layer_protocol_negotiation (len=11)
        [JA3S Fullstring: 771,49161,65281-11-5-16]
        [JA3S: 27e1e2fec5836ff082ab9f327502067a]
```

v. For the connection, also specify which communication partner(s) is/are authenticated.

https://stackoverflow.com/questions/25085100/can-you-check-monitor-the-client-certificates-sent-in-requests-using-wireshark#25130004

https://datatracker.ietf.org/doc/html/rfc8446#page-11



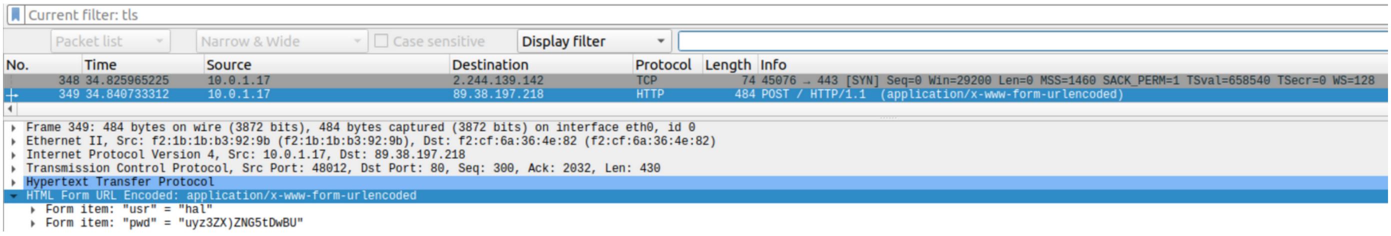
```
tls
No.     Time         Source           Destination      Protocol  Length Info
      8 0.085378642  10.0.1.17        199.82.234.226   TLSv1.2    234 Client Hello
     10 0.119521742  199.82.234.226   10.0.1.17        TLSv1.2   1502 Server Hello
     12 0.120334176  199.82.234.226   10.0.1.17        TLSv1.2   2950 Certificate [TCP segment of a reassembled PDU]
     14 0.120395527  199.82.234.226   10.0.1.17        TLSv1.2    696 Certificate Status, Server Key Exchange, Server Hello Done
     16 0.137202503  10.0.1.17        199.82.234.226   TLSv1.2    204 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
     18 0.170556261  199.82.234.226   10.0.1.17        TLSv1.2    129 Change Cipher Spec, Encrypted Handshake Message
     19 0.170917911  10.0.1.17        199.82.234.226   TLSv1.2    411 Application Data
     21 0.204379222  199.82.234.226   10.0.1.17        TLSv1.2   1131 Application Data
     30 0.289309489  10.0.1.17        199.82.234.226   TLSv1.2    238 Client Hello
     32 0.324125173  199.82.234.226   10.0.1.17        TLSv1.2   1502 Server Hello
     36 0.324659915  199.82.234.226   10.0.1.17        TLSv1.2   2172 Certificate, Certificate Status, Server Key Exchange, Server Hello Done
     38 0.337549905  10.0.1.17        199.82.234.226   TLSv1.2    204 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
     40 0.370620286  199.82.234.226   10.0.1.17        TLSv1.2    129 Change Cipher Spec, Encrypted Handshake Message
     41 0.371020427  10.0.1.17        199.82.234.226   TLSv1.2    459 Application Data
     43 0.404520941  199.82.234.226   10.0.1.17        TLSv1.2   1403 Application Data
     44 0.405164483  199.82.234.226   10.0.1.17        TLSv1.2   1502 Application Data
     46 0.405660560  199.82.234.226   10.0.1.17        TLSv1.2   1304 Application Data
     47 0.406567911  199.82.234.226   10.0.1.17        TLSv1.2    299 Application Data
```

```
▾ Transport Layer Security
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 2961
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 2957
        Certificates Length: 2954
      ▾ Certificates (2954 bytes)
          Certificate Length: 1815
        ▾ Certificate: 30820713308205fba00302010202121121a225ba0402d791854854c8ba60686a9b300d06… (id-at-commonName=*.wikipedia.org,id-at-organizationName=Wikimedia Foundation, Inc.,id-at-localityName=San Francisco,id-at-stateOrProvinceName=Califor
          ▾ signedCertificate
              version: v3 (2)
              serialNumber: 0x1121a225ba0402d791854854c8ba60686a9b
            ▸ signature (sha256WithRSAEncryption)
            ▸ issuer: rdnSequence (0)
            ▸ validity
            ▸ subject: rdnSequence (0)
            ▸ subjectPublicKeyInfo
            ▸ extensions: 9 items
          ▸ algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted: b2c6af4b8831c244333720480171068139a503bc160f217b292362a184fcd0f5f92d0a26…
          Certificate Length: 1133
        ▾ Certificate: 308204693082035 1a003020102020b040000000001444ef04247300d06092a864886f70d… (id-at-commonName=GlobalSign Organization Validation CA - SHA256,id-at-organizationName=GlobalSign nv-sa,id-at-countryName=BE)
          ▾ signedCertificate
              version: v3 (2)
              serialNumber: 0x040000000001444ef04247
            ▸ signature (sha256WithRSAEncryption)
            ▸ issuer: rdnSequence (0)
            ▸ validity
            ▸ subject: rdnSequence (0)
            ▸ subjectPublicKeyInfo
            ▸ extensions: 7 items
          ▸ algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted: 462aee5ebdae0160373111867174b64649c81016fe2f622317ab1f87f882edcadf0e2cdf…
```

vi. There is an HTTP connection to the IP `89.38.197.218`. A user name and a password were transmitted unencrypted. Find them out and write them down.

- **username:** `hal`

- **password:** `uyz3ZX)ZNG5tDwBU`



Current filter: tls

```
No.     Time            Source          Destination       Protocol  Length  Info
    348 34.825965225    10.0.1.17       2.244.139.142     TCP           74 45076 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=658540 TSecr=0 WS=128
    349 34.840733312    10.0.1.17       89.38.197.218     HTTP         484 POST / HTTP/1.1  (application/x-www-form-urlencoded)
```

```
▸ Frame 349: 484 bytes on wire (3872 bits), 484 bytes captured (3872 bits) on interface eth0, id 0
▸ Ethernet II, Src: f2:1b:1b:b3:92:9b (f2:1b:1b:b3:92:9b), Dst: f2:cf:6a:36:4e:82 (f2:cf:6a:36:4e:82)
▸ Internet Protocol Version 4, Src: 10.0.1.17, Dst: 89.38.197.218
▸ Transmission Control Protocol, Src Port: 48012, Dst Port: 80, Seq: 300, Ack: 2032, Len: 430
▸ Hypertext Transfer Protocol
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
   ▸ Form item: "usr" = "hal"
   ▸ Form item: "pwd" = "uyz3ZX)ZNG5tDwBU"
```