SHARKFEST '13

Wireshark Developer and User Conference

SSL Troubleshooting with Wireshark

Mak Kolybabi
Tenable Network Security

Why am I here?

Substitute for Sake Blok

Today’s material is based on his from 2012.

Who am I?

- Nessus (Engine, SSL, SSH)
- Nmap (SSL, Stuxnet, ProFTPD, Dropbox)
- SkullSpace Hackerspace
- BSides Winnipeg
- DangerZone Learning CTF

Who am I not?

- Cryptographer
- Server Admin
- Network Admin

How’s this going to work?

- If you have a question, just interrupt me by saying my name: Mak.
  - If a question requires a lengthy answer, I’ll let you know and answer it later
- Questions are awesome, please ask questions.
- Seriously.

Have you ever…

- …created a certificate?
- …created a CA?
- …used Wireshark?
- …used the SSL dissector?
- …decrypted SSL?
- …written an SSL implementation?

What does SSL provide?

- Confidentiality
  - Encryption & Decryption
- Integrity
  - MACs and HMACs
- Authenticity & Non-repudiation
  - X.509 and CAs

What could possibly go wrong?

- SSL is composed of two **very** complicated pieces:
  1. Cryptography
  2. Certificates
- Either can be the source of countless problems

What's the plan?

1. Cryptography, Certificates, and SSL Basics
   - LAB: Looking at certificates in Wireshark and OpenSSL
2. SSL Handshakes
   - LAB: Looking at SSL handshakes in Wireshark
3. SSL Decryption
   - LAB: Decrypting SSL sessions in Wireshark
4. Common Problems
   - LAB: Decrypting (more) SSL sessions in Wireshark
5. Discussions

Cryptography & Certificates

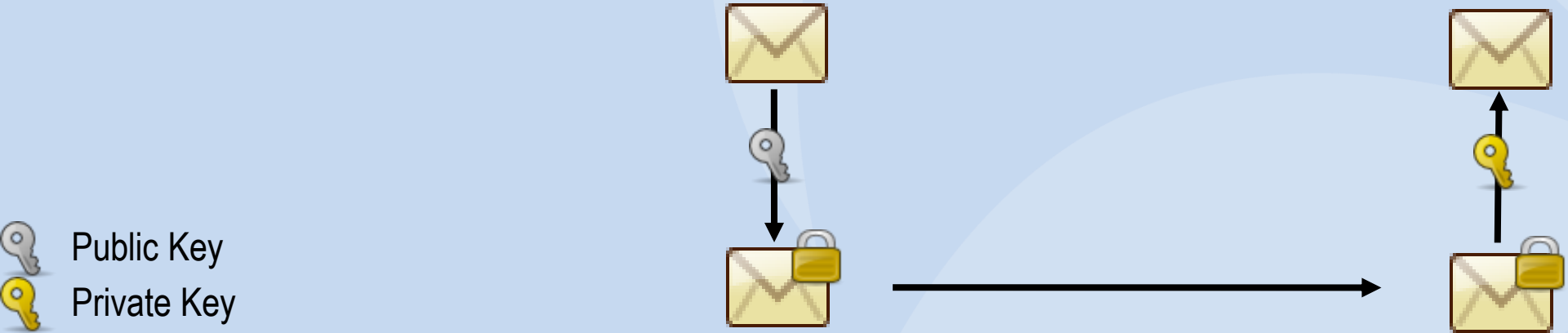
Symmetric Encryption

- One key for everything
- Short keys
- Fast
- DES, 3DES, AES, RC4

Private Key

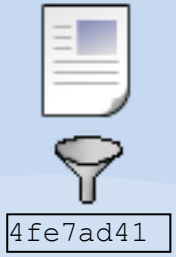
Asymmetric Encryption

- Key pairs
- Long keys
- Slow
- RSA, DSA, ECDSA



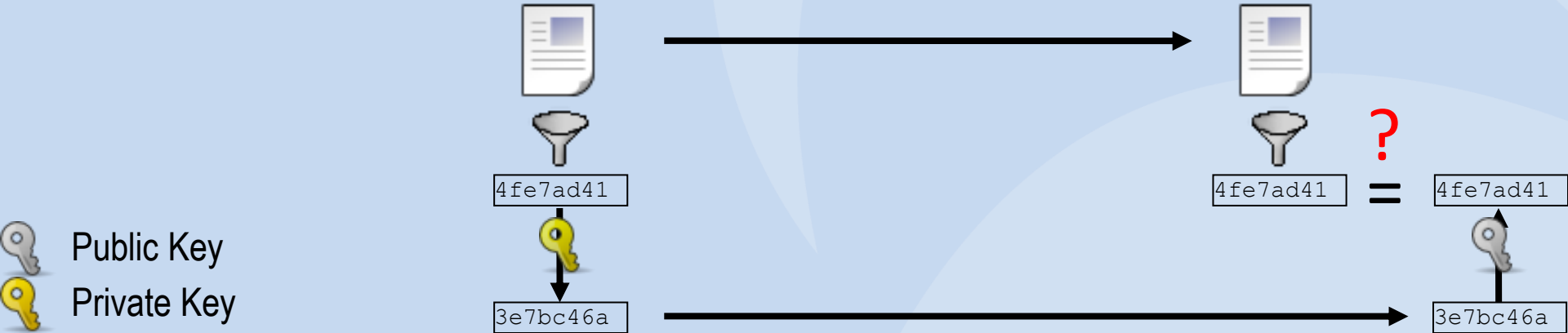
Hashes or Message Digests

- Fingerprint
- Irreversible
  - Can’t get the original input from the output
- Preimage resistant
  - Can’t find any input that matches a given output
- Collision resistant
  - Can’t find two inputs that have the same output
- MD5, SHA-1, SHA-2, SHA-3



Message Signing

- Hash message
- Encrypt hash private key, creating a signature
- Send the message and the signature
- Decrypt signature with public key, retrieving hash
- Hash message and compared to decrypted hash



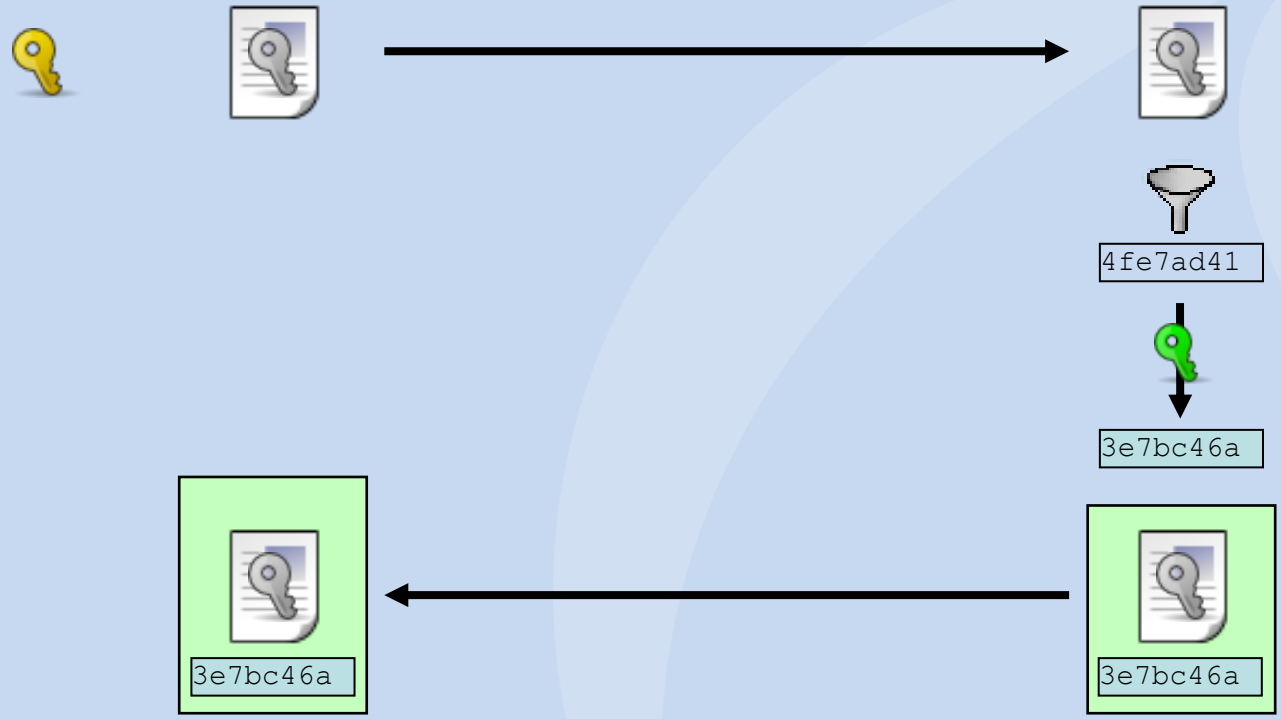
Certificates (1)

- Contain:
  - Public key
  - Identity information
  - Issuer information
  - Restrictions
- Associates public keys with:
  - Servers and services
  - People and roles

Certificate Authorities

- Trusted by the SSL client
- CAs can be chained
  - Middle of chain are Intermediate CAs
  - Top of chain is self-signed Root CA
- Nobody really likes the CA system
  - Increasing government surveillance
  - Too many trusted parties
  - Many recent debacles

Certificate Creation



Public Key

Private Key

CA Private Key

Recommended Reading

- Wikipedia
- RFCs
- A Few Thoughts on Cryptographic Engineering
- Twitter

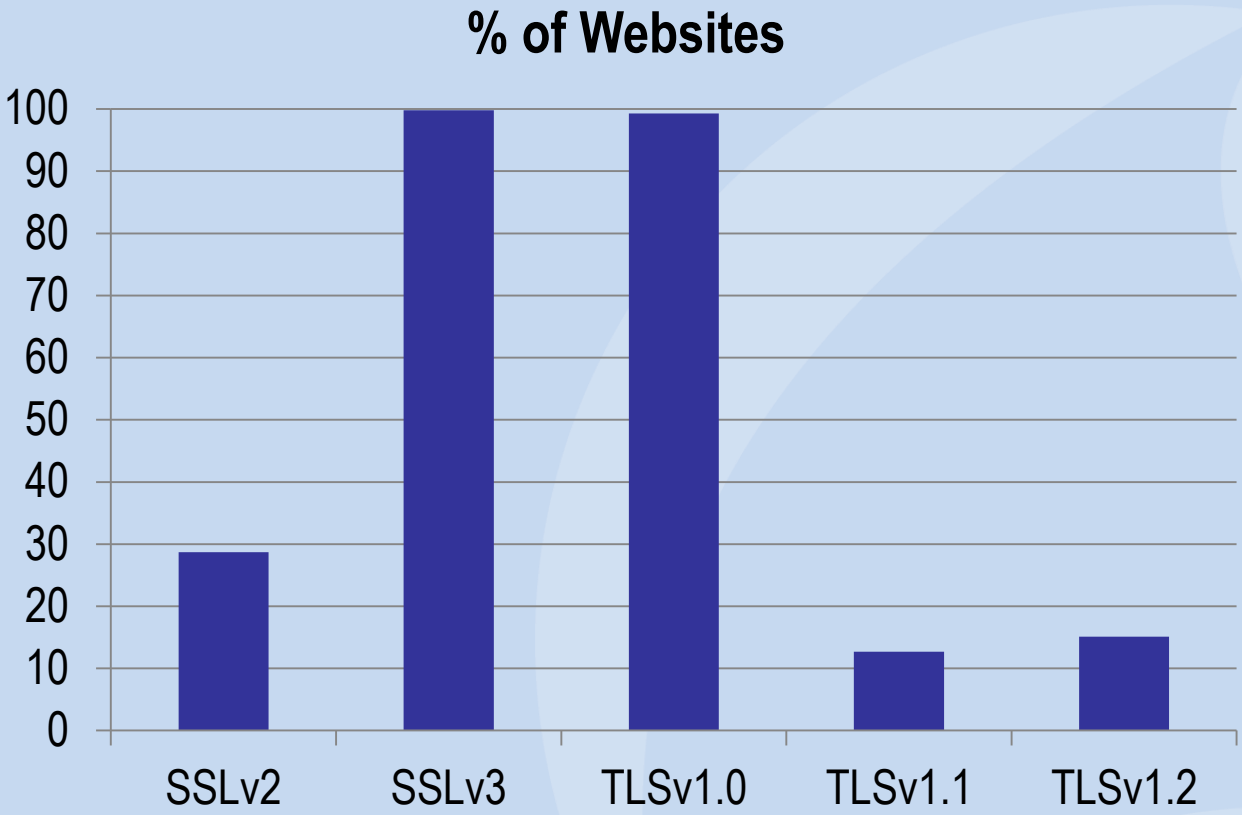
SSL Basics

SSL Timeline

1. 1994: SSLv1, Netscape
2. 1994: SSLv2, Netscape
3. 1995: SSLv3, Netscape
4. 1999: TLSv1.0, IETF
5. 2006: TLSv1.1, IETF
6. 2008: TLSv1.2, IETF

Differences explained at http://tinyurl.com/ssl-vers

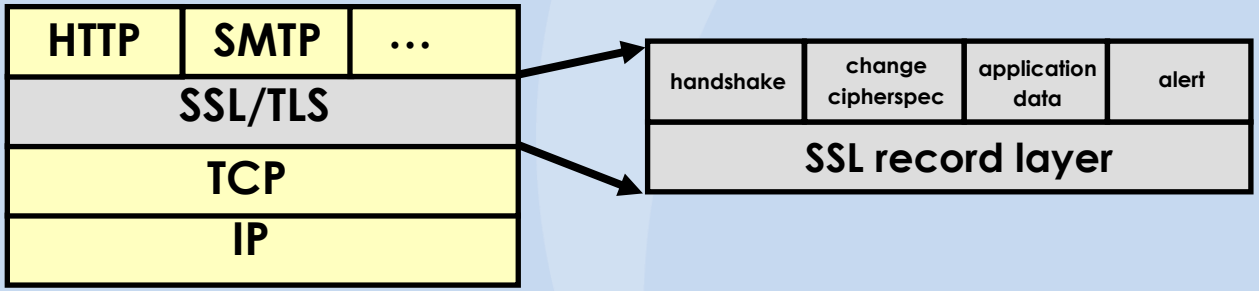
SSL Support



https://www.trustworthyinternet.org/ssl-pulse/

Place in protocol stack

- TCP/IP Stack: Above TCP
- OSI Stack: Presentation layer
- Protocol independent, encapsulate anything



SSL Layering

- TCP Packet Layer
  - Provides a continuous, reliable stream of data
  - Packets may contain multiple records
- SSL Record Layer
  - Provides fragmentation
  - Records may contain multiple messages
- SSL Message Layer
  - Provides control and data messages for the protocol

SSL Record Content Types

- Handshake Protocol
  - Authentication and key exchange
- ChangeCipherSpec Protocol
  - Starts encryption
- Alert Protocol
  - Reports warnings and errors
- Application Protocol
  - Encrypted data

LAB: Setup

Lab Material

1. basic.pcap
2. basic.pem
3. tricky.pcap
4. tricky.pem

All of these files are available at http://mogigoma.com/

Choosing the right settings (1)

- In protocol preferences, enable:
  - IPv4
    - Reassemble fragmented IPv4 datagrams
  - TCP
    - Allow subdissector to reassemble TCP streams
  - SSL
    - Reassemble SSL records spanning multiple TCP segments
    - Reassemble SSL Application Data spanning multiple SSL records

Choosing the right settings (2)

- In protocol preferences, disable:
  - TCP
    - Validate the TCP checksum if possible

LAB: Server Certificates

Filtering

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.3.1	192.168.3.3	TCP	66	18736 > https [SYN] Seq=0 Win=
2	0.000309	192.168.3.3	192.168.3.1	TCP	66	https > 18736 [SYN, ACK] Seq=0
3	0.000357	192.168.3.1	192.168.3.3	TCP	54	18736 > https [ACK] Seq=1 Ack=
4	0.011511	192.168.3.1	192.168.3.3	TLSv1	124	Client Hello
5	0.011876	192.168.3.3	192.168.3.1	TCP	54	https > 18736 [ACK] Seq=1 Ack=
6	0.017431	192.168.3.3	192.168.3.1	TLSv1	1514	Server Hello

Filter: ssl Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
4	0.011511	192.168.3.1	192.168.3.3	TLSv1	124	Client Hello
6	0.017431	192.168.3.3	192.168.3.1	TLSv1	1514	Server Hello
7	0.017782	192.168.3.3	192.168.3.1	TLSv1	1019	Certificate
9	0.026711	192.168.3.1	192.168.3.3	TLSv1	252	Client Key Exchange, Cha
10	0.038327	192.168.3.3	192.168.3.1	TLSv1	113	Change Cipher Spec, Encr
11	0.040173	192.168.3.1	192.168.3.3	TLSv1	491	Application Data

Record Fragmentation

```
⊞ Frame 7: 1019 bytes on wire (8152 bits), 1019 bytes captured (8152
⊞ Ethernet II, Src: Vmware_5d:c5:66 (00:0c:29:5d:c5:66), Dst: Vmware
⊞ Internet Protocol Version 4, Src: 192.168.3.3 (192.168.3.3), Dst:
⊞ Transmission Control Protocol, Src Port: https (443), Dst Port: 18
⊞ [2 Reassembled TCP Segments (2337 bytes): #6(1381), #7(956)]
⊟ Secure Sockets Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Certificate
⊟ Secure Sockets Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Server Hello Done
```

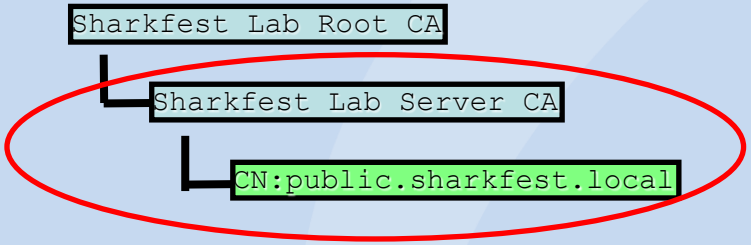
Certificate Message

```
⊟ Secure Sockets Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 2332
    ⊟ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 2328
        Certificates Length: 2325
      ⊟ Certificates (2325 bytes)
          Certificate Length: 1079
        ⊞ Certificate (pkcs-9-at-emailAddress=co@sharkfest.local,id-at-c
          Certificate Length: 1240
        ⊞ Certificate (pkcs-9-at-emailAddress=so@sharkfest.local,id-at-c
```

Subject and Issuer

```
issuer: rdnSequence (0)
  rdnSequence: 5 items (pkcs-9-at-emailAddress=so@sharkfest.local,id-at-c
    RDNSequence item: 1 item (id-at-countryName=NL)
    RDNSequence item: 1 item (id-at-stateOrProvinceName=Noord-Holland)
    RDNSequence item: 1 item (id-at-organizationName=Sharkfest Lab)
    RDNSequence item: 1 item (id-at-commonName=Sharkfest Lab Server CA)
    RDNSequence item: 1 item (pkcs-9-at-emailAddress=so@sharkfest.local)
validity
subject: rdnSequence (0)
  rdnSequence: 5 items (pkcs-9-at-emailAddress=co@sharkfest.local,id-at-c
    RDNSequence item: 1 item (id-at-countryName=NL)
    RDNSequence item: 1 item (id-at-stateOrProvinceName=Noord-Holland)
    RDNSequence item: 1 item (id-at-organizationName=Sharkfest Lab)
    RDNSequence item: 1 item (id-at-commonName=public.sharkfest.local)
    RDNSequence item: 1 item (pkcs-9-at-emailAddress=co@sharkfest.local)
```

Certificate Chain



Restrictions

```
validity
  notBefore: utcTime (0)
    utcTime: 09-03-15 23:06:49 (UTC)
  notAfter: utcTime (0)
    utcTime: 10-03-15 23:06:49 (UTC)
```

```
Extension (id-ce-basicConstraints)
  Extension Id: 2.5.29.19 (id-ce-basicConstraints)
  BasicConstraintsSyntax
```

SSL Handshakes

ClientHello Message (1)

- Initiates a connection
- Informs the server of the client's capabilities:
  - Supported protocol version
  - Ciphersuites (authentication, encryption, MAC)
  - Compression algorithms
- All you need to produce for port scanning
  - EFF SSL Observatory

ClientHello Message (2)

```
⊟ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 61
    Version: TLS 1.0 (0x0301)
  ⊟ Random
      gmt_unix_time: Apr 19, 2009 10:43:26.000000000 Central Daylight Time
      random_bytes: dd819516fc5dddd097428d410d7852e2579e2e8903cdb331...
    Session ID Length: 0
    Cipher Suites Length: 16
  ⊟ Cipher Suites (8 suites)
      Cipher Suite: TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0084)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
      Cipher Suite: TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0041)
      Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
      Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Cipher Suite: SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA (0xfeff)
      Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
    Compression Methods Length: 1
  ⊟ Compression Methods (1 method)
      Compression Method: null (0)
    Extensions Length: 4
  ⊟ Extension: SessionTicket TLS
      Type: SessionTicket TLS (0x0023)
      Length: 0
      Data (0 bytes)
```

ServerHello Message (1)

- Responds to a connection request
- Informs the client of its choices:
  - Protocol version
  - Ciphersuite
  - Compression algorithm

ServerHello Message (2)

```
⊟ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 70
    Version: TLS 1.0 (0x0301)
  ⊟ Random
      gmt_unix_time: Mar 15, 2009 20:30:23.000000000 Central Daylight Time
      random_bytes: d6f56969813144fdb2340a273f419e463bf915549b0740df...
    Session ID Length: 32
    Session ID: db00c2aad79cfda109ce4f65a9801aa8d5f1bbeb9e1f848f...
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Compression Method: null (0)
```

Certificate Message

- Every certificate the client will need to create a chain to reach the CA
- Certificates should be ordered in ascending order
  1. Server certificate
  2. Intermediate certificates
- Shouldn’t contain unused certificates

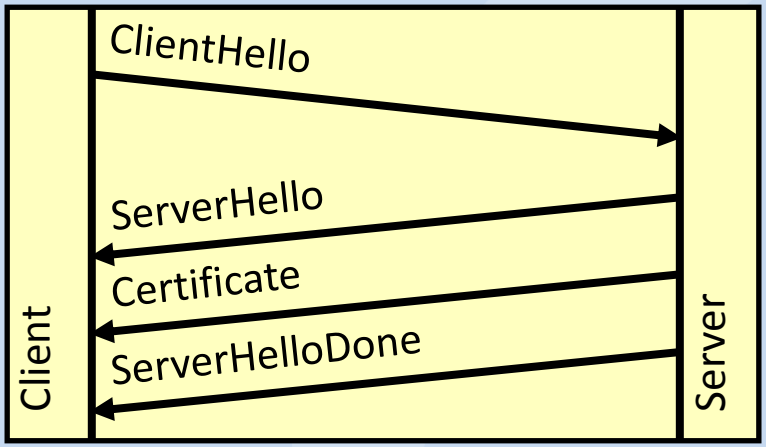
ServerHelloDone Message (1)

- “Your move…”

ServerHelloDone Message (2)

```
⊟ Handshake Protocol: Server Hello Done
    Handshake Type: Server Hello Done (14)
    Length: 0
```

Recap

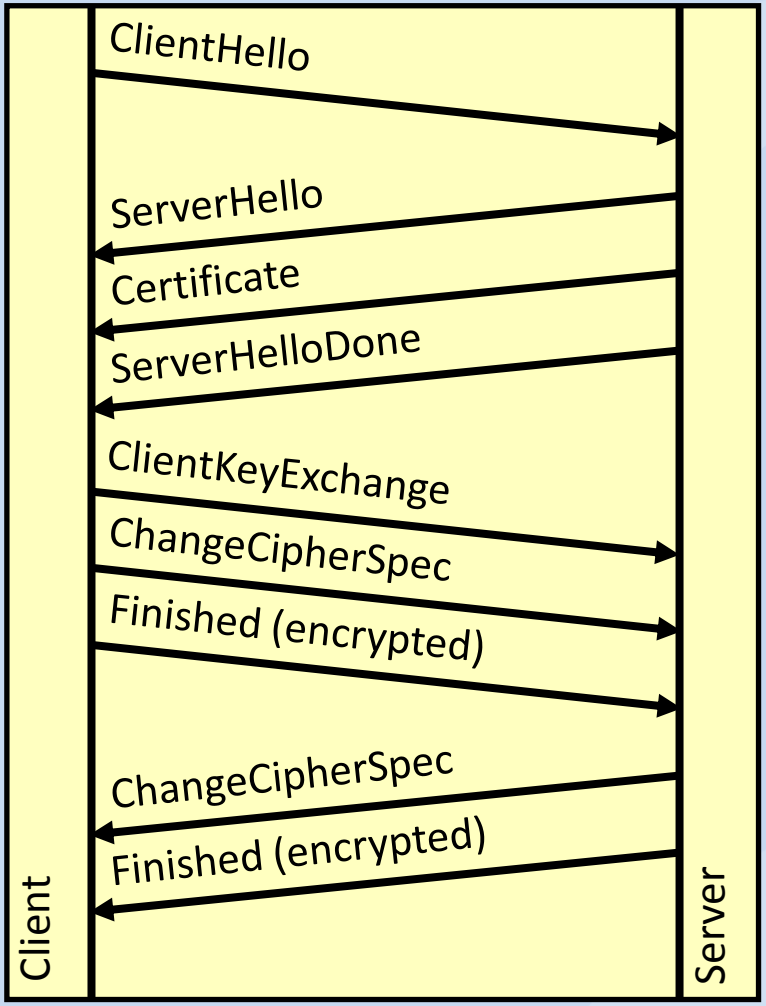

Client
ClientHello
ServerHello
Certificate
ServerHelloDone
Server

Handshake Scenarios

1. Normal handshake
2. Ephemeral handshake
3. Client authentication

- Reusing SSL sessions
  1. Reused SSL session
  2. Expired SSL session
  3. No SSL reuse

Normal Handshake (1)



Normal Handshake (2)

4	0.011511	192.168.3.1	192.168.3.3	TLSv1	124	Client Hello
6	0.017431	192.168.3.3	192.168.3.1	TLSv1	1514	Server Hello
7	0.017782	192.168.3.3	192.168.3.1	TLSv1	1019	Certificate
9	0.026711	192.168.3.1	192.168.3.3	TLSv1	252	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.038327	192.168.3.3	192.168.3.1	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message

ClientKeyExchange (1)

- Contents are encrypted asymmetrically
- Contains the PreMaster Secret
  - Used to derive the Master Secret
    - Master Secret produces all symmetric keys
  - If you know this, you can decrypt the session

ClientKeyExchange (2)

```
⊟ Handshake Protocol: Client Key Exchange
    Handshake Type: Client Key Exchange (16)
    Length: 130
  ⊟ RSA Encrypted PreMaster Secret
      Encrypted PreMaster length: 128
      Encrypted PreMaster: 761b1beac35e59de9a3bb9f74ebf9109b738e8ad346c3ce8...
```

ChangeCipherSpec :: Client

```
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
  ⊟ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.0 (0x0301)
      Length: 1
      Change Cipher Spec Message
  ⊞ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

Finished :: Client

Without decryption:

```
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
  ⊞ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊟ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 48
      Handshake Protocol: Encrypted Handshake Message
```

With decryption:

```
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
  ⊞ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊟ TLSv1 Record Layer: Handshake Protocol: Finished
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 48
    ⊟ Handshake Protocol: Finished
        Handshake Type: Finished (20)
        Length: 12
        Verify Data
```

ChangeCipherSpec :: Server

```
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.0 (0x0301)
      Length: 1
      Change Cipher Spec Message
  ⊞ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

Finished :: Server

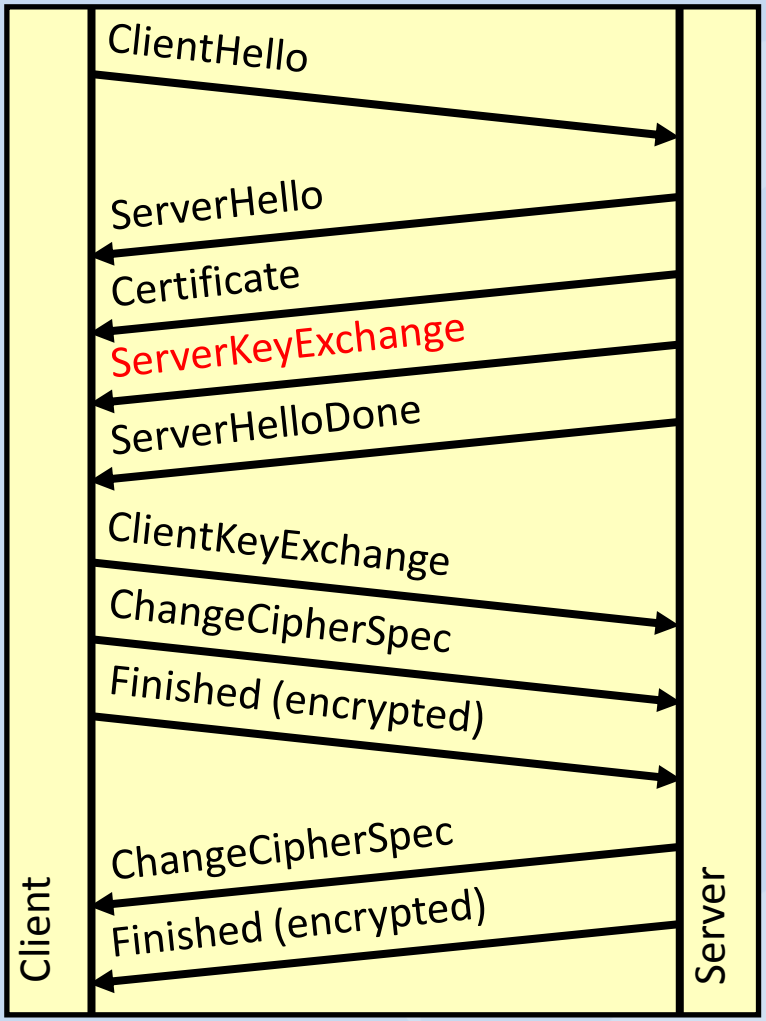
Without decryption:

```
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊟ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 48
      Handshake Protocol: Encrypted Handshake Message
```

With decryption:

```
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊟ TLSv1 Record Layer: Handshake Protocol: Finished
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 48
    ⊟ Handshake Protocol: Finished
        Handshake Type: Finished (20)
        Length: 12
        Verify Data
```

Ephemeral Handshake (1)



Ephemeral Handshake (2)

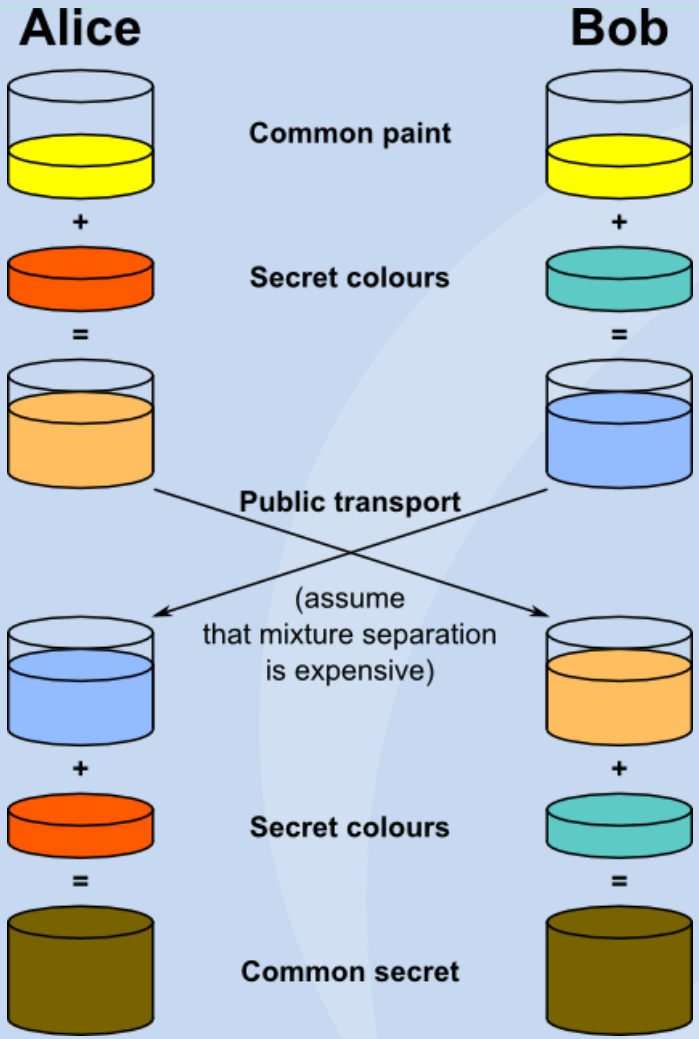
No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.1	192.168.3.3	TCP	42370 > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
2	0.000577	192.168.3.3	192.168.3.1	TCP	https > 42370 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
3	0.000618	192.168.3.1	192.168.3.3	TCP	42370 > https [ACK] Seq=1 Ack=1 Win=128000 Len=0
4	0.026109	192.168.3.1	192.168.3.3	SSL	Client Hello
5	0.026465	192.168.3.3	192.168.3.1	TCP	https > 42370 [ACK] Seq=1 Ack=107 Win=5840 Len=0
6	0.070925	192.168.3.3	192.168.3.1	TLSv1	Server Hello,
7	0.071108	192.168.3.3	192.168.3.1	TLSv1	Certificate, Server Key Exchange, Server Hello Done
8	0.071172	192.168.3.1	192.168.3.3	TCP	42370 > https [ACK] Seq=107 Ack=2828 Win=128000 Len=0
9	0.090279	192.168.3.1	192.168.3.3	TLSv1	Client Key Exchange, Change Cipher Spec, Encrypted Handshak
10	0.090657	192.168.3.3	192.168.3.1	TCP	https > 42370 [ACK] Seq=2828 Ack=305 Win=6912 Len=0
11	0.110494	192.168.3.3	192.168.3.1	TLSv1	Change Cipher Spec, Encrypted Handshake Message

ServerKeyExchange (1)

- Sends the data the client will need to create the PreMaster Secret
- Keys will be thrown out after the session expires

You cannot decrypt this session with Wireshark

ServerKeyExchange (2)

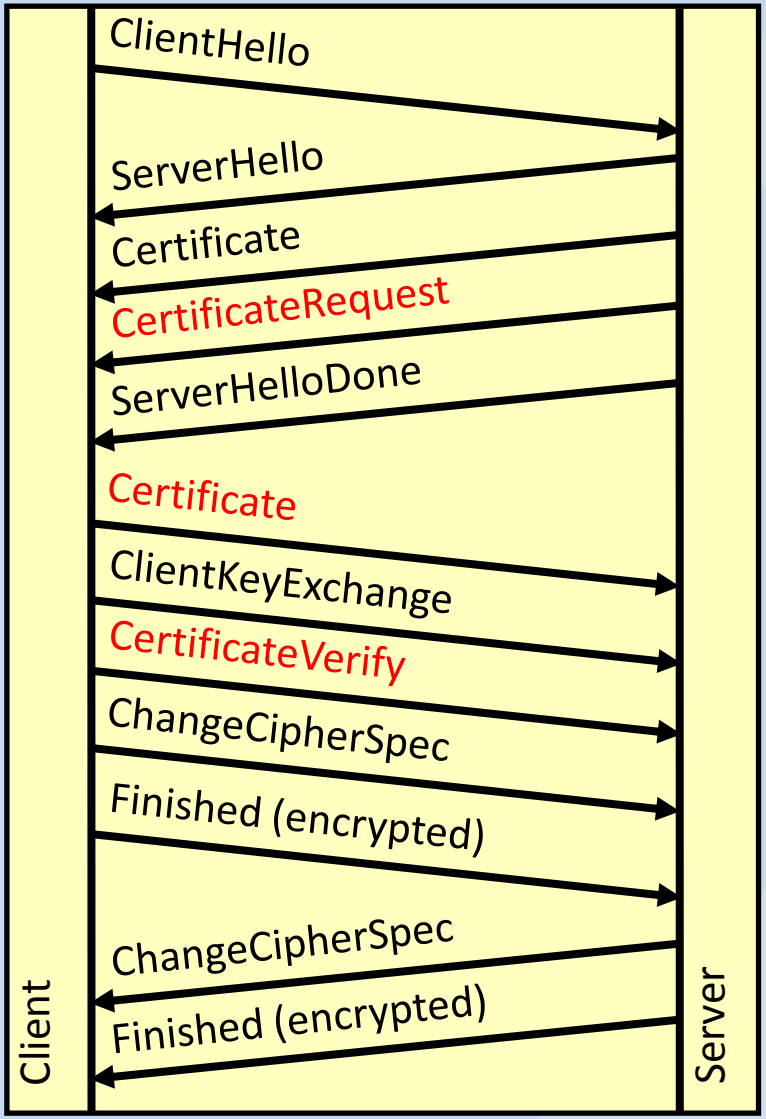


Explanation available at **http://tinyurl.com/dh-paint**

ServerKeyExchange (3)

```
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Certificate
  ⊟ TLSv1 Record Layer: Handshake Protocol: Server Key Exchange
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 397
    ⊟ Handshake Protocol: Server Key Exchange
        Handshake Type: Server Key Exchange (12)
        Length: 393
  ⊞ TLSv1 Record Layer: Handshake Protocol: Server Hello Done
```

Client Authentication (1)



Client Authentication (2)

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.1	192.168.3.4	TCP	14980 > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
2	0.000372	192.168.3.4	192.168.3.1	TCP	https > 14980 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
3	0.000400	192.168.3.1	192.168.3.4	TCP	14980 > https [ACK] Seq=1 Ack=1 Win=128000 Len=0
4	0.015645	192.168.3.1	192.168.3.4	SSLv2	Client Hello
5	0.015824	192.168.3.4	192.168.3.1	TCP	https > 14980 [ACK] Seq=1 Ack=52 Win=5840 Len=0
6	0.017894	192.168.3.4	192.168.3.1	SSLv3	Server Hello,
7	0.017988	192.168.3.4	192.168.3.1	SSLv3	Certificate, Certificate Request, Server Hello Done
8	0.018015	192.168.3.1	192.168.3.4	TCP	14980 > https [ACK] Seq=52 Ack=2590 Win=128000 Len=0
9	4.089191	192.168.3.1	192.168.3.4	TCP	[TCP segment of a reassembled PDU]
10	4.089622	192.168.3.4	192.168.3.1	TCP	https > 14980 [ACK] Seq=2590 Ack=1512 Win=8768 Len=0
11	4.089949	192.168.3.1	192.168.3.4	SSLv3	Certificate, Client Key Exchange, Certificate Verify, Chang
12	4.107141	192.168.3.4	192.168.3.1	SSLv3	Change Cipher Spec, Encrypted Handshake Message

CertificateRequest

```
⊟ Secure Socket Layer
  ⊞ SSLv3 Record Layer: Handshake Protocol: Certificate
  ⊟ SSLv3 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 167
    ⊟ Handshake Protocol: Certificate Request
        Handshake Type: Certificate Request (13)
        Length: 159
        Certificate types count: 2
      ⊟ Certificate types (2 types)
          Certificate type: RSA Sign (1)
          Certificate type: DSS Sign (2)
        Distinguished Names Length: 154
      ⊟ Distinguished Names (154 bytes)
          Distinguished Name Length: 152
        ⊟ Distinguished Name: ()
          ⊟ RDNSequence: 1 item ()
            ⊟ RelativeDistinguishedName
                Id: 2.5.4.3 (id-at-commonName)
              ⊟ DirectoryString: printableString (1)
                  printableString: Sharkfest Lab Root CA
          ⊞ RDNSequence: 1 item ()
          ⊞ RDNSequence: 1 item ()
          ⊞ RDNSequence: 1 item ()
          ⊞ RDNSequence: 1 item ()
          ⊞ RDNSequence: 1 item ()
    ⊞ Handshake Protocol: Server Hello Done
```

Certificate

```
⊟ Secure Socket Layer
  ⊟ SSLv3 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 2579
    ⊟ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 2309
        Certificates Length: 2306
      ⊟ Certificates (2306 bytes)
          Certificate Length: 1060
        ⊞ Certificate ()
          Certificate Length: 1240
        ⊞ Certificate ()
    ⊞ Handshake Protocol: Client Key Exchange
    ⊞ Handshake Protocol: Certificate Verify
  ⊞ SSLv3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊞ SSLv3 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

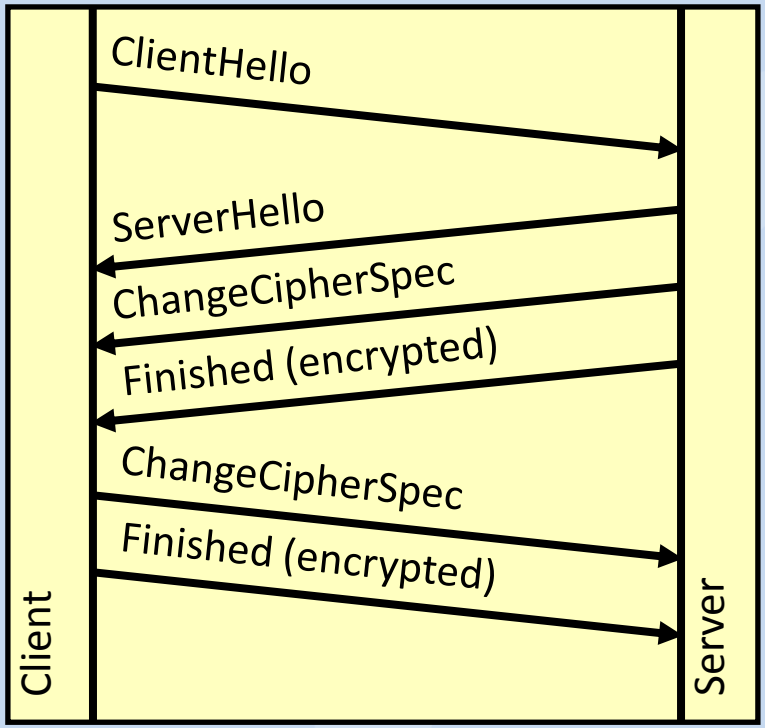
CertificateVerify

```
⊟ Secure Socket Layer
  ⊟ SSLv3 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 2579
    ⊞ Handshake Protocol: Certificate
    ⊞ Handshake Protocol: Client Key Exchange
    ⊟ Handshake Protocol: Certificate Verify
        Handshake Type: Certificate Verify (15)
        Length: 130
  ⊞ SSLv3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊞ SSLv3 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

Session Resume

- Key negotiation expensive
- Servers can cache session keys for reuse
- Session ID used to retrieve keys from cache
- Cache has an absolute timeout, not idle
  - Ensures eventual re-keying

Handshake of a Reused Session



No.	Time	Source	Destination	Protocol	Info
23	39.68772	192.168.3.1	192.168.3.3	TCP	18774 > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
24	39.68810	192.168.3.3	192.168.3.1	TCP	https > 18774 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=
25	39.68814	192.168.3.1	192.168.3.3	TCP	18774 > https [ACK] Seq=1 Ack=1 Win=128000 Len=0
26	39.68871	192.168.3.1	192.168.3.3	TLSv1	Client Hello
27	39.68898	192.168.3.3	192.168.3.1	TCP	https > 18774 [ACK] Seq=1 Ack=103 Win=5840 Len=0
28	39.69430	192.168.3.3	192.168.3.1	TLSv1	Server Hello, Change Cipher Spec, Encrypted Handshake Message
29	39.71735	192.168.3.1	192.168.3.3	TLSv1	Change Cipher Spec, Encrypted Handshake Message, Application Dat

Session Resume in Action

Filter: ssl.handshake | Expression... | Clear | Apply | Save

No.	Time	Source	Destination	Protocol	Session ID	Info
4	0.011511	192.168.3.1	192.168.3.3	TLSv1		Client Hello
6	0.017431	192.168.3.3	192.168.3.1	TLSv1	db00c2aad7	Server Hello
7	0.017782	192.168.3.3	192.168.3.1	TLSv1		Certificate
9	0.026711	192.168.3.1	192.168.3.3	TLSv1		Client Key Exchange, Change Ciphe
10	0.038327	192.168.3.3	192.168.3.1	TLSv1		Change Cipher Spec, Encrypted Har
26	39.688711	192.168.3.1	192.168.3.3	TLSv1	db00c2aad7	Client Hello
28	39.694301	192.168.3.3	192.168.3.1	TLSv1	db00c2aad7	Server Hello, Change Cipher Spec,
29	39.717354	192.168.3.1	192.168.3.3	TLSv1		Change Cipher Spec, Encrypted Har
41	111.192869	192.168.3.1	192.168.3.3	TLSv1	db00c2aad7	Client Hello
43	111.197758	192.168.3.3	192.168.3.1	TLSv1	fbcf322128	Server Hello
44	111.197984	192.168.3.3	192.168.3.1	TLSv1		Certificate
46	111.205534	192.168.3.1	192.168.3.3	TLSv1		Client Key Exchange, Change Ciphe
47	111.217564	192.168.3.3	192.168.3.1	TLSv1		Change Cipher Spec, Encrypted Har

Without SSL Resume

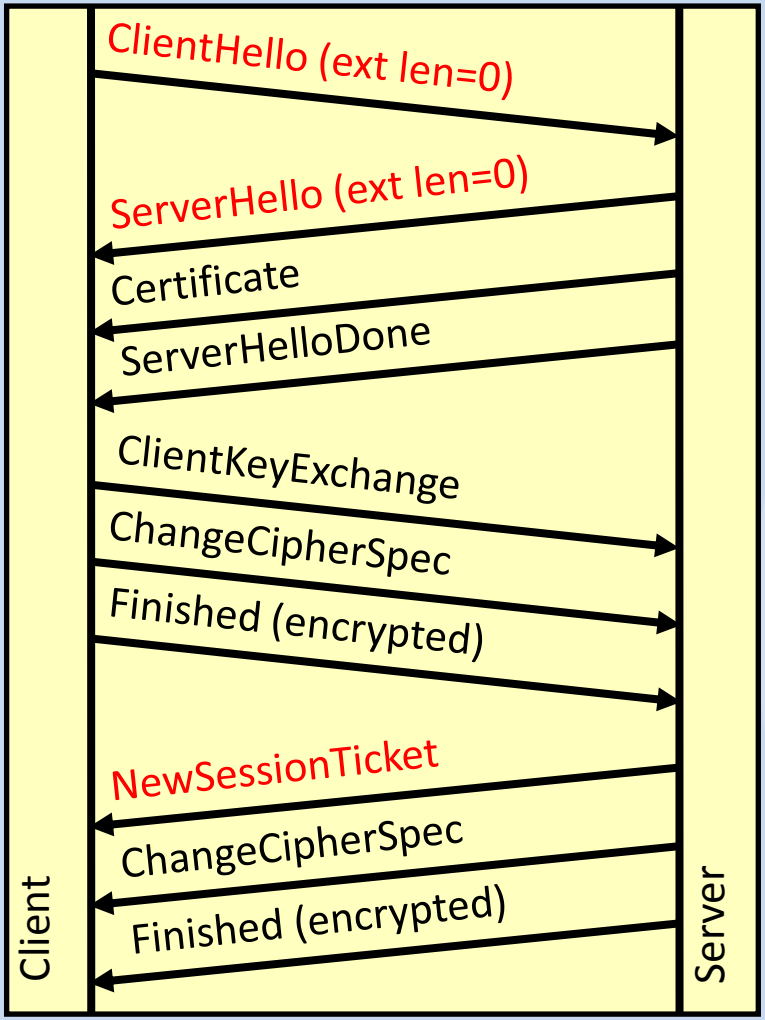
No.	Time	Source	Destination	Protocol	ssl-id len	ssl-id	Info
4	0.011833	192.168.3.1	192.168.3.3	TLSv1	32	5186BC	Client Hello
6	0.018800	192.168.3.3	192.168.3.1	TLSv1	0		Server Hello,
7	0.019128	192.168.3.3	192.168.3.1	TLSv1			Certificate
9	0.026392	192.168.3.1	192.168.3.3	TLSv1			Client Key Exchange, Change Cipher Spec, Encryp
10	0.037500	192.168.3.3	192.168.3.1	TLSv1			Change Cipher Spec, Encrypted Handshake Message

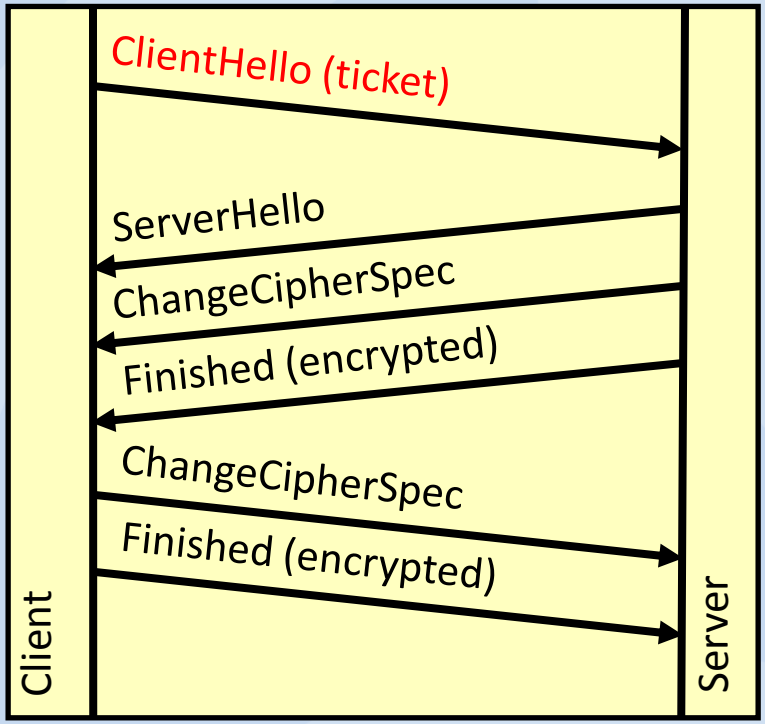
```
⊞ Frame 6 (1514 bytes on wire, 1514 bytes captured)
⊞ Ethernet II, Src: Vmware_5d:c5:66 (00:0c:29:5d:c5:66), Dst: Vmware_c0:00:01 (00:50:56:c0:00:01)
⊞ Internet Protocol, Src: 192.168.3.3 (192.168.3.3), Dst: 192.168.3.1 (192.168.3.1)
⊞ Transmission Control Protocol, Src Port: https (443), Dst Port: 17788 (17788), Seq: 1, Ack: 103, Len: 1460
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 42
    ⊟ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 38
        Version: TLS 1.0 (0x0301)
      ⊞ Random
        Session ID Length: 0
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
        Compression Method: null (0)
```

TLS Session Tickets (1)

- No state on server, only on client
  - Good for load balancing
- TLS extension in ClientHello and ServerHello
- New SSL HandshakeType: **NewSessionTicket**

TLS Session Tickets (2)





TLS Session Tickets (3)

4	0.015145	192.168.1.22	74.125.132.19	TLSv1	164	Client Hello
6	0.032365	74.125.132.19	192.168.1.22	TLSv1	1484	Server Hello
7	0.032767	74.125.132.19	192.168.1.22	TLSv1	350	Certificate, Server Hello Done
9	0.033752	192.168.1.22	74.125.132.19	TLSv1	252	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.051951	74.125.132.19	192.168.1.22	TLSv1	292	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
22	2.363423	192.168.1.22	74.125.132.19	TLSv1	360	Client Hello
26	2.383264	74.125.132.19	192.168.1.22	TLSv1	199	Server Hello, Change Cipher Spec, Encrypted Handshake Message

```
▽ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 89
    Version: TLS 1.0 (0x0301)
```

```
▽ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 53
    Version: TLS 1.0 (0x0301)
  ▷ Random
    Session ID Length: 0
    Cipher Suite: TLS_RSA_WITH_RC4_128_SH
    Compression Method: null (0)
    Extensions Length: 13
  ▷ Extension: server_name
  ▷ Extension: renegotiation_info
  ▽ Extension: SessionTicket TLS
```

```
▽ TLSv1 Record Layer: Handshake Proto
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 174
  ▽ Handshake Protocol: New Session Ticket
      Handshake Type: New Session Ticket (4)
      Length: 170
```

```
▽ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 285
    Version: TLS 1.0 (0x0301)
  ▷ Random
    Session ID Length: 32
    Session ID: 73d2a649be4542fefe7b5cb6f4b15b5a48ae87f7597390b0...
    Cipher Suites Length: 20
  ▷ Cipher Suites (10 suites)
    Compression Methods Length: 1
  ▷ Compression Methods (1 method)
    Extensions Length: 192
  ▷ Extension: server_name
  ▽ Extension: SessionTicket TLS
      Type: SessionTicket TLS (0x0023)
      Length: 164
      Data (164 bytes)
```

TLS Session Tickets (4)

- Dangerous to trust client with server's state
- ASP.net vulnerability (MS10-070, CVE-2010-3332)

Alerts

Without decryption:

14	12.494568	192.168.3.1	192.168.3.3	TLSv1	Application Data
15	12.495834	192.168.3.3	192.168.3.1	TLSv1	Application Data, Application Data
17	27.530927	192.168.3.3	192.168.3.1	TLSv1	Encrypted Alert
20	32.811207	192.168.3.1	192.168.3.3	TLSv1	Encrypted Alert

```
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Encrypted Alert
      Content Type: Alert (21)
      Version: TLS 1.0 (0x0301)
      Length: 32
      Alert Message: Encrypted Alert
```

With decryption:

14	12.494568	192.168.3.1	192.168.3.3	HTTP	GET / HTTP/1.1
15	12.495834	192.168.3.3	192.168.3.1	HTTP	HTTP/1.1 200 OK (text/html)
17	27.530927	192.168.3.3	192.168.3.1	TLSv1	Alert (Level: Warning, Description: Close Notify)
20	32.811207	192.168.3.1	192.168.3.3	TLSv1	Alert (Level: Warning, Description: Close Notify)

```
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Alert (Level: Warning, Description: Close Notify)
      Content Type: Alert (21)
      Version: TLS 1.0 (0x0301)
      Length: 32
    ⊟ Alert Message
        Level: Warning (1)
        Description: Close Notify (0)
```

LAB: SSL Handshake

Things to look for in the PCAP

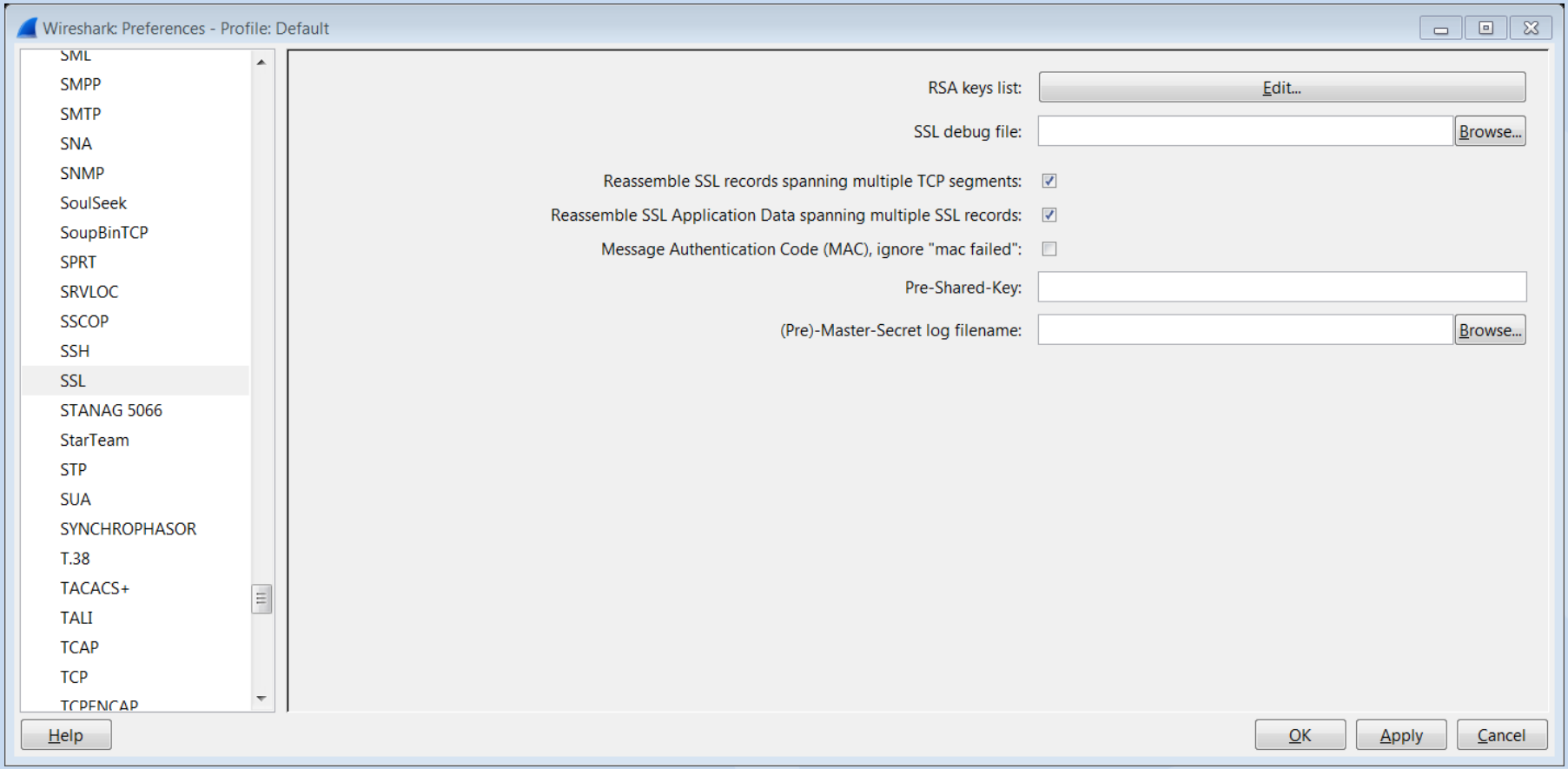
- Which ciphersuites does the client support?
- Which ciphersuite did the server choose?
  - How could we find the ciphersuites the server supports?
- How many SSL sessions are in the file?
  - How many of them were full handshakes?
  - Why?
- Were the client and server clocks in sync?

SSL Decryption

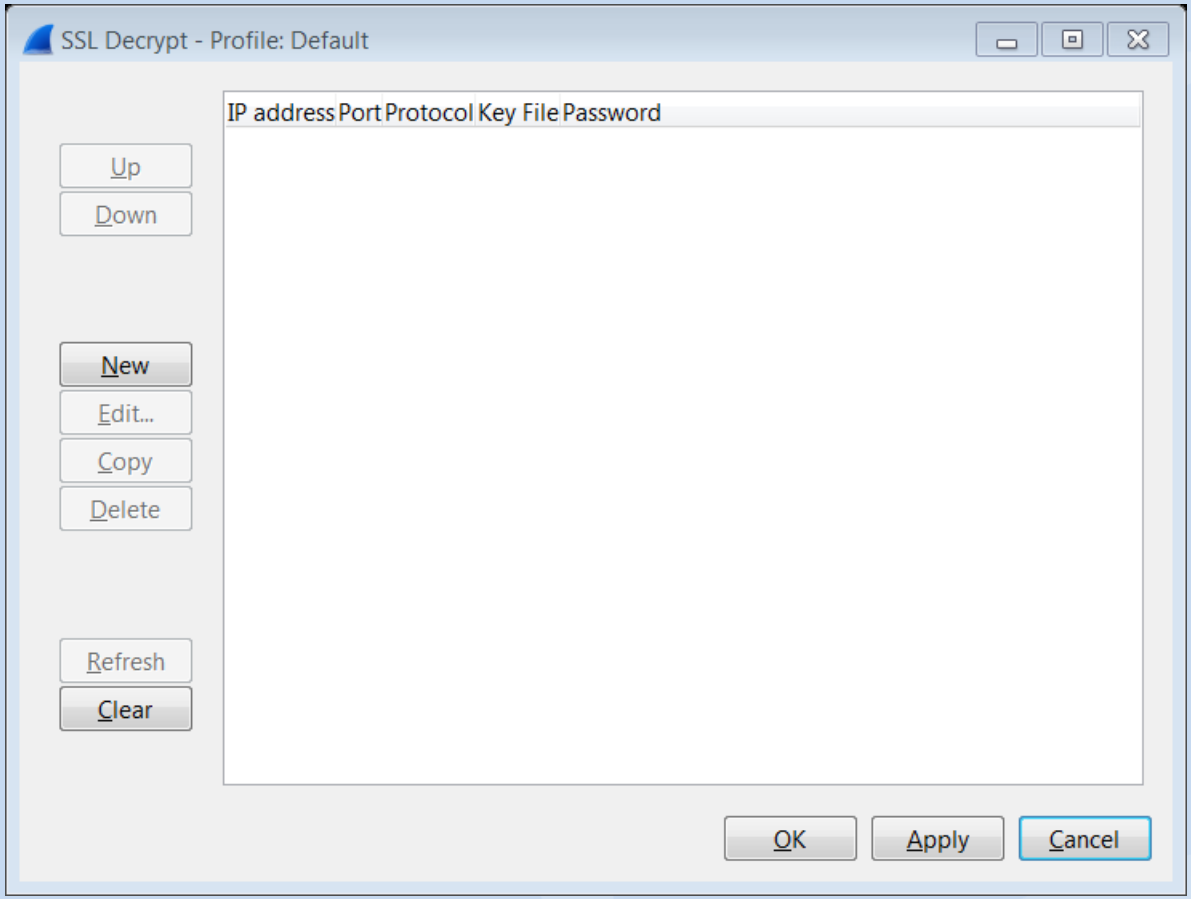
Decrypting SSL Sessions

1. You need the server’s private key
2. You need the entire session
3. Session must not be ephemeral

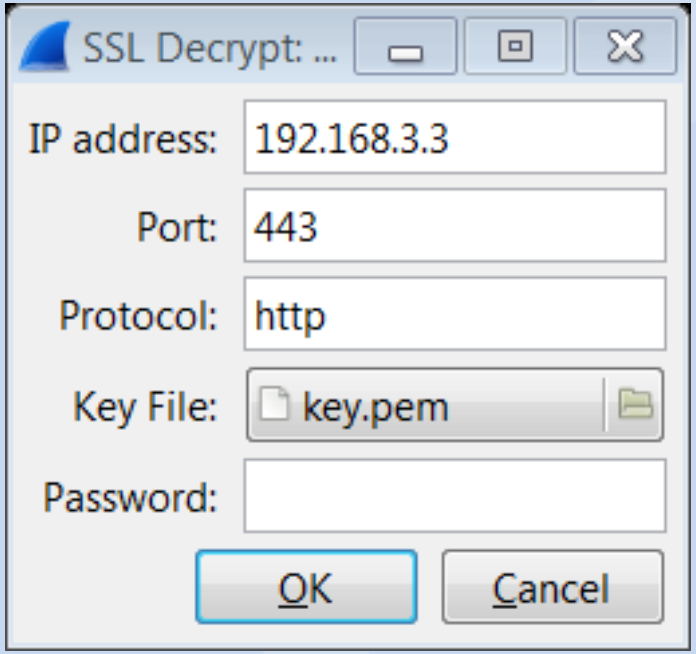
Providing the Server's Private Key (1)



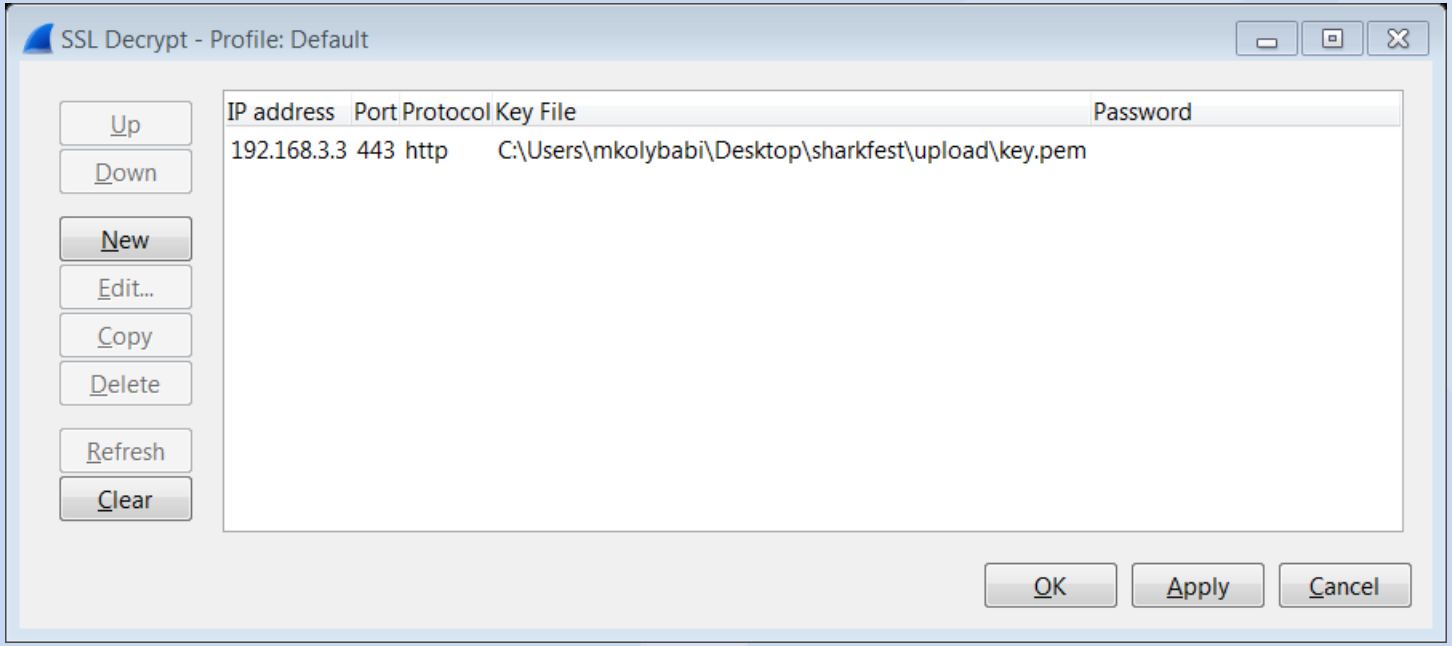
Providing the Server's Private Key (2)



Providing the Server's Private Key (3)



Providing the Server's Private Key (4)



Decryption in Action (1)

Client Hello
Server Hello
Certificate
Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
Change Cipher Spec, Encrypted Handshake Message
Application Data
Application Data, Application Data
Application Data
Application Data, Application Data
Encrypted Alert
Encrypted Alert

Client Hello
Server Hello
Certificate
Client Key Exchange, Change Cipher Spec, Finished
Change Cipher Spec, Finished
GET / HTTP/1.1
HTTP/1.1 200 OK (text/html)
GET / HTTP/1.1
HTTP/1.1 200 OK (text/html)
Alert (Level: Warning, Description: Close Notify)
Alert (Level: Warning, Description: Close Notify)

Decryption in Action (2)

```
⊟ Secure Sockets Layer
  ⊟ TLSv1 Record Layer: Application Data Protocol: http
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 432
      Encrypted Application Data: c0d1c49a5e8119fc1b21ef547592476df61aa48a11c445
⊟ Hypertext Transfer Protocol
  ⊞ GET / HTTP/1.1\r\n
    Host: 192.168.3.3\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.8) Geck
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    \r\n
```

Decrypting IMAPS

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.46	192.168.1.20	TCP	22446 > imaps [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
2	0.001820	192.168.1.20	192.168.1.46	TCP	imaps > 22446 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=7
3	0.001857	192.168.1.46	192.168.1.20	TCP	22446 > imaps [ACK] Seq=1 Ack=1 Win=128000 Len=0
4	0.010231	192.168.1.46	192.168.1.20	SSL	Client Hello
5	0.011625	192.168.1.20	192.168.1.46	TCP	imaps > 22446 [ACK] Seq=1 Ack=103 Win=5888 Len=0
6	0.012351	192.168.1.20	192.168.1.46	TLSv1	Server Hello, Certificate, Server Hello Done
7	0.013831	192.168.1.46	192.168.1.20	TLSv1	Client Key Exchange, Change Cipher Spec, Finished
8	0.019822	192.168.1.20	192.168.1.46	TLSv1	Change Cipher Spec, Finished
9	0.168748	192.168.1.46	192.168.1.20	TCP	22446 > imaps [ACK] Seq=285 Ack=978 Win=127022 Len=0
10	0.170301	192.168.1.20	192.168.1.46	IMAP	Response: * OK Dovecot ready.
11	0.172574	192.168.1.46	192.168.1.20	IMAP	Request: g7fg CAPABILITY

```
⊞ Frame 10 (96 bytes on wire, 96 bytes captured)
⊞ Ethernet II, Src: JuniperN_bb:d1:3b (00:12:1e:bb:d1:3b), Dst: IntelCor_61:3a:ad (00:1c:bf:61:3a:ad)
⊞ Internet Protocol, Src: 192.168.1.20 (192.168.1.20), Dst: 192.168.1.46 (192.168.1.46)
⊞ Transmission Control Protocol, Src Port: imaps (993), Dst Port: 22446 (22446), Seq: 978, Ack: 285, Len: 42
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Application Data Protocol: imap
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 37
      Encrypted Application Data: F8260B9E9D0597A3CE35E176BA3EDB28D588E004F6B57F74...
⊟ Internet Message Access Protocol
  ⊟ * OK Dovecot ready.\r\n
      Response Tag: *
      Response: OK Dovecot ready.
```

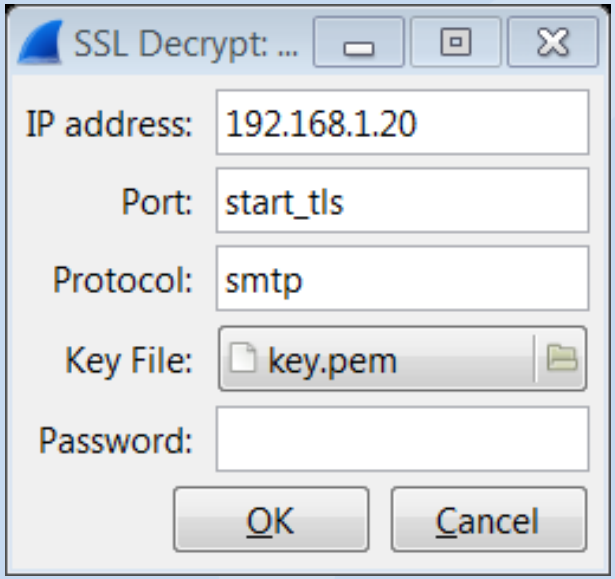
Decrypting STARTTLS (1)

Filter: smtp || ssl Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
4	0.021653	192.168.1.20	192.168.1.46	SMTP	S: 220 brutus.netcc.local ESMTP Postfix (Ubuntu)
5	0.023320	192.168.1.46	192.168.1.20	SMTP	C: EHLO HTRQ93J
7	0.025077	192.168.1.20	192.168.1.46	SMTP	S: 250-brutus.netcc.local | 250-PIPELINING | 250-SIZE 10240000 | :
8	0.025868	192.168.1.46	192.168.1.20	SMTP	C: STARTTLS
9	0.027373	192.168.1.20	192.168.1.46	SMTP	S: 220 2.0.0 Ready to start TLS
11	0.262273	192.168.1.46	192.168.1.20	TLSv1	Client Hello
12	0.264832	192.168.1.20	192.168.1.46	TLSv1	Server Hello, Certificate, Server Hello Done
13	0.266373	192.168.1.46	192.168.1.20	TLSv1	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Mess.
14	0.281296	192.168.1.20	192.168.1.46	TLSv1	Change Cipher Spec, Encrypted Handshake Message

```
⊞ Frame 13 (236 bytes on wire, 236 bytes captured)
⊞ Ethernet II, Src: IntelCor_61:3a:ad (00:1c:bf:61:3a:ad), Dst: JuniperN_bb:d1:3b (00:12:1e:bb:d1:3b)
⊞ Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 192.168.1.20 (192.168.1.20)
⊞ Transmission Control Protocol, Src Port: 38477 (38477), Dst Port: smtp (25), Seq: 95, Ack: 1153, Len: 182
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
  ⊞ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊟ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 32
      Handshake Protocol: Encrypted Handshake Message
```

Decrypting STARTTLS (2)



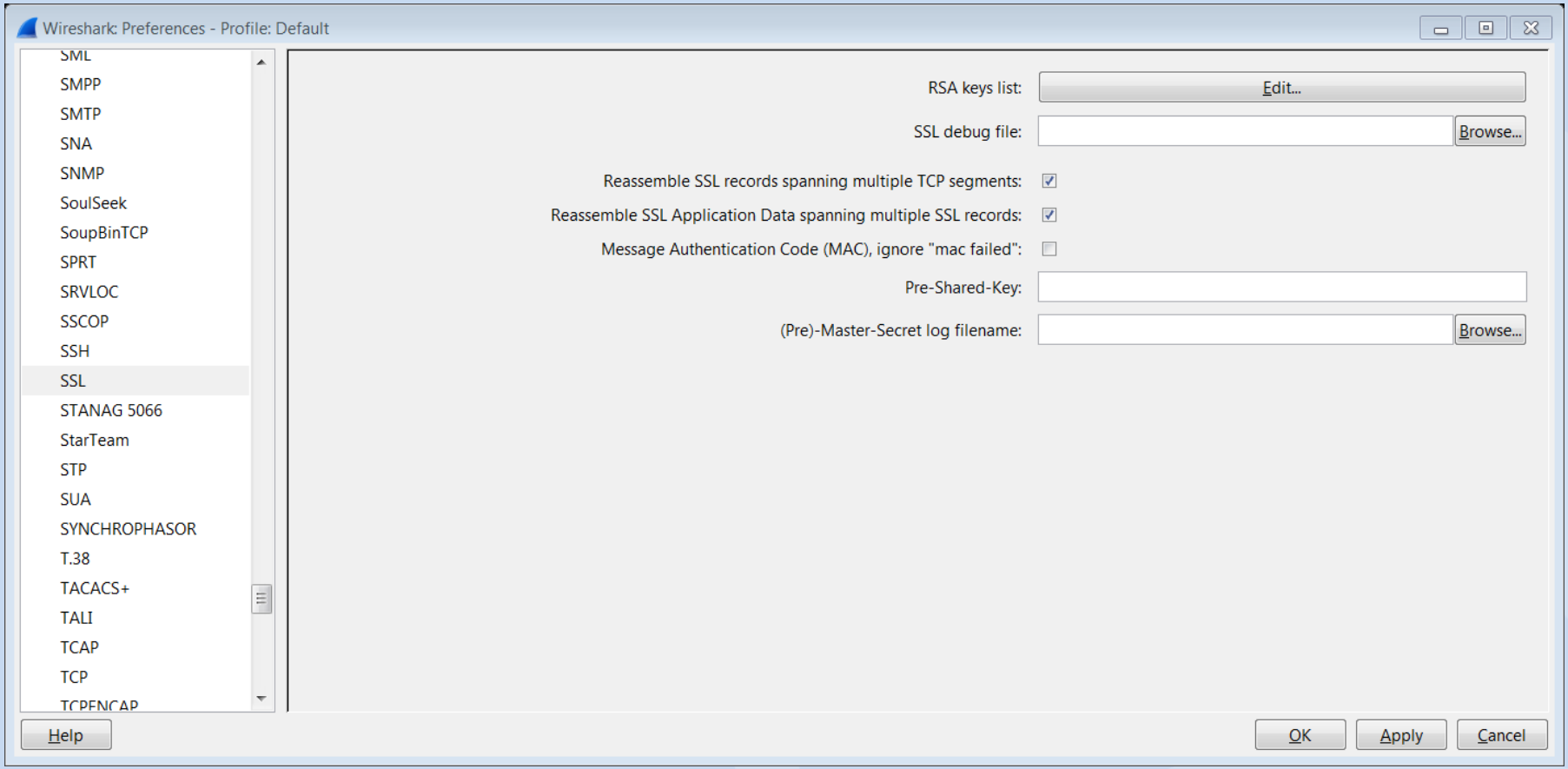
Decrypting STARTTLS (3)

Filter: smtp || ssl Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
4	0.021653	192.168.1.20	192.168.1.46	SMTP	S: 220 brutus.netcc.local ESMTP Postfix (Ubuntu)
5	0.023320	192.168.1.46	192.168.1.20	SMTP	C: EHLO HTRQ93J
7	0.025077	192.168.1.20	192.168.1.46	SMTP	S: 250-brutus.netcc.local | 250-PIPELINING | 250-SIZE 10240000 |
8	0.025868	192.168.1.46	192.168.1.20	SMTP	C: STARTTLS
9	0.027373	192.168.1.20	192.168.1.46	SMTP	S: 220 2.0.0 Ready to start TLS
11	0.262273	192.168.1.46	192.168.1.20	TLSv1	Client Hello
12	0.264832	192.168.1.20	192.168.1.46	TLSv1	Server Hello, Certificate, Server Hello Done
13	0.266373	192.168.1.46	192.168.1.20	TLSv1	Client Key Exchange, Change Cipher Spec, Finished
14	0.281296	192.168.1.20	192.168.1.46	TLSv1	Change Cipher Spec, Encrypted Handshake Message

```
⊞ Frame 13 (236 bytes on wire, 236 bytes captured)
⊞ Ethernet II, Src: IntelCor_61:3a:ad (00:1c:bf:61:3a:ad), Dst: JuniperN_bb:d1:3b (00:12:1e:bb:d1:3b)
⊞ Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 192.168.1.20 (192.168.1.20)
⊞ Transmission Control Protocol, Src Port: 38477 (38477), Dst Port: smtp (25), Seq: 95, Ack: 1153, Len: 182
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
  ⊞ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ⊟ TLSv1 Record Layer: Handshake Protocol: Finished
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 32
    ⊟ Handshake Protocol: Finished
        Handshake Type: Finished (20)
        Length: 12
        Verify Data
```

SSL Debug File



Decrypt Problem (1)

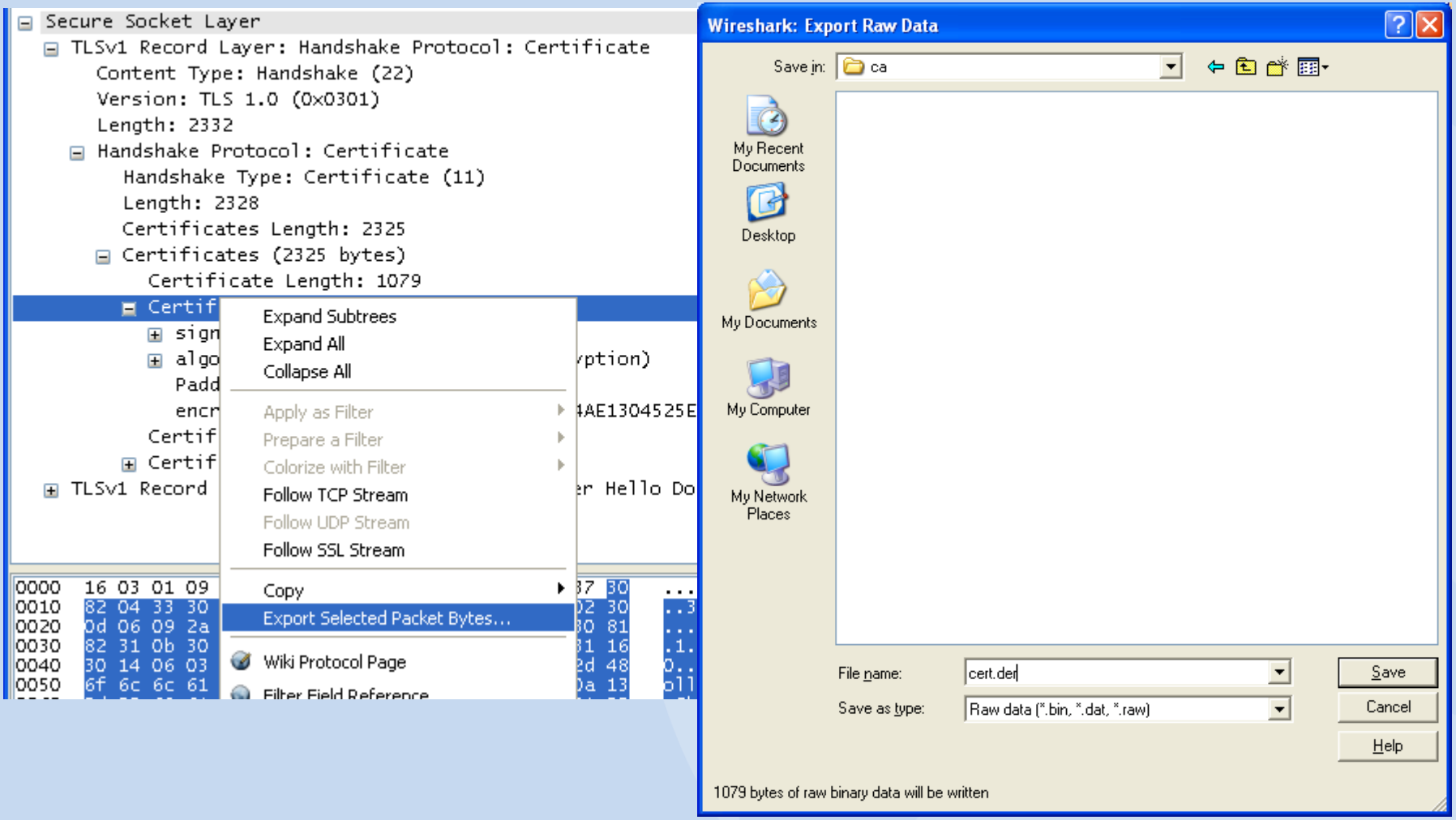
Checking ssl debug log:

```
ssl_association_remove removing TCP 443 - http handle 04086F30
ssl_init keys string:
192.168.3.3,443,http,c:\temp\public.sharkfest.local.key
ssl_init found host entry 192.168.3.3,443,http,c:\temp\public.sharkfest.local.key
ssl_init addr '192.168.3.3' port '443' filename 'c:\temp\public.sharkfest.local.key' password(only for p12
file) '(null)'
Private key imported: KeyID FA:56:73:A4:38:9C:A1:4F:28:23:88:76:83:42:13:86:...
ssl_init private key file c:\temp\public.sharkfest.local.key successfully loaded
association_add TCP port 443 protocol http handle 04086F30

[...]

ssl_decrypt_pre_master_secret:RSA_private_decrypt
pcry_private_decrypt: stripping 0 bytes, decr_len zd
decrypted_unstrip_pre_master[128]:
6a f7 2a 4b 45 17 72 47 c2 11 d1 dd ad dc af b6
04 76 cb 3c 32 1c d1 01 57 4a 83 79 af d9 40 af
aa a8 71 1f bd 6f 70 d5 cc 49 e6 be 44 42 07 7c
45 b7 5b 5b 52 de 3e 58 d3 42 8d 5f bc 99 3e 13
f5 7d 27 a1 3e 7f b2 3f 8b 9d e5 fb 60 ec 40 26
87 8f 24 41 fb d4 ec f7 0e ea 04 46 c2 d7 5f 7b
4a d2 40 47 07 7b 0d 63 d8 d6 0f e6 9e 98 92 02
58 13 51 72 1b 85 69 04 52 42 74 12 40 e2 a5 bb
ssl_decrypt_pre_master_secret wrong pre_master_secret length (128, expected 48)
dissect_ssl3_handshake can't decrypt pre master secret
```

Decrypt Problem (2)



Decrypt Problem (3)

In wireshark preferences:

```
ssl.keys_list: 192.168.3.3,443,http,c:\temp\public.sharkfest.local.key
```

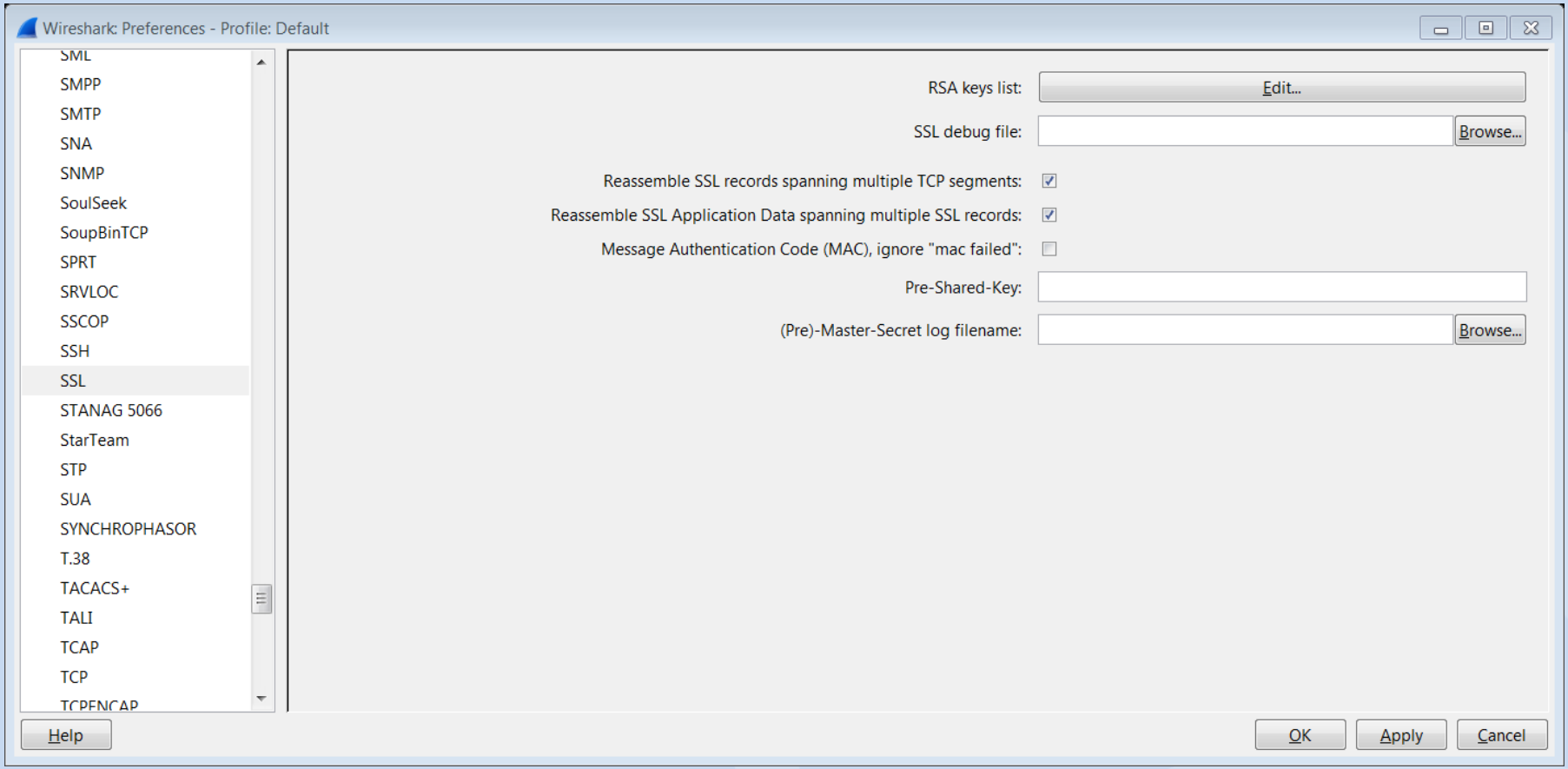
Checking whether certificate and key match:

```
$ openssl x509 -in cert.der -inform DER -noout -text | grep "Subject:"
        Subject: C=NL, ST=Noord-Holland, O=Sharkfest Lab,
CN=public.sharkfest.local/emailAddress=co@sharkfest.local
$
$ openssl x509 -noout -modulus -inform DER -in cert.der | openssl md5
a29682af822b4cd064d39d4ccd1e0e6c
$
$ openssl rsa -noout -modulus -in public.sharkfest.local.key | openssl md5
ce71158d3851a885314c264863142389
$
$ openssl rsa -noout -modulus -in private.sharkfest.local.key | openssl md5
a29682af822b4cd064d39d4ccd1e0e6c
$
```

Decryption Without the Private Key (1)

- Allows debugging when you control only the client
- Needs the PreMaster Secret, from either:
  - Debug version of Firefox/Chrome
  - OpenSSL's s_client

Decryption Without the Private Key (2)



OpenSSL's s_client

```
$ openssl s_client -cipher AES256-SHA -no_ticket -connect imap.syn-bit.nl:993 | tee openssl-s_client.txtdepth=1
C = GB, ST = Greater Manchester, L = Salford, O = COMODO CA Limited, CN = COMODO High-Assurance Secure Server
CAverify error:num=20:unable to get local issuer certificateverify return:0CONNECTED(00000003)---[...]
SSL-Session:    Protocol  : TLSv1    Cipher    : AES256-SHA    Session-ID:
5EF3E7EDCC46993E51935914ACC1CBE6723259121248F958BC223D54FA84CFA0    Session-ID-ctx:    Master-Key:
066512
None
Start
certif
```

15	0.180186	46.30.211.94	192.168.1.22	IMAP	119	Response: * OK IMAP4 ready
17	2.631302	192.168.1.22	46.30.211.94	IMAP	140	Request: HELP
18	2.669607	46.30.211.94	192.168.1.22	IMAP	135	Response: * BAD invalid command

```
▷ Frame 15: 119 bytes on wire (952 bits), 119 bytes captured (952 bits)
▷ Ethernet II, Src: JuniperN_bb:d1:32 (00:12:1e:bb:d1:32), Dst: Apple_d8:87:48 (f8:1e:df:d8:87:48)
▷ Internet Protocol Version 4, Src: 46.30.211.94 (46.30.211.94), Dst: 192.168.1.22 (192.168.1.22)
▷ Transmission Control Protocol, Src Port: imaps (993), Dst Port: 64400 (64400), Seq: 2965, Ack: 425, Len: 53
▽ Secure Sockets Layer
  ▽ TLSv1 Record Layer: Application Data Protocol: imap
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 48
      Encrypted Application Data: 74a980b1955b4f74c1be949df97da0f4a25f27704ed7b66a...
▽ Internet Message Access Protocol
  ▷ * OK IMAP4 ready\r\n
```

```
$ awk '$1~"Session-ID:" {printf("RSA %s%s ",$1,$2)} $1~"Master-Key:" {printf("%s%s\n",$1,$2)}' openssl-s_client.txt > openssl-s_client.keys$ cat openssl-s_client.keys RSA Session-ID:5EF3E7EDCC46993E51935914ACC1CBE6723259121248F958BC223D54FA84CFA0 Master-Key:0665121ADB266864CDEF89E32A6F1A39677D540DB5B362BC351D3B08EE3059800F9A218E6601710CE774AFB2CE3166C9$
```

SSL
STANAG 5066

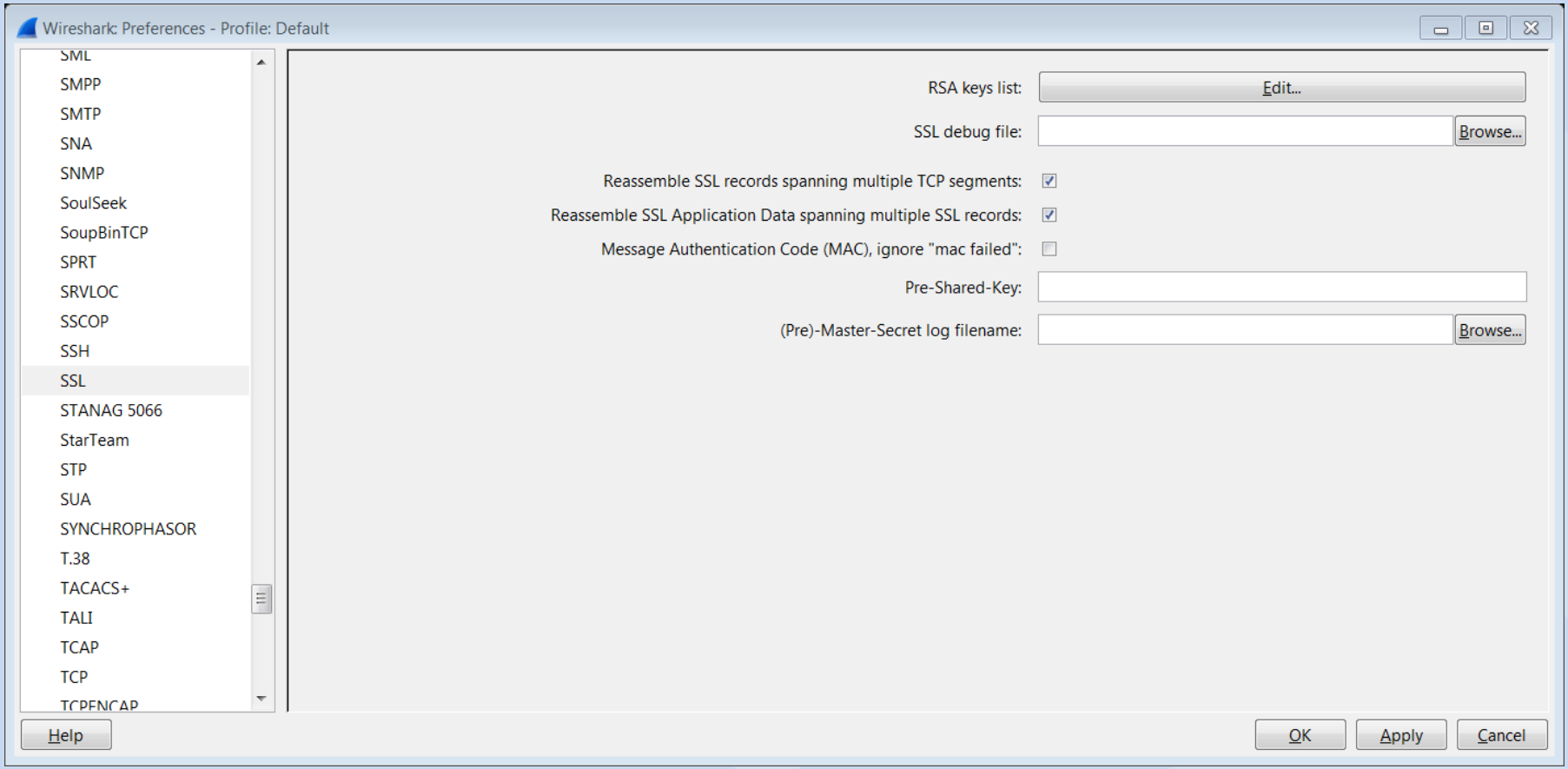
(Pre)-Master-Secret log filename: 012/traces/openssl-s_client.keys Browse...

Exporting Session Keys

- Export:
  - File → Export → SSL Session Keys (1.6.x)
  - File → Export SSL Session Keys (1.8 / 1.10)

Let a third party decrypt the session without giving them the private key

Importing Session Keys



LAB: SSL Decryption

Questions

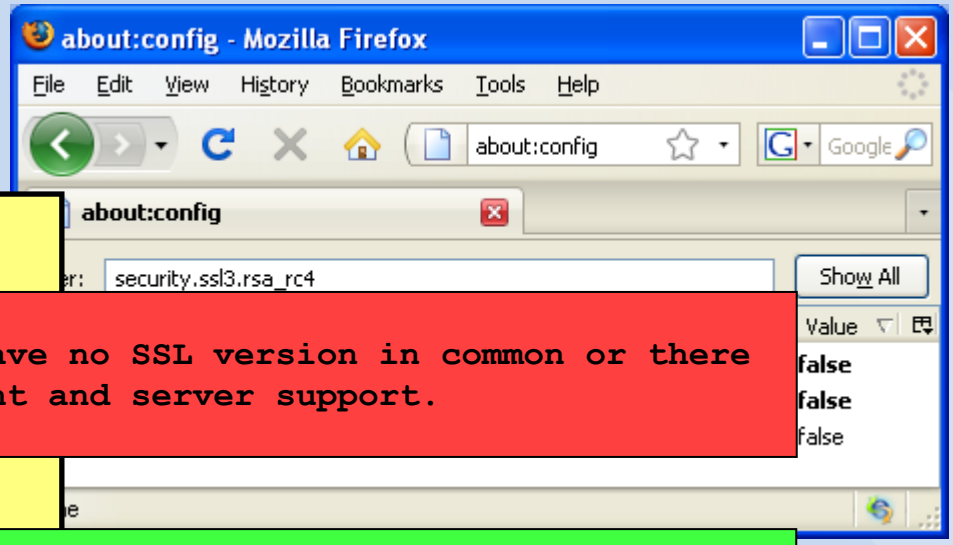
- Add the private key to Wireshark:
  - 192.168.3.3, 443, http, basic.pem
  - Can you see the encrypted messages, now?
- Are all the SSL sessions decrypted? Why?
- What are the contents of the page accessed?
- Does the HTTP Host header match the certificate's Common Name? Would the user have noticed?

Common Connection Problems

Common SSL problems I (1)

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.1	192.168.3.3	TCP	24269 > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
2	0.000667	192.168.3.3	192.168.3.1	TCP	https > 24269 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=4
3	0.000716	192.168.3.1	192.168.3.3	TCP	24269 > https [ACK] Seq=1 Ack=1 Win=128000 Len=0
4	0.020817	192.168.3.1	192.168.3.3	SSLv3	Client Hello
5	0.021173	192.168.3.3	192.168.3.1	TCP	https > 24269 [ACK] Seq=1 Ack=65 Win=5840 Len=0
6	0.024816	192.168.3.3	192.168.3.1	SSLv3	Alert (Level: Fatal, Description: Handshake Failure)
7	0.025488	192.168.3.3	192.168.3.1	TCP	https > 24269 [FIN, ACK] Seq=8 Ack=65 Win=5840 Len=0
8	0.025536	192.168.3.1	192.168.3.3	TCP	24269 > https [ACK] Seq=65 Ack=9 Win=127992 Len=0
9	0.031750	192.168.3.1	192.168.3.3	TCP	24269 > https [FIN, ACK] Seq=65 Ack=9 Win=127992 Len=0
10	0.032001	192.168.3.3	192.168.3.1	TCP	https > 24269 [ACK] Seq=9 Ack=66 Win=5840 Len=0

Common SSL problems I (2)



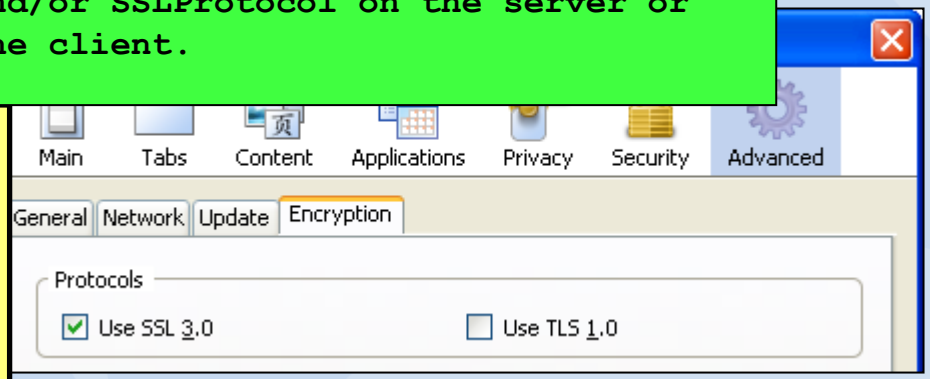
In apache2:

SSLCi

RC4+RSA

The client and the server have no SSL version in common or there is no cipher that both client and server support.

Reconfigure SSLCipherSuite and/or SSLProtocol on the server or adjust the SSL settings on the client.

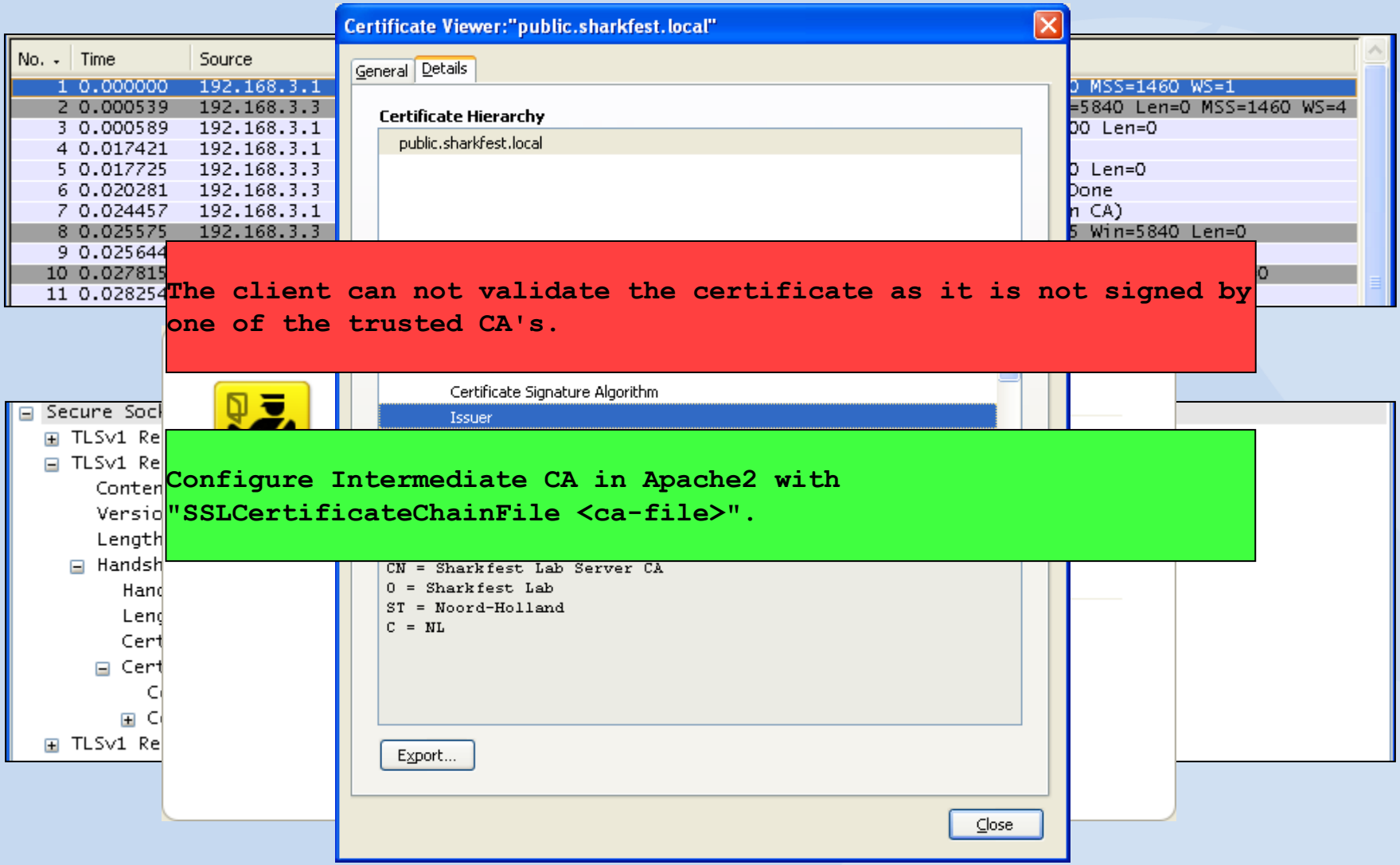


In apache2:

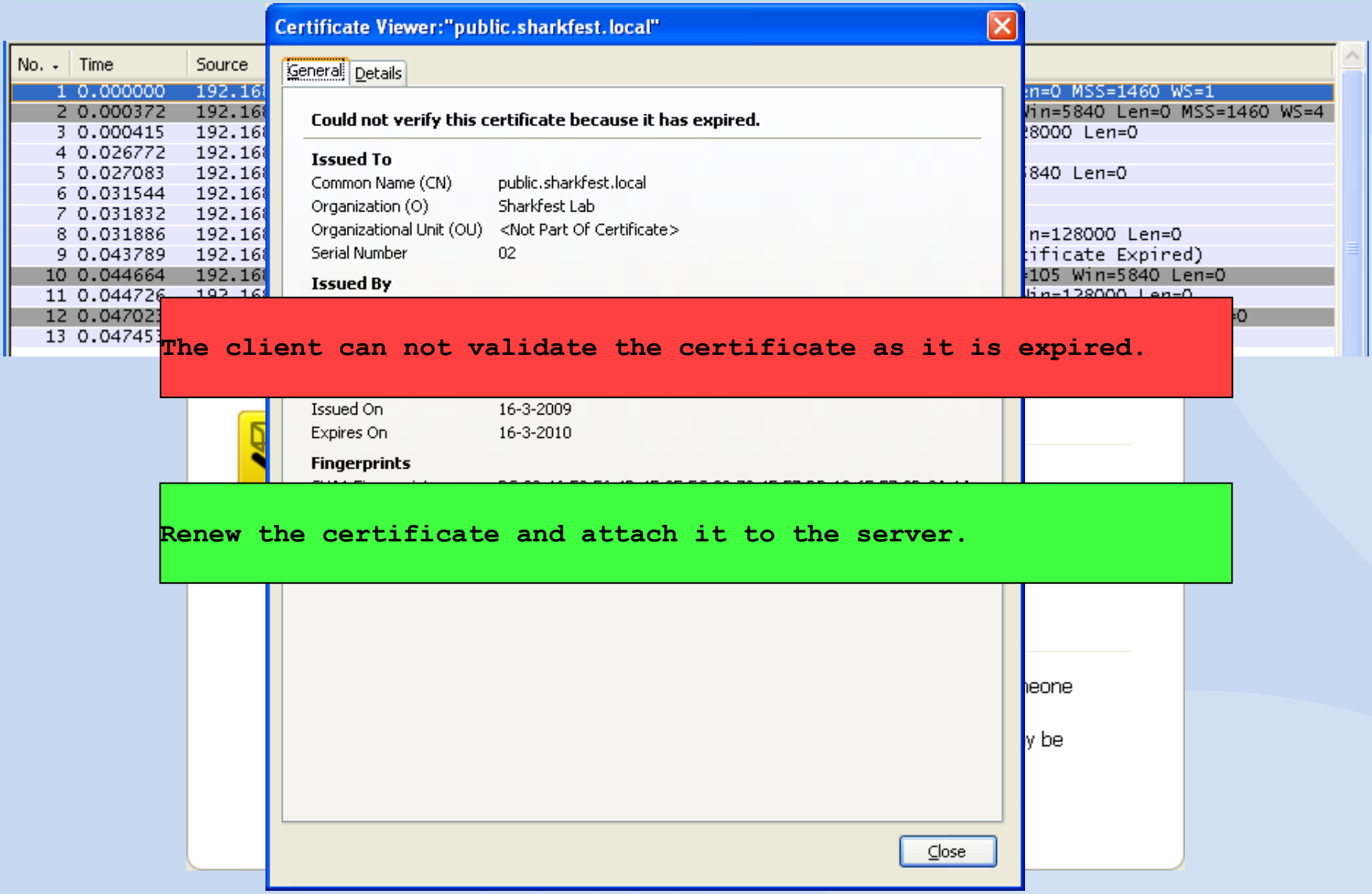
SSLProtocol

TLSv1

Common SSL Problems II



Common SSL Problems III (1)



Common SSL Problems III (2)

```
Secure Socket Layer
  TLSv1 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 92
    Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 88
        Version: TLS 1.0 (0x0301)
      Random
          gmt_
          rand
        Sessio
        Cipher
      Cipher
          Cipher Suite: TLS_RSA_WITH_CAMELLIA_256_
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_
          Ciph
          Ciph
          Ciph
        Compression Methods Length: 1
      Compression Methods (1 method)
        Extensions Length: 35
      Extension: server_name
      Extension: SessionTicket TLS
```

```
Secure Socket Layer
  TLSv1 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Random
          gmt_unix_time: May 21, 2009 15:49:33.000000000
          random_bytes: E036DED536B73FC46F947D99AD604196CEACA680E42F3083
                                                                  999F8A87...
```

The client can not validate the certificate as it's clock is not set correctly.

Set the correct time on the client.

Common SSL Problems IV

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.1	192.168.3.3	TCP	28051 > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
2	0.000264	192.168.3.3	192.168.3.1	TCP	https > 28051 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=4
3	0.000304				
4	0.019417				
5	0.019790				
6	0.025173				
7	0.025326				
8	0.025376				
9	0.055480				Len=0
10	0.056264				
11	0.056306				

Secure Connection Failed

www.sharkfest.local uses an invalid security certificate.

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.

The client can not validate the certificate as the common name in the certificate does not match the hostname.

Make sure the site you are trying to visit is indeed the site you intended to visit.

```
Secure Sock
  TLSv1 Recor
    Content T
    Versio
    Length
    Handsh
      Hand
      Leng
      Certif
      Certificates (2325 bytes)
        Certificate Length: 1079
        Certificate ()
        Certificate Length: 1240
        Certificate ()
  TLSv1 Record Layer: Handshake Protocol: Server Hello Done
```

Common SSL Problems V (1)

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.1	192.168.3.4	TCP	30245 > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
2	0.000289	192.168.3.4	192.168.3.1	TCP	https > 30245 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=4
3	0.000314	192.168.3.1	192.168.3.4	TCP	30245 > https [ACK] Seq=1 Ack=1 Win=128000 Len=0
4	0.017233	192.168.3.1	192.168.3.4	SSL	Client Hello
5	0.017657	192.168.3.4	192.168.3.1	TCP	https > 30245 [ACK] Seq=1 Ack=99 Win=5840 Len=0
6	0.019863	192.168.3.4	192.168.3.1	TLSv1	Server Hello,
7	0.019939	192.168.3.4	192.168.3.1	TLSv1	Certificate, Certificate Request, Server Hello Done
8	0.019966	192.168.3.1	192.168.3.4	TCP	30245 > https [ACK] Seq=99 Ack=2572 Win=128000 Len=0
9	3.299274	192.168.3.1	192.168.3.4	TLSv1	Certificate, Client Key Exchange, Certificate Verify, Change Ciph
10	3.300415	192.168.3.4	192.168.3.1	TLSv1	Alert (Level: Fatal, Description: Unknown CA)
11	3.300763	192.168.3.4	192.168.3.1	TCP	https > 30245 [FIN, ACK] Seq=2579 Ack=1501 Win=8768 Len=0
12	3.300791	192.168.3.1	192.168.3.4	TCP	30245 > https [ACK] Seq=1501 Ack=2580 Win=127992 Len=0
13	3.310232	192.168.3.1	192.168.3.4	TCP	30245 > https [FIN, ACK] Seq=1501 Ack=2580 Win=127992 Len=0
14	3.310386	192.168.3.4	192.168.3.1	TCP	https > 30245 [ACK] Seq=2580 Ack=1502 Win=8768 Len=0

Secure Connection Failed

An error occurred during a connection to private.sharkfest.local.

Peer does not recognize and trust the CA that issued your certificate.

(Error code: ssl_error_unknown_ca_alert)

The page you are trying to view can not be shown because the authenticity of the received data could not be verified.

- Please contact the web site owners to inform them of this problem.

Try Again

Common SSL Problems V (2)

```
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Certificate
  ⊟ TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 149
    ⊟ Handshake Protocol: Certificate Request
        Handshake Type: Certificate Request (13)
        Length: 141
        Certificate types count: 3
    ⊞ Cert
      Dist
    ⊟ Dist
        Di
      ⊟ Di
        ⊞ RDNSequence: 1 item ()
        ⊞ RDNSequence: 1 item ()
        ⊞
        ⊟
                printableString: Sharkfest Lab Client CA
        ⊞ RDNSequence: 1 item ()
  ⊞ Handshake Protocol: Server Hello Done
```

The server can not validate the client certificate as it does not have the Root CA configured.

Add the Root Ca to the certificate bundle that is pointed to by "SSLCACertificateFile <trusted-ca-bundle>".

```
[Thu May 21 10:29:45 2009] [error] Certificate Verification: Error (2): unable to
get issuer certificate
```

Common SSL Problems VI

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.1	192.168.3.4	TCP	30824 > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
2	0.000178	192.168.3.4	192.168.3.1	TCP	https > 30824 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=4
3	0.000214	192.168.3.1	192.168.3.4	TCP	30824 > https [ACK] Seq=1 Ack=1 Win=128000 Len=0
4	0.008474	192.168.3.1	192.168.3.4	TLSv1	Client Hello
5	0.008656	192.168.3.4	192.168.3.1	TCP	https > 30824 [ACK] Seq=1 Ack=99 Win=5840 Len=0
6	0.010907	192.168.3.4	192.168.3.1	TLSv1	Server Hello,
7	0.011001	192.168.3.4	192.168.3.1	TLSv1	Certificate, Certificate Request, Server Hello Done
8	0.011040	192.168.3.1	192.168.3.4	TCP	30824 > https [ACK] Seq=99 Ack=2726 Win=128000 Len=0
9	3.441257	192.168.3.1	192.168.3.4	TLSv1	Certificate, Client Key Exchange, Certificate Verify, Change Ciph
10	3.443320	192.168.3.4	192.168.3.1	TLSv1	Alert (Level: Fatal, Description: Certificate Unknown)
11	3.443762				
12	3.443796				
13	3.445834				=0
14	3.445982				

The server can not validate the client certificate as the CA chain used is larger than the allowed depth.

Secure Connection Failed

Configure the correct CA verify depth in Apache2 with "SSLCertificateChainFile <ca-file>".

(Error code: ssl_error_certificate_unknown_alert)

The page you are trying to view can not be shown because the authenticity of

```
[Thu May 21 10:38:30 2009] [error] Certificate Verification: Certificate Chain too long (chain has 2 certificates, but maximum allowed are only 1)
```

Try Again

Common SSL Problems VII

```
Secure Socket Layer
  TLSv1 Record Layer: Handshake Protocol: Certificate
  TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 168
    Handshake Protocol: Certificate Request
      Distinguished Names (154 bytes)
        Distinguished Name Length: 152
                printableString: Sharkfest Lab Root CA
        RDNSequence: 1 item ()
        RDNSequence: 1 item ()
        RDNSequence: 1 item ()
        RDNSequence: 1 item ()
        RDNSequence: 1 item ()
    Handshake Protocol: Server Hello Done
  TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

The client did not send a certificate as it could not find one that was signed by the presented CA's.

Make sure the client has the Intermediate CA in it's certificate store, so it can find a matching certificate.

Common SSL Problems VIII

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.1	192.168.3.4	TCP	32123 > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
2	0.000085	192.168.3.4	192.168.3.1	TCP	https > 32123 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=4
3	0.000108	192.168.3.1	192.168.3.4	TCP	32123 > https [ACK] Seq=1 Ack=1 Win=128000 Len=0
4	0.009474	192.168.3.1	192.168.3.4	SSL	Client Hello
5	0.009680	192.168.3.4	192.168.3.1	TCP	https > 32123 [ACK] Seq=1 Ack=99 Win=5840 Len=0
6	0.011632	192.168.3.4	192.168.3.1	TLSv1	Server Hello,
7	0.011719	192.168.3.4	192.168.3.1	TLSv1	Certificate, Certificate Request, Server Hello Done
8	0.011744	192.168.3.1	192.168.3.4	TCP	32123 > https [ACK] Seq=99 Ack=2591 Win=128000 Len=0
9	5.275850	192.168.3.1	192.168.3.4	TCP	[TCP segment of a reassembled PDU]
10	5.275889	192.168.3.1	192.168.3.4	TLSv1	Certificate, Client Key Exchange, Certificate Verify, Change Ciph
11	5.276312				
12	5.283642				
13	5.2844				0
14	5.2844				
15	5.2874				=0
16	5.2880				

The server rejected the client certificate because it has been revoked by the signing CA.

An error occurred during a connection to private.sharkfest.local.

The client needs to request a new certificate.

The page you are trying to view can not be shown because the authenticity of the received data could not be verified.

Try Again

```
[Thu May 21 10:57:57 2009] [error] Certificate Verification: Error (23): certificate revoked
```

Common SSL Problems IX

```
Certificates (2306 bytes)
  Certificate Length: 1060
  Certificate ()
    signedCertificate
      version: v3 (2)
      serialNumber: 1
      signature (shaWithRSAEncryption)
      issuer: rdnSequence (0)
      validity
        notBefore: utcTime (0)
```

The CRL file on the server is expired. This results in revoking all certificates until the CRL is updated.

```
      extensions: 4 items
    algorithmIdentifier (shaWithRSAEncryption)
    Padding: 0
```

Make sure the CRL file pointed to by "SSLCARevocationFile <crl-file>" stays up to date.

```
      serialNumber: 2
      signature (shaWithRSAEncryption)
```

```
[Thu May 21 11:01:15 2009] [warn] Found CRL is expired - revoking all certificates
until you get updated CRL
[Thu May 21 11:01:15 2009] [error] Certificate Verification: Error (12): CRL has
expired
```

```
          utcTime: 10-03-15 23:03:14 (UTC)
      subject: rdnSequence (0)
```

Questions

- Use tricky.pcap and tricky.pem
- Can you decrypt all the sessions? Why?
  - Could an attacker change the session to be decrypted if they had the private key? Why?
  - Could an attacker man-in-the-middle any session?
- How many certificates did the client send?
  - How many certificates should the client have sent?
- What is the name of the user that is connecting?
  - Can eavesdroppers see who is connecting, too?

Conclusion

The moral

Don’t trust that SSL is working how you intended to configure it until you’ve verified it by reading a packet capture.

Suggested tools

1. OpenSSL
   - x509, req, s_client, s_server
2. Ncat
   - What netcat wants to be when it grows up
3. Stunnel
   - Great for creating servers with specific configurations to test clients against
4. Nessus
   - Easy way to test lots of common problems

Questions?

Anything you want to discuss?

Thank you.