SSL Troubleshooting with Wireshark and Tshark

Sake Blok

Application Delivery Networking Consultant and Troubleshooter

sake.blok@SYN-bit.nl
Fokke & Sukke
Sluiten de maand van de filosofie af

Geloof jij in de cloud?

Nou, mwah...

Ik geloof wel dat er "iets" is
Columbia Internet Tech Support. Greg speaking, please state the nature of your problem.

Hello. How secure is my e-mail?

Very secure, m'am. Why?

Well, I'm a centerfold model and my husband is overseas and we send each other very steamy letters. My privacy is important to me.

We're the only ones who can get to your mail.

Phew. Thank you. That's a relief.

Say, what's your account number?

Copyright (c) Illiad 1997, 1998
About you?

• Who…

...thinks SSL is just about encryption?  
...troubleshooted SSL traffic before?  
...knows the purpose of each handshake message?  
...tried to decrypt SSL traffic before?  
...and ran into problems decrypting?  
...troubleshooted client authentication problems?
About me?

• Started SYN-bit in 2009
• Application Delivery Networking Consultant & Troubleshooter (F5 Networks, Cisco ACE, Alteon)
• Have used SSL extensively in customer projects
• Using Ethereal since 1999, developing since 2006, member core-developers since 2007
• Enjoy scuba diving and art-house movies
Challenges

• Confidentiality
  – Encryption and Decryption

• Message Integrity
  – Message Digest and Message Signing

• Endpoint Authentication & Non-repudiation
  – Certificates and Certificate Authorities
Agenda

• Cryptology overview
• The SSL protocol
• Analyzing SSL with Wireshark
• Analyzing SSL with Tshark
• Common SSL connection problems
• Further reading
• Questions & Discussion
Agenda

• Cryptology overview
• The SSL protocol
• Analyzing SSL with Wireshark
• Analyzing SSL with Tshark
• Common SSL connection problems
• Further reading
• Questions & Discussion
Symmetric Encryption

- Same key for encryption and decryption
- Computationally "cheap"
- Short keys (typically 40-256 bits)
- DES, 3DES, AESxxx, RC4

- Confidentiality?
- One-to-many?
Asymmetric Encryption

• One key for encryption, second key for decryption (both keys form a pair)
• Computatively "expensive"
• Long keys (typically 512-4096 bits)
• RSA, DSA

• Confidentiality?
• Authentication?
• One-to-many?
Hashing / Message Digest

• Irreversible
  — original text not reproducible from the digest
• Collision-resistance
  — "Not possible" to create a message M' so that it has the same digest as message M
• MD5, SHA-1, SHA-2
Message Signing

- Create digest of message
- Encrypt digest with private key
- Authenticity and sender of message can be checked with public key

4fe7ad41 = 4fe7ad41
3e7bc46a = 3e7bc46a
Digital Certificates

"In cryptography, a public key certificate (or identity certificate) is an **electronic document** which utilizes a **digital signature** to bind together a **public key** with an identity."


But who is signing???
Certificate Authorities

- Mutually trusted by sender and receiver
- "Solves" key exchange problems
- CA's can be chained
- Top of chain is "self-signed" (and is called the "Root CA")
Creating a certificate
Agenda

- Cryptology overview
- The SSL protocol
- Analyzing SSL with Wireshark
- Analyzing SSL with Tshark
- Common SSL connection problems
- Further reading
- Questions & Discussion
SSL History

- SSLv1 by Netscape (unreleased, 1994)
- SSLv2 by Netscape (v2-draft, 1994)
- SSLv3 by Netscape (v3-draft, 1995)
- TLSv1.0, IETF (RFC 2246, 1999)
- TLSv1.1, IETF (RFC 4346, 2006)
- TLSv1.2, IETF (RFC 5246, 2008)
- Risks and differences explained at:
Place in TCP/IP stack

- Between transport and application layer
- Protocol independent

<table>
<thead>
<tr>
<th>HTTP</th>
<th>SMTP</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL/TLS</td>
<td>TCP</td>
<td>IP</td>
</tr>
</tbody>
</table>

SSL record layer

- handshake
- change cipherspec
- application data
- alert
SSL Record Layer

- Provides fragmentation (max size $2^{14}$)
- Multiple SSL messages (of one content type) per SSL Record allowed

- SSL Record can be split over multiple TCP-segments ($2^{14} >$ MSS!)
- One TCP-segment can contain multiple SSL Records (or fragments)
SSL Content Types

• Handshake Protocol (0x16)
  – responsible for authentication and session key setup
• ChangeCipherSpec Protocol (0x14)
  – Notify start of encryption
• Alert Protocol (0x15)
  – Reporting of warnings and fatal errors
• Application Protocol (0x17)
  – Actual encryption and transport of data
Agenda

• Cryptology overview
• The SSL protocol
• Analyzing SSL with Wireshark
• Analyzing SSL with Tshark
• Common SSL connection problems
• Further reading
• Questions & Discussion
Lab setup

Client
WinXP - Firefox
Client-cert 1: Neo
Client-cert 2: Trinity

192.168.3.0/24

(virtual) Server 1
Ubuntu - Apache
Server-cert: public.sharkfest.local

(virtual) Server 2
Ubuntu - Apache
Server-cert: private.sharkfest.local
Require Client-Certificate

Sharkfest Lab Root CA

Sharkfest Lab Server CA

CN:public.sharkfest.local
CN:private.sharkfest.local

Sharkfest Lab Client CA

CN:Neo
CN:Trinity
Choosing the right settings

- **ip.defragment**: TRUE
- **tcp.check_checksum**: FALSE
- **tcp.desegment_tcp_streams**: TRUE
- **ssl.desegment_ssl_records**: TRUE
- **ssl.desegment_ssl_application_data**: TRUE
Analyzing the SSL record layer (1)

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.011511</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>SSL</td>
<td>Client Hello</td>
</tr>
<tr>
<td>5</td>
<td>0.011876</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 18736 [ACK] Seq=1 Ack=71 Win=5840 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>0.017431</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Server Hello,</td>
</tr>
<tr>
<td>7</td>
<td>0.017782</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TLSv1</td>
<td>Certificate, Server Hello Done</td>
</tr>
<tr>
<td>8</td>
<td>0.017890</td>
<td>192.168.3.3</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>18736 &gt; https [ACK] Seq=71 Ack=2426 Win=128000 Len=0</td>
</tr>
</tbody>
</table>

Frame 4 (124 bytes on wire, 124 bytes captured)

- Ethernet II, Src: Vmware_c0:00:01 (00:50:56:c0:00:01), Dst: Vmware_s5d:c5:66 (00:0c:29:5d:c5:66)
- Internet Protocol, Src: 192.168.3.1 (192.168.3.1), Dst: 192.168.3.3 (192.168.3.3)
- Transmission Control Protocol, Src Port: 18736 (18736), Dst Port: https (443), Seq: 1, Ack: 1, Len: 70

Secure Socket Layer

- TLsv1 Record Layer: Handshake Protocol: Client Hello

  - Content Type: Handshake (22)
  - Version: TLS 1.0 (0x0301)
  - Length: 65

  - Handshake Protocol: Client Hello

0000 00 0c 29 5d c5 66 00 50 56 c0 00 01 08 00 45 00
0010 00 6e 42 29 40 00 80 06 31 0c c0 a8 03 01 c0 a8
0020 03 03 49 30 01 bb 21 62 08 73 02 3e 54 89 50 18
0030 fa 00 67 eb 00 00 16 03 01 00 41 01 00 00 3d 03
0040 01 49 eb 46 9e dd 81 95 18 fc 5d dd d0 97 42 8d
0050 41 0d 78 52 e2 57 9e 2e 89 03 cd b3 31 c7 63 dc
0060 a9 00 00 10 00 84 00 35 00 41 00 04 00 05 00 2f
0070 fe ff 00 0a 01 00 00 04 00 23 00 00

File: "C:\cygwin\home\sablo\sharkfest\2009\traces\session-reuse.cap" 13 KB 00:02:17

Packets: 56 Displayed: 56 ...

woensdag 27 juni 12
Analyzing the SSL record layer (2)

- 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: 18736 (18736), Seq: 1, Ack: 71, Len: 1460
- Secure Socket Layer
  - TLSv1 Record Layer: Handshake Protocol: Server Hello

**0x091c = 2332 bytes**

---

woensdag 27 juni 12

Sake Blok

sake.blok@SYN-bit.nl
Analyzing the SSL record layer (3)

\[
(5+2332) + (5+4) = 2346
\]
Analyzing the SSL handshake

• Normal RSA handshake
• Ephemeral RSA (or DH) handshake
• SSL session with client authentication
• Reusing SSL sessions
  – Reused SSL session (partial handshake)
  – Expired SSL session
  – No SSL reuse
Normal RSA handshake

ClientHello

ServerHello

Certificate

ServerHelloDone

ClientKeyExchange

ChangeCipherSpec

Finished (encrypted)

ChangeCipherSpec

Finished (encrypted)
### ... in Wireshark

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>18736 &gt; https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1</td>
</tr>
<tr>
<td>2</td>
<td>0.000309</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 18736 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=1</td>
</tr>
<tr>
<td>3</td>
<td>0.000357</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>18736 &gt; https [ACK] Seq=1 Ack=1 Win=128000 Len=0</td>
</tr>
<tr>
<td>4</td>
<td>0.011511</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TLSv1</td>
<td>Client Hello</td>
</tr>
<tr>
<td>5</td>
<td>0.011876</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 18736 [ACK] Seq=1 Ack=71 Win=5840 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>0.017431</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Server Hello,</td>
</tr>
<tr>
<td>7</td>
<td>0.017782</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Certificate, Server Hello Done</td>
</tr>
<tr>
<td>8</td>
<td>0.017890</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>18736 &gt; https [ACK] Seq=71 Ack=2426 Win=128000 Len=0</td>
</tr>
<tr>
<td>9</td>
<td>0.026711</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TLSv1</td>
<td>Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message</td>
</tr>
<tr>
<td>10</td>
<td>0.038327</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Change Cipher Spec, Encrypted Handshake Message</td>
</tr>
</tbody>
</table>
ClientHello

Secure Socket Layer
  TLSv1 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 65
  Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 61
    Version: TLS 1.0 (0x0301)
  Random
    gmt_unix_time: Apr 19, 2009 17:43:26.000000000
    random_bytes: DD819516FC5DDD097428D410D7852E2579E28903CDB331...
    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 (0x0f00)
    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

Secure Socket Layer
  TLSv1 Record Layer: Handshake Protocol: Server Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 74
  Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 70
    Version: TLS 1.0 (0x0301)
  Random
    gmt_unix_time: Mar 16, 2009 02:30:23.000000000
    random_bytes: D6F69B9813144FDB2340A273F419E463BF915549B074DF...
    Session ID Length: 32
    Session ID: DB00C2A0D79CFDA109CE4F65A9801AA8D5F1BBEB9E1F848F...
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Compression Method: null (0)
Certificate (1)

Secure Socket 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 ()
  - Certificate Length: 1240
  - Certificate ()
- TLSv1 Record Layer: Handshake Protocol: Server Hello Done
Certificate (2)

Handshake Protocol: Certificate
Handshake Type: Certificate (11)
Length: 2328
Certificates Length: 2325
Certificates (2325 bytes)
Certificate Length: 1079

Certificate ()

signedCertificate

version: v3 (2)
serialNumber: 2
signature (shaWithRSAEncryption)
issuer: rdnSequence (0)
validity
subject: rdnSequence (0)
subjectPublicKeyInfo
extensions: 4 items
algorithmIdentifier (shaWithRSAEncryption)

Algorithm Id: 1.2.840.113549.1.1.5 (shaWithRSAEncryption)
Padding: 0
encrypted: 739D20C79873ADD406549E824AE1304525EEA1A5E185FB0B...

Certificate Length: 1240
Certificate ()
Certificate (3)

```
validity

subject: rdnSequence (0)

- rdnSequence: 5 items()
  - RDNSequence: 1 item()
    - RelativeDistinguishedName
      - Id: 2.5.4.6 (id-at-countryName)
        - CountryName: NL
    - RDNSequence: 1 item()
      - RelativeDistinguishedName
        - Id: 2.5.4.8 (id-at-stateOrProvinceName)
        - DirectoryString: printableString (1)
          - printableString: Noord-Holland
    - RDNSequence: 1 item()
      - RelativeDistinguishedName
        - Id: 2.5.4.10 (id-at-organizationName)
        - DirectoryString: printableString (1)
          - printableString: Sharkfest Lab
    - RDNSequence: 1 item()
      - RelativeDistinguishedName
        - Id: 2.5.4.3 (id-at-commonName)
        - DirectoryString: printableString (1)
          - printableString: public.sharkfest.local
  - RDNSequence: 1 item()
    - RelativeDistinguishedName
      - Id: 1.2.840.113549.1.9.1 (pkcs-9-at-emailAddress)
      - SyntaxIA5String: co@sharkfest.local

subjectPublicKeyInfo
```
Certificate (4)

```
signedCertificate
  version: v3 (2)
  serialNumber: 2
  signature (shaWithRSAEncryption)
  issuer: rdnSequence (0)
    rdnSequence: 5 items ()
      RDNSequence: 1 item ()
        RDNSequence: 1 item ()
        RDNSequence: 1 item ()
        RDNSequence: 1 item ()
        RDNSequence: 1 item ()
      RelativeDistinguishedName
        Id: 2.5.4.3 (id-at-commonName)
        DirectoryString: printableString (1)
        printableString: Sharkfest Lab Server CA
    RDNSequence: 1 item ()
  validity
  subject: rdnSequence (0)
  subjectPublicKeyInfo
  extensions: 4 items
  algorithmIdentifier (shaWithRSAEncryption)
    Padding: 0
    encrypted: 739D2OC79873ADD406549E824AE1304525EEA1A5E1B5FB0B...
Certificate Length: 1240
```
ServerHelloDone

- Secure Socket Layer
  - TLSv1 Record Layer: Handshake Protocol: Certificate
  - TLSv1 Record Layer: Handshake Protocol: Server Hello Done
    - Content Type: Handshake (22)
    - Version: TLS 1.0 (0x0301)
    - Length: 4
  - Handshake Protocol: Server Hello Done
    - Handshake Type: Server Hello Done (14)
    - Length: 0
Secure Socket Layer
- TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
  - Content Type: Handshake (22)
  - Version: TLS 1.0 (0x0301)
  - Length: 134
- Handshake Protocol: Client Key Exchange
  - Handshake Type: Client Key Exchange (16)
  - Length: 130
- TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
- TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
ChangeCipherSpec (C)

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 (C)

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

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
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 (S)

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 RSA (or DH) handshake

ClientHello

ServerHello

Certificate

ServerKeyExchange

ServerHelloDone

ClientKeyExchange

ChangeCipherSpec

Finished (encrypted)

ChangeCipherSpec

Finished (encrypted)

Client

Server
... in Wireshark

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>42370 &gt; https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1</td>
</tr>
<tr>
<td>2</td>
<td>0.000377</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 42370 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146</td>
</tr>
<tr>
<td>3</td>
<td>0.000618</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>42370 &gt; https [ACK] Seq=1 Ack=1 Win=12800 Len=0</td>
</tr>
<tr>
<td>4</td>
<td>0.026109</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>SSL</td>
<td>Client Hello</td>
</tr>
<tr>
<td>5</td>
<td>0.026465</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 42370 [ACK] Seq=1 Ack=107 Win=5840 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>0.070925</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Server Hello, Certificate, Server Key Exchange, Server Hello Done</td>
</tr>
<tr>
<td>7</td>
<td>0.071108</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Certificate, Server Key Exchange, Change Cipher Spec, Encrypted Handshak</td>
</tr>
<tr>
<td>8</td>
<td>0.071172</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>42370 &gt; https [ACK] Seq=107 Ack=2828 Win=128000 Len=0</td>
</tr>
<tr>
<td>9</td>
<td>0.090279</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TLSv1</td>
<td>Client Key Exchange, Change Cipher Spec, Encrypted Handshak</td>
</tr>
<tr>
<td>10</td>
<td>0.090657</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 42370 [ACK] Seq=2828 Ack=305 Win=6912 Len=0</td>
</tr>
<tr>
<td>11</td>
<td>0.110494</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Change Cipher Spec, Encrypted Handshake Message</td>
</tr>
</tbody>
</table>
ServerKeyExchange

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

Client

CertificateRequest

Certificate

CertificateVerify

ClientKeyExchange

ChangeCipherSpec

Finished (encrypted)

Server

ServerHelloDone

ServerHello

ClientHello

Certificate

ChangeCipherSpec

Finished (encrypted)
... in Wireshark

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>14980 &gt; https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1</td>
</tr>
<tr>
<td>2</td>
<td>0.000372</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 14980 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146</td>
</tr>
<tr>
<td>3</td>
<td>0.000400</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>14980 &gt; https [ACK] Seq=1 Ack=1 Win=128000 Len=0</td>
</tr>
<tr>
<td>4</td>
<td>0.015645</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>SSLv2</td>
<td>Client Hello</td>
</tr>
<tr>
<td>5</td>
<td>0.015824</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 14980 [ACK] Seq=1 Ack=52 Win=5840 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>0.017894</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>SSLv3</td>
<td>Server Hello</td>
</tr>
<tr>
<td>7</td>
<td>0.017988</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>SSLv3</td>
<td>Certificate, Certificate Request, Server Hello Done</td>
</tr>
<tr>
<td>8</td>
<td>0.018015</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>14980 &gt; https [ACK] Seq=52 Ack=2590 Win=128000 Len=0</td>
</tr>
<tr>
<td>9</td>
<td>4.089191</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>10</td>
<td>4.089622</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 14980 [ACK] Seq=2590 Ack=1512 Win=8788 Len=0</td>
</tr>
<tr>
<td>11</td>
<td>4.089949</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>SSLv3</td>
<td>Certificate, Client Key Exchange, Certificate Verify, Change Encryption Spec, Encrypted Handshake Message</td>
</tr>
</tbody>
</table>
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

Handshake Protocol: Server Hello Done
Certificate (C)

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
Reusing SSL sessions

• Key negotiation "expensive"
• Cache session keys and re-use for new TCP sessions
• SSL session ID is used as Index
• Timeout on SSL session ID is an "absolute timeout" not an "idle timeout"
  —Old IE: 2 minutes, now 10 hours
Handshake of a Reused Session

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

Inter-Process Session Cache:
Configure the SSL Session Cache: First the mechanism
to use and second the expiring timeout (in seconds).

# Inter-Process Session Cache:
# Configure the SSL Session Cache: First the mechanism
# to use and second the expiring timeout (in seconds).

#SSLSessionCache  dbm:/var/run/apache2/ssl_scache
SSLSessionCache  shmcb:/var/run/apache2/ssl_scache(512000)
SSLSessionCacheTimeout  60
No SSL session reuse

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

Frame 6 (1514 bytes on wire, 1514 bytes captured)
Ethernet II, Src: 00:0C:29:5d:c5:66 (00:0C:29:5d:c5:66), Dst: 00:50:56: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)
    Version: TLS 1.0 (0x0301)
    Length: 38
    Random
  Session ID Length: 0
  Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
  Compression Method: null (0)
TLS session Tickets (1)

- TLS session tickets (RFC 5077)
- Do not keep state on server, only on client
- TLS extension in ClientHello and ServerHello
- New SSL HandshakeType: **NewSessionTicket**
TLS session Tickets (2)
TLS session Tickets (3)

- 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_SHA
      - Compression Method: null (0)
      - Extensions Length: 13
        - Extension: server_name
        - Extension: renegotiation_info
        - Extension: SessionTicket TLS

- TLSv1 Record Layer: Handshake Protocol
  - 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
      - 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)
Analyzing SSL Alerts

Without decryption:

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Version</th>
<th>Content Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>12.494568</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>TLSv1</td>
<td>Application Data</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>12.495834</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>TLSv1</td>
<td>Application Data, Application Data</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>27.530927</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>TLSv1</td>
<td>Encrypted Alert</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>32.811207</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>TLSv1</td>
<td>Encrypted Alert</td>
<td></td>
</tr>
</tbody>
</table>

```
Secure Socket Layer
  TLSv1 Record Layer: Encrypted Alert
    Content Type: Alert (21)
    Version: TLS 1.0 (0x301)
    Length: 32
    Alert Message: Encrypted Alert
```

With decryption:

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Version</th>
<th>Content Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>12.494568</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>TLSv1</td>
<td>HTTP</td>
<td>GET / HTTP/1.1</td>
</tr>
<tr>
<td>15</td>
<td>12.495834</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>TLSv1</td>
<td>HTTP</td>
<td>HTTP/1.1 200 OK (text/html)</td>
</tr>
<tr>
<td>17</td>
<td>27.530927</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>TLSv1</td>
<td>Alert</td>
<td>Level: Warning, Description: Close Notify</td>
</tr>
<tr>
<td>20</td>
<td>32.811207</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>TLSv1</td>
<td>Alert</td>
<td>Level: Warning, Description: Close Notify</td>
</tr>
</tbody>
</table>

```
Secure Socket Layer
  TLSv1 Record Layer: Alert (Level: Warning, Description: Close Notify)
    Content Type: Alert (21)
    Version: TLS 1.0 (0x301)
    Length: 32
    Alert Message
      Level: Warning (1)
      Description: Close Notify (0)
```
Analyzing SSL Application Data

<table>
<thead>
<tr>
<th>Time</th>
<th>Sequence</th>
<th>IP Address</th>
<th>Port</th>
<th>Protocol</th>
<th>Length</th>
<th>Type Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0.040173</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TLSv1</td>
<td>491</td>
<td>Application Data</td>
</tr>
<tr>
<td>12</td>
<td>0.042446</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>496</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>14</td>
<td>12.494568</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TLSv1</td>
<td>491</td>
<td>Application Data</td>
</tr>
<tr>
<td>15</td>
<td>12.495834</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>496</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>29</td>
<td>39.717354</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TLSv1</td>
<td>550</td>
<td>Change Cipher Spec, Encrypted Handshake Message, Application Data</td>
</tr>
<tr>
<td>30</td>
<td>39.720262</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>496</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>48</td>
<td>111.230987</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TLSv1</td>
<td>491</td>
<td>Application Data</td>
</tr>
<tr>
<td>49</td>
<td>111.233419</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>496</td>
<td>Application Data, Application Data</td>
</tr>
</tbody>
</table>

- Secure Sockets Layer
  - TLSv1 Record Layer: Application Data Protocol: http
    - Content Type: Application Data (23)
    - Version: TLS 1.0 (0x0301)
    - Length: 384
    - Encrypted Application Data: 94c662e11c5c01813955dfc675754583ab4a70d65fdff8e9...
  - TLSv1 Record Layer: Application Data Protocol: http
    - Content Type: Application Data (23)
    - Version: TLS 1.0 (0x0301)
    - Length: 48
    - Encrypted Application Data: 635e2a228ddc1aa5d7a2a89c809e6e699ec01f4cf5746fee...
Decrypting SSL traffic

- Provide **server** private key to Wireshark
- Only works when whole session (including **full handshake**) is in the tracefile
- Does **not** work with Ephemeral RSA or DH ciphers (ServerKeyExchange present)
- **Does** works with Client Authentication
Providing the server private key (1)

Wireshark preferences file:

- ssl.keys_list: 192.168.3.3,443,http,c:\key.pem
- ssl.debug_file: c:\temp\ssl-debug.log

When using Tshark:

tshark -r file.cap -o ssl.keys_list:192.168.3.3,443,http,"c:\key.pem" \
-o ssl.debug_file:"c:\ssl-debug.log" -V -R http
Providing the server private key (2)

$ cat ~/.wireshark/ssl_keys
# This file is automatically generated, DO NOT MODIFY.
"192.168.3.3","443","http","/Users/sake/tmp/ssl-deflate.key",""
$

$ cat ~/.wireshark/ssl_keys
# This file is automatically generated, DO NOT MODIFY.
"192.168.3.3","443","http","/Users/sake/tmp/ssl-deflate.key",""
Providing the server private key (3)

SSL debug log:

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)'
ssl_load_key: can't import pem data

PEM keyfile *without* passphrase:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDrHdbb+yGE6mEZ03bXURpZCjch2H6g97YAkJVGvmjZFFettBASA
EyA8vYXywaf8KbPzEsks2CzdA9MnU2H6DjzqvOa5WfeXMAr4OsCOpauStpreg7
q1h8ioqy+f4K1jRrrWP1h1QW1A8gtS1q137pyUUhW+7afwHwm2zjGIC1SwIDAQAB
AoGBAMneA9U6KJzjb+JUg/99c7h9W6wEvTYHTXjfl6paWA+hpuQ82E65/2Jdssl6
...b6QRMh1er5wd6smQ+CmhOEnqQyT5AIww12RIr9GbfIpTbtbRQw/EcQoC9wF1Efo
tOGpEF7zrkH+DpJqRI9kEA72gdyXKgPfHOS3rIQ08ScI8mBvuS8CBA4cuUIX3I/
No93a8v9V]'#7/yZm4xvB8o5z2YyBepawHVIv26jDD0aXw==
-----END RSA PRIVATE KEY-----
```

PEM keyfile *with* passphrase:

```
-----BEGIN RSA PRIVATE KEY-----
FR2cmkkkFHH45Dcsty1qDiIUy/uXn+9m/xeQMVrxtiSAmBmnUDUFIFCDDIc9yif
EROk2jPr2BzAaz15RbXS2TY+/7x0/dHD1lsF3LnJuOsNruo77TERxqgzO1O1VDRA
...ygw5Js1xqIN18F36E/cEP5rKvVvYfEPMa6IsiRhf2k1jLAu2ihWve7JodDF+6RKV
yBXRk/bDd8ih+bOnYu+2DrjAzn29GhggCW4QMnoDpTxxrYPkJ5Nw==
-----END RSA PRIVATE KEY-----
```
Converting keys

Removing passphrase:

```
root@mgmt# openssl rsa -in encrypted.key -out cleartext.key
Enter pass phrase for encrypted.key: <passphrase>
writing RSA key
root@mgmt#
```

Converting from DER to PEM (and removing passphrase):

```
root@mgmt# openssl rsa -inform DER -in der.key -out pem.key
Enter pass phrase for encrypted.key: <passphrase>
writing RSA key
root@mgmt#
```

Converting from PEM to PKCS12 (and adding passphrase):

```
root@mgmt# openssl pkcs12 -in pem.cert -inkey pem.key -export -out cert.pkcs12
Enter Export Password: <new-passphrase>
Verifying - Enter Export Password: <new-passphrase>
root@mgmt#
```
Decryption in Action

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.03825</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Change Cipher Spec, Finished</td>
</tr>
<tr>
<td>11</td>
<td>0.040173</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>HTTP</td>
<td>GET / HTTP/1.1</td>
</tr>
<tr>
<td>12</td>
<td>0.042446</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>HTTP</td>
<td>HTTP/1.1 200 OK (text/html)</td>
</tr>
<tr>
<td>13</td>
<td>0.238304</td>
<td>10.168.2.1</td>
<td>10.168.2.3</td>
<td>TCP</td>
<td>18736 &gt; <a href="https://www.mack.nl/seq-706">https://www.mack.nl/seq-706</a>, Ack-2077, win-127498, len-0</td>
</tr>
</tbody>
</table>

Frame 11 (491 bytes on wire, 491 bytes captured)
- Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_5d:c5:66 (00:0c:29:5d:c5:66)
- Internet Protocol, Src: 192.168.3.1 (192.168.3.1), Dst: 192.168.3.3 (192.168.3.3)

Secure Socket Layer
- TLSv1 Record Layer: Application Data Protocol: http
  Content Type: Application Data (23)
  Version: TLS 1.0 (0x0301)
  Length: 432
  Encrypted Application Data: C0D1C49A5E8119FC1B21EF547592476DF61AA48A11C44522...

Hypertext Transfer Protocol
- GET / HTTP/1.1
  Host: 192.168.3.3
  User-Agent: Mozilla/5.0 (windows; u; windows NT 5.1; en-us; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
  Accept-Language: en-us,en;q=0.5
  Accept-Encoding: gzip,deflate
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
  Keep-Alive: 300
  Connection: keep-alive
  Pragma: no-cache
  Cache-Control: no-cache
Decrypting IMAPS

ssl.keys_list: 192.168.1.20,993,imap,C:\key.pem

* OK Dovecot ready.
  Response Tag: *
  Response: OK Dovecot ready.
Decryption "STARTTLS" (1)

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.021653</td>
<td>192.168.1.20</td>
<td>192.168.1.46</td>
<td>SMTP</td>
<td>S: 220 brutus.netcc.local ESMTP Postfix (Ubuntu)</td>
</tr>
<tr>
<td>5</td>
<td>0.023320</td>
<td>192.168.1.46</td>
<td>192.168.1.20</td>
<td>SMTP</td>
<td>C: EHLO HELO93J</td>
</tr>
<tr>
<td>7</td>
<td>0.025077</td>
<td>192.168.1.20</td>
<td>192.168.1.46</td>
<td>SMTP</td>
<td>S: 250-brutus.netcc.local</td>
</tr>
<tr>
<td>8</td>
<td>0.025868</td>
<td>192.168.1.46</td>
<td>192.168.1.20</td>
<td>SMTP</td>
<td>C: STARTTLS</td>
</tr>
<tr>
<td>9</td>
<td>0.027373</td>
<td>192.168.1.20</td>
<td>192.168.1.46</td>
<td>SMTP</td>
<td>S: 220 2.0.0 Ready to start TLS</td>
</tr>
<tr>
<td>11</td>
<td>0.262273</td>
<td>192.168.1.46</td>
<td>192.168.1.20</td>
<td>TLSv1</td>
<td>Client Hello</td>
</tr>
<tr>
<td>12</td>
<td>0.264832</td>
<td>192.168.1.20</td>
<td>192.168.1.46</td>
<td>TLSv1</td>
<td>Server Hello, Certificate, Server Hello Done</td>
</tr>
<tr>
<td>13</td>
<td>0.266373</td>
<td>192.168.1.46</td>
<td>192.168.1.20</td>
<td>TLSv1</td>
<td>Client Key Exchange, Change Cipher Spec, Encrypted Handshake Mess.</td>
</tr>
<tr>
<td>14</td>
<td>0.281296</td>
<td>192.168.1.20</td>
<td>192.168.1.46</td>
<td>TLSv1</td>
<td>Change Cipher Spec, Encrypted Handshake Message</td>
</tr>
</tbody>
</table>

Frame 13 (236 bytes on wire, 236 bytes captured)
- Ethernet II, Src: IntelCor_61:3a:ad (00:1c:bf:61:3a:ad), Dst: Juniper
- 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)

**ssl.keys_list**: 192.168.1.20, 25, smtp, C:\key.pem

Frame 13 (236 bytes on wire, 236 bytes captured)
- Ethernet II, Src: IntelCor_61:3a:4d (00:14:28:04:0c:7a)

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
Decrypting "STARTTLS" (3)

Frame 13 (236 bytes on wire, 236 bytes captured)
ssl.keys_list: 192.168.1.20,start_tls,smtp,C:\key.pem
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 (Ox0301)
  Length: 32
  Handshake Protocol: Finished
    Handshake Type: Finished (20)
    Length: 12
    Verify Data
Decrypt-problem I (1)

Frame 7 (550 bytes on wire, 550 bytes captured)
Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_5d:c5:66 (00:0c:29:5d:c5:66)
Internet Protocol, Src: 192.168.3.1 (192.168.3.1), Dst: 192.168.3.3 (192.168.3.3)
Transmission Control Protocol, Src Port: 18774 (18774), Dst Port: 443 (443), Seq: 103, Ack: 139, Len: 496

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
      - TLSv1 Record Layer: Application Data Protocol: http

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 B8:2B:EA:B8:F8:BD:62:50:E3:0C:2D:3D:06:09:91:64:...
ssl_init private key file c:\temp\public.sharkfest.local.key successfully loaded
association_add TCP port 443 protocol http handle 04086228
Decrypt-problem 1 (2)

Checking ssl debug log:

```
[...] 
dissect_ssl enter frame #7 (first time)
    conversation = 07411870, ssl_session = 07411BC8
    record: offset = 0, reported_length_remaining = 496
  dissect_ssl3_record: content_type 20 
  dissect_ssl3_change_cipher_spec
  association_find: TCP port 18774 found 00000000
  packet_from_server: is from server - FALSE
  ssl_change_cipher CLIENT
    record: offset = 6, reported_length_remaining = 490
  dissect_ssl3_record: content_type 22 
  decrypt_ssl3_record: app_data len 48 ssl, state 0x17
  association_find: TCP port 18774 found 00000000
  packet_from_server: is from server - FALSE
  decrypt_ssl3_record: using client decoder
  decrypt_ssl3_record: no decoder available
  association_find: TCP port 18774 found 00000000
  "association_find: TCP port 443 found 047AF518"
[...] 
```

Make sure that the whole SSL session (which can be made out of multiple TCP streams) is in the tracefile. Starting with the handshake and up to the current frame.
Decrypt-problem II (1)

Checking ssl debug log:

```plaintext
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)'
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 II (2)
Decrypt-problem II (3)

In wireshark preferences:

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

Checking whether certificate and key match:

```plaintext
$ 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 a29682af822b4cd064d39d4ccd1e0e6c
$
$ openssl ce7158d3851a885314c264863142389
$
$ openssl rsa -noout -modulus -in private.sharkfest.local.key | openssl md5
  a29682af822b4cd064d39d4ccd1e0e6c
$
```

Make sure that the private key matches the (server) certificate that is used in the tracefile.

woensdag 27 juni 12
Decryption Without The Private Key

• Using the unencrypted (pre-)Master-Secret
• First use: Debug version of Firefox/Chrome (see: https://bugs.wireshark.org/bugzilla/show_bug.cgi?id=4349)
• Second use: openssl s_client

• SSL preferences: (Pre)-Master-Secret log filename
openssl s_client

$ openssl s_client -cipher AES256-SHA -no_ticket -connect imap.syn-bit.nl:993 | tee openssl-s_client.txt
depth=1 C = GB, ST = Greater Manchester, L = Salford, O = COMODO CA Limited, CN = COMODO High-Assurance Secure Server CA
verify error:num=20:unable to get local issuer certificate
verify return:0
CONNECTED(00000003)
---
[...]
SSL-Session:
Protocol  : TLSv1
Cipher    : AES256-SHA
Session-ID: 5EF3E7EDCC46993E51935914ACC1CBE6723259121248F958BC223D54FA84CFA0 Master-Key:
0665121ADB266864CDEF89E32A6F1A39677D540DB5B362BC351D3B08EE3059800F9A2186601710CE774AFB2CE3166C9
Session-ID-ctx:
Master-Key: 0665121ADB266864CDEF89E32A6F1A39677D540DB5B362BC351D3B08EE3059800F9A2186601710CE774AFB2CE3166C9
Key-Arg   : None
PSK identity: None
PSK identity hint: None
SRP username: None
Compression: 1 (zlib compression)
Start Time: 1339857082
Timeout   : 300 (sec)
Verify return code: 20 (unable to get local issuer certificate)
---
* OK IMAP4 ready
HELP
* BAD invalid command
QUIT
DONE

$ awk '{ if ($1~"Session-ID:" || $1~"Master-Key:" || $1~"RSA") print $1}' openssl-s_client.txt

RSA Session-ID:5EF3E7EDCC46993E51935914ACC1CBE6723259121248F958BC223D54FA84CFA0 Master-Key: 0665121ADB266864CDEF89E32A6F1A39677D540DB5B362BC351D3B08EE3059800F9A2186601710CE774AFB2CE3166C9

$ awk '{ if ($1~"Session-ID:" && $2 ~ /IMAP/) {printf("RSA %s%s ", $1, $2)} }' openssl-s_client.txt > openssl-s_client.keys

75 27 2012
Sake Blok sake.blok@SYN-bit.nl
Export SSL Session Keys

- **Export:**
  - File -> Export -> SSL Session Keys (1.6.x)
  - File -> Export SSL Session Keys (1.8.x)

- **Import:**
  - SSL preferences: (Pre)-Master-Secret log filename

*Provide SSL decryption in Wireshark to a 3rd party without having to share the private key!*
Agenda

• Cryptology overview
• The SSL protocol
• Analyzing SSL with Wireshark
• Analyzing SSL with Tshark
• Common SSL connection problems
• Further reading
• Questions & Discussion
Analyzing SSL with Tshark (1)

- -V to show whole tree (and decrypted application data)

- tshark -G fields | fgrep "ssl."
  tshark -R ssl.alert_message

- tshark -G currentprefs | egrep "^#?ssl."
  tshark -o ssl.keys_list:<ip>,<port>,<proto>,<keyfile> \ 
  -o ssl.debug_file:<log-file>
Analyzing SSL with Tshark (2)

```
tshark -r file.cap -o ssl.keys_list:192.168.3.3,443,http,"c:\key.pem" \
   -o ssl.debug_file:"c:\ssl-debug.log" -V -R http
```

```
$ tshark -o ssl.keys_list:192.168.3.3,443,http,"c:\tmp.key" \
   -r session-reuse.cap -R ssl.alert_message
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)
32  54.756406  192.168.3.3 -> 192.168.3.1  TLSv1 Alert (Level: Warning, Description: Close Notify)
35  62.809496  192.168.3.1 -> 192.168.3.3  TLSv1 Alert (Level: Warning, Description: Close Notify)
51 126.272833  192.168.3.3 -> 192.168.3.1  TLSv1 Alert (Level: Warning, Description: Close Notify)
54 137.815000  192.168.3.1 -> 192.168.3.3  TLSv1 Alert (Level: Warning, Description: Close Notify)
$ 
```
Agenda

• Cryptology overview
• The SSL protocol
• Analyzing SSL with Wireshark
• Analyzing SSL with Tshark

• Common SSL connection problems
• Further reading
• Questions & Discussion
Common SSL problems I (1)

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>24269 &gt; https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1</td>
</tr>
<tr>
<td>2</td>
<td>0.000667</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 24269 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=4</td>
</tr>
<tr>
<td>3</td>
<td>0.000716</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>24269 &gt; https [ACK] Seq=1 Ack=1 Win=128000 Len=0</td>
</tr>
<tr>
<td>4</td>
<td>0.020817</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>SSLv3</td>
<td>Client Hello</td>
</tr>
<tr>
<td>5</td>
<td>0.021173</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 24269 [ACK] Seq=1 Ack=65 Win=5840 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>0.024816</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>SSLv3</td>
<td>Alert (Level: Fatal, Description: Handshake Failure)</td>
</tr>
<tr>
<td>7</td>
<td>0.025488</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 24269 [FIN, ACK] Seq=8 Ack=65 Win=5840 Len=0</td>
</tr>
<tr>
<td>8</td>
<td>0.025536</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>24269 &gt; https [ACK] Seq=65 Ack=9 Win=127992 Len=0</td>
</tr>
<tr>
<td>9</td>
<td>0.031750</td>
<td>192.168.3.1</td>
<td>192.168.3.3</td>
<td>TCP</td>
<td>24269 &gt; https [FIN, ACK] Seq=65 Ack=9 Win=127992 Len=0</td>
</tr>
<tr>
<td>10</td>
<td>0.032001</td>
<td>192.168.3.3</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 24269 [ACK] Seq=9 Ack=66 Win=5840 Len=0</td>
</tr>
</tbody>
</table>

Secure Connection Failed

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

Cannot communicate securely with peer: no common encryption algorithm(s).

/Error code: ssl_error_no_cypher_overlap/

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 I (2)

In apache2:

SSLCipherSuite
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.
Common SSL problems II

The client can not validate the certificate as it is not signed by one of the trusted CA's.

Configure Intermediate CA in Apache2 with "SSLCertificateChainFile <ca-file>".
Common SSL problems III (1)

The client can not validate the certificate as it is expired.

Renew the certificate and attach it to the server.
Common SSL problems III (2)

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

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

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

Make sure the site you are trying to visit is indeed the site you intended to visit.
Common SSL problems V (1)

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>30245 &gt; https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1</td>
</tr>
<tr>
<td>2</td>
<td>0.000289</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 30245 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=4</td>
</tr>
<tr>
<td>3</td>
<td>0.000314</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>30245 &gt; https [ACK] Seq=1 Ack=1 Win=128000 Len=0</td>
</tr>
<tr>
<td>4</td>
<td>0.017233</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>SSL</td>
<td>Client Hello</td>
</tr>
<tr>
<td>5</td>
<td>0.017657</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 30245 [ACK] Seq=1 Ack=99 Win=5840 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>0.019863</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>SSL</td>
<td>Server Hello</td>
</tr>
<tr>
<td>7</td>
<td>0.019939</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Certificate, Certificate Request, Server Hello Done</td>
</tr>
<tr>
<td>8</td>
<td>0.019966</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>30245 &gt; https [ACK] Seq=99 Ack=2572 Win=128000 Len=0</td>
</tr>
<tr>
<td>9</td>
<td>3.299274</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Certificate, Client Key Exchange, Certificate Verify, Change Ciph...</td>
</tr>
<tr>
<td>10</td>
<td>3.300815</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TLSv1</td>
<td>Alert (Level: Fatal, Description: Unknown CA)</td>
</tr>
<tr>
<td>11</td>
<td>3.300763</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 30245 [FIN, ACK] Seq=2579 Ack=1501 Win=8768 Len=0</td>
</tr>
<tr>
<td>12</td>
<td>3.300791</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>30245 &gt; https [ACK] Seq=1501 Ack=2580 Win=127992 Len=0</td>
</tr>
<tr>
<td>13</td>
<td>3.310232</td>
<td>192.168.3.1</td>
<td>192.168.3.4</td>
<td>TCP</td>
<td>30245 &gt; https [FIN, ACK] Seq=1501 Ack=2580 Win=127992 Len=0</td>
</tr>
<tr>
<td>14</td>
<td>3.310386</td>
<td>192.168.3.4</td>
<td>192.168.3.1</td>
<td>TCP</td>
<td>https &gt; 30245 [ACK] Seq=2580 Ack=1502 Win=8768 Len=0</td>
</tr>
</tbody>
</table>

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
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>".

Common SSL problems VI

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

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)
Common SSL problems VII

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

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

The client needs to request a new certificate.

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.

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

[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)
Agenda

- Cryptology overview
- The SSL protocol
- Analyzing SSL with Wireshark
- Analyzing SSL with Tshark
- Common SSL connection problems
- Further reading
- Questions & Discussion
Further Reading about SSL

SSL and TLS: Designing and Building Secure Systems
by Eric Rescorla

SSL and TLS Essentials: Securing the Web
by Stephen A. Thomas
Questions & Discussion
FIN/ACK, ACK, FIN/ACK, ACK!

Thank You!

If you would like to receive the tracefiles (and keys!) that I used, please mail me: sake.blok@SYN-bit.nl