SSL decryption using Wireshark

Peter Wu
peter@lekensteyn.nl
https://lekensteyn.nl

January 12, 2016

Overview

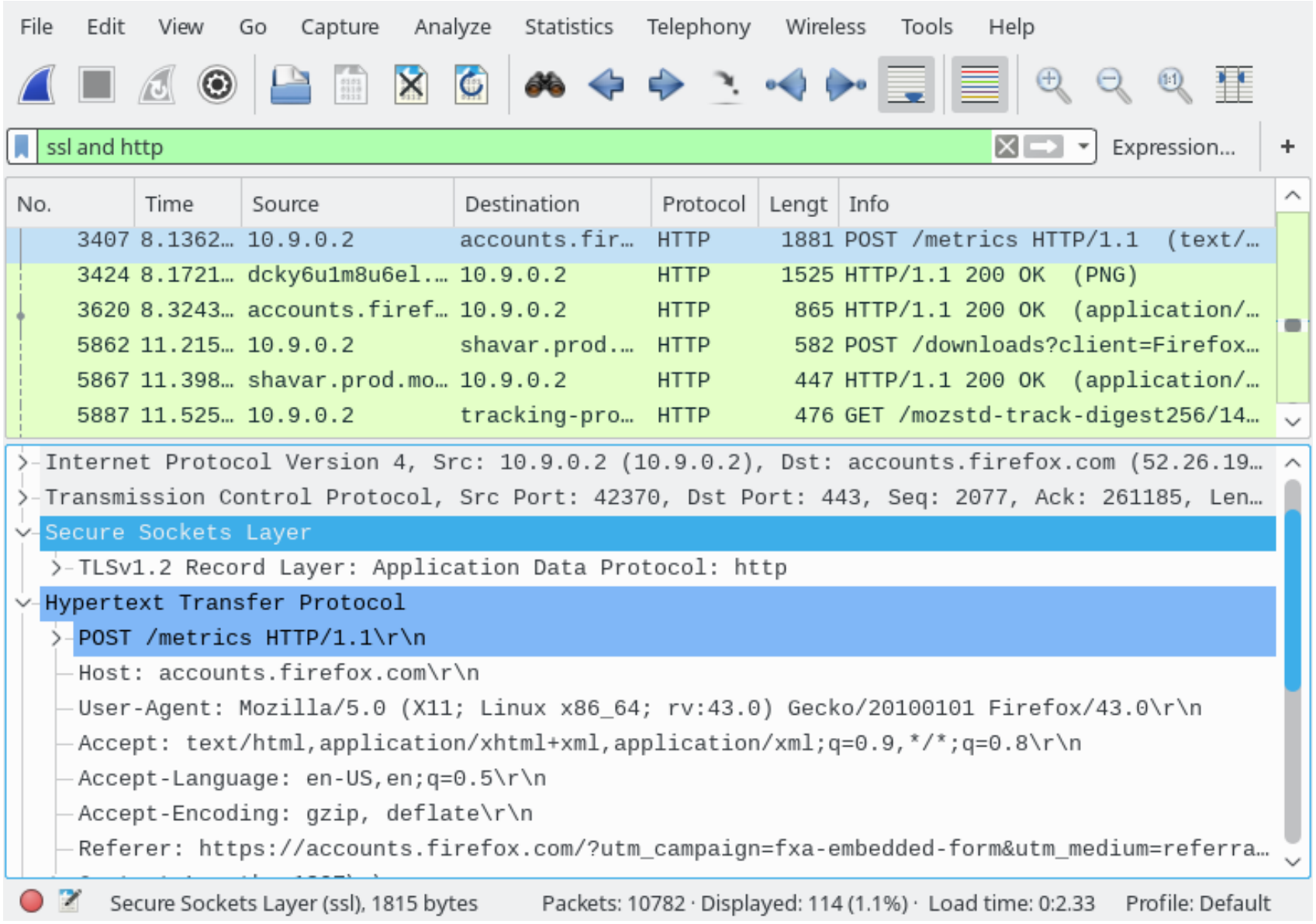
Introduction

SSL/TLS

SSL Decryption using Wireshark

Conclusion

Wireshark: network protocol analyzer



Why decrypt SSL with Wireshark?

- Debug applications that use SSL.
- Packet captures contain a full view of all network traffic.
- Wireshark supports many (application) protocols.

Methods for obtaining plaintext

Active:

- MITM, replace certificate.

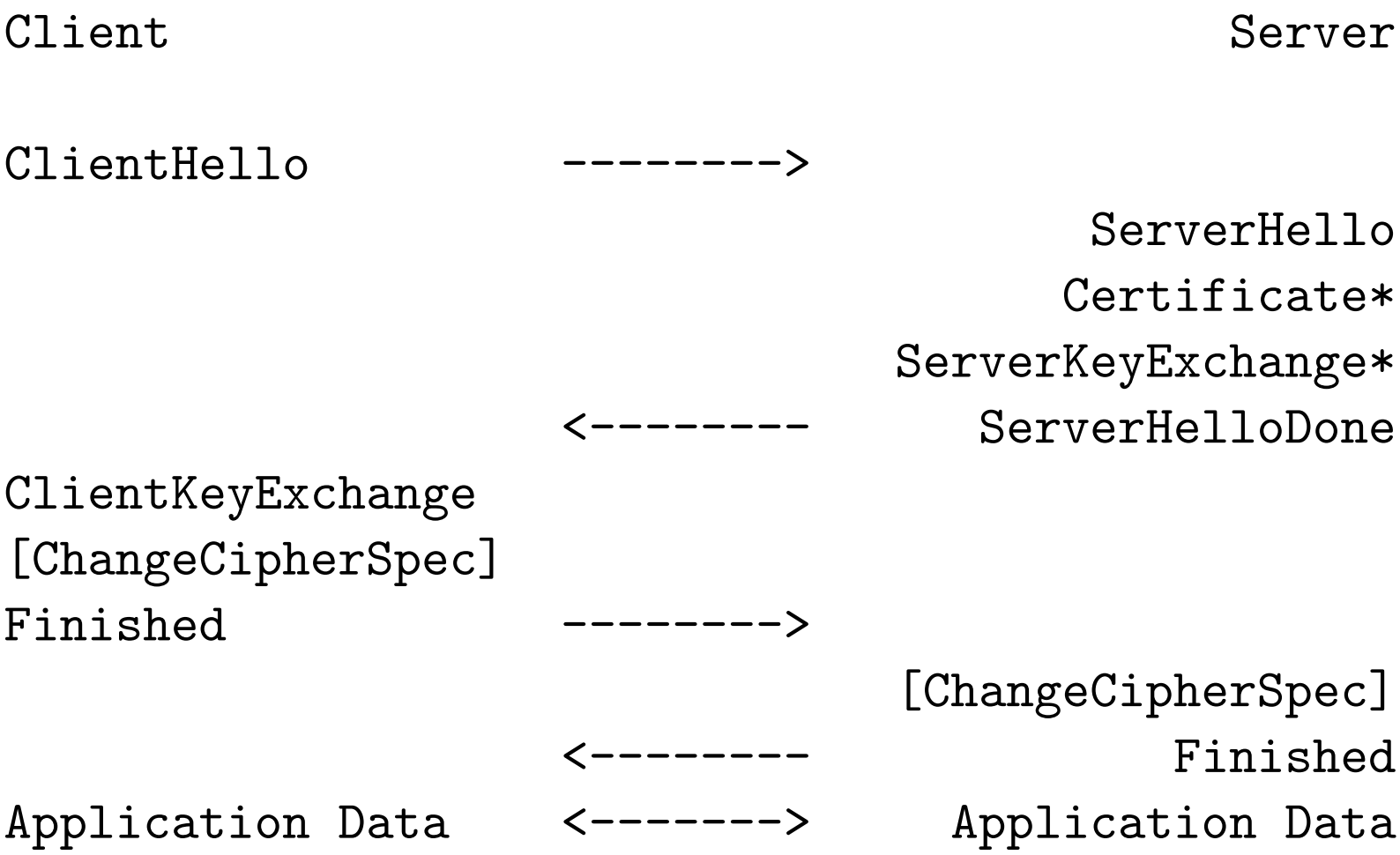
Passive:

- Option 1: after decryption (e.g. Web developer tools)
- Option 2: obtain secrets and capture packets (Wireshark)

SSL protocol overview

- SSLv3/TLS: basically the same protocol.
- Handshake establishing master secret ("session key").
- Master secret is used for symmetric encryption of Application Data (HTTP, SMTP, etc.).

Handshake overview



Simplified SSL handshake (adapted from RFC 5246 (TLS 1.2))

Relevant security parameters

- Client Hello: Client Random, list of supported cipher suites.
- Server Hello: Server Random, selected cipher suite.
- ServerKeyExchange: needed for DHE cipher suites.
- ClientKeyExchange: **encrypted** pre-master secret.
- (Public key from Certificate is only used for authentication.)
- Master secret: calculated from pre-master secret and two Randoms.

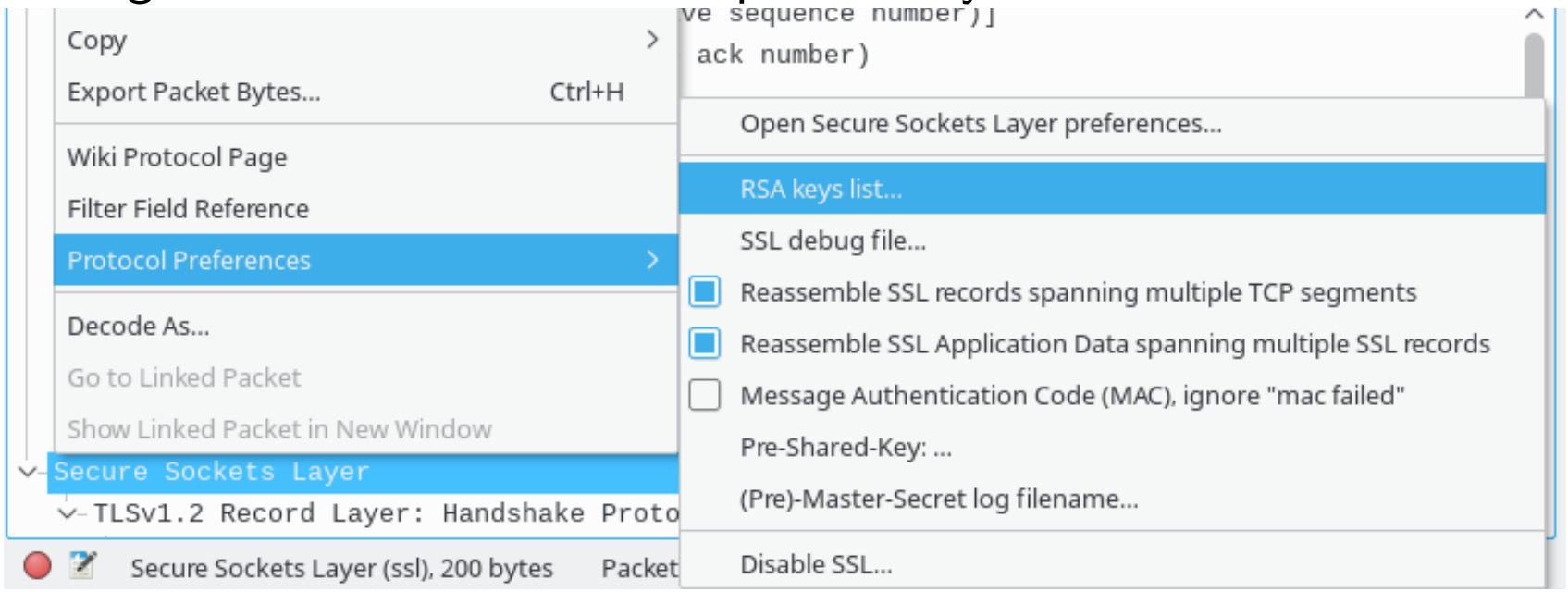
Pre-master secret

Value of pre-master secret depends on key exchange algorithm:

- RSA: 48 random bytes from **client**, encrypted using public key from Server Certificate.
- Diffie-Hellman: compute shared secret using parameters from Server Certificate (or ServerKeyExchange) and ClientKeyExchange.

Decryption using private RSA key of server

Configure Wireshark with RSA private key[1]:



Limitations:

- Does not work with Diffie-Hellman key exchanges.
- Requires RSA private key of the **server**, i.e. cannot be used for decryption of traffic as a client.

[1] See `https://wiki.wireshark.org/SSL#Preference_Settings`

SSL key logfile

- Text file containing (pre-)master secrets from SSL libraries[2].
- Configure file in Wireshark preferences: Edit → Preferences; Protocols → SSL; (Pre-)Master Secret log filename.
- Works also for clients!
- Supported by Firefox (via NSS), Chrome (via patched BoringSSL library).
- Also supported by cURL when built with NSS.
- For other libraries: dump keys from memory.

[2] Documented at `https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS/Key_Log_Format`

Example with Firefox (and other NSS users)

- Set environment variable `SSLKEYLOGFILE` to the output file before starting a program:
  - `SSLKEYLOGFILE=$PWD/premaster.txt firefox`
- Set file in Wireshark preferences.
- Start live capture.

Example with OpenSSL

- Does not support `SSLKEYLOGFILE` method.
- Solution: intercept OpenSSL library calls and dump keys[3].
- Implementations[4]:
  - `LD_PRELOAD` approach: teach OpenSSL `SSLKEYLOGFILE` by injecting code.
    - `SSLKEYLOGFILE=premaster.txt LD_PRELOAD=./sslkeylog.so curl https://example.com`
  - Debugger (GDB) approach: allows extraction of keys from running programs.

[3] `https://security.stackexchange.com/questions/80158/extract-pre-master-keys-from-an-openssl-application`

[4] `https://git.lekensteyn.nl/peter/wireshark-notes/tree/src`

Concluding remarks

- Remember that RSA keys cannot be used for decryption of SSL sessions using DH key exchanges.
- SSL keylog files (`SSLKEYLOGFILE`) also works for DH key exchanges and can be used on clients too (Firefox, Chrome).
- Use the latest version (currently Wireshark 2.0.1) when possible, fixes various bugs.

Other resources

- Main website: `https://www.wireshark.org/`
- Q&A: `https://ask.wireshark.org/`
- IRC: `#wireshark` at Freenode
- Wireshark wiki on SSL: `https://wiki.wireshark.org/SSL`
- Sample captures: `https://wiki.wireshark.org/SampleCaptures#SSL_with_decryption_keys`