Transport security layer

Questions and answers: "sslyze analysis"

Results via `sslyze localhost`

1. Are only strong cipher suites supported?
    - I think some ciphers are no longer considered strong (e.g. TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, considered weak by https://www.ssllabs.com/.. 128 bit is not really strong) Various ciphers are listed as "should be rejected"
2. Does the server support and prefer cipher suites with forward secrecy?
    - yes, it is supported and - from my understanding - TLS always prefers the strongest cipher. Forward Secrecy means: Not possible to break encryption later even if long-term secret which is used to exchange session keys is compromised (known to the attacker)
3. Does the server support strong protocol versions?
    - It supports TLS 1.0 - 1.2 (but not 1.3). TLS 1.2 is ok, but support for the older TLS versions should be stopped
4. Does the server support downgrade detection?
    - Not sure what is meant by this. Does it refer to the TLS_FALLBACK_SCSV? (not found any indicator that it is activated though)
5. What is TLS_FALLBACK_SCSV?
    - It is a flag passed during the handshake that a client might send to tell the server that it supports a higher version of TLS than advertised. (According to the article below, this can prevent a downgrade attack in edge cases, for example when a server aborts a connection during a handshake for some reason and client will then try with a lower protcol version..)
        - https://security.stackexchange.com/questions/112531/is-tls-fallback-scsv-useless-if-only-tls-1-0-1-1-1-2-is-supported
        - https://crashtest-security.com/de/tls-fallback-scsv/
6. Does the server support secure TLS renegotiation?
    - yes
7. Does the server support client-initiated renegotiation?
    - No as the server is NOT vulnerable to "client renegotiation Dos Attack".
    - According to https://crashtest-security.com/secure-client-initiated-ssl-renegotiation/#:~:text=The%20SSL%2FTLS%20renegotiation%20vulnerability,attack%20into%20the%20HTTPS%20sessions. client initiated renegotiation is disabled to prevent such a DOS attack
8. Is TLS compression support enabled?
    - no
    - there is a known attack (CRIME) that is only possible when the TLS compression feature is enabled (feature was dropped in TLS 1.3 i believe)
9. Is the server vulnerable to the Heartbleed attack?
    - no
    - Heartbleed is a known attack on incorrect implementation of TLS in the OpenSSL library.
10. Is the server vulnerable to the OpenSSL CCS injection attack?
    - no
    - A known vulnerability of the OpenSSL library

    - https://crashtest-security.com/prevent-ccs-injection/
11. Is the server vulnerable to the ROBOT attack?
    - no
    - This is an attack on certain RSA ciphers that allow an attacker to decrypt the traffic.
    - https://crashtest-security.com/prevent-robot-attack/
12. Does the Domain use CAA to specify CAs, which can be used to issue certificates for it?
    - Since i tested "localhost" with a certificate from a local CA, I'm quite certain that no CAA was specified. (However, hacking-lab.com uses CAA)

Remark: Quick analyzes results via https://www.ssllabs.com/ssltest/analyze.html?d=hacking-lab.com: Only supports TLS 1.2 but supports some week ciphery (128 bit, EDES) but overall rating still A

Notes and varia

- TLS provides:
    - Confidentialy
    - Authenticity
        - Client and Server supported
        - Usually: unilateral..just server is authenticated
    - Integrity
- often used for HTTP, FTP, IMAP, POP3, SMTP
    - HTTS is just HTTP wrapped in TLS (no new protocol)
- Encryption
    1. Assyemtric encrption to establish connection
        - keay exchange
    2. Symmetric encryption for actual data

TLS handshake

1. client -> server: hello
2. server -> client:
3. ... etc.. (see diagram in powerpoint or links)

https://de.wikipedia.org/wiki/Transport_Layer_Security https://tls.ulfheim.net/

CLR Certificate revokation list Client (browser) can download list and check if certificate is on list

OCSP During the handshake: the client (browser) can ask the OCSP responder if server certificate is still valid. (Alternative: OCSP Stapling: server asks OCSP responder and caches answer from some time and can then send its OCSP status to the client during handshake. Not widely supported/used yet)

-> generally: CRL, OCSP browser will still accept certificate if CLR or OCSP servers not reachable. When using OCSP stapling: Flag "Must Staple" (X.509 extension), makes browser abort connection when no OCSP response is present.

analyte TLS configuration of a server

- Linux tool. "sslyze"
  - `sslyze localhost` (did not work: `sslyze --regular localhost`)
  - https://www.kali.org/tools/sslyze/
  - Doku: https://nabla-c0d3.github.io/sslyze/documentation/
- Website
  - https://www.ssllabs.com/ssltest/analyze.html?d=hacking-lab.com

Changes in TLS 1.3

- Removed support for: Weak ciphers
- Added: Improved handshake, downgrade protected, new algorithms (etc.)

Varia

- Perfect forward secrecy: Not possible to break encryption later even if long-term secret which is used to exchange session keys is compromissed (known to the attacker).
- CAA (Certificate Authority Authorization)
  - A DNS entry (CAA) that specifies one or more CA's to issue certificates for a certain domain
  - https://www.websecurity.digicert.com/security-topics/what-is-certificate-authority-authorization