DI-FCT-UNL
Segurança de Redes e Sistemas de Computadores
Network and Computer Systems Security

Mestrado Integrado em Engenharia Informática
MSc Course: Informatics Engineering
1º Semestre, 2019/2020

Web Security Beyond TLS

Outline

- Web Security: Beyond TLS
  - TLS complexity and weak TLS Ciphersuites
  - Evolution of TLS: From TLS 1.2 to TLS 1.3
- TLS Implementations and attacks (as time goes by)
- Web Applications Security beyond TLS

Outline

- Web Security: Beyond TLS
  - TLS complexity and weak TLS Ciphersuites
  - Evolution of TLS: From TLS 1.2 to TLS 1.3
- TLS Implementations and attacks (as time goes by)
- Web Applications Security beyond TLS

Web insecurity vs. TLS Cryptosuites

TLS Cryptographic Suites:

Negotiation options (handshake), flexibility, complexity (design vs. implementation)

vs. Security vs. Insecurity

One relevant issue for Web Security concerns:

See (ex.):

OWASP (Open Web Application Security Project) Foundation

> Top Ten Vulnerability Rank (2010, 2013, 2017)

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

OWASP:
Ten Most Critical Web App Security Risks: 2017

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure

   Weak-Ciphers, No PBS/PFS provisioning, unsecure PWD-encryption/hashing w/ impact on TLS misconfigurations
4. XML External Entities (XXE)
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting (XSS)
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging & Monitoring

Outline

- Web Security: Beyond TLS
  - TLS complexity and weak TLS Ciphersuites
  - Evolution of TLS: From TLS 1.2 to TLS 1.3
- TLS Implementations and attacks (as time goes by)
- Web Applications Security beyond TLS

TLS and SSL Versions (Installation Base)

After Apr/2016, latest versions of major browsers adopt TLS V1.1, 1.2, 1.3
… but many vulnerabilities are induced by old browsers and old versions of OSes and many implementations (libraries or App packaged implementations)

TLS v1.3 is recent: in Safari 12.0, Opera v60, Firefox v66, Google Chrome v73
See here: https://en.wikipedia.org/wiki/Transport_Layer_Security

Protocol version	Website support[59]	Security[59][60]
SSL 2.0	1.9%	Insecure
SSL 3.0	7.8%	Insecure[61]
TLS 1.0	68.8%	Depends on cipher[n 1] and client mitigations[n 2]
TLS 1.1	77.9%	Depends on cipher[n 1] and client mitigations[n 2]
TLS 1.2	95.0%	Depends on cipher[n 1] and client mitigations[n 2]
TLS 1.3	13.6%	Secure

Client and Server Endpoints must agree In the protocol version

Ciphersuites and related parameterizations

- The established *ciphersuites* (standardized cryptography) are defined in different versions of SSL and TLS
  - Dynamically negotiable in different TLS and SSL versions and Handshake Sub-protocols, between clients (ex., browsers) and servers (ex., HTTPS servers):
    - Clients: propose supported ciphersuites (typically in a set) and Keysizes
    - Servers: accept the ciphersuite (from the client set)
    - Relevant issue: possible bad default settings

- Standardization of different client or server certificate types, digital signatures supported: correct verification in implementations and operational trust assumptions are very important issues !

- Padding processing and insufficient mitigation of DoS/DDoS is another security standardization issue (remember the base RLP message format and design implications)

TLS Authentication and Key-Exhange Methods

Key exchange/agreement and authentication

Algorithm	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3	Status
RSA	Yes	Yes	Yes	Yes	Yes	No	Defined for TLS 1.2 in RFCs
DH-RSA	No	Yes	Yes	Yes	Yes	No	
DHE-RSA (forward secrecy)	No	Yes	Yes	Yes	Yes	Yes	
ECDH-RSA	No	No	Yes	Yes	Yes	No	
ECDHE-RSA (forward secrecy)	No	No	Yes	Yes	Yes	Yes	
DH-DSS	No	Yes	Yes	Yes	Yes	No	
DHE-DSS (forward secrecy)	No	Yes	Yes	Yes	Yes	No[45]	
ECDH-ECDSA	No	No	Yes	Yes	Yes	No	
ECDHE-ECDSA (forward secrecy)	No	No	Yes	Yes	Yes	Yes	
PSK	No	No	Yes	Yes	Yes		
PSK-RSA	No	No	Yes	Yes	Yes		
DHE-PSK (forward secrecy)	No	No	Yes	Yes	Yes		
ECDHE-PSK (forward secrecy)	No	No	Yes	Yes	Yes		
SRP	No	No	Yes	Yes	Yes		
SRP-DSS	No	No	Yes	Yes	Yes		
SRP-RSA	No	No	Yes	Yes	Yes		
Kerberos	No	No	Yes	Yes	Yes		
DH-ANON (insecure)	No	Yes	Yes	Yes	Yes		
ECDH-ANON (insecure)	No	No	Yes	Yes	Yes		
GOST R 34.10-94 / 34.10-2001[46]	No	No	Yes	Yes	Yes		Proposed in RFC drafts

Cipher security against publicly known feasible attacks

Cipher			Protocol version						Status
Type	Algorithm	Nominal strength (bits)	SSL 2.0	SSL 3.0 [n 1][n 2][n 3][n 4]	TLS 1.0 [n 1][n 3]	TLS 1.1 [n 1]	TLS 1.2 [n 1]	TLS 1.3	
Block cipher with mode of operation	AES GCM[47][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	AES CCM[48][n 5]		N/A	N/A	N/A	N/A	Secure	Secure	
	AES CBC[n 6]		N/A	N/A	Depends on mitigations	Depends on mitigations	Depends on mitigations	N/A	
	Camellia GCM[49][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	N/A	
	Camellia CBC[50][n 6]		N/A	N/A	Depends on mitigations	Depends on mitigations	Depends on mitigations	N/A	
	ARIA GCM[51][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	N/A	
	ARIA CBC[51][n 6]		N/A	N/A	Depends on mitigations	Depends on mitigations	Depends on mitigations	N/A	
	SEED CBC[52][n 6]	128	N/A	N/A	Depends on mitigations	Depends on mitigations	Depends on mitigations	N/A	
	3DES EDE CBC[n 6][n 7]	112[n 8]	Insecure	Insecure	Insecure	Insecure	Insecure	N/A	
	GOST 28147-89 CNT[46][n 7]	256	N/A	N/A	Insecure	Insecure	Insecure	N/A	Defined in RFC 4357
	IDEA CBC[n 6][n 7][n 9]	128	Insecure	Insecure	Insecure	Insecure	N/A	N/A	Removed from TLS 1.2
	DES CBC[n 6][n 7][n 9]	56	Insecure	Insecure	Insecure	Insecure	N/A	N/A	
		40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	Forbidden in TLS 1.1 and later
	RC2 CBC[n 6][n 7]	40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
Stream cipher	ChaCha20-Poly1305[57][n 5]	256	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	RC4[n 11]	128	Insecure	Insecure	Insecure	Insecure	Insecure	N/A	Prohibited in all versions of TLS by RFC 7465
		40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
None	Null[n 12]	–	N/A	Insecure	Insecure	Insecure	Insecure	N/A	Defined for TLS 1.2 in RFCs

HMACs

HMACs standardized by RFC 2104

Data integrity							
Algorithm	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3	Status
HMAC-MD5	Yes	Yes	Yes	Yes	Yes	No	Defined for TLS 1.2 in RFCs
HMAC-SHA1	No	Yes	Yes	Yes	Yes	No	
HMAC-SHA256/384	No	No	No	No	Yes	No	
AEAD	No	No	No	No	Yes	Yes	
GOST 28147-89 IMIT[46]	No	No	Yes	Yes	Yes		Proposed in RFC drafts
GOST R 34.11-94[46]	No	No	Yes	Yes	Yes		

Standardized Functions in TLS endpoints

- Cryptographic computations are different in different SSL and TLS versions

- Key and MAC Generation for Cryptogrphic Computations MACs: TLS v1.2 (RFC 5246)

- Other critical cryptographic computations:
  - PRE-MASTER Secrets
  - MASTER SECRET CREATION : 48 bytes (384 bits)
  - KEY-BLOCK generation by PRF Pseudorandom Function based on HMACs from previous random seeds and shared secrets along the handshake exchanged parameters

See the bibliography

Ciphersuites and related parameterizations

- Important security measures (default baseline):
  - Avoidance of SSL versions and TLS 1.0
  - Avoidance of considered "Weak Cryptosuites"
  - Appropriate key sizes (RSA, DSA keys >= 2048 bits) for the proper protection of secure envelopes for the establishment of session or MAC keys and security transport and session association parameters
  - The problem of "possibly unsecure ECCs" (on going problem)
  - Only Ephemeral Diffie Hellman Agreements with parameterizations for public and private numbers >= 2048 bits
    - Trade-off for Efficiency: fixed shared initialization parameters (primitive root and prime number for the modular operations)
  - Problem: scale, installation base vs. "relaxed" TLS server configurations

See the bibliography and also

- LABs (hands-on study and verifications in tracing Handshake Protocol)
- Security auditing on possible weak ciphersuites and vulnerabilities

Some references on Web App. Security Risks and TLS Vulnerabilities

- Ref. OWASP (Open Web Application Security Project)
- https://www.owasp.org
- https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
- https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html

Security checks of TLS robustness

- Management and validation of X509 certificates and certification chains
- Incomplete Certificate Validation Issues
  - Complete and correct validation of critical attributes
  - Complete and correct validation of chains (all certificates involved in certification chains)
- Problem of weak ciphersuites

- Some interesting tools and practical verifications:
  - https://www.ssllabs.com/
  - https://webcheck.pt/pt/
  - https://internet.nl/
  - https://github.com/drwetter/testssl.sh
  - https://testssl.sh
  - https://www.ssllabs.com/ssltest/
  - https://www.immuniweb.com/ssl/

TLS Security Auditing

- Typical roadmap for TLS endpoint auditing:
  - Testing Validity of Certificates / Certification Chains
  - Testing TLS protocol enabled versions
  - Testing considered weak ciphersuites
  - Testing robustness for perfect secrecy
  - Testing ciphersuites ordering of acceptance for the enabled TLS protocol versions
  - Testing for keysizes (or avoidance of considered weak keys)
  - Testing enforcements for required Extended Verifiable Certificates
  - Testing critical and other required attributes as security requirements for certificates
  - Testing available CRL and OCSP endpoints
  - Testing App Level Protocols encapsulated on TLS enabled sessisons
  - Testing for auditable vulnerabilities
  - Testing specifically ciphersuites, used crypto algorithms and key sizes (according to expected requirements)
  - Testing security compliance face to different client – environments and systems

Outline

- Web Security: Beyond TLS
  - TLS complexity and weak TLS Ciphersuites
  - Evolution of TLS: From TLS 1.2 to TLS 1.3
- TLS Implementations and attacks (as time goes by)
- Web Applications Security beyond TLS

SSL/TLS Attacks

SSL/TLS: Other Attacks and Impact

- Design implications
- Implementation implications

TLS and SSL Attacks vs. Countermeasures

The history of SSL (versions 1., 2., 3) and TLS (versions 1.1 and 1.2) attacks and related countermeasures (as many other protocols) that the "perfect secure protocol" and "the perfect implementation strategy for security vs. flexibility vs. usability tradeoffs" have not been achieved.

Constant back-and-forth between threats and counter-measures has been a constant struggle ….

New complexities and tradeoffs =>
New threats and threat models =>
New adversarial conditions =>
New counter-measures (patching ?) =>
Evolution/Revision of standardization =>
Evolution/Revision of Implementations

TLS Security Auditing (Vulnerabilities): A possible / typical verification roadmap

- **CCS** (CVE-2014-0224)
- **Ticketbleed** (CVE-2016-9244)
- **ROBOT**
- **Secure Renegotiation (RFC 5746)**
- **Secure Client-Initiated Renegotiation**
- **CRIME, TLS** (CVE-2012-4929)
- **BREACH** (CVE-2013-3587)
- **POODLE, SSL** (CVE-2014-3566)
- **TLS_FALLBACK_SCSV** (RFC 7507) Downgrade attack prevention
- **SWEET32** (CVE-2016-2183, CVE-2016-6329**)**
- **FREAK** (CVE-2015-0204) **DROWN** (CVE-2016-0800, CVE-2016-0703)
- **LOGJAM** (CVE-2015-4000), no DH EXPORT ciphers
- **BEAST** (CVE-2011-3389)
- **LUCKY13** (CVE-2013-0169)
- **RC4** (CVE-2013-2566, CVE-2015-2808)
- **HACKERSCHOICE**

TLS and SSL Attacks

Attacks involving PKI and X509 Certificates' Management and Validation

Attacks against the Handshake Protocol

Attacks on the record layer protocol

- **BEAST** (Browser-Exploit Against SSL/TLS): Crypto Attack (Chosen-Plaintext Crypto. Attack)
- **CRIME** Attack (Compression Ratio Info-Leak Cookies): Session Hijacking on TLS protected cookies and compression/decompression processing, can break the authentication of TLS sessions
- Attacks on PKIs and Certification-Chain validations in many libraries, overtime:
  - OpenSSL, GnuTLS, JSSE, ApacjeHttpCLient, Weberknetch, cURL, PHP, Python, and other Applications with integrated Packaged TLS processing
- **HackersChoice Attack**: DoS against the Handshake Proecessing Computations for usual Server-Only Authentication Modes currently used

TLS and SSL Attacks

Heatbleed Attack:
Endpoint from client side TLS negotiation of Heartbeat messages

Attack against TLS SW implementations (Bad TLS Heartbeat implementation) causing access to "memory mapped" security association parameters

https://en.wikipedia.org/wiki/Heartbleed

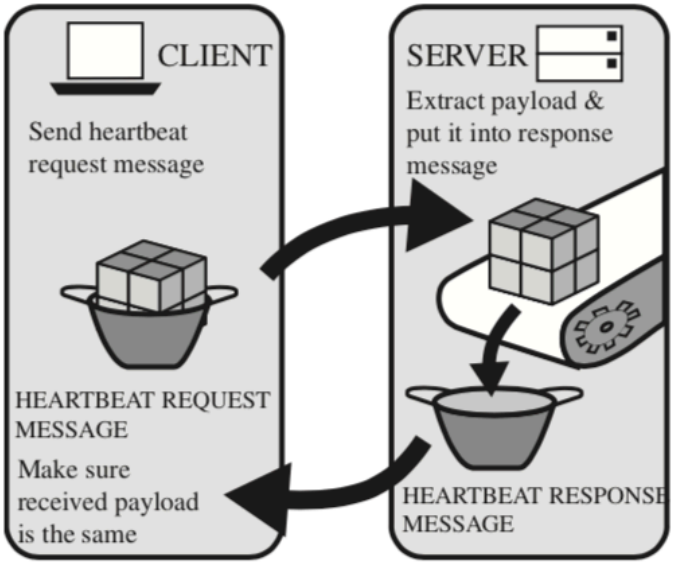
POODLE (Padding Oracle On Downgraded Legacy Encryption)

Man in the Middle Attack: exploit which takes advantage of Internet and security software clients' fallback to "weak-ciphersuites' negotitated and accepted by the HTTPS server endpoint

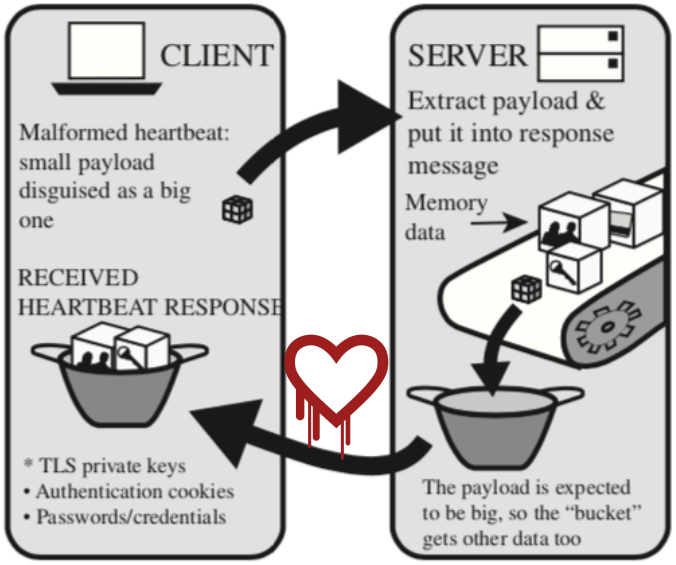
https://en.wikipedia.org/wiki/POODLE

Heartbeat Protocol vs. Heartbleed Attack

Heartbleed -The Open SSL Heartbeat Exploit" Copyright © 2014 BAE Systems Applied Intelligence



(a) How TLS Heartbeat Protocol works



(b) How TLS Heartbleed exploit works

Attacker sends a message indicating maximum payload length (64 KB) but only includes minimum payload (16 bytes).

Almost 64 KB of the buffer is not overwritten and whatever happened to be in memory at the time will be sent to the requestor:
Repeated attacks can result in the exposure of significant amounts of memory on the vulnerable system: private keys, user identification information, authentication data, passwords, or other sensitive data

TLS vulnerabilities and impact

Attacks	Security				
	Insecure		Depends	Secure	Other
Renegotiation attack	1.2% (–0.1%) support insecure renegotiation		0.4% (±0.0%) support both	96.2% (+0.1%) support secure renegotiation	2.2% (±0.0%) no support
RC4 attacks	<0.1% (±0.0%) support only RC4 suites	6.0% (–0.3%) support RC4 suites used with modern browsers	28.5% (–0.7%) support some RC4 suites	65.5% (+1.0%) no support	N/A
CRIME attack	2.4% (–0.1%) vulnerable		N/A	N/A	N/A
Heartbleed	0.1% (±0.0%) vulnerable		N/A	N/A	N/A
ChangeCipherSpec injection attack	0.8% (±0.0%) vulnerable and exploitable		4.7% (–0.2%) vulnerable, not exploitable	92.6% (+0.4%) not vulnerable	1.9% (+0.1%) unknown
POODLE attack against TLS (Original POODLE against SSL 3.0 is not included)	2.1% (–0.1%) vulnerable and exploitable		N/A	97.1% (+0.2%) not vulnerable	0.8% (–0.1%) unknown
Protocol downgrade	23.2% (–0.4%) TLS_FALLBACK_SCSV not supported		N/A	67.6% (+0.7%) TLS_FALLBACK_SCSV supported	9.1% (–0.4%) unknown

Current relevance of TLS 1.3

TLS 1.3, IETF Defined in 2014

(Today coexisting w/ TLS 1.2 …)

TLS 1.3 removes:

- Compression
- Not Authenticated Modes and Handshake Exchanges
- Considered Weak Chiphers
- Static RSA and DH Key Exchange Methods
- 32 bit timestamps as part of Random parameters in Client/Server Hello Handshake Messages
- Renegotiation of secrets from previous established parameters
- Heartbeat Protocol
- Change Cipher Spec Protocol
- RC4
- Use of MD5, SHA-1 and SHA-224

Current relevance of TLS 1.3

TLS 1.3, IETF Defined in 2014

(Today coexisting w/ TLS 1.2)

TLS 1.3 includes (for improving the tradeoff security and efficiency):

- DH and EC-DH for Key Exchanges (no RSA Key Exchange)
- Simplification of "one-shot" Handshake rounds (one round trip time handshake), by reordering/piggybacking (or pipelining) the handshake sequence
- Client side must send authenticated parameters, before the negotiation of cipher suites when client-authentication or mutual-aiuthentication is adopted

A Bibliography on TLS security research

- The most dangerous code in the world: validating SSL certificates in non-browser software, M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh and V. Shmatikov, *ACM CCS 2012*
- Forward Secrecy and TLS Renegotiation: F. Giesen et al., On the Security of TLS Renegotiation, *ACM CCS 2013*

- T. Jager et al., On the Security of TLS v1.3 and QUIC against Weaknesses in PKCS#1 .5 Encryption, *ACM CCS 2015*

- The 9 Lives of Bleichenbacher's CAT: New Cache Attacks on TLS Implementations , Eyal Ronen, Robert Gillham, Daniel Genkin, Adi Shamir, David Wong, and Yuval Yarom, Dec 2018

See also:

- https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2019/february/downgrade-attack-on-tls-1.3-and-vulnerabilities-in-major-tls-libraries/ , Nov 2018

- Selfie: reflections on TLS 1.3 with PSK, Nir Drucker and Shay Gueron , https://eprint.iacr.org/2019/347.pdf ,

A Recent Research Bibliog. … (TLS Vulnerabilities and Proposed Solutions)

ACM CCS 2018

•Pseudo Constant Time Implementations of TLS Are Only Pseudo Secure
Eyal Ronen (Weizmann Institute of Science), Kenny Paterson (Royal Holloway, University of London), Adi Shamir (Weizmann Institute of Science)

•Partially specified channels: The TLS 1.3 record layer without elision
Christopher Patton (University of Florida), Thomas Shrimpton (University of Florida)

•The Multi-user Security of GCM, Revisited: Tight Bounds for Nonce Randomization
Viet Tung Hoang (Florida State University), Stefano Tessaro (University of California Santa Barbara), Aishwarya Thiruvengadam (University of California Santa Barbara)

Usenix Sec. Symp. 2018:

•Return Of Bleichenbacher's Oracle Threat (ROBOT), H. Bock et al.,

IEEE Sympo. On Security and Privacy 2018

•A Formal Treatment of Accountable Proxying over TLS, Karthikeyan Bhargavan at al.

IEEE Synp. On Sec and Privacy 2019:

•The 9 Lives of Bleichenbacher's CAT: New Cache Attacks on TLS Implementations, E. Ronen et al.

NDSS 2018:

•Removing Secrets from Android's TLS. Jaeho Lee *(Rice University)* and Dan S. Wallach *(Rice University)*.

•TLS-N: Non-repudiation over TLS Enablign Ubiquitous Content Signing. Hubert Ritzdorf *(ETH Zurich)*, Karl Wust *(ETH Zurich)*, Arthur Gervais *(Imperial College London)*, Guillaume Felley *(ETH Zurich)*, and Srdjan Capkun *(ETH Zurich)*.

NDSS 2019:

•The use of TLS in Censorship Circumvention. Sergey Frolov, Eric Wustrow

TLS in current practice …

- TLS v1.2 and v 1.3 is the base of current baseline security

- A strict control on considered secure ciphersuites, and parameterizations must be addressed as baseline countermeasures against the more prevalent attacks:

Hands-on (Ref. Example):
https://www.ssllabs.com/ssltest/
https://www.ssllabs.com/ssltest/viewMyClient.html
https://www.ssllabs.com/projects/index.html

See also: https://www.howsmyssl.com
Hands-on with TLS checking Tools: https://testssl.sh

Outline

- Web Security: Beyond TLS
  - TLS complexity and weak TLS Ciphersuites
  - Evolution of TLS: From TLS 1.2 to TLS 1.3
- TLS Implementations and attacks (as time goes by)
- Web Applications Security beyond TLS

Threats beyond TLS

- Remember: TLS is designed to protect transport-based communication channels (UDP or TCP)

- TLS and HTTPS is not a panacea for WEB Security: it is just one of the security elements for WEB Security
  - See: OWASP Web Security Attacks and Top-Ten Vulnerabilities
  - OWASP: See https://www.owasp.org/index.php/Main_Page

- Relates with communication security properties, not considering intrusions on endpoints

- The required secure processing in implementing the TLS endpoints (transport and session states and sensitive security association parameters and correct and trusted TLS state-machine execution control) is out of scope of TLS protocols' security standardization effort

Threats beyond TLS

- SW and Application Level Security
  - Can use TLS but with Application-Level Vulnerabilities
  - Bad or unmatched use of TLS Parameterizations
- PKI SW based vulnerabilities
- Related Attacks: Attacks against Time Synchronization Protocols
- Unsecure management of X509 certificates and incorrect verification and validation of x509 (namely X509v3 extension attributes) in the TLS handshake of Certification chains: Recurrent vulnerabilities in many TLS libraries
  - This included deficient management of the "trusted root assumption" in acceptance or pre-installed X509 certificates (including CA certificates)
  - Incorrect operation and management of X509 certificates' life-cycles – include lack of proper control for CRLs and management of OCSP endpoints
- DoS or DDoS
  - No effective protection on TLS…. It Can be aggravated w/ TLS

Web / SW Security auditing and assessment tools

- Suggestions for the interested students:
  - OWASP Flagship Projects / Tools and Code Projects
    - https://www.owasp.org/index.php/Main_Page
    - https://www.owasp.org/index.php/
  - OWASP Mobile Security Testing Guide
    - https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide

Revision: Complementary Readings

See references on the slides