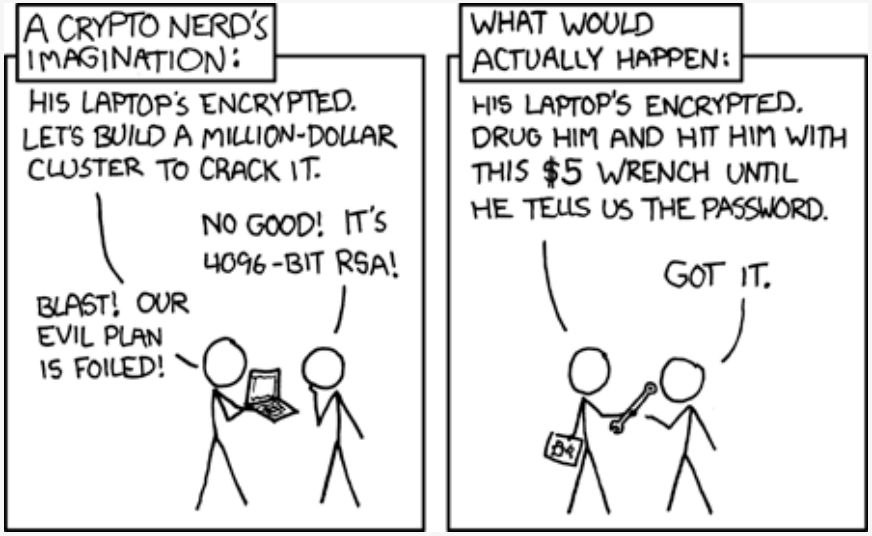
SSL in der Praxis, sicher?

Achim Hoffmann
achim@owasp.org
München 25. Oktober 2013
sic[!]sec GmbH

Sicher?



http://xkcd.com/538/

Disclaimer

Problem ist nicht NSA, GCHQ, BND, PRISM, Tempora, XKeyScore, ...

Angriffspunkte

- Wo?
  - Kryptographie (Cipher, etc.)
  - SSL-Protokoll
  - PKI
  - (SSL-Konfiguration)

Angriffspunkte

- Wo?
  - Kryptographie (Cipher, etc.)
  - SSL-Protokoll
  - PKI
  - (SSL-Konfiguration)
- Was?
  - Integrität
  - Verfügbarkeit
  - Authentizität

Agenda

- Lösungen
- Übersicht der Angriffe
- Übersicht der Probleme
- Einige Angriffe im Detail
- Glossar
- Referenzen

Lösungen?

- PKI: Umstellung auf "SSH-Modell"
- PKI: Umstellung auf Convergence Concept
- PKI: unabhängige Notare (siehe Problem mit "Trust" später)
  - Modelle: "Public Key Pinning", "Trust Assertions for Certificate Keys"
- Certificate Pinning
- Check: Datum der Signatur, dann kann Client alte Zertifikate selbst erkennen

Und bis das umgesetzt ist ...

Lösungen Agenda

- Server-Konfiguration
- Browser-Konfiguration

Server-Konfiguration

- Protokoll
- Cipher
- Zertifikat
- Webserver

Server-Konfiguration: Protokoll

1. SSLv2 deaktivieren
2. SSLv3 und TLS v1.0 nur benutzen, wenn unbedingt nötig
3. TLS v1.1 und TLS v1.2 aktivieren (RC4, BEAST)
4. keine Renegotiation vom Client erlauben
5. Kompression in SSL abschalten (CRIME)

Server-Konfiguration: Cipher

1. keine NULL-Cipher
2. keine EXPORT-Cipher
3. Keine "WEAK"-Cipher
4. keine ADH-Cipher
5. Keysize min. 128 Bits
6. Cipher mit EDH Key Exchange (wegen PFS)
7. Cipher mit CBC-Mode meiden (wegen Oracle Attack usw.)
8. Default: den stäksten Cipher anbieten
9. in Zukunft: GCM-Cipher

Server-Konfiguration: Cipher

Teufel oder Belzebub

1. ISM-Compliant: **keine** Cipher mit RC4 erlauben
2. FIPS-140-Compliant: **keine** Cipher mit RC4 erlauben
3. BEAST: **nur** Cipher mit RC4 erlauben

UPDATE: November 2013

- RC4 nicht empfohlen gemäß BSI TR-02102-2
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2_pdf.pdf?__blob=publicationFile
- RC4 gilt als gebrochen und kann (angeblich) in echtzeit entschlüsselt werden.

Server-Konfiguration: Zertifikat

1. vertrauenswürdige CA auswählen (ist Vertrauen bezahlbar?)
2. Vertraunskette muss stimmen (Trust Chain)
3. Gültigkeit (Datum, Ablauf, Fingerprint)
4. kein MD5 Fingerprint
5. Wildcard-Zertifikate sind unsicher!
6. EV-Zertifikate (Extended Validation) benutzen
7. OCSP: ja, nein, nein, vielleicht, ...

Server-Konfiguration: Webserver

1. wenn SSL, dann **keinen** Inhalt unter `http://` anbieten
   auch keine Weiterleitung (Redirect) auf `https://`
2. **alle** Daten (Seiten, Bilder, Skripte, CSS) mit https ausliefern
3. HSTS benutzen, am Besten mit "certificate pinning"
4. bei Cookies **immer** das `secure`-Flag setzen
5. bei Cookies **immer** das `HttpOnly`-Flag setzen
6. keine fremden Source (z.B. Skripte von anderen Servern)
7. alle Daten mit **korrektem** `Content-Type` ausliefern

Siehe OWASP:

- https://www.owasp.org/index.php?title=Transport_Layer_Protection_Cheat_Sheet
- https://www.owasp.org/index.php/HTTP_Strict_Transport_Security
- https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning

Server-Konfiguration: Apache

schlecht

```
SSLProtocol all -SSLv2
SSLCipherSuite ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM: \
   +LOW:+SSLv2:+EXP:+eNULL
```

besser

```
SSLProtocol -ALL +SSLv3 +TLSv1 +TLSv1.1 +TLSv1.2
SSLCompression           Off
SSLInsecureRenegotiation Off
SSLHonorCipherOrder On
SSLCipherSuite HIGH:!RC4-SHA:!ADH
 # oder
SSLCipherSuite DHE-RSA-AES256-SHA:AES128-SHA \
   :DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES256-SHA
```

UPDATE November 2013: kein RC4

Siehe SSL/TLS Deployment Best Practices [7]

Server-Konfiguration: nginx

```
ssl_protocols SSLv3 TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_ciphers ECDHE-ECDSA-AES256-GCM-SHA384:\
    ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256
    ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM
    ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128
    ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-RC4-SHA:\
    ECDHE-RSA-AES256-SHA;
ssl_session_cache  builtin:1000  shared:SSL:10m;
 # Compression per Default off bei nginx 1.1.6+/1.
```

UPDATE November 2013: kein RC4

Server-Konfiguration: IIS

- gut (aber umständlich)

```
  \Ciphers\NULL]        "Enabled"=dword:00000000
  \Ciphers\DES 56/56]   "Enabled"=dword:00000000
  \Ciphers\RC4 56/128] "Enabled"=dword:00000000
  \Ciphers\RC4 64/128] "Enabled"=dword:00000000
```

in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL`

UPDATE November 2013: kein RC4

Server-Konfiguration: Test-Tools

chkcert.pl – einfacher Test der Chain (RC4-SHA hardcodet!)
cnark.pl – einfacher Test der Cipher-Suite (fest vorgegeben!)
o-saft.pl – umfangreiche Tests der Cipher-Suite, Zertifikat, Schwachstellen
smtp_tls_cert.pl – Dump des Zertifikates bei STARTTLS
ssldiagnos – i.W. Test der Cipher-Suite; https, sip, smtpls, popssl, ftpssl
sslscan – i.W. Test der Cipher-Suite
ssltest.pl – i.W. Test der Cipher-Suite
SSLAudit.exe –
SSl Vulnerabilities Analyzer – GUI fuer sslscan (Windows)
TestSSLServer.jar – sehr einfache Tests (Protokoll, Cipher-Suite)
//www.ssllabs.com/ – online Test mit Scoring (Protokoll, Cipher-Suite)

Browser-Konfiguration: Alptraum

- Auswahl der Cipher
  - Mozilla (fast) nur mit `about:config`
  - IE nur via Registry
  - Chrome?
  - Opera?
  - auf Smartphones, Tablets? siehe auch [B-AN]
- Cipher-Reihenfolge festlegen
  - geht nur im IE via Registry

Browser-Konfiguration: Fallen

UPDATE: November 2013

- IE unter Windows XP: kann kein PFS
- IE unter Windows 8: TLS 1.2 geht nur mit EC-Cipher

(Danke, Boris!)

Browser-Konfiguration: Test-Tools

?

http://xkcd.com/538/

Es gibt Browser-Plugins (Calomel, Certificate Watch, Certificate Patrol)

Angriffe: SSL-Handshake-Protokoll

(stolen from [P-RU])

- Cipher suite rollback
- ChangeCipherSpec message drop
- Key exchange algorithm confusion
- Version rollback
- Bleichenbacher Attack on PKCS#1
- Timing based attacks
- ECC based timing attacks
- ECC-based key exchange algorithm confusion attack
- Renegotiation
- THC-SSL-DoS [A-TH]

Angriffe: SSL-ApplicationData-Protokoll

(stolen from [P-RU])

- MAC does not cover padding length
- Weaknesses through CBC usage (aka Padding Oracle Attack)
- Information leakage by the use of compression
- Chosen-Plain-text Attacks on SSL reloaded
- Practical IV Chaining vulnerability (aka BEAST)
- Practical compression based attacks (aka CRIME)

Angriffe: PKI

(stolen from [P-RU])

- Weak cryptographic primitives lead to colliding certificates (MD5 collisions)
- Weaknesses in X.509 certificate constraint checking (siehe auch [P-CC])
- Attacks on Certificate Issuer Application Logic (0-Bytes im Subject)
- Attacking the PKI (Manipulation der OSCP-Response)

Exploits: ~2007

- 1998: Bleichenbacher Attack on PKCS#1 (decrypt preshared master secret in RSA cipher)
- 2002: Chosen-plaintext attck with IV against cipher with CBC mode [A-CP]
- 2002: MSIE nutzt jedes Zertifikat ohne Basic Constraints CA als Zwischenzertifikat, Verisign liefert die Zertifikate dazu
  http://www.thoughtcrime.org/ie-ssl-chain.txt ⇒ 2011: Bug taucht in iOS wieder auf
- 2005: IDN homograph spoofing [A-IS]
- 2007: Certificate spoofing with subjectAltName [A-SA]
- 2007: Secret Backdoor in New Encryption Standard? (Dual_EC_DRBG) [A-BS]

⇒ **6 Probleme in 10 Jahren**

Exploits: ~2008

- 2008: Zertifikat mit MD5 Kollisionen (seit 90er Jahre bekannt)
  → Reaktion: MD5 wird nicht mehr empfohlen
- 2008: CVE-2008-2809: Spoofing via user-trusted subjectAltName (seit 2004 bekannt)
- 2008: CA: Thawte erstellt Zertifikat für www.live.com
- 2008: CA: Comodo erstellt Zertifikat für mozilla.com

⇒ **4 Probleme in einem Jahren**

Exploits: 2009, 2010

- 2009: null-Prefix im subjectName ”*\00doxpara.com”, sslsniff; OCSP kompromitiert [A-00]
- 2009: OCSP Attacken
  → Reaktion Browser deaktivieren OCSP teilweise wieder
- 2009: sslstrip
  → Reaktion ist HSTS
- 2009: Renegotiation Attacke [P-RE]
- 2010: EFF SSL Observatory zeigt Chaos der CA
  → Reaktion Gedanken über "gesetzliche" Regelungen

⇒ 5 Exploits in 2 Jahren

Exploits: 2011

- 03/2011: CA: Comodo Hack
  → Reaktion: Certificate-Pinning in Chrome
- 07/2011: CA: DigiNotar Hack
  → Reaktion: CA von Regierung übernommen
- 07/2011: PeerJacking (Problem in PHP's cURL) [A-PJ]
- 09/2011: CA: Einbruch bei GlobalSign (Folge von DigiNotar Hack)
- 09/2011: BEAST
  → Reaktion: Empfehlung RC4
- 09/2011: weitere gefälschte Zertifikate (Folge von DigiNotar Hack)
- 11/2011: CA: Einbruch bei KPN

⇒ **7 Exploits in einem Jahr**

Exploits: 2012

- 02/2012: Trustwave verkauft HSM-Box mit Feature für MiTM
- 06/2012: Malware Flame mit "einem" MS Cert
- 09/2012: Adobe hacked "Inappropriate Use of Adobe Code Signing Certificate" [A-AD]
- 09/2012: CRIME (betroffen: Chrome, Firefox, nicht IE) [A-CR]
  → Reaktion: TLS/SSL/level compression deaktivieren

⇒ 4 Probleme in 1 Jahr

Exploits: 2013

- 01/2013: CA: Türktrust stellt *.google.com aus
- 02/2013: Lucky 13 attack
  → Reaktion: RC4 **nicht** empfohlen
- 03/2013: RC4 attack [A-RC]
  → Reaktion: RC4 **nicht** empfohlen
- 06/2013: How to botch TLS forward secrecy (PFS) [A-FS]
- 07/2013: TIME: A Perfect CRIME? Only TIME Will Tell [A-TI]
  → Reaktion: Workarounds nur in der Applikation: Anti-CSRF-Token, X-Frame-Option
- 08/2013: BREACH:
  → Reaktion: große Ratlosigkeit (Workarounds nur in der Applikation, aber nicht in SSL möglich!)
- 09/2013: Dual EC_DRBG von NSA kompromitiert
  → Reaktion: NIST zieht Algorithmus(-Empfehlung) zurück [M-RS]

⇒ **7 Probleme in 1 Jahr**

PKI: Was sind die Probleme?

- ("Trust"): keine Transparenz, viele private Firmen
- (CA): jeder darf alles, z.B. Rechte an andere weitergeben
- (Chain): Prinzip des schwächsten Glieds
- (OCSP): nicht performant, daher meist abgeschalten; Privacy!
- (Revokation): schlechter Mechanismus
- keine proaktive Sicherheit
  - Beispiel wäre eine "Certificate Positiv List" statt CRL
- mehr als 600 Root-CA im Browser
- und dann kommt PaloAlto, NSA, MI6, ...

SSL: "broken security design"

- CRL: was ist wenn Server nicht erreichbar ist?
- CRL: was ist wenn alte CRL geliefert wird?
- CRL: Problem mit DNS-Spoofing
- OCSP: Problem mit DNS-Spoofing
- Sicherheit der Root-CA hängt am schwächsten Glied:
  ⇒ eine der 600 CAs ...
- **eine** kaputte CA betrifft **alle** User

SSL/TLS : ein Standard?

Es gibt viele Implementierungen:

- openssl (nur TLS)
- GnuTLS
- Mozilla NSS
- Microsoft SSPI
- Chrome Speedy
- Adobe, Opera, Safari, ...
- Java (JSEE und javax.net.ssl)
- viele propritäre Systeme ...

SSL/TLS : ein Standard?

Es gibt viele Implementierungen:

- openssl (nur TLS)
- GnuTLS
- Mozilla NSS
- Microsoft SSPI
- Chrome Speedy
- Adobe, Opera, Safari, ...
- Java (JSEE und javax.net.ssl)
- viele propritäre Systeme ...

und die müssen alle miteinander reden → Kompatibiltät

Angriff: Renegotiation

- kann der Server initiieren: [✔]
- kann der Client initiieren: [✗] → DoS
  Ursache – Handshake ist nicht symetrisch, d.h. kryptographische Berechnung auf dem Server viel komplexer als auf dem Client [P-RE], [IETF].
  Angriff – Client öffnet wenige (400) Sockets, baut Verbindung auf und fordert jeweils Renegotiation an.
  Maßnahme – Renegotiation nur vom Server initiieren

Siehe Server based DoS vulnerabilities in SSL/TLS Protocols [W-DO] und THC-SSL-DoS [A-TH].

Angriff: BEAST

BEAST ist ein "block-wise chosen-plaintext"-Angriff.

Ursache – Der Browser verschlüsselt den Text und kann dann auf die Entropie schließen. Dadurch wird das Brechen der Verschlüsselung erheblich vereinfacht. [A-CB], [A-CP]

Angriff – Angreifer fügt bekannten Text am Anfang ein, z.B. in URL (mittels JavaScript im Browser). [A-BE]

Maßnahme – TLSv1.2 verwenden oder Cipher mit CBC deaktivieren, nur RC4 Cipher.

→ Problem seit 2006 (oder sogar 2002 bekannt)

Gute Beschreibung http://blogs.msdn.com/b/kaushal/archive/2011/10/03/taming-the-beast-browser-exploit-against-ssl-tls.aspx

Angriff: BREACH

BREACH ist ähnlich wie CRIME, benutzt aber die Server-Response.

Ursache – HTTP-Kompression verrät wo in den verschlüsselten Daten bestimmte Informationen stehen. [A-BR]

Angriff – Browser sendet Daten, die in der Response reflektiert werden. Es gibt Tools dafür [C-BR].

Maßnahme – HTTP-Kompression abschalten sonst nur mit Hilfe der Anwendung möglich, z.B. zufälliger Wert im HTTP-Header.

Angriff: CRIME

Ursache – TLS-Kompression verrät wo in den verschlüsselten Daten bestimmte Informationen stehen.

Angriff – Browser sendet Pakete mit unterschiedlich langen Payloads, aus der Größe der verschlüsselten Pakete lässt sich erkennen was die Daten sind. [A-CR]

Maßnahme – Kompression deaktivieren.

Angriff: Padding Oracle Attack

Problem bereits 2002 beschrieben: CBC nicht Teil der Prüfsumme [A-OA]. 2010 Exploit durch Schwachstellen (Information Disclosure) verschiedener SSL-Implementierungen.

Ursache – SSL liefert unterschiedliche Fehlermeldungen, wenn CBC nicht stimmt.

Angriff – verschiedene Pakete senden und Fehler analysieren; daraus kann letztendlich der Schlüssel (eigenlich der IV) zurückgerechnet werden. [A-OP]

Maßnahme – Cipher mit CBC deaktivieren, nur RC4 Cipher.

Angriff: Lucky 13 Attack

Ist i.W. Erweiterung der Padding Oracel Attack. Wenn keine Fehlermeldungen geliefert werden, kann man u.U. über das Zeitverhalten Rückschlüsse auf korrekte oder falsche Pakete schließen. [A-13]

Maßnahme – Cipher mit "authenticated encryption algorithm" verwenden (z.B. AES-GCM, AES-CCM) nur mit TLS 1.2 möglich.

Angriff: RC4 Bias Attack

Prinzip ähnlich wie Lucky13.

Ursache – Statistische Analyse vieler identischer Pakete erlaubt Rückschlüsse auf die Verschlüsselung.

Angriff – Angreifer sendet sehr viele identische (plaintext) Pakete und analysiert die Antworten.

Maßnahme – RC-Cipher deaktivieren, oder korrigierte (patched) Versionen vewenden.

Siehe Beschreibung in [A-RC].

Angriff: TIME

Erweiterung der CRIME Attacke, Analyse der Response statt des Requests.

Ursache – Kompression verrät wo in den verschlüsselten Daten bestimmte Informationen stehen.

Angriff – Durch das Zeitverhalten kann der ursprüngliche Text zeichenweise bestimmt werde. Funktioniert als MiTM oder mittels JavaScript im Borwser. [A-TI]

Maßnahme – nur mit Hilfe der Anwendung möglich, z.B. zufälliger Wert im HTTP-Header; CSRF verhindern.

Angriff: HSTS

- MiTM is bei allererstem Zugriff erfolgreich!
- Widerspruch bei `maxage`:
  - grosser Wert → Gefahr von ausgelaufenen, zurückgezogenen Zertifikaten
  - kleiner Wert → Gefahr von neuer MiTM
- Timeout beim Tests: viele Browser akzeptieren dann die Chain
- DNS-Spoofing (in Kombination mit Timeouts)

Tools

```
nmap --script ssl-enum-ciphers -p 443 localhost
sslscan --no-failed localhost:443
ssltest.pl          localhost 443
ssltest.pl   -g     localhost 443
SSLAudit.pl         localhost
sslyze              localhost --sslv3 --tlsv1 --tlsv1_1 --tlsv1_2
sslyze --hide_rejected_ciphers localhost --sslv3 ...
TestSSLServer.jar   localhost
ssldiagnos.exe      localhost
o-saft.pl           localhost
o-saft.pl           localhost:443
o-saft.pl  -p 443   localhost --enabled
```

Online-Tools

http://www.ssllabs.com/ https://sslguru.com/ssl-tools/check-ssl-certificate.html http://certlogik.com/ssl-checker/ http://www.sslshopper.com/ssl-checker.html

O-Saft

help ToDo Glossar

--host demo --port

\+ + + +

version, ciphers, info, cipher, check, sizes, sni, sni_check, quick, http, ...

--cipher yeast ALL, MEDIUM, !EDH, DEFAULT, yeast ...

--sslv2 --sslv3 --tlsv1

--no-sslv2 --no-sslv3 --no-tlsv1 --nullsslv2

--sni --no-sni --http

--dns --no-dns --no-cert --force-openssl

--enabled --disabled

--legacy compact, full, quick, ssltest, sslaudit, sslyze, ssldiagnos, ...

--format hex, raw --short

--separator --timeout

-help={commands,check,legacy,compliance,score}

https://owasp.org/index.php/O-Saft https://github.com/OWASP/O-Saft

Glossar

ADH – Anonymous Diffie-Hellman (auch DH_anon)
CRL – Certificate Revocation List
DHE – Diffie-Hellman Ephemeral
EDH – Ephemeral Diffie-Hellman
EV – Extended Validation
FIPS – FIPS Security Requirements for Cryptographic Modules
GCM – Galois/Counter Mode (block cipher mode)
HSTS – HTTP Strict Transport Security
OSCP – Online Certificate Status Protocol
SNI – Server Name Indication

Glossar: Attacks

BEAST – Browser Exploit Against SSL/TLS
BREACH – Browser Reconnaissance & Exfiltration via Adaptive Compression of Hypertext
CRIME – Compression Ratio Info-leak Made Easy (Exploit SSL/TLS)
Lucky13 – Attack RC4
RC4 – Attack RC4
TIME – Timing Info-leak Made Easy
A Perfect CRIME? TIME Will Tell

Referenzen

[P-CC] **The Most Dangerous Code in the World:**
Validating SSL Certificates in Non-Browser Software
shmat_ccs12.pdf

[P-RE] **TLS / SSLv3 renegotiation vulnerability explained**
Thierry Zoller, http://www.g-sec.lu, http://blog.zoller.lu
practicaltls.pdf

[A-OA] **Security Flaws Induced by CBC Padding Applications to SSL...**
Serge Vaudenay, EUROCRYPT 2002
http://www.iacr.org/cryptodb/archive/2002/EUROCRYPT/2850/2850.pdf

Referenzen

[IETF] **TLS Renegotiation Vulnerability**
http://tools.ietf.org/agenda/76/slides/tls-7.pdf

[W-DO] **Server based DoS vulnerabilities in SSL/TLS Protocols**
Sukalp Bhople (Master Thesis), Eindhoven University of Technology

[A-AD] **Adobe hacked**
http://blogs.adobe.com/asset/2012/09/inappropriate-use-of-adobe-code-signing-certificate.html

[W-CP] **Chosen-plaintext attack**
http://en.wikipedia.org/wiki/Chosen-plaintext_attack

Referenzen: Standards

[W-FS] **Perfect Forward Secrecy**
http://en.wikipedia.org/wiki/Perfect_forward_secrecy

[W-TS] **TLS - Transport Layer Security**
http://en.wikipedia.org/wiki/Transport_Layer_Security

[P-CA] **Guide to Webserver SSL Certificates**
https://calomel.org/ssl_certs.html

Referenzen: Standards

[RFC2246] **The Transport Layer Security (TLS) Protocol Version 1.0**
http://www.ietf.org/rfc/rfc2246.txt

[RFC4346] **The Transport Layer Security (TLS) Protocol Version 1.1**
http://www.ietf.org/rfc/rfc4346.txt

[RFC5246] **The Transport Layer Security (TLS) Protocol Version 1.2**
http://www.ietf.org/rfc/rfc5246.txt

[RFC2818] **HTTP Over TLS**
http://www.ietf.org/rfc/rfc2818.txt

[RFC3546] **Transport Layer Security (TLS) Extensions (SNI)**
http://www.ietf.org/rfc/rfc3546.txt

Referenzen: Empfehlungen

[W-CC] **Convergence Concept**
http://en.wikipedia.org/wiki/Convergence_%28SSL%29

[P-RU] **Lessons Learned From Previous SSL/TLS Attacks**
A Brief Chronology Of Attacks And Weaknesses
Christopher Meyer und Jörg Schwenk, Horst Görtz Institute for IT-Security, Ruhr-University Bochum

[7] **SSL/TLS Deployment Best Practices** ,Ivan Ristić
https://www.ssllabs.com/downloads/SSL_TLS_Deployment_Best_Practices_1.2.pdf

[P-SG] **X.509 Style Guide** , Peter Gutmann
http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt

[P-UF] **Security Usability Fundamentals** , Peter Gutmann
http://www.cs.auckland.ac.nz/~pgut001/pubs/usability.pdf

[P-CP] **Certificate Pinning Extension for HSTS** , Chris Evans, Chris Palmer
http://tools.ietf.org/html/draft-evans-palmer-hsts-pinning-00
pdfnSTRd9kYcY.pdf

[P-BSI] **BSI Technischiche Richtlinie: TR-02102-2** , 2012
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2_pdf.pdf?__blob=publicationFile

Referenzen: Attacks

[A-BE] **BEAST** , Juliano Rizzo, Thai Duong
http://www.kb.cert.org/vuls/id/864643
http://blogs.msdn.com/b/kaushal/archive/2011/10/03/taming-the-beast-browser-exploit-against-ssl-tls.aspx
A Challenging but Feasible Blockwise-Adaptive Chosen-Plaintext Attack on SSL

[A-BR] **BREACH** , Yoel Gluck, Neal Harris, Ángel Prado
http://www.kb.cert.org/vuls/id/987798
http://breachattack.com/resources/BREACH%20-%20SSL,%20gone%20in%2030%20seconds.pdf

[A-CR] **The CRIME attack** , Juliano Rizzo, Thai Duong
https://docs.google.com/presentation/d/11eBmGiHbYcHR9gL5nDyZChu_-lCa2GizeuOfaLU2HOU
/edit?pli=1#slide=id.g1de53288_0_16
http://arstechnica.com/security/2012/09/crime-hijacks-https-sessions/

Referenzen: Attacks

[A-00] **Null Prefix Attacks Against SSL/TLS Certificates**
Moxie Marlinspike, 07/2009 (using sslsniff)
http://www.thoughtcrime.org/papers/null-prefix-attacks.pdf

[A-13] **Lucky Thirteen attack**
Nadhem J. AlFardan and Kenneth G. Paterson, Royal Holloway, University of London
http://www.isg.rhul.ac.uk/tls/Lucky13.html
https://www.imperialviolet.org/2013/02/04/luckythirteen.html

[A-PJ] **PeerJacking (Problem in PHP's cURL)**
http://www.unrest.ca/peerjacking

[A-OP] **Practical Padding Oracle Attacks**
Juliano Rizzo, Thai Duong, USENIX WOOT 2010.
http://www.usenix.org/event/woot10/tech/full_papers/Rizzo.pdf

Referenzen: Attacks

[A-RC] **RC4 attack**
Nadhem J. AlFardan, Dan Bernstein, Kenny G. Paterson, Bertram Poettering and Jacob Schuldt, Royal Holloway, University of London
http://www.isg.rhul.ac.uk/tls/
http://blog.cryptographyengineering.com/2013/03/attack-of-week-rc4-is-kind-of-broken-in.html

[A-SA] **Certificate spoofing with subjectAltName and domain name wildcards**
Nils Toedtmann
http://nils.toedtmann.net/pub/subjectAltName.txt

[A-TI] **A Perfect CRIME? TIME will tell** , Tal Be'ery, Amichai Shulman
https://www.owasp.org/images/e/eb/A_Perfect_CRIME_TIME_Will_Tell_-_Tal_Beery.pdf

Referenzen: Attacks

[A-TH] **THC-SSL-DoS** , The Hackers Choice
http://www.thc.org/thc-ssl-dos/

[A-IS] **IDN homograph spoofing** , Eric Johanson
http://www.shmoo.com/idn/homograph.txt

[A-BS] **Did NSA Put a Secret Backdoor in New Encryption Standard?** , Bruce Schneier
http://www.wired.com/politics/security/commentary/securitymatters/2007/11/securitymatters_1115
based on:
http://eprint.iacr.org/2006/190, http://eprint.iacr.org/2007/048

[A-FS] **How to botch TLS forward secrecy**
https://www.imperialviolet.org/2013/06/27/botchingpfs.html

Referenzen: Maßnahmen

[M-BE] **Mitigating the BEAST attack on TLS** , Ivan Ristić
https://community.qualys.com/blogs/securitylabs/2011/10/17/mitigating-the-beast-attack-on-tls
https://community.qualys.com/blogs/securitylabs/2013/08/07/defending-against-the-breach-attack

[M-BR] **Defending against the BREACH attack** , Ivan Ristić
https://community.qualys.com/blogs/securitylabs/2013/08/07/defending-against-the-breach-attack

[M-RC] **RC4 in TLS is Broken: Now What?** , Ivan Ristić
https://community.qualys.com/blogs/securitylabs/2013/03/19/rc4-in-tls-is-broken-now-what

Referenzen: Maßnahmen

[M-RS] **RSA Tells Its Developer Customers: Stop Using NSA-Linked Algorithm**
http://www.wired.com/threatlevel/2013/09/rsa-advisory-nsa-algorithm/

[M-CR] **How to beat the BEAST successor?**
http://security.stackexchange.com/questions/19911/crime-how-to-beat-the-beast-successor/19914

[M-SS] **ATTACKS ON SSL A COMPREHENSIVE STUDY**
www.isecpartners.com/media/106031/ssl_attacks_survey.pdf

Referenzen: unsortiert

[x-xx] **Use of RSA Algorithm without OAEP**
http://cwe.mitre.org/data/definitions/780.html

[A-CB] **Not Using a Random IV with CBC Mode**
http://cwe.mitre.org/data/definitions/329.html

[A-CP] **Chosen-plaintext attck with IV against cipher with CBC mode**
http://www.mail-archive.com/openssl-dev@openssl.org/msg10664.html

[C-BR] **BREACH code**
https://github.com/nealharris/BREACH

[x-MZ] **Not binding X.509 certificate to originating domain name allows certificate spoofing**
https://bugzilla.mozilla.org/show_bug.cgi?id=402347

[B-AN] **Android-Verschlüsselung wurde verschlimmbessert**
http://www.heise.de/security/meldung/Android-Verschluesselung-wurde-verschlimmbessert-1979572.html