Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Bakalářská práce

Zabezpečený síťový protokol HTTPS

Jiří Navrátil

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jiří Navrátil

Informatika

Název práce

Zabezpečený síťový protokol HTTPS

Název anglicky

Secured network protocol HTTPS

Cíle práce

Cílem práce je porovnat existující protokoly rozšiřující end-to-end HTTP komunikaci v počítačových sítích TCP o zabezpečení. Dílčím cílem je vytvořit přehled zranitelností jednotlivých protokolů a stanovit sadu doporučení pro správce serverů služby WWW k praktickému zabezpečení komunikace na bázi HTTPS.

Metodika

Autor analyzuje odbornou literaturu a s využitím konzultací s odborníky z praxe stanoví jednotlivé použitelné protokoly. Následně je jednotlivě charakterizuje, včetně jejich vývojových stupňů a vzájemných vazeb. Autor dále vybere vhodnou komparační metodu pro porovnání vybraných protokolů v prostředí typického serveru/klienta a vytvoří doporučení pro jejich nasazení v praxi. V praktické části pak na základě tohoto doporučení vytvoří modelovou konfiguraci WWW serveru pro zabezpečenou komunikaci s využitím protokolu HTTPS.

Doporučený rozsah práce

40

Klíčová slova

HTTP, HTTPS, SSL, TLS, WWW

Doporučené zdroje informací

DIERKS, Tim a Eric RESCORLA (eds.). Request for Comments: 5246 – The Transport Layer Security (TLS) Protocol Version 1.2 [online]. 2008 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc5246.txt

DIERKS, Tim, ALLEN, Christopher (ed.). Request for Comments: 2246 – The TLS Protocol Version 1.0 [online]. 1999 [cit. 2016-03-03]. Dostupné z: https://www.ietf.org/rfc/rfc2246.txt

FREIER, Alan O., Philip KARLTON a Paul C. KOCHER. Request for Comments: 6101 – The Secure Sockets Layer (SSL) Protocol Version 3.0 [online]. 2011 [cit. 2016-03-03]. ISBN ISSN: 2070-1721. Dostupné z: https://tools.ietf.org/rfc/rfc6101.txt

RESCORLA, Eric and contributors . The Transport Layer Security (TLS) Protocol Version 1.3: draft-ietf-tls-tls13-16 [online]. 22.09.2016. [cit. 2016-10-20]. Dostupné z: https://tools.ietf.org/id/draft-ietf-tls-tls13-16.txt

RESCORLA, Eric, DIERKS, Tim (ed.). Request for Comments: 4346 – The Transport Layer Security (TLS) Protocol Version 1.1 [online]. 2006 [cit. 2016-03-03]. ISBN smazat. Dostupné z: https://www.ietf.org/rfc/rfc4346.txt

RISTIC, Ivan. Bulletproof SSL and TLS. London: Feisty Duck, 2015. ISBN 9781907117046

SWORD, Helen. Stylish academic writing. Cambridge, Mass.: Harvard University Press, 2012, viii, 220 p. ISBN 9780674064485

Předběžný termín obhajoby

2016/17 ZS – PEF

Vedoucí práce

Ing. Tomáš Vokoun

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 31. 10. 2016

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 1. 11. 2016

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 11. 11. 2016

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Zabezpečený síťový protokol HTTPS" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce.

V Praze dne 21.09.2016 _____

Poděkování

Dovoluji si tímto poděkovat Ing. Tomáši Vokounovi za vedení a rady. Dále bych chtěl poděkovat mým dětem Dušanovi, Barborce, Kačence, Elišce a Pepíčkovi za podporu a pochopení, které mi věnovaly během tvorby této práce i celého studia. Barborce dále děkuji za nakreslení obrázků.

Zabezpečený síťový protokol HTTPS

Bakalářská práce nejprve vysvětluje důvody pro zavedení zabezpečeného síťového protokolu HTTPS jako rozšíření síťového protokolu HTTP.

Následně popisuje a porovnává protokoly, které toto rozšíření umožňují, včetně upozornění na známé zranitelnosti. Výsledky jsou použity k praktickému doporučení pro správce serverů WWW služeb. Dále jsou uvedeny doporučení pro uživatele webových prohlížečů.

Závěrem je poskytnuta modelová konfigurace WWW služeb pro využití zabezpečeného síťového protokolu HTTPS.

Klíčová slova: HTTP. HTTPS, SSL, TLS, WWW

Secured network protocol HTTPS

Bachelor thesis explain the motivation behind extension of network protocol HTTP to the secured network protocol HTTPS.

Thesis describe and compare protocols, which can be used for such extension. These protocols are compared and their known vulnerabilities are listed and commented. Results are used for recommendations to WWW servers' administrators and to regular web browser s' users.

Thesis also include practical example of WWW server configuration with enabled secured network protocol HTTPS.

Keywords: HTTP. HTTPS, SSL, TLS, WWW

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Informační technologie mají uživatelům pomáhat při řadě činností. Jejich užívání ale přináší i různá rizika.

Počítače, chytré telefony, tablety a další zařízení umožňují přístup k veřejným informacím a také k informacím důvěrným: emailům, bankovním účtům, fotografiím, textovým či tabulkovým dokumentům a dalším osobním či firemním datům. Data, zejména ta důvěrná, je nutné chránit.

Jedna z oblastí, kde je potřeba data chránit, je zveřejňování a případné zpracování dat s využitím webových stránek. Při tomto modelu výměny dat zajistí webový server reprezentování dat ve formě webových stránek a webový klient se postará o jejich prohlížení a případné editování. Ke komunikaci mezi webovým serverem a webovým prohlížečem byl navržen protokol HTTP. Tento protokol neřeší bezpečnost, proto byl následně navržen zabezpečený síťový protokol HTTPS, který rozšiřuje HTTP komunikaci o zabezpečení. HTTPS k tomu využívá různé verze protokolů SSL (Secure Socket Layer) a TLS (Transport Layder Security). Následující kapitoly se HTTPS, SSL a TLS podrobně věnují.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je porovnat existující protokoly rozšiřující end-to-end HTTP komunikaci v počítačových sítích TCP o zabezpečení. Dílčím cílem je vytvořit přehled zranitelností jednotlivých protokolů a stanovit sadu doporučení pro správce serverů služby WWW k praktickému zabezpečení komunikace na bázi HTTPS.

2.2 Metodika

Autor analyzuje odbornou literaturu a s využitím konzultací s odborníky z praxe stanoví jednotlivé použitelné protokoly. Následně je jednotlivě charakterizuje, včetně jejich vývojových stupňů a vzájemných vazeb. Autor dále vybere vhodnou komparační metodu pro porovnání vybraných protokolů v prostředí typického serveru/klienta a vytvoří doporučení pro jejich nasazení v praxi. V praktické části pak na základě tohoto doporučení vytvoří modelovou konfiguraci WWW serveru pro zabezpečenou komunikaci s využitím protokolu HTTPS.

3 Teoretická východiska

3.1 HTTP

Protokol HTTP (anglicky celým názvem „Hypertext Transfer Protocol“) se využívá k obousměrnému přenosu strukturovaného textu a dalších dat mezi webovým serverem a webovým prohlížečem. Pochází z dob, kdy mezi jeho uživatele patřily především univerzity a uživatelům se důvěřovalo. Bezpečnost se v té době neřešila. S rozvojem Internetu se rozšířila skupina uživatelů a také způsoby použití WWW (anglicky celým názvem „World Wide Web“). Pro přenos dat jako heslo, rodné číslo (číslo sociálního zabezpečení osoby), číslo platební karty nebo provedení bankovního příkazu a tak podobně, bylo potřeba doplnit protokol HTTP o zabezpečení. Byl navržen zabezpečený síťový protokol HTTPS.

3.2 HTTPS

Tato práce popisuje zabezpečený síťový protokol HTTPS definovaný v RFC2818 „HTTP Over TLS“ a využívající pro zabezpečenou komunikaci port 443 a URI (anglicky celým názvem „Uniform Resource Identitifer“) schéma HTTPS resp. https. Bezpečnostní rozšíření zajišťují protokoly SSL (anglicky celým názvem „Secure Sockets Layer“) a TLS (anglicky celým názvem „Transport Layer Security“).

Existuje také RFC2817 „Upgrading to TLS Within HTTP/1.1“, který definuje, jak využít TLS v rámci HTTP/1.1 (anglicky celým názvem „Hypertext Transfer Protocol“) na portu 80 v rámci URI schématu HTTP resp. http. Tato varianta zabezpečení HTTP není v práci popisována.

3.3 HTTP a HTTPS v rámci TCP/IP

Internet je založen na TCP (Transmission Control Protocol) / IP (Internet Protocol). IP protokol zajišťuje přenos datagramů na síťové vrstvě. Tento přenos je nespolehlivý a nezabezpečený. Na transportní vrstvě výše je TCP protokol, který zajišťuje transportní služby. Je navržen jako kontrolovaný a spolehlivý, ale nikoli jako zabezpečený. V aplikační vrstvě nad TCP protokolem pracuje HTTP. HTTP protokol byl původně navržen jako nezabezpečený, ale v současné době již existují možnosti, jak zabezpečení

doplnit. Tato práce se zaměřuje na přidání bezpečnosti do transportní vrstvy pomocí protokolů SSL (Secure Sockets Layer) a TLS (Transport Layer Security). Takto zabezpečený protokol HTTP je označován HTTPS, používá URI schéma HTTPS resp. https a má přiřazený port 443.

Existují možnosti, jak přidat zabezpečení do síťové vrstvy. Jedna z možností je IPsec. Z pohledu bezpečnosti je ideální přidání zabezpečení do více vrstev. V této práci není využití IPsec popisováno. V případě zabezpečení síťové vrstvy pomocí IPsec bude toto zabezpečení pro transportní i aplikační vrstvu transparentní, tj. na způsobu nasazení HTTPS se nic nezmění.

3.4 Rizika při komunikaci HTTP

Pro popsání rizik u nezabezpečené HTTP komunikace je využit obrázek č. 1 s osobami reprezentující komunikaci po nezabezpečeném Internetu.

Alice a Bob spolu komunikují. Na trase mezi Alicí a Bobem se k infrastruktuře dostaly Eve a Mallory. Obě mají špatné úmysly. Eve se snaží komunikaci odposlouchávat (anglicky „eavesdropping“) a Mallory (jméno podobné anglickému „malicious“) se snaží do komunikace zasahovat, například měnit obsah zpráv. Eve se může dozvědět, kam Alice pozvala Boba na večeři a její telefonní číslo. Mallory může zaměnit odpověď Boba. Bob sice na pozvání od Eve odpoví, že na večeři přijde, ale Mallory odpověď změní a Eve obdrží od Mallory odpověď, že Bob nepřijde a navíc, že už Eve nechce nikdy vidět. Eve netuší, že byla odpověď změněna, a tak považuje odpověď za autentickou, tj. od Boba a následně se podle toho zařídí.

Chování Eve i Mallory je pro komunikaci Alice a Boba nežádoucí. Mallory znamená větší a horší zásah do komunikace než Eve. Pokud by Alice a Bob použili pro komunikaci zabezpečení pomocí HTTPS, tak by sice Eve i Mallory viděly probíhající komunikaci, ale ta by pro ně byla nečitelná a Mallory by nemohla zprávy zaměňovat.

Obrázek 1 – Alice a Bob

3.5 SSL a TLS

Protokoly SSL (Secure Sockets Layer) a TLS (Transport Layer Security) umožňují zabezpečit HTTP přidáním bezpečnosti do transportní vrstvy. Takto zabezpečený protokol HTTP je označován HTTPS, používá URI schéma HTTPS, resp. https a má přiřazený port 443.

První verze 1.0 SSL protokolu byla vypracována společností společnost Netscape Communications Corp., ale nebyla nikdy zveřejněna, neboť obsahovala v návrhu vážné bezpečnostní chyby.

Druhá verze 2.0 SSL protokolu byla zveřejněna dne 09.02.1995. Specifikaci „The SSL Protocol" vydala společnost Netscape Communications Corp. Tato verze byla pro bezpečnostní nedostatky zakázána RFC6176 „Prohibiting Secure Sockets Layer (SSL) Version 2.0“ v březnu 2011.

Třetí a současně poslední verze 3.0 SSL protokolu byla zveřejněna dne 18.11.1996 společnosti Netscape Communications Corp. a později vydána IETF (anglicky celým názvem Internet Engineering Task Force) v nezměněné podobě jako RFC6101 v srpnu 2011. Tato verze byla napsána znovu, aby se vyhnula chybám předchozí verze. Přesto byla i tato verze pro bezpečnostní nedostatky zakázána, a to RFC7568 „Deprecating Secure Sockets Layer Version 3.0“ v červnu 2015.

```
SSLv2 zakázána RFC6176 v srpnu 2011
SSLv3 zakázána RFC7568 v červnu 2015
```

Tabulka 1 – zakázané verze SSL

Protokol TLS navazuje na SSL. Protokol SSL vydávala společnost Netscape Communications Corp. a protokoly TLS začala vydávat IETF (anglicky celým názvem Internet Engineering Task Force).

První verze 1.0 TLS vycházela ze SSL 3.0 a byla zveřejněna v lednu 1999. RFC2246 „The TLS Protocol Version 1.0“ vydala IETF (anglicky celým názvem Internet Engineering Task Force). Mezi slabiny patří možnost ponížit spojení na SSLv3, což snižuje bezpečnost.

Druhá verze 1.1 TLS byla zveřejněna v dubnu 2006. RFC4346 „The Transport Layer Security (TLS) Protocol Version 1.1“ vydala IETF (anglicky celým názvem Internet Engineering Task Force). Oproti verzi 1.0 zakazuje zpětnou kompatibilitu s SSL a vyřazuje další nebezpečné vlastnosti.

Třetí a zatím poslední vydaná verze 1.2 TLS byla zveřejněna v srpnu 2008. RFC5246 „The Transport Layer Security (TLS) Protocol Version 1.2“ vydala IETF (anglicky celým názvem Internet Engineering Task Force).

Čtvrtá verze 1.3 TLS zatím nebyla zveřejněna. Doposud je v připomínkování IETF (anglicky celým názvem Internet Engineering Task Force). První pracovní verze byla zveřejněna 17.04.2014, druhá pracovní verze byla zveřejněna 07.07.2014, třetí pracovní verze byla zveřejněna 27.10.2014, čtvrtá pracovní verze byla zveřejněna 03.01.2015, pátá pracovní verze byla zveřejněna 09.03.2015, šestá pracovní verze byla zveřejněna 29.06.2015, sedmá pracovní verze byla zveřejněna 08.07.2015, osmá pracovní verze byla zveřejněna 28.08.2015, devátá pracovní verze byla zveřejněna 05.10.2015, desátá pracovní verze byla zveřejněna 19.10.2015, jedenáctá pracovní verze byla zveřejněna 28.12.2015, dvanáctá pracovní verze byla zveřejněna 21.03.2016, třináctá pracovní verze byla zveřejněna 22.05.2016, čtrnáctá pracovní verze byla zveřejněna 11.07.2016, patnáctá pracovní verze byla zveřejněna 17.08.2016, šestnáctá pracovní verze byla zveřejněna 22.09.2016, sedmnáctá pracovní verze byla zveřejněna 20.10.2016 a osmnáctá pracovní verze byla zveřejněna 26.10.2016. Návrh vychází z verze 1.2 a snaží se odstranit jeho známé či potenciální slabosti. Níže jsou citovány některé navrhované změny oproti verzi 1.2. Kompletní seznam navrhovaných změn je umístěn v příloze 8.1.

- Restructure PSK key exchange negotiation modes
- Harmonize requirements about cipher suite matching: for resumption you need to match KDF but for 0-RTT you need whole cipher suite. This allows PSKs to actually negotiate cipher suites.
- Revise version negotiation
- Forbid CertificateRequest with 0-RTT and PSK.
- Clearer guidance on what is needed for TLS 1.2.
- Explicitly require checking that handshake records not span key changes.
- Allow cookies to be longer.

- Remove the "context" from EarlyDataIndication as it was undefined and nobody used it.
- Define ecdsa_sha1.
- Allow resumption even after fatal alerts. This matches current practice.
- Add a section describing the data limits for each cipher.
- Remove (EC)DHE 0-RTT.
- Provide a list of the PSK cipher suites.
- Revise signature algorithm negotiation to group hash, signature algorithm, and curve together. This is backwards compatible.
- Make ticket lifetime mandatory and limit it to a week.
- Make the purpose strings lower-case. This matches how people are implementing for interop.
- Port the CFRG curves & signatures work from RFC4492bis.
- Add support for version anti-downgrade mechanism.
- Unify authentication modes. Add post-handshake client authentication.
- Change to RSA-PSS signatures for handshake messages.
- Remove support for DSA.
- Update key schedule per suggestions by Hugo, Hoeteck, and Bjoern Tackmann.
- Add support for per-record padding.
- Switch to encrypted record ContentType.
- Change HKDF labeling to include protocol version and value lengths.
- Shift the final decision to abort a handshake due to incompatible certificates to the client rather than having servers abort early.
- Deprecate SHA-1 with signatures.
- Add MTI algorithms.
- Remove support for weak and lesser used named curves.
- Remove support for MD5 and SHA-224 hashes with signatures.
- Update lists of available AEAD cipher suites and error alerts.
- Reduce maximum permitted record expansion for AEAD from 2048 to 256 octets.
- Require digital signatures even when a previous configuration is used.
- Relax certificate_list ordering requirement to match current practice.

- Integration of semi-ephemeral DH proposal.
- Add initial 0-RTT support.
- Prohibit RC4 negotiation for backwards compatibility.
- Prohibit SSL negotiation for backwards compatibility.
- Remove ChangeCipherSpec.
- Remove renegotiation.
- Remove point format negotiation.
- Remove GMT time.
- Merge in support for ECC from RFC 4492 but without explicit curves.
- Rework handshake to provide 1-RTT mode.
- Remove custom DHE groups.
- Remove support for compression.
- Remove support for static RSA and DH key exchange.
- Remove support for non-AEAD ciphers.

Z výše uvedeného výčtu je zřejmé, že návrh TLS verze 1.3 se snaží odstranit známé slabosti protokolu TLS verze 1.2.

Vzhledem k tomu, že jsou všechny SSL verze zakázané, tak bude v následujícím textu upřednostňována zkratka TLS.

3.6 TLS komunikace

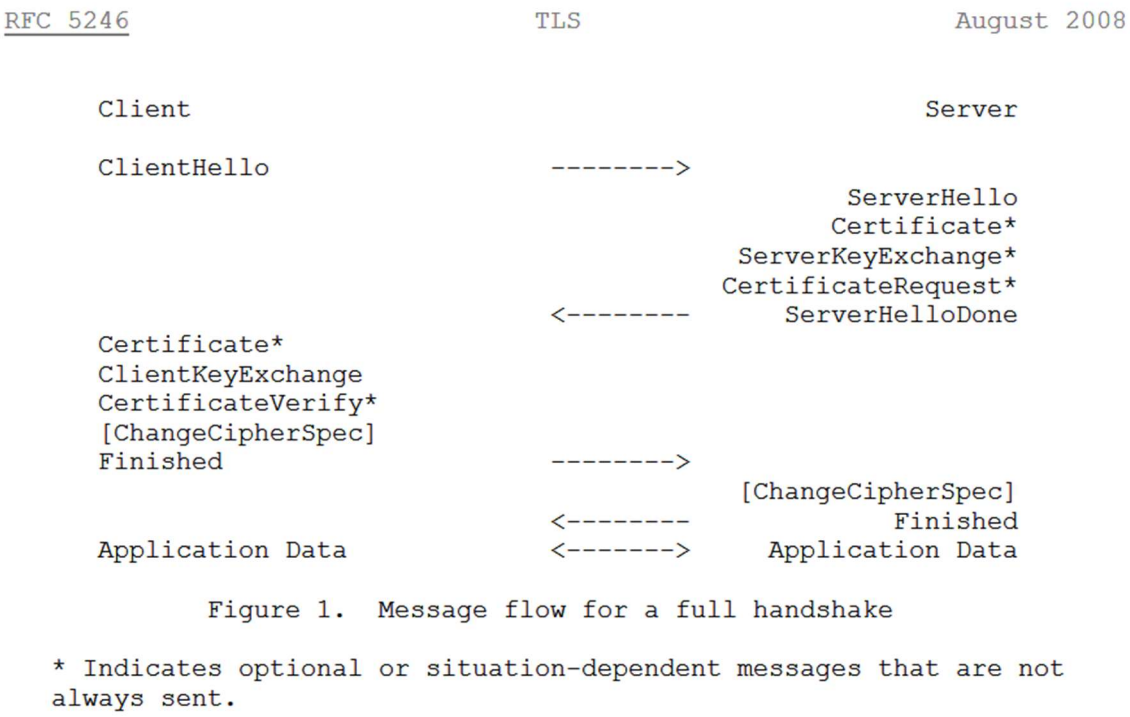
Komunikace zabezpečená pomocí TLS probíhá vždy napřímo mezi dvěma konkrétními zařízeními v Internetu. První zařízení (dále klient) zahajuje připojení s druhým zařízením (dále server) pomocí tzv. „full handshake“. Po jeho úspěšném provedení je možné zasílání dat mezi klientem a serverem zabezpečeně.

Varianta, při které je autentizovaný pouze server, vypadá u TLS 1.2 následovně:

1. Klient spustí „handshake“ a oznámí serveru svoje schopnosti
2. Server vybere parametry
3. Server pošle svůj certifikát
4. Server může poslat další informace pro generování hlavního tajného klíče.
5. Server potvrdí, že je s touto fází hotov.

6. Klient pošle další informace pro generování hlavního tajného klíče
7. Klient začíná posílat zprávy šifrované a informuje o tom server.
8. Klient pošle MAC (anglicky celým názvem „message authentication code“) výše přijatých a odeslaných zpráv.
9. Server se přepne na šifrování a informuje klienta.
10. Server pošle MAC (anglicky celým názvem „message authentication code“) výše přijatých a odeslaných zpráv.
11. Následuje přenos zašifrovaných dat

Na obrázku níže je vidět TLS verze 1.2 „full handshake“ tak, jak je popsaný v RFC5246.



Obrázek 2 – TLS 1.2 full handshake

3.7 Klíče pro šifrování

Pro šifrování HTTPS jsou v současnosti nejvhodnější RSA klíče a doporučovaná délka klíče je 2048 bitů. DSA klíče jsou totiž prakticky omezeny kvůli Internet Exploreru na velikost 1024 bitů a ECDSA klíče zatím nejsou dostatečně podporovány certifikačními autoritami. Klíč je nutné zajistit u spolehlivé certifikační autority. Používání „self-signed“

certifikátu není doporučováno. Bezplatně a automatizovaně pomocí skriptů lze certifikát obdržet od Let's Encrypt certifikační autority na https://letsencrypt.org/.

3.8 Šifry

Symetrické šifrování je varianta šifrování, kdy se používá jeden klíč jak pro zašifrování, tak pro dešifrování dat. Je doporučeno používat šifry kombinující šifrování a kontrolu integrity v rámci jednoho algoritmu – AEAD (anglicky celým názvem „Authenticated Encryption with Associated Data"). U blokových šifer je nejznámější AES (anglicky celým názvem „Advanced Encryption Standard"). U proudových šifer se nedoporučuje používat RC4 a doporučuje se používat Salsa20 a Chacha. U režimu s blokovými šiframi se nedoporučuje používat ECB a doporučuje se použít GCM.

Asymetrické šifrování je varianta šifrování, při které se používá veřejný klíč a důvěrný klíč. Pomocí veřejného klíče se data zašifrují a pomocí důvěrného klíče se mohou dešifrovat. Je doporučeno používat RSA (pojmenované dle osob Ron Rivers, Adi Shamir a Leonard Adleman) s délkou 2048 bitů. RSA je možné také použít pro elektronický podpis na rozdíl od jiných asymetrických šifer jako DSA a ECDSA.

3.9 Komprese

Původní návrhy počítaly s kompresí dat před šifrováním. V praxi se komprimace příliš nepoužívala a navíc byla zneužita v CRIME útoku, proto není doporučována a v návrhu TLS verze 1.3 se komprese dat zakazuje.

3.10 Chyby

Bezpečnostní chyba CVE-2014-016 (23) v knihovně OpenSSL známá jako chyba **Heartbleed**, dopustila zranitelnost půl milionu webových serverů. Při implementaci rozšíření nazvaného TLS heartbeat nebyl ošetřen rozsah poskytovaných dat, čehož mohl zneužít útočník a načíst z postiženého serveru až 64 KB paměti. Tato paměť mohla obsahovat důvěrné informace jako hesla a privátní klíče. Oficiální záznam zveřejněný 07.04.2014 uvádí (23) „The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that

trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug.“ Chyba obdržela obrázek a doménu s webem http://heartbleed.com/.

POODLE útok je bezpečnostní chyba publikovaná pod CVE-2014-3566 (23), která může zpřístupnit útočníkovi důvěrná data vynucením degradování spojení na SSLv3. Oficiální záznam vytvořený 14.05.2014 uvádí (23) „The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other products, uses nondeterministic CBC padding, which makes it easier for man-in-the-middle attackers to obtain cleartext data via a padding-oracle attack, aka the "POODLE" issue.“

DROWN útok je bezpečnostní chyba publikovaná pod CVE-2016-0800 (23) může zpřístupnit útočníkovi důvěrná data pomocí kombinování TLS a SSLv2. Oficiální záznam vytvořený 16.12.2015 uvádí „The SSLv2 protocol, as used in OpenSSL before 1.0.1s and 1.0.2 before 1.0.2g and other products, requires a server to send a ServerVerify message before establishing that a client possesses certain plaintext RSA data, which makes it easier for remote attackers to decrypt TLS ciphertext data by leveraging a Bleichenbacher RSA padding oracle, aka a "DROWN" attack.“. Chyba obdržela obrázek a doménu s webem https://drownattack.com/.

3.11 **Implementace**

Existují různé implementace knihoven a nástrojů pro nasazení bezpečnostních protokolů SSL a TLS.

OpenSSL je nejznámější a nejrozšířenější implementace SSL a TLS. Nejsilnější stránkou je velká rozšířenost. Nejslabší stránkou podpora zakázaného protokolu SSL v3.

Knihovna **LibreSSL** je fork OpenSSL od vývojářů OpenBSD. Z původního kódu byl odstraněn SSL protokol a kód pro Ebcdic, DOS, MacOS Classic, Wind16 a VMS. Byly přidány šifry Brainpool, ChaCha, poly1305 a ANSSI FRP256v1. Místa s ASN.1 byla přepsána přímo do kódu. V knihovně LibreSSL byl vyřešen problém s datovým typem time_t . Knihovna bude fungovat i po úterý 19. ledna 2038 03:14:07.

Kompatibilita s OpenSSL je udržována na úrovni API a funkcionalita je opravována a odstraňování uvnitř funkcí. Kód je rozdělen podle funkcionality na

- libssl - TLS knihovna zpětně kompatibilní s OpenSSL
- libtls - nová TLS knihovna poskytující POSIX API a snadnější použití TLS

- libcrypto - knihovna pro kryptografii
- openssl utilita

Bezpečnostní knihovna **GnuTLS** vznikla již před chybou Heartbleed v roce 2003 kvůli poptávce GNU projektů po bezpečnostní knihovně s GPL licencí. Mezi aplikace, které využívají tuto knihovnu se řadí například openconnect.

Firma Google se rozhodla pro vlastní implementaci **BoringSSL**, která vychází z OpenSSL. Google s týmy kolem OpenSSL a LibreSSL spolupracuje.

3.12 **Operační systém**

Bezpečnostní knihovna je v řadě ohledů odkázána na hostující operační systém. Proto je důležité, aby operační systém poskytoval kvalitní generátor pseudonáhodných čísel, náhodné alokování paměti (bloky paměti nejsou alokované za sebou), odkládací paměť byla šifrována, volně dostupný zdrojový kód pro možnost auditování, varování při kompilaci zdrojového kódu bylo zapnuto a využíváno, změny byly před zahrnutím do systému verzování zdrojového kódu kontrolovány a schvalovány a bez restrikcí na export šifrovacích algoritmů a šifer.

Dále jsou užitečné vlastnosti jako W^X (anglicky celým názvem „Write XOR Execute“), kdy je možné do paměti psát nebo ji spouštět, ale nikoli obojí současně nebo například funkce pledge, která poskytuje aplikaci pouze ta práva, která pro svoji práci potřebuje a žádná navíc.

3.13 **Bezpečnostní pravidla**

Výše uvedené je vhodné doplnit o řadu dalších pravidel pro udržení bezpečnosti dat. Software je nutné pravidelně aktualizovat, hesla by měla být složitá, pro každou službu / server jiná a pravidelně obměňovaná, privátní klíče by měly být opatřené heslem a nikdy by neměly opustit chráněné prostředí majitele, fyzický přístup k počítačům a datům by měl kontrolovaný a zabezpečený, při přistupování na web servery, by měla být řádně ověřena jejich identita a nepovolovány bezpečnostní výjimky a nezabezpečené WiFi by se neměly využívat. Tento výčet není úplný. Slouží k připomenutí, že bezpečnost v informačních technologiích je komplexní pojem, a že používání HTTPS bezpečnost zvyšuje, ale nesmí zůstat osamocené.

3.14 **Porovnání SSL a TLS**

Díky výše uvedeným informacím je možné udělat jednoduchá porovnání. V porovnání SSL a TLS je na tom lépe TLS, protože všechny SSL verze jsou v současnosti zakázané. V porovnání konkrétních verzí, je nejnovější TLS verze 1.2 ta nejbezpečnější, a přitom plně podporovaná na straně web serverů i na straně webových klientů. Dále je zřejmé, že dlouho připravovaná TLS verze 1.3 ji nahradí.

Při porovnání SSL a TLS pomocí vícekriteriální analýzy variant, jsou zachyceny v kriteriální matici vlastnosti jako platnost (zda je protokol zakázaný, nedoporučený, povolený či doporučený), podpora ve výše zmíněných knihovnách (počet knihoven, které danou verzi podporují) a podpora ve výše uvedených webových prohlížečích (počet klientů, kteří danou verzi podporují). Bodovací metodou na škále od 1 (nejhorší) do 10 (nejlepší) jsou následně protokoly porovnány.

	platnost	podpora knihovnami (max)	podpora klienty (max)
SSL 2.0	zakázaný	0	0
SSL 3.0	nedoporučený	2	0
TLS 1.0	povolený	4	5
TLS 1.1	povolený	4	5
TLS 1.2	doporučený	4	5

Tabulka 2 – kriteriální matice

	platnost	podpora knihovnami (max)	Podpora klienty (max)	Celkem
SSL 2.0	1	1	1	3
SSL 3.0	2	5	1	8
TLS 1.0	5	10	10	25
TLS 1.1	7	10	10	27
TLS 1.2	9	10	10	29

Tabulka 3 – porovnání SSL a TLS bodovací metodou

Nejvyšší počet bodů 29 obdržel protokol TLS verze 1.2 následovaný TLS verze 1.1 s 27 body a dále TLS verze 1.0 s 25 body. Protokoly SSL verze 3.0 a SSL verze 2.0 obdržely pouze 8 bodů, resp. 3 body, a tudíž nejsou doporučeny k používání.

4 Vlastní práce

Pro modelovou konfiguraci WWW serveru pro zabezpečenou komunikaci s využitím protokolu HTTPS a pro testování nezabezpečené a zabezpečené komunikace byl použit jako server stroj PC Engines T40E s procesorem AMD G-T40E 4GB RAM a operačním systémem OpenBSD amd64 -current (dále gildor.navratil.cz) a jako client notebook ASUS UX305LA s procesorem Intel Core i7-5500 8GM RAM s operačními systémy OpenBSD amd64 -current resp. Microsoft Windows 10 Pro Verze 1607 Build 14393.447 (dále jako duilin.navratil.lan).

Operační systémy OpenBSD byly aktualizovány na poslední amd64 -current verzi s implementací LibreSSL 2.5.1 a na operačním systému Microsoft Windows 10 Pro byly aplikovány všechny dostupné aktualizace.

Server byl umístěn do DMZ (anglicky celým názvem „demilitarized zone“) s veřejnou IP adresou 89.22.65.152 a byl vytvořen záznam na name serverech pro jméno gildor.navratil.cz. Notebook byl připojen do vnitřní vnitřní sítě WiFi s IP adresou 192.168.1.111 přidělovanou pomocí DHCP (anglicky celým názvem „Dynamic Host Configuration Protocol“).

Sledování síťového provozu proběhlo pouze v rámci sítě vlastněné a provozované autorem. Byl využit 1000baseT Ethernet a 802.11n WiFi.

4.1 Konfigurace web serveru pro HTTPS komunikaci

Pro modelové nakonfigurování web serveru pro HTTPS komunikaci byl zvolen OpenBSD httpd server. Tato aplikace je součástí standardní instalace operačního systému a její konfigurace je jednoduchá a přehledná. Dále byla využita knihovna LibreSSL, která je součástí OpenBSD. Knihovna neobsahuje zakázané SSL protokoly a obsahuje řadu bezpečnostních vylepšení zmíněných v předešlém textu.

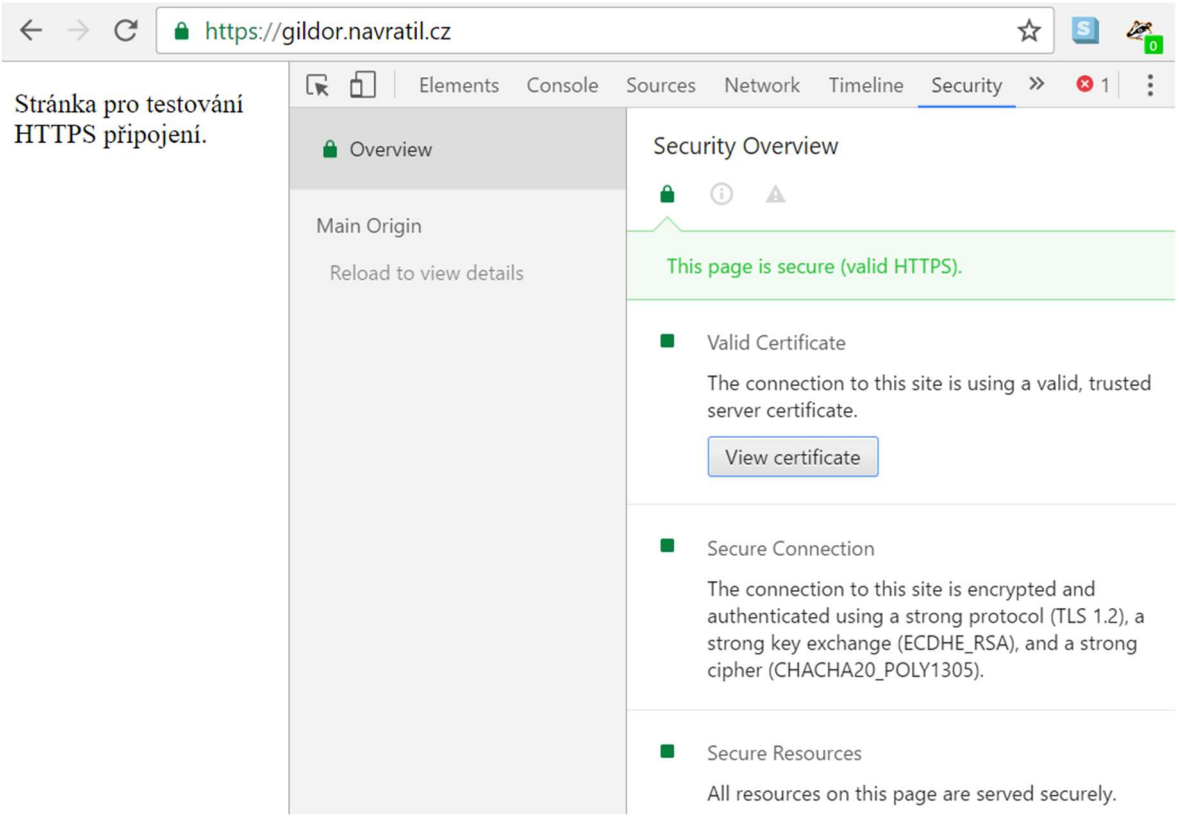
O certifikát bude požádána nezisková certifikační autorita Let’s Encrypt. Pro automatizované vyřízení žádosti je zapotřebí běžící web server na adrese, pro kterou bude certifikát vystaven. Prokazuje se tak, že žadatel doménu vlastní. Z tohoto důvodu je nejprve vytvořen dočasný self-signed certifikát pomocí příkazů openssl a na popředí spuštěn web server.

```
openssl req -new -key /etc/ssl/private/server.key -out /etc/ssl/private/server.csr
openssl x509 -sha256 -req -days 3 -in /etc/ssl/private/server.csr  -signkey /etc/ssl/private/server.key  -out /etc/ssl/server.crt

cat /etc/httpd.conf
ext_if="egress"

server gildor.navratil.cz {
        listen on $ext_if port 80
        listen on $ext_if tls port 443
        tls certificate "/etc/ssl/server.crt"
        tls key "/etc/ssl/private/server.key"
        location "/.well-known/acme-challenge/*" {
                root "/acme"
                root strip 2
        }
}

httpd -dv
```

Následně je použit nástroj acme-client pro vyřízení certifikátu. Po obdržení certifikátu a upravení konfigurace web serveru je spuštěn web server pro trvalý provoz.

```
acme-client -vNn gildor.navratil.cz
acme-client: /etc/ssl/acme/private/privkey.pem: generated RSA domain key
acme-client: /etc/acme/privkey.pem: generated RSA account key
acme-client: https://acme-v01.api.letsencrypt.org/directory: directories
acme-client: acme-v01.api.letsencrypt.org: DNS: 104.123.211.71
acme-client: https://acme-v01.api.letsencrypt.org/acme/new-authz: req-auth: gildor.navratil.cz
acme-client: /var/www/acme/5wIC0i20wykYcP2FDUj78USmqUuO8KMjW6hjyL8PP8Q: created
acme-client: https://acme-v01.api.letsencrypt.org/acme/challenge/KqFbIcsEUF9WTZg3gAO9N8f_LL9pRqKd5MR4kHkaOY4/355301681: challenge
acme-client: https://acme-v01.api.letsencrypt.org/acme/challenge/KqFbIcsEUF9WTZg3gAO9N8f_LL9pRqKd5MR4kHkaOY4/355301681: status
acme-client: https://acme-v01.api.letsencrypt.org/acme/new-cert: certificate
acme-client: http://cert.int-x3.letsencrypt.org/: full chain
acme-client: cert.int-x3.letsencrypt.org: DNS: 23.219.91.32
acme-client: /etc/ssl/acme/chain.pem: created
acme-client: /etc/ssl/acme/cert.pem: created
acme-client: /etc/ssl/acme/fullchain.pem: created

cat /etc/httpd.conf
ext_if="egress"

server gildor.navratil.cz {
      listen on $ext_if port 80
      block return 301 "https://gildor.navratil.cz/"
}

server gildor.navratil.cz {
        listen on $ext_if tls port 443
        tls certificate "/etc/ssl/acme/fullchain.pem"
        tls key "/etc/ssl/acme/private/privkey.pem"
        location "/.well-known/acme-challenge/*" {
                root "/acme"
                root strip 2
        }
}

rcctl enable httpd
rcctl start httpd
httpd(ok)
```

Při otevření webu https://gildor.navratil.cz/ v prohlížeči Google Chrome je možné zobrazit detaily o HTTPS připojení.



Další otestování webového serveru bylo realizováno z příkazové řádky pomocí aplikace w3m. Na výpisu (výpis byl zkrácen a částečně přeformátován) níže je patrné automatické přesměrování z HTTP na HTTPS, platný certifikát, certifikační autorita Let's Encrypt Authority X3, seriové číslo, použitý algoritmus sha256WithRSAEncryption, platnost certifikátu od 21.11.2016 01:28:00 GMT do 19.02.2017 01:28:00 GMT, vystavení pro gildor.navratil.cz, algoritmus pro veřejný klíč rsaEncryption s délkou 4096 bitů a další informace.

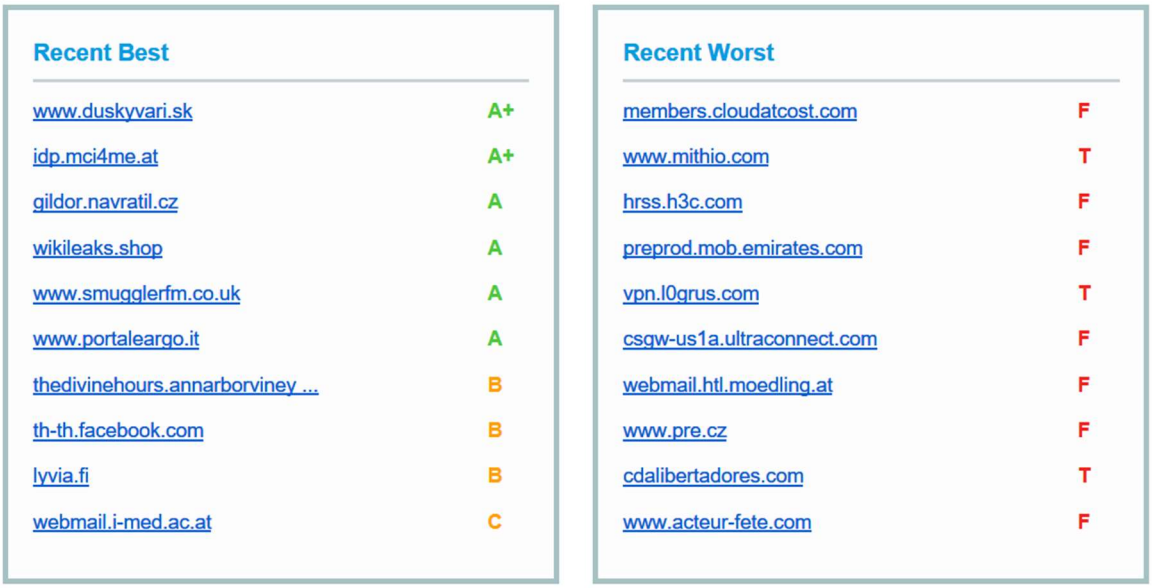
```
w3m -T text/html -dump_extra http://gildor.navratil.cz/
W3m-current-url: https://gildor.navratil.cz/
W3m-document-charset: US-ASCII
W3m-ssl-certificate: valid certificate
         subject=gildor.navratil.cz: issuer=Let's Encrypt Authority X3
        Certificate:
            Data:
                Version: 3 (0x2)
                Serial Number:
                    03:9c:28:d2:da:b8:bf:0c:1b:bb:0f:63:6a:d8:94:9f:fb:3c
            Signature Algorithm: sha256WithRSAEncryption
                Issuer: C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
                Validity
                    Not Before: Nov 21 01:28:00 2016 GMT
                    Not After : Feb 19 01:28:00 2017 GMT
                Subject: CN=gildor.navratil.cz
                Subject Public Key Info:
                    Public Key Algorithm: rsaEncryption
                        Public-Key: (4096 bit)
                        Modulus:
                            00:dd:f8:42:25:44:7c:c1:8d:d8:a4:13:a4:06:d7:
                            b4:bf:c7:c3:df:0c:d4:c9:b2:80:c2:f3:88:ed:cf:
                            b2:26:a4:26:db:48:a0:82:1c:26:ef:2a:77:27:02:
                0c:24:21
            Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Key Usage: critical
            Digital Signature, Key Encipherment
        X509v3 Extended Key Usage:
            TLS Web Server Authentication, TLS Web Client Authentication
        X509v3 Basic Constraints: critical
            CA:FALSE
        X509v3 Subject Key Identifier:
            B3:35:1C:12:83:3E:F4:C4:52:87:39:18:91:F3:02:03:7B:16:2B:9A
        X509v3 Authority Key Identifier:
            keyid:A8:4A:6A:63:04:7D:DD:BA:E6:D1:39:B7:A6:45:65:EF:F3:A8:EC:A1
        Authority Information Access:
            OCSP - URI:http://ocsp.int-x3.letsencrypt.org/
            CA Issuers - URI:http://cert.int-x3.letsencrypt.org/
        X509v3 Subject Alternative Name:
            DNS:gildor.navratil.cz
        X509v3 Certificate Policies:
            Policy: 2.23.140.1.2.1
            Policy: 1.3.6.1.4.1.44947.1.1.1
              CPS: http://cps.letsencrypt.org
              User Notice:
                Explicit Text: This Certificate may only be relied upon by Relying Parties
                and only in accordance with the Certificate Policy found at https://letsencrypt.org/repository/
Signature Algorithm: sha256WithRSAEncryption
     54:37:20:f1:cd:46:48:03:9d:25:8c:fc:b8:f6:b7:a1:8b:ec:
```

```
                    94:86:0b:64
HTTP/1.0 200 OK
Connection: close
Content-Length: 222
Content-Type: text/html
Date: Mon, 21 Nov 2016 03:48:03 GMT
Last-Modified: Mon, 21 Nov 2016 03:12:32 GMT
Server: OpenBSD httpd

<!doctype html>
<html lang=cs>
<head>
    <meta charset=utf-8>
    <title>OpenBSD httpd nakonfigurovaný pro HTTPS připojení.</title>
</head>
<body>
    <p>Stránka pro testování HTTPS připojení.</p>
</body>
</html>
```

Certifikát je vystavován na dobu 3 měsíců. Prodloužení je možné a nutné realizovat s blížícím se koncem jeho platnosti. Pro tento účel byl nainstalován skript a zajištěno jeho denní spuštění. Certifikát bude automaticky obnoven, jakmile to certifikační autorita dovolí a web server bude následně restartován, aby začal používat nový certifikát.

```
acme-client -vNn gildor.navratil.cz
acme-client: /etc/ssl/acme/private/privkey.pem: domain key exists (not creating)
acme-client: /etc/acme/privkey.pem: account key exists (not creating)
acme-client: /etc/ssl/acme/cert.pem: certificate valid: 89 days left

cat /usr/bin/renew_the_certificates.sh
#! /bin/sh
acme-client gildor.navratil.cz
if [ $? -eq 0 ]
then
     /etc/rc.d/httpd reload
fi

chmod u+x /usr/bin/renew_the_certificates.sh

crontab -l
SHELL=/bin/sh
27      4       *       *       *       /usr/bin/renew_the_certificates.sh
```

Server byl také otestován online nástrojem SSL Server Test společnosti Qualys, Inc. na adrese https://www.ssllabs.com/ssltest/. Server gildor.navratil.cz se zařadil v porovnání s ostatními servery testovanými během podobného času mezi nejlépe hodnocené. Porovnání bylo dostupné po provedeném testu na adrese https://www.ssllabs.com/ssltest/



Recent Best	
www.duskyvari.sk	A+
idp.mci4me.at	A+
gildor.navratil.cz	A
wikileaks.shop	A
www.smugglerfm.co.uk	A
www.portaleargo.it	A
thedivinehours.annarborviney ...	B
th-th.facebook.com	B
lyvia.fi	B
webmail.i-med.ac.at	C

Recent Worst	
members.cloudatcost.com	F
www.mithio.com	T
hrss.h3c.com	F
preprod.mob.emirates.com	F
vpn.l0grus.com	T
csgw-us1a.ultraconnect.com	F
webmail.htl.moedling.at	F
www.pre.cz	F
cdalibertadores.com	T
www.acteur-fete.com	F

Detailní protokol z testu je umístěn v přílohách.

4.2 Test webových prohlížečů

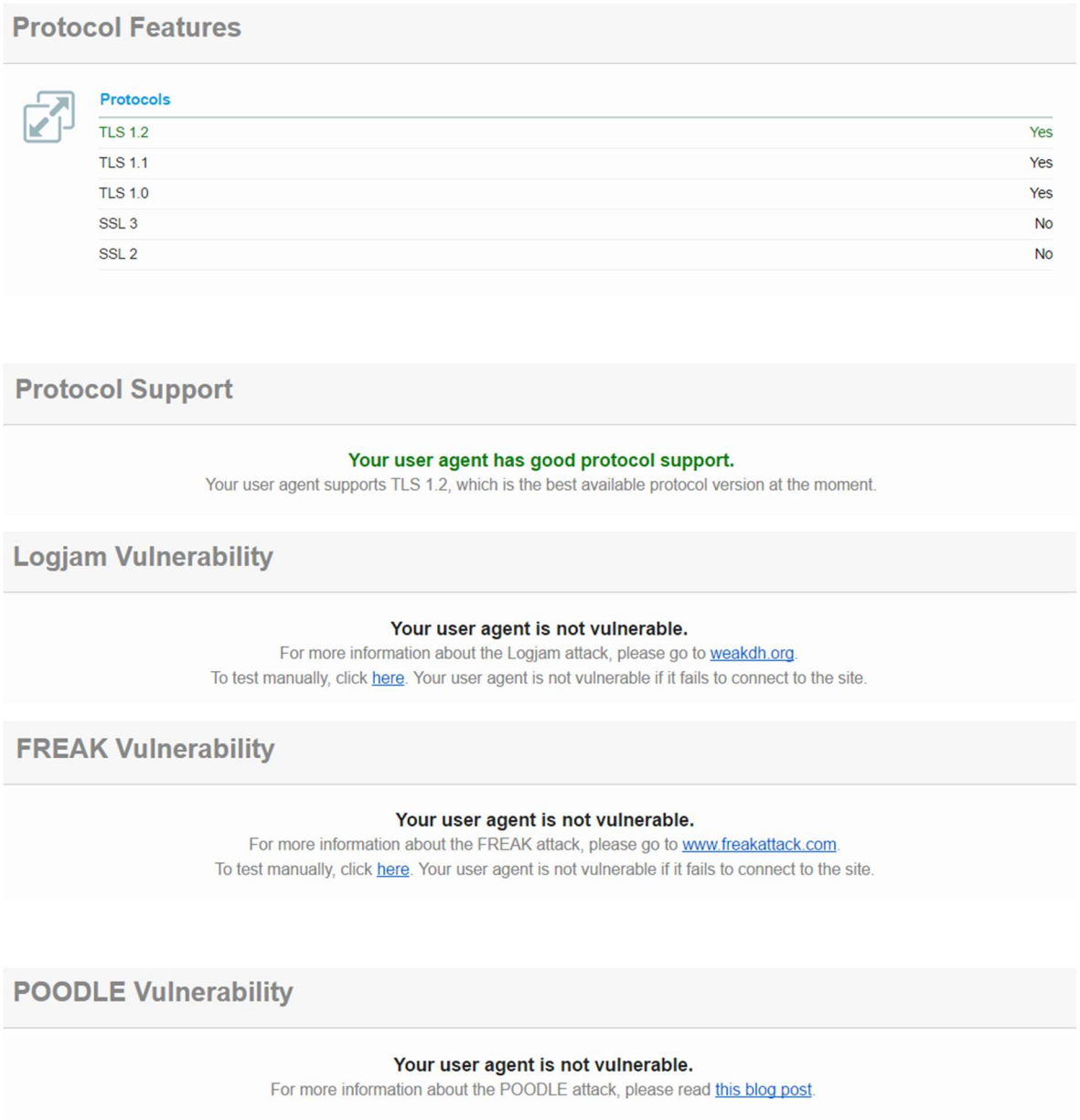
Pro testy webových prohlížečů byl použit 64bitový operační systém Microsoft Windows 10 Pro Verze 1607 Build 14393.447, webové prohlížeče instalované a aktualizované na poslední verze (viz. tabulka) a online nástroj SSL Client Test společnosti Qualys, Inc. na adrese https://www.ssllabs.com/ssltest/viewMyClient.html.

Otestovány byly tyto prohlížeče:

- Firefox 50.0
- Google Chrome Verze 54.0.2840.99 m (64-bit)
- Opera 41.0.2353.69
- Microsoft Edge 38.14393.0.0 Microsoft EdgeHTML 14.14393
- Internet Explorer 11.447.14393.0 Verze Aktualizace 11.0.37

Tabulka 4 – testované webové prohlížeče

Všechny prohlížeče měly v rámci testu shodné tyto parametry:



Protocol Features

Protocols	
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Protocol Support

Your user agent has good protocol support.
Your user agent supports TLS 1.2, which is the best available protocol version at the moment.

Logjam Vulnerability

Your user agent is not vulnerable.
For more information about the Logjam attack, please go to weakdh.org.
To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

FREAK Vulnerability

Your user agent is not vulnerable.
For more information about the FREAK attack, please go to www.freakattack.com.
To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

POODLE Vulnerability

Your user agent is not vulnerable.
For more information about the POODLE attack, please read this blog post.

Prohlížeče se lišily v obsahu a pořadí „Cipher Suites“. Detaily, včetně kompletních protokolů z testů, jsou v přílohách.

4.3 Sledování HTTP a HTTPS komunikace

Na počítači duilin.navratil.lan bylo provedeno zachycení HTTP i HTTPS komunikace pomocí nástroje tshark (textová verze populárního grafického nástroje pro analyzování síťové komunikace - wireshark).

Instalace byla provedena příkazem

```
pkg_add -vi tshark
```

V jednom terminálu byl spuštěn příkaz

```
w3m -T text/html -dump_extra http://gildor.navratil.cz/
```

a ve vedlejším terminálu byla zachycena HTTP komunikace příkazem

```
tshark -i iwm0 -f "port 80"
```

```
tshark -i iwm0 -f "port 80"
Capturing on 'iwm0'
  1   0.000000 192.168.1.111 ? 89.22.65.152 TCP 78 12901?80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=64 TSval=1242722184 TSecr=0
  2   0.271256 89.22.65.152 ? 192.168.1.111 TCP 78 80?12901 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=64 TSval=366147240 TSecr=1242722184
  3   0.271439 192.168.1.111 ? 89.22.65.152 TCP 66 12901?80 [ACK] Seq=1 Ack=1 Win=16384 Len=0 TSval=1242722185 TSecr=366147240
  4   0.271684 192.168.1.111 ? 89.22.65.152 HTTP 257 GET / HTTP/1.0
  5   0.348127 89.22.65.152 ? 192.168.1.111 HTTP 707 HTTP/1.0 301 Moved Permanently  (text/html)
  6   0.348752 192.168.1.111 ? 89.22.65.152 TCP 66 12901?80 [FIN, ACK] Seq=192 Ack=642 Win=16384 Len=0 TSval=1242722185 TSecr=366147240
  7   0.348808 89.22.65.152 ? 192.168.1.111 TCP 66 80?12901 [FIN, ACK] Seq=642 Ack=192 Win=17344 Len=0 TSval=366147240 TSecr=1242722185
```

a následně byla obdobně zaznamenána HTTP a HTTPS komunikace příkazem

```
tshark -i iwm0 -f "port 80" -f "port 443"
```

```
tshark -i iwm0 -f "port 80" -f "port 443"
Capturing on 'iwm0'
  1   0.000000 192.168.1.111 ? 89.22.65.152 TCP 78 28584?443 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=64 TSval=4171223177 TSecr=0
  2   0.546599 89.22.65.152 ? 192.168.1.111 TCP 78 443?28584 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=64 TSval=1084823770 TSecr=4171223177
  3   0.546782 192.168.1.111 ? 89.22.65.152 TCP 66 28584?443 [ACK] Seq=1 Ack=1 Win=16384 Len=0 TSval=4171223178 TSecr=1084823770
  4   0.579588 192.168.1.111 ? 89.22.65.152 SSL 583 Client Hello
  5   1.178225 89.22.65.152 ? 192.168.1.111 TLSv1.2 1514 Server Hello
  6   1.180946 89.22.65.152 ? 192.168.1.111 TLSv1.2 1514 Certificate[TCP segment of a reassembled PDU]
  7   1.181206 192.168.1.111 ? 89.22.65.152 TCP 66 28584?443 [ACK] Seq=518 Ack=2897 Win=14912 Len=0 TSval=4171223179 TSecr=1084823771
  8   1.187901 89.22.65.152 ? 192.168.1.111 TLSv1.2 663 Server Key ExchangeServer Hello Done
  9   1.209291 192.168.1.111 ? 89.22.65.152 TLSv1.2 264 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
 10   1.779066 89.22.65.152 ? 192.168.1.111 TLSv1.2 332 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
 11   1.783614 192.168.1.111 ? 89.22.65.152 TLSv1.2 278 Application Data
 12   2.316611 89.22.65.152 ? 192.168.1.111 TLSv1.2 277 Application Data
 13   2.326755 89.22.65.152 ? 192.168.1.111 TLSv1.2 309 Application Data
 14   2.326913 192.168.1.111 ? 89.22.65.152 TCP 66 28584?443 [ACK] Seq=928 Ack=4215 Win=16128 Len=0 TSval=4171223182 TSecr=1084823774
 15   2.327748 192.168.1.111 ? 89.22.65.152 TCP 66 28584?443 [FIN, ACK] Seq=928 Ack=4215 Win=16384 Len=0 TSval=4171223182 TSecr=1084823774
 16   2.819311 89.22.65.152 ? 192.168.1.111 TCP 66 443?28584 [ACK] Seq=4215 Ack=929 Win=17344 Len=0 TSval=1084823775 TSecr=4171223182
```

Na výpisech je klient (duilin.navratil.lan) reprezentován IP adresou 192.168.1.111 a server (gildor.navratil.cz) IP adresou 89.22.65.152. Na výpisech je zachyceno trvalé přesměrování z HTTP na HTTPS, TLS handshake a použití TLSv1.2.

5 Výsledky a diskuse

Bezpečnostní rizika při používání nezašifrované komunikace prostřednictvím HTTP protokolu by měla motivovat správce web služeb i samotné uživatele k přechodu na šifrovanou komunikaci prostřednictvím HTTPS. Software i hardware je k takovému přechodu připravený. Výpočetní výkon na straně serverů i klientů umožňuje, aby byla komunikace šifrována po celou dobu komunikace.

Bezpečnostní chyby v protokolech SSL a TLS jsou často přičítány vývojářům. Jedním z výsledků práce je zjištění, že nemalý podíl na těchto chybách mají špatné specifikace a také způsob nakonfigurování služeb.

Existují online nástroje, které umožňují administrátorům i uživatelům rychlou kontrolu, zda je nakonfigurovaný, resp. navštěvovaný server imunní proti známým bezpečnostním hrozbám.

Výsledkem teoretických východisek a vlastní práce jsou tato doporučení.

5.1 Doporučení pro správce web služeb

- Webové služby by měly být nabízeny výhradně na portu 443 pomocí HTTPS (viz. kapitoly 3.4 a 3.2)
- Na portu 80 by mělo být zapnuto trvalé přesměrování na port 443 pomocí „HTTP response status code 301 - Moved Permanently“ (viz. kapitola 4.1)
- Zapnout TLS verze 1.2 a vypnout všechny SSL verze. Dále zvážit vypnutí i TLS verze 1.1 a TLS verze 1.0 (viz. 4.2, 3.14 a 3.5)
- Používání certifikátu podepsaného certifikační autoritou. (viz. 3.7, 4.1 a 4.2)
- Nasazení web serveru na OpenBSD operačním systému s využitím LibreSSL knihovny. (viz. 4.1, 3.11 a 3.12)
- Otestování serveru na https://www.ssllabs.com/ssltest/ (viz. 4.1)

5.2 Doporučení pro uživatele

- Nasazení nejaktuálnější verze webového prohlížeče, který podporuje TLS verze 1.2, TLS verze 1.1 a TLS verze 1.0 a všechny SSL verze má zakázané. (viz. 4.2, 3.14 a 3.5)

- Otestování klienta na https://www.ssllabs.com/ssltest/viewMyClient.html (viz. 4.2)
- Používání schématu HTTPS místo HTTP, kdekoli je to možné. Buď ručně, nebo s využitím rozšíření pro prohlížeče, které takové přesměrování zajistí automaticky. Mezi taková rozšíření patří například „HTTPS Everywhere“ z https://www.eff.org/https-everywhere/ (viz. 3.3 a 3.4)
- Akceptovat pouze certifikát podepsaný certifikační autoritou. To znamená neakceptovat „self-signed“ certifikáty. (viz. 3.7)

6 Závěr

Bezpečnostní protokoly SSL a TLS se jmenují odlišně, ale z technického pohledu TLS navazuje na SSL. Pojem SSL je známější než TLS, ale protokol SSL již není bezpečné používat. Doporučený je pouze TLS, a to zejména v poslední verzi 1.2.

Bezpečnostní knihovny a webové servery TLS verze 1.2 podporují. Stejně tak všechny testované webové prohlížeče TLS verze 1.2 podporují, a navíc nepodporují SSL. Definitivní opuštění SSL protokolů a přechod na TLS 1.2 spolu s kvalitními a dostatečně dlouhými klíči, bezpečnými šiframi, kvalitní implementací a správně realizovanou konfigurací umožní bezpečnější komunikaci prostřednictvím HTTPS.

Software i hardware je připravený na přechod od HTTP k HTTPS. S tím, jak narůstá počet serverů poskytujících výhradně zabezpečenou komunikaci přes HTTPS, se dle autora blíží doba, kdy bude možné říkat Internet bezpečnější, Internet HTTPS.

7 Seznam použitých zdrojů

7.1 Knižní publikace

1. SWORD, Helen. Stylish academic writing. Cambridge, Mass.: Harvard University Press, 2012. ISBN 9780674064485.
2. RISTIC, Ivan. Bulletproof SSL and TLS. London: Feisty Duck, 2016. ISBN 9781907117046

7.2 Elektronické dokumenty a online zdroje

3. BARNES, Richard, Martin THOMSON, Alfredo PIRONTI a Adam LANGLEY. Request for Comments: 7568 - Deprecating Secure Sockets Layer Version 3.0 [online]. 2015 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc7568.txt
4. BECK, Bob. LibreSSL - An OpenSSL replacement: The first 30 days, and where we go from there [online]. 2014 [cit. 2016-03-03]. Dostupné z: http://www.openbsd.org/papers/bsdcan14-libressl/
5. BLAKE-WILSON, Simon, Magnus NYSTROM, David HOPWOOD, Jan MIKKELSEN a Tim WRIGHT. Request for Comments: 4366 - Transport Layer Security (TLS) Extensions [online]. 2006 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc4366.txt
6. DE RAADT, Theo. OpenBSD 5.6 [online]. 2014 [cit. 2016-03-03]. ISBN 978-0-9881561-4-2. Dostupné z: http://www.openbsd.org/56.html
7. DE RAADT, Theo. OpenBSD 5.7 [online]. 2015 [cit. 2016-03-03]. ISBN 978-0-9881561-5-9. Dostupné z: http://www.openbsd.org/57.html
8. DE RAADT, Theo. OpenBSD 5.8 [online]. 2015 [cit. 2016-03-03]. ISBN 978-0-9881561-6-6. Dostupné z: http://www.openbsd.org/58.html
9. DIERKS, Tim a Eric RESCORLA (eds.). Request for Comments: 5246 - The Transport Layer Security (TLS) Protocol Version 1.2 [online]. 2008 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc5246.txt
10. DIERKS, Tim, ALLEN, Christopher (ed.). Request for Comments: 2246 - The TLS Protocol Version 1.0 [online]. 1999 [cit. 2016-03-03]. Dostupné z: https://www.ietf.org/rfc/rfc2246.txt

11. FREIER, Alan O., Philip KARLTON a Paul C. KOCHER. Request for Comments: 6101 - The Secure Sockets Layer (SSL) Protocol Version 3.0 [online]. 2011 [cit. 2016-03-03]. ISSN: 2070-1721. Dostupné z: https://tools.ietf.org/rfc/rfc6101.txt
12. HENDERSON, Stuart. Re: SSL version 3 is still mentioned in man pages: email in openbsd-tech email list [online]. In: . 2015 [cit. 2016-03-03]. Dostupné z: https://marc.info/?l=openbsd-tech&m=145139429104668
13. MCGREW, David a Eric RESCORLA. Request for Comments: 5764 - Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP) [online]. 2010 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc5764.txt
14. MOELLER Bodo, LANGLEYAdam. Request for Comments: 7507 - TLS Fallback Signaling Cipher Suite Value (SCSV) for Preventing Protocol Downgrade Attacks [online]. 2015 [cit. 2016-03-03]. ISSN 2070-1721. Dostupné z: https://tools.ietf.org/rfc/rfc7507.txt
15. PHELAN, Tom. Request for Comments: 5238 - Datagram Transport Layer Security (DTLS) over the Datagram Congestion Control Protocol (DCCP) [online]. 2008 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc5238.txt
16. RESCORLA, Eric and contributors . The Transport Layer Security (TLS) Protocol Version 1.3: draft-ietf-tls-tls13-18 [online]. 22.09.2016. [cit. 2016-10-20]. Dostupné z: https://tools.ietf.org/id/draft-ietf-tls-tls13-18.txt
17. RESCORLA, Eric a Nagendra MODADUGU. Request for Comments: 4347 - Datagram Transport Layer Security [online]. 2006 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc4347.txt
18. RESCORLA, Eric a Nagendra MODADUGU. Request for Comments: 6347 - Datagram Transport Layer Security Version 1.2 [online]. 2012 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc6347.txt
19. RESCORLA, Eric, DIERKS, Tim (ed.). Request for Comments: 4346 - The Transport Layer Security (TLS) Protocol Version 1.1 [online]. 2006 [cit. 2016-03-03].. Dostupné z: https://www.ietf.org/rfc/rfc4346.txt
20. SALOWEY, Joseph, Hao ZHOU, Pasi ERONEN a Hannes TSCHOFENIG. Request for Comments: 5077 - Transport Layer Security (TLS) Session Resumption without

Server-Side State [online]. 2008 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc5077.txt

21. SALTER, Margaret a Russ HOUSLEY. Request for Comments: 6460 - Suite B Profile for Transport Layer Security (TLS) [online]. 2012 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc6460.txt
22. SHIREY, Robert W. Request for Comments: 4949 - Internet Security Glossary, Version 2 [online]. 2007 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc4949.txt
23. TUEXEN, Michael, Robin SEGGELMANN a Eric RESCORLA. Request for Comments: 6083 - Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP) [online]. 2011 [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc6083.txt
24. TURNER, Sean a Tim POLK. Request for Comments: 6176 - Prohibiting Secure Sockets Layer (SSL) Version 2.0 [online]. [cit. 2016-03-03]. Dostupné z: https://tools.ietf.org/rfc/rfc6176.txt

7.3 Online databáze

25. Common Vulnerabilities and Exposures: The Standard for Information Security Vulnerability Names. [online]. [cit. 2016-03-03]. Dostupné z: http://cve.mitre.org/
26. CVE Details: The ultimate security vulnerability datasource. [online]. [cit. 2016-03-03]. Dostupné z: http://www.cvedetails.com/

8 Přílohy

8.1 Přehled dosavadních změn TLS v1.3 oproti TLS v1.2

TLS verze 1.3 zatím nebyla zveřejněna a nachází se v připomínkovacím režimu u IETF (anglicky celým názvem Internet Engineering Task Force). Od první pracovní verze zveřejněné 17.04.2014 do osmnácté pracovní verze zveřejněné 26.10.2016, došlo k níže uvedeným změnám. V řadě případů se jedná o úpravy oproti TLS verze 1.2. Převzato z https://tools.ietf.org/html/draft-ietf-tls-tls13-18

- Remove unnecessary resumption_psk which is the only thing expanded from the resumption master secret.
- Fix signature_algorithms entry in extensions table.
- Restate rule from RFC 6066 that you can't resume unless SNI is the same.
- Remove the 0-RTT Finished, resumption_context, and replace with a psk_binder field in the PSK itself.
- Restructure PSK key exchange negotiation modes
- Add max_early_data_size field to TicketEarlyDataInfo
- Add a 0-RTT exporter and change the transcript for the regular exporter
- Merge TicketExtensions and Extensions registry. Changes ticket_early_data_info code point.
- Replace Client.key_shares in response to HRR
- Remove redundant labels for traffic key derivation.
- Harmonize requirements about cipher suite matching: for resumption you need to match KDF but for 0-RTT you need whole cipher suite. This allows PSKs to actually negotiate cipher suites.
- Move SCT and OCSP into Certificate.extensions
- Explicitly allow non-offered extensions in NewSessionTicket
- Explicitly allow predicting ClientFinished for NST
- Clarify conditions for allowing 0-RTT with PSK
- Revise version negotiation

- Change RSASSA-PSS and EdDSA SignatureScheme codepoints for better backwards compatibility.
- Move HelloRetryRequest.selected_group to an extension
- Clarify the behavior of no exporter context and make it the same as an empty context.
- New KeyUpdate format that allows for requesting/not-requesting an answer. This also means changes to the key schedule to support independent updates.
- New certificate_required alert
- Forbid CertificateRequest with 0-RTT and PSK.
- Relax requirement to check SNI for 0-RTT.
- New negotiation syntax as discussed in Berlin
- Require CertificateRequest.context to be empty during handshake.
- Forbid empty tickets.
- Forbid application data messages in between post-handshake messages from the same flight.
- Clean up alert guidance.
- Clearer guidance on what is needed for TLS 1.2.
- Guidance on 0-RTT time windows.
- Rename a bunch of fields.
- Remove old PRNG text.
- Explicitly require checking that handshake records not span key changes.
- Allow cookies to be longer.
- Remove the "context" from EarlyDataIndication as it was undefined and nobody used it.
- Remove 0-RTT EncryptedExtensions and replace the ticket_age extension with an obfuscated version. Also necessitates a change to NewSessionTicket.
- Move the downgrade sentinel to the end of ServerHello.Random to accommodate tlsdate.
- Define ecdsa_sha1.
- Allow resumption even after fatal alerts. This matches current practice.
- Remove non-closure warning alerts. Require treating unknown alerts as fatal.
- Make the rules for accepting 0-RTT less restrictive.
- Clarify 0-RTT backward-compatibility rules.
- Clarify how 0-RTT and PSK identities interact.

- Add a section describing the data limits for each cipher.
- Major editorial restructuring.
- Replace the Security Analysis section with a WIP draft.
- Allow server to send SupportedGroups.
- Remove 0-RTT client authentication
- Remove (EC)DHE 0-RTT.
- Flesh out 0-RTT PSK mode and shrink EarlyDataIndication
- Turn PSK-resumption response into an index to save room
- Move CertificateStatus to an extension
- Extra fields in NewSessionTicket.
- Restructure key schedule and add a resumption_context value.
- Require DH public keys and secrets to be zero-padded to the size of the group.
- Remove the redundant length fields in KeyShareEntry.
- Define a cookie field for HRR.
- Provide a list of the PSK cipher suites.
- Remove the ability for the ServerHello to have no extensions (this aligns the syntax with the text).
- Clarify that the server can send application data after its first flight (0.5 RTT data)
- Revise signature algorithm negotiation to group hash, signature algorithm, and curve together. This is backwards compatible.
- Make ticket lifetime mandatory and limit it to a week.
- Make the purpose strings lower-case. This matches how people are implementing for interop.
- Define exporters.
- Editorial cleanup
- Port the CFRG curves & signatures work from RFC4492bis.
- Remove sequence number and version from additional_data, which isnnow empty.
- Reorder values in HkdfLabel.
- Add support for version anti-downgrade mechanism.
- Update IANA considerations section and relax some of the policies.
- Unify authentication modes. Add post-handshake client authentication.
- Remove early_handshake content type. Terminate 0-RTT data with an alert.

- Reset sequence number upon key change (as proposed by Fournet et al.)
- Remove ClientCertificateTypes field from CertificateRequest and add extensions.
- Merge client and server key shares into a single extension.
- Change to RSA-PSS signatures for handshake messages.
- Remove support for DSA.
- Update key schedule per suggestions by Hugo, Hoeteck, and Bjoern Tackmann.
- Add support for per-record padding.
- Switch to encrypted record ContentType.
- Change HKDF labeling to include protocol version and value lengths.
- Shift the final decision to abort a handshake due to incompatible certificates to the client rather than having servers abort early.
- Deprecate SHA-1 with signatures.
- Add MTI algorithms.
- Remove support for weak and lesser used named curves.
- Remove support for MD5 and SHA-224 hashes with signatures.
- Update lists of available AEAD cipher suites and error alerts.
- Reduce maximum permitted record expansion for AEAD from 2048 to 256 octets.
- Require digital signatures even when a previous configuration is used.
- Merge EarlyDataIndication and KnownConfiguration.
- Change code point for server_configuration to avoid collision with server_hello_done.
- Relax certificate_list ordering requirement to match current practice.
- Integration of semi-ephemeral DH proposal.
- Add initial 0-RTT support.
- Remove resumption and replace with PSK + tickets.
- Move ClientKeyShare into an extension.
- Move to HKDF.
- Prohibit RC4 negotiation for backwards compatibility.
- Freeze & deprecate record layer version field.
- Update format of signatures with context.
- Remove explicit IV.
- Prohibit SSL negotiation for backwards compatibility.
- Fix which MS is used for exporters.

- Modify key computations to include session hash.
- Remove ChangeCipherSpec.
- Renumber the new handshake messages to be somewhat more consistent with existing convention and to remove a duplicate registration.
- Remove renegotiation.
- Remove point format negotiation.
- Remove GMT time.
- Merge in support for ECC from RFC 4492 but without explicit curves.
- Remove the unnecessary length field from the AD input to AEAD ciphers.
- Rename {Client,Server}KeyExchange to {Client,Server}KeyShare.
- Add an explicit HelloRetryRequest to reject the client's.
- Increment version number.
- Rework handshake to provide 1-RTT mode.
- Remove custom DHE groups.
- Remove support for compression.
- Remove support for static RSA and DH key exchange.
- Remove support for non-AEAD ciphers.

8.2 Externí test web serveru gildor.navratil.cz

Pro test serveru gildor.navratil.cz byl použit online nástroj SSL Server Test společnosti Qualys, Inc. na adrese https://www.ssllabs.com/ssltest/.

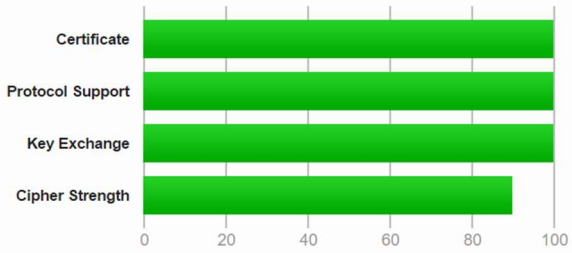
SSL Report v1.25.2 obsahoval mimo jiné tyto informace:

```
SSL Report: gildor.navratil.cz (89.22.65.152)
 Assessed on:  Sat, 26 Nov 2016 09:00:34 UTC
```

Summary

Overall Rating

A



Protocols

TLS 1.2	Yes
TLS 1.1	No
TLS 1.0	No
SSL 3	No
SSL 2	No

Cipher Suites (SSL 3+ suites in server-preferred order; deprecated and SSL 2 suites at the end)

TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	256
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d)	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	256
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256 (0xc0)	256
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x84)	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c)	128
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256 (0xba)	128
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x41)	128

Certificate #1: RSA 4096 bits (SHA256withRSA)

Server Key and Certificate #1

Subject	gildor.navratil.cz Fingerprint SHA1: ebc471ce8dfd02a4a1943131ac5da7a969fefcb3 Pin SHA256: 38kE0Io/QEGU1pLNFS64Ev15afYKvcGTFuzB64ohm6A=
Common names	gildor.navratil.cz
Alternative names	gildor.navratil.cz
Valid from	Mon, 21 Nov 2016 01:28:00 UTC
Valid until	Sun, 19 Feb 2017 01:28:00 UTC (expires in 2 months and 23 days)
Key	RSA 4096 bits (e 65537)
Weak key (Debian)	No
Issuer	Let's Encrypt Authority X3 AIA: http://cert.int-x3.letsencrypt.org/
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	No
OCSP Must Staple	No
Revocation information	OCSP OCSP: http://ocsp.int-x3.letsencrypt.org/
Revocation status	Good (not revoked)
Trusted	Yes

Additional Certificates (if supplied)

Certificates provided	2 (2722 bytes)
Chain issues	None

#2

Subject	Let's Encrypt Authority X3 Fingerprint SHA1: e6a3b45b062d509b3382282d196efe97d5956ccb Pin SHA256: YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg=
Valid until	Wed, 17 Mar 2021 16:40:46 UTC (expires in 4 years and 3 months)
Key	RSA 2048 bits (e 65537)
Issuer	DST Root CA X3
Signature algorithm	SHA256withRSA

Protocol Details

DROWN (experimental)	No, server keys and hostname not seen elsewhere with SSLv2 **(1) For a better understanding of this test, please read this longer explanation** (2) Key usage data kindly provided by the Censys network search engine; original DROWN test here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete
Secure Renegotiation	**Supported**
Secure Client-Initiated Renegotiation	Yes
Insecure Client-Initiated Renegotiation	No
BEAST attack	Mitigated server-side (more info)
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Downgrade attack prevention	Unknown (requires support for at least two protocols, excl. SSL2)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
Forward Secrecy	**Yes (with most browsers) ROBUST** (more info)
ALPN	No
NPN	No
Session resumption (caching)	**No (IDs assigned but not accepted)**
Session resumption (tickets)	Yes
OCSP stapling	No
Strict Transport Security (HSTS)	No
HSTS Preloading	**Not in: Chrome Edge Firefox IE**
Public Key Pinning (HPKP)	No
Public Key Pinning Report-Only	No
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No
Incorrect SNI alerts	No
Uses common DH primes	No, DHE suites not supported
DH public server param (Ys) reuse	No, DHE suites not supported
SSL 2 handshake compatibility	No

8.3 Detailní protokoly z testů Qualys SSL LABS

Následují detailní protokoly vygenerované pomocí https://www.ssllabs.com/ssltest/ a https://www.ssllabs.com/ssltest/viewMyClient.html

SSL Report: gildor.navratil.cz (89.22.65.152)

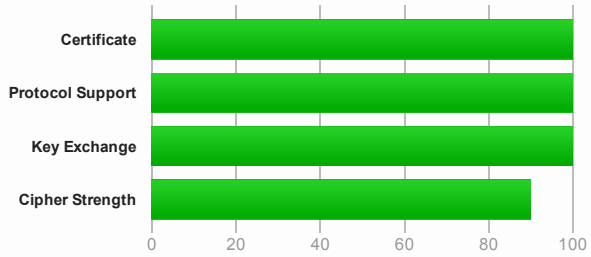
Assessed on: Mon, 21 Nov 2016 14:02:23 UTC | Hide | Clear cache

Scan Another »

Summary

Overall Rating

A



Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

Certificate #1: RSA 4096 bits (SHA256withRSA)

Server Key and Certificate #1

Subject	gildor.navratil.cz Fingerprint SHA1: ebc471ce8dfd02a4a1943131ac5da7a969fefcb3 Pin SHA256: 38kE0lo/QEGU1pLNFS64Ev15afYKvcGTFuzB64ohm6A=
Common names	gildor.navratil.cz
Alternative names	gildor.navratil.cz
Valid from	Mon, 21 Nov 2016 01:28:00 UTC
Valid until	Sun, 19 Feb 2017 01:28:00 UTC (expires in 2 months and 28 days)
Key	RSA 4096 bits (e 65537)
Weak key (Debian)	No
Issuer	Let's Encrypt Authority X3 AIA: http://cert.int-x3.letsencrypt.org/
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	No
OCSP Must Staple	No
Revocation information	OCSP OCSP: http://ocsp.int-x3.letsencrypt.org/
Revocation status	Good (not revoked)
Trusted	Yes

Additional Certificates (if supplied)

Certificates provided	2 (2722 bytes)
Chain issues	None

#2

Subject	Let's Encrypt Authority X3 Fingerprint SHA1: e6a3b45b062d509b3382282d196efe97d5956ccb Pin SHA256: YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg=
Valid until	Wed, 17 Mar 2021 16:40:46 UTC (expires in 4 years and 3 months)
Key	RSA 2048 bits (e 65537)
Issuer	DST Root CA X3
Signature algorithm	SHA256withRSA

Cipher Suites (sorted by strength; we could not determine if the server has a preference)

Certification Paths

Click here to expand

Configuration

Protocols

TLS 1.2	Yes
TLS 1.1	No
TLS 1.0	No
SSL 3	No
SSL 2	No

Cipher Suites (sorted by strength; we could not determine if the server has a preference)

TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012) ECDH sect571r1 (eq. 15360 bits RSA) FS	112
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x41)	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH sect571r1 (eq. 15360 bits RSA) FS	128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c)	128
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	128
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256 (0xba)	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH sect571r1 (eq. 15360 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH sect571r1 (eq. 15360 bits RSA) FS	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	256
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x84)	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH sect571r1 (eq. 15360 bits RSA) FS	256
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH sect571r1 (eq. 15360 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH sect571r1 (eq. 15360 bits RSA) FS	256
OLD_TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc13) ECDH sect571r1 (eq. 15360 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) ECDH sect571r1 (eq. 15360 bits RSA) FS	256

Handshake Simulation

Android 2.3.7 No SNI [2]	Server closed connection		
Android 4.0.4	Server closed connection		
Android 4.1.1	Server closed connection		
Android 4.2.2	Server closed connection		
Android 4.3	Server closed connection		
Android 4.4.2	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp521r1 FS
Android 5.0.0	RSA 4096 (SHA256)	TLS 1.2	OLD_TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp521r1 FS
Android 6.0	RSA 4096 (SHA256)	TLS 1.2	OLD_TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp384r1 FS
Android 7.0	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp384r1 FS
Baidu Jan 2015	Server closed connection		
BingPreview Jan 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH sect571r1 FS
Chrome 49 / XP SP3	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp384r1 FS
Chrome 51 / Win 7 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp384r1 FS
Firefox 31.3.0 ESR / Win 7	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp521r1 FS
Firefox 47 / Win 7 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp521r1 FS
Firefox 49 / XP SP3	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp521r1 FS
Firefox 49 / Win 7 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp521r1 FS
Googlebot Feb 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH sect571r1 FS

Handshake Simulation

IE 6 / XP No FS [1] No SNI [2]	Server closed connection		
IE 7 / Vista	Server closed connection		
IE 8 / XP No FS [1] No SNI [2]	Server closed connection		
IE 8-10 / Win 7 R	Server closed connection		
IE 11 / Win 7 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp384r1 FS
IE 11 / Win 8.1 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp384r1 FS
IE 10 / Win Phone 8.0	Server closed connection		
IE 11 / Win Phone 8.1 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp384r1 FS
IE 11 / Win Phone 8.1 Update R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp384r1 FS
IE 11 / Win 10 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp384r1 FS
Edge 13 / Win 10 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp384r1 FS
Edge 13 / Win Phone 10 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp384r1 FS
Java 6u45 No SNI [2]	Server closed connection		
Java 7u25	Server closed connection		
Java 8u31	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH sect571r1 FS
OpenSSL 0.9.8y	Server closed connection		
OpenSSL 1.0.1l R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH sect571r1 FS
OpenSSL 1.0.2e R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH sect571r1 FS
Safari 5.1.9 / OS X 10.6.8	Server closed connection		
Safari 6 / iOS 6.0.1	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp521r1 FS
Safari 6.0.4 / OS X 10.8.4 R	Server closed connection		
Safari 7 / iOS 7.1 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp521r1 FS
Safari 7 / OS X 10.9 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp521r1 FS
Safari 8 / iOS 8.4 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp521r1 FS
Safari 8 / OS X 10.10 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp521r1 FS
Safari 9 / iOS 9 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp521r1 FS
Safari 9 / OS X 10.11 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp521r1 FS
Safari 10 / iOS 10 R	Protocol or cipher suite mismatch TLS 1.2 | TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384		
Safari 10 / OS X 10.12 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp521r1 FS
Apple ATS 9 / iOS 9 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp521r1 FS
Yahoo Slurp Jan 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp384r1 FS
YandexBot Jan 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH sect571r1 FS

(1) Clients that do not support Forward Secrecy (FS) are excluded when determining support for it.

(2) No support for virtual SSL hosting (SNI). Connects to the default site if the server uses SNI.

(3) Only first connection attempt simulated. Browsers sometimes retry with a lower protocol version.

(R) Denotes a reference browser or client, with which we expect better effective security.

(All) We use defaults, but some platforms do not use their best protocols and features (e.g., Java 6 & 7, older IE).

Protocol Details

DROWN (experimental)	No, server keys and hostname not seen elsewhere with SSLv2 **(1) For a better understanding of this test, please read this longer explanation** (2) Key usage data kindly provided by the Censys network search engine; original DROWN test here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete
Secure Renegotiation	**Supported**
Secure Client-Initiated Renegotiation	Yes
Insecure Client-Initiated Renegotiation	No
BEAST attack	Mitigated server-side (more info)
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Downgrade attack prevention	Unknown (requires support for at least two protocols, excl. SSL2)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)

1 https://gildor.navratil.cz/ (HTTP/1.0 200 OK)

OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
Forward Secrecy	Yes (with most browsers) ROBUST (more info)
ALPN	No
NPN	No
Session resumption (caching)	No (IDs assigned but not accepted)
Session resumption (tickets)	Yes
OCSP stapling	No
Strict Transport Security (HSTS)	No
HSTS Preloading	Not in: Chrome Edge Firefox IE
Public Key Pinning (HPKP)	No
Public Key Pinning Report-Only	No
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No
Incorrect SNI alerts	No
Uses common DH primes	No, DHE suites not supported
DH public server param (Ys) reuse	No, DHE suites not supported
SSL 2 handshake compatibility	No

HTTP Requests

1 https://gildor.navratil.cz/ (HTTP/1.0 200 OK)

Miscellaneous

Test date	Mon, 21 Nov 2016 13:59:06 UTC
Test duration	197.117 seconds
HTTP status code	200
HTTP server signature	OpenBSD httpd
Server hostname	bilbo.navratil.cz

SSL Report v1.25.2

SSL/TLS Capabilities of Your Browser

User Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0

Other User Agents »

Protocol Support

Your user agent has good protocol support.

Your user agent supports TLS 1.2, which is the best available protocol version at the moment.

Logjam Vulnerability

Your user agent is not vulnerable.

For more information about the Logjam attack, please go to weakdh.org.

To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

FREAK Vulnerability

Your user agent is not vulnerable.

For more information about the FREAK attack, please go to www.freakattack.com.

To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

POODLE Vulnerability

Your user agent is not vulnerable.

For more information about the POODLE attack, please read this blog post.

Protocol Features

Protocols

TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) Forward Secrecy	256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) Forward Secrecy	128
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) Forward Secrecy	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	256

Cipher Suites (in order of preference)

TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	112

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very first connection to this site. To see the suites, close all browser windows, then open this exact page directly. Don't refresh.

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA256/RSA, SHA384/RSA, SHA512/RSA, SHA1/RSA, SHA256/ECDSA, SHA384/ECDSA, SHA512/ECDSA, SHA1/ECDSA, SHA384/DSA, SHA256/DSA, SHA1/DSA
Elliptic curves	secp256r1, secp384r1, secp521r1
Next Protocol Negotiation	Yes
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

Mixed Content Handling

Mixed Content Tests

Images	Passive	Yes
CSS	Active	No
Scripts	Active	No
XMLHttpRequest	Active	No
WebSockets	Active	No
Frames	Active	No

(1) These tests might cause a mixed content warning in your browser. That's expected.
(2) If you see a failed test, try to reload the page. If the error persists, please get in touch.

Related Functionality

Upgrade Insecure Requests request header (more info)	Yes

SSL/TLS Capabilities of Your Browser

Other User Agents »

User Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36

Protocol Support

Your user agent has good protocol support.
Your user agent supports TLS 1.2, which is the best available protocol version at the moment.

Logjam Vulnerability

Your user agent is not vulnerable.
For more information about the Logjam attack, please go to weakdh.org.
To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

FREAK Vulnerability

Your user agent is not vulnerable.
For more information about the FREAK attack, please go to www.freakattack.com.
To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

POODLE Vulnerability

Your user agent is not vulnerable.
For more information about the POODLE attack, please read this blog post.

Protocol Features

Protocols

TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) Forward Secrecy	256
OLD_TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc14) Forward Secrecy	256
OLD_TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc13) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) Forward Secrecy	128

Cipher Suites (in order of preference)

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) Forward Secrecy	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	112

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very first connection to this site. To see the suites, close all browser windows, then open this exact page directly. Don't refresh.

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA512/RSA, SHA512/ECDSA, SHA384/RSA, SHA384/ECDSA, SHA256/RSA, SHA256/ECDSA, SHA1/RSA, SHA1/ECDSA
Elliptic curves	x25519, secp256r1, secp384r1
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

Mixed Content Handling

Mixed Content Tests

Images	Passive	Yes
CSS	Active	No
Scripts	Active	No
XMLHttpRequest	Active	No
WebSockets	Active	No
Frames	Active	No

(1) These tests might cause a mixed content warning in your browser. That's expected.
(2) If you see a failed test, try to reload the page. If the error persists, please get in touch.

Related Functionality

Upgrade Insecure Requests request header (more info)	Yes

SSL/TLS Capabilities of Your Browser

Other User Agents »

User Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36 OPR/41.0.2353.69

Protocol Support

Your user agent has good protocol support.
Your user agent supports TLS 1.2, which is the best available protocol version at the moment.

Logjam Vulnerability

Your user agent is not vulnerable.
For more information about the Logjam attack, please go to weakdh.org.
To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

FREAK Vulnerability

Your user agent is not vulnerable.
For more information about the FREAK attack, please go to www.freakattack.com.
To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

POODLE Vulnerability

Your user agent is not vulnerable.
For more information about the POODLE attack, please read this blog post.

Protocol Features

Protocols

TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) Forward Secrecy	256
OLD_TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc14) Forward Secrecy	256
OLD_TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc13) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) Forward Secrecy	128

Cipher Suites (in order of preference)

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) Forward Secrecy	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	112

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very first connection to this site. To see the suites, close all browser windows, then open this exact page directly. Don't refresh.

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA512/RSA, SHA512/ECDSA, SHA384/RSA, SHA384/ECDSA, SHA256/RSA, SHA256/ECDSA, SHA1/RSA, SHA1/ECDSA
Elliptic curves	x25519, secp256r1, secp384r1
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

Mixed Content Handling

Mixed Content Tests

Images	Passive	Yes
CSS	Active	No
Scripts	Active	No
XMLHttpRequest	Active	No
WebSockets	Active	No
Frames	Active	No

(1) These tests might cause a mixed content warning in your browser. That's expected.
(2) If you see a failed test, try to reload the page. If the error persists, please get in touch.

Related Functionality

Upgrade Insecure Requests request header (more info)	Yes

SSL/TLS Capabilities of Your Browser

Other User Agents »

User Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.79 Safari/537.36 Edge/14.14393

Protocol Support

Your user agent has good protocol support.
Your user agent supports TLS 1.2, which is the best available protocol version at the moment.

Logjam Vulnerability

Your user agent is not vulnerable.
For more information about the Logjam attack, please go to weakdh.org.
To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

FREAK Vulnerability

Your user agent is not vulnerable.
For more information about the FREAK attack, please go to www.freakattack.com.
To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

POODLE Vulnerability

Your user agent is not vulnerable.
For more information about the POODLE attack, please read this blog post.

Protocol Features

Protocols

TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f) Forward Secrecy	256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) Forward Secrecy	128

Cipher Suites (in order of preference)

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) Forward Secrecy	128
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) Forward Secrecy	256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) Forward Secrecy	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	128
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d)	256
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c)	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	112
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 (0x6a) Forward Secrecy[2]	256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 (0x40) Forward Secrecy[2]	128
TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x38) Forward Secrecy[2]	256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x32) Forward Secrecy[2]	128
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x13) Forward Secrecy[2]	112

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very first connection to this site. To see the suites, close all browser windows, then open this exact page directly. Don't refresh.

(2) Cannot be used for Forward Secrecy because they require DSA keys, which are effectively limited to 1024 bits.

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA256/RSA, SHA384/RSA, SHA1/RSA, SHA256/ECDSA, SHA384/ECDSA, SHA1/ECDSA, SHA1/DSA, SHA512/RSA, SHA512/ECDSA
Elliptic curves	secp256r1, secp384r1
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

Mixed Content Handling

Mixed Content Tests

Images	Passive	Yes
CSS	Active	No
Scripts	Active	No
XMLHttpRequest	Active	No
WebSockets	Active	No
Frames	Active	No

(1) These tests might cause a mixed content warning in your browser. That's expected.

(2) If you see a failed test, try to reload the page. If the error persists, please get in touch.

Related Functionality

Upgrade Insecure Requests request header (more info)	No

SSL/TLS Capabilities of Your Browser

User Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko

Other User Agents »

Protocol Support

Your user agent has good protocol support.

Your user agent supports TLS 1.2, which is the best available protocol version at the moment.

Logjam Vulnerability

Your user agent is vulnerable. Upgrade as soon as possible.

For more information about the Logjam attack, please go to weakdh.org.

To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

FREAK Vulnerability

Your user agent is not vulnerable.

For more information about the FREAK attack, please go to www.freakattack.com.

To test manually, click here. Your user agent is not vulnerable if it fails to connect to the site.

POODLE Vulnerability

Your user agent is not vulnerable.

For more information about the POODLE attack, please read this blog post.

Protocol Features

Protocols

TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f) Forward Secrecy	256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) Forward Secrecy	256

Cipher Suites (in order of preference)

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) Forward Secrecy	128
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) Forward Secrecy	256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) Forward Secrecy	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	128
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d)	256
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c)	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	112
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 (0x6a) Forward Secrecy[2]	256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 (0x40) Forward Secrecy[2]	128
TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x38) Forward Secrecy[2]	256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x32) Forward Secrecy[2]	128
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x13) Forward Secrecy[2]	112

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very first connection to this site. To see the suites, close all browser windows, then open this exact page directly. Don't refresh.

(2) Cannot be used for Forward Secrecy because they require DSA keys, which are effectively limited to 1024 bits.

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA256/RSA, SHA384/RSA, SHA1/RSA, SHA256/ECDSA, SHA384/ECDSA, SHA1/ECDSA, SHA1/DSA, SHA512/RSA, SHA512/ECDSA
Elliptic curves	secp256r1, secp384r1
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

Mixed Content Handling

Mixed Content Tests

Images	Passive	Yes
CSS	Active	Yes
Scripts	Active	Yes
XMLHttpRequest	Active	No
WebSockets	Active	No
Frames	Active	Yes

(1) These tests might cause a mixed content warning in your browser. That's expected.

(2) If you see a failed test, try to reload the page. If the error persists, please get in touch.

Related Functionality

Upgrade Insecure Requests request header (more info)	No