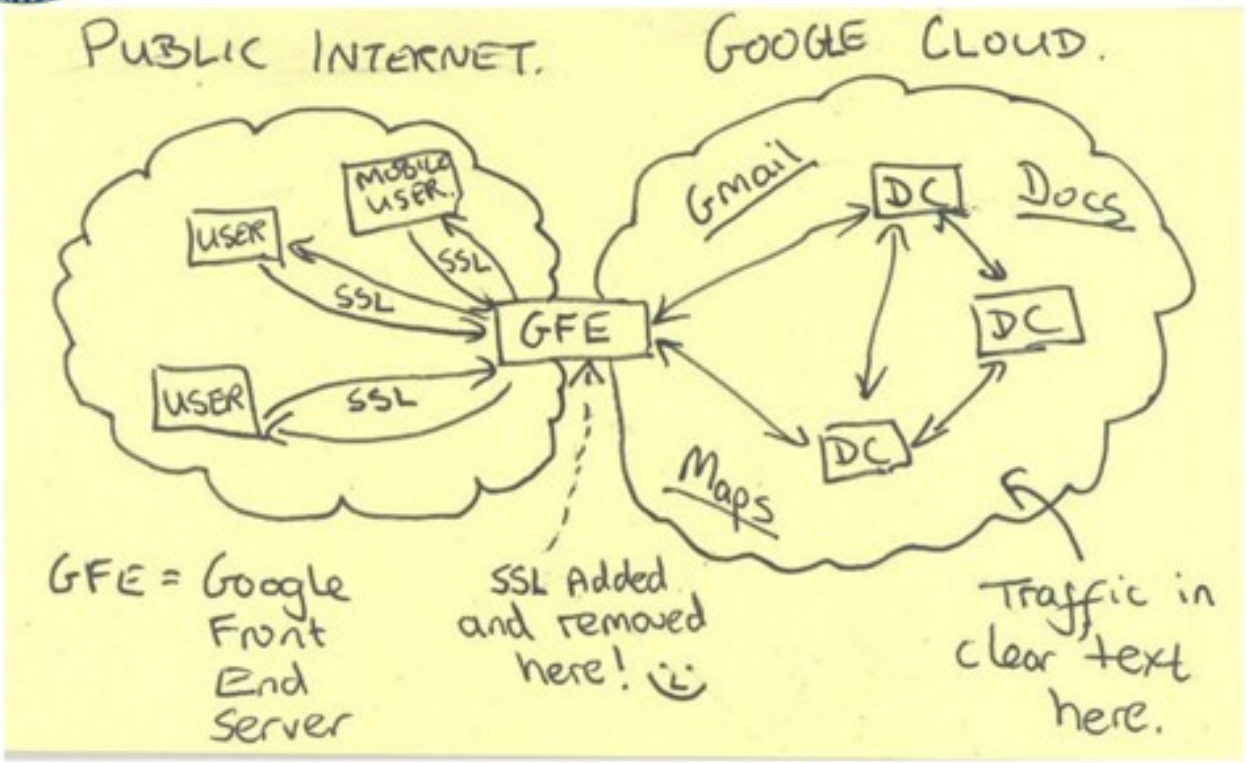
Varnost HTTPS povezav

Matej Kovačič
https://telefoncek.si

HTTPS deluje
(če je pravilno implementiran)

TOP SECRET//SI//NOFORN

Current Efforts - Google



TOP SECRET//SI//NOFORN

+ Lavabit!

HTTPS deluje
(če je pravilno implementiran)

"*Encryption works. Properly implemented strong crypto systems are one of the few things that you can rely on.*
Unfortunately, endpoint security is so terrifically weak that NSA can frequently find ways around it."

--Edward Snowden

Kaj je HTTPS?

- HTTPS (*HyperText Transfer Protocol Secure*) je protokol za šifrirano spletno komunikacijo.
- HTTP: TCP/80, HTTPS: TCP/443.
- Šifriranje poteka preko SSL (*Secure Sockets Layer*) ali TLS (*Transport Layer Security*) kriptografskega protokola, pri samem šifriranju pa se uporabljajo različni šifrirni algoritmi z različnimi dolžinami ključev.

Nekateri napadi na HTTPS

Nekateri napadi na HTTPS

- Ponovno pogajanje za vzpostavitev HTTPS seje (ang. *session renegotiation attack*).
- Napad z vsiljevanjem šibkejše različice protokola (ang. *version rollback attack*).
- BEAST napad.
- CRIME in BREACH napadi.
- Napadi na bitno zapolnjevanje.
- Napadi na RC4.
- Napadi na kriptografsko implementacijo.
- Napad s posrednikom.

Ponovno pogajanje za vzpostavitev HTTPS seje
(*session renegotiation attack*)

- Napadalec:
  - `GET /index.php?narocilo=burek&vrsta=sirni&naslov=napadalcev%20naslov HTTP/1.1`
  - `X-Ignore-This:` **[brez *newline* znaka!; sledi *session renegotiation*]**
- Žrtev napada:
  - `GET /index.php?narocilo=burek&vrsta=mesni&naslov=zrtvin%20naslov HTTP/1.1`
  - `Cookie: zrtvin_piskotek`
- Spletni strežnik:
  - `GET /index.php?narocilo=`**`burek`**`&vrsta=`**`sirni`**`&naslov=napadalcev%20naslov HTTP/1.1`
  - `X-Ignore-This: GET /index.php?narocilo=`**`burek`**`&vrsta=`**`mesni`**`&naslov=zrtvin%20naslov HTTP/1.1`
  - `Cookie: zrtvin_piskotek`
- Ranljivi so SSL 3 in TLS 1.0 do 1.2.
- Zaščita: konfiguracija, ki ne dovoli ponovnega pogajanja za vzpostavitev HTTPS seje (RFC 5746).

Napad z vsiljevanjem šibkejše različice protokola

(*version rollback attack*)

- Napadalec se postavi med strežnik in odjemalca ter odjemalca prepriča, da strežnik podpira samo starejše, ranljive različice HTTPS protokola.
- Hkrati tudi strežnik prepriča, da odjemalec podpira samo starejše, ranljive različice HTTPS protokola.
- Komunikacija je zato sicer šifrirana, a steče s pomočjo ranljivega protokola.

- Zaščita: ne dovolimo uporabe šibkih protokolov in algoritmov.

BEAST napad
(Browser Exploit Against SSL/TLS)

- Gre za napad s pomočjo tim. znanega čistopisa (ang. *known plaintext*) oz. vnaprej izbranega čistopisa (ang. *chosen-plaintext attack*).
- Napadalec s pomočjo JavaScript vstavka preko brskalnika strežniku pošilja vnaprej znane podatke (čistopis), hkrati pa te podatke prestreže tudi v šifrirani obliki (kriptogram).
- Na podlagi čistopisa in kriptograma nato izvede kriptoanalizo ter s tem rekonstruira sejni ključ.
- Ranljivost je bila odpravljena v TLS 1.1.

CRIME in BREACH napadi

(Compression Ratio Info-Leak Mass Exploitation* ter *Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext)

- Napada napadalcu omogočata kriptoanalizo šifriranih podatkov oz. rekonstrukcijo spletnih piškotkov v primeru, da je pri šifriranju uporabljena kompresija podatkov.
- S pomočjo BREACH napada, je HTTPS šifrirano sejo mogoče ugrabiti v manj kot minuti.
- Pri BREACH napadih gre za serijo ranljivosti in ne en sam napad.
- Rešitev pred napadom je izklop kompresije šifriranih podatkov (pomemben je vrstni red – *najprej kompresija, nato šifriranje!*).

Napadi na bitno zapolnjevanje
(*padding oracle attack*)

- Gre za napad, ki izkorišča ugibanje vsebine tim. bitnega zapolnjevanja (ang. *padding*) pri šifriranju sporočil.
- Ranljivi so navadno ECB (*Electronic Code Book*; vsak blok sporočila šifriramo z istim ključem) in CBC (*Cipher Block Chaining*; začetni blok sporočila ter naključno število imenovano inicializacijski vektor (IV) seštejemo z operacijo XOR, vsak naslednji blok sporočila seštejemo s šifriranim prejšnjim blokom in ga zašifriramo).
- Eden bolj znanih napadov na bitno zapolnjevanje je Lucky 13 napad. Ranljivost je bila odpravljena v OpenSSL 1.0.1d.

Napadi na RC4

- RC4 je, da je v določenih načinih uporabe zelo ranljiv (uporaba ne-naključnih šifrirnih ključev oz. Večkratna uporaba istih ključev).
- RC4 se je v preteklosti celo priporočal kot možna obramba pred BEAST napadom.
- Microsoft je konec 2013 priporočil, da se RC4 preneha uporabljati.
- Napadi na RC4 postanejo uporabni v praksi ko ima napadalec na voljo dovolj veliko količino prestreženih šifriranih podatkov. => PFS!
- *Tradeoff*: RC4 uporabljajo nekateri starejši brskalniki.

Napadi na kriptografsko implementacijo

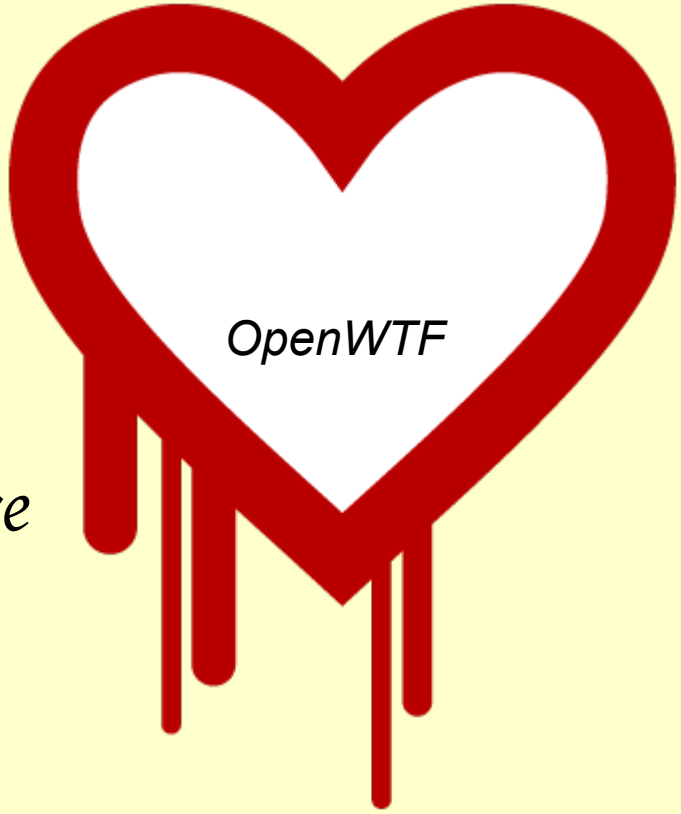
- Močna kriptografija sicer močno pripomore k varnosti, vendar pa do večine napak pride pri implementaciji.
- S pomočjo napak v implementaciji je mogoče odkriti notranje stanje kriptografskega sistema, kar napadalcu v končni fazi lahko omogoči celo rekonstrukcijo dešifriranih podatkov ali šifrirnih ključev.

- Rešitev: poskrbimo, da bo programska oprema na našem sistemu vedno posodobljena.

Napadi na kriptografsko implementacijo

- Primer: Heartbleed napad.
  Preverjanje ranljivosti:
  <http://filippo.io/Heartbleed/>

- *»To mitigate against the openssl heartbleed vulnerability, we now require all passwords to be longer than 64k.«*



Napad s posrednikom

- Rešitev: uporaba zaupanja vrednih digitalnih potrdil. Rešitve so bolj na strani odjemalca.
- Problem: je CA res zaupanja vreden?

Ukrepi za povečanje HTTPS varnosti

Kaj je pomembno?

- Varnost strežniških digitalnih potrdil (vsaj 2048 bitov, še bolje 4096 bitov; samopodpisana ali ne (StartSSL)?).
- Kriptografski protokoli (onemogočimo SSL v2, smiselno je onemogočiti tudi SSL v3, vklop TLS 1.2).
- Kriptografski algoritmi (nastavimo preferenčni red).
- Vključimo poudarjeno zaupnost (ang. *perfect forward security* oz. *perfect forward secrecy*).
  - Poudarjena zaupnost pomeni, da se šifrirni ključi samodejno spreminjajo vsako novo sejo oz. na določeno časovno obdobje. Če napadalec uspe razbiti enega izmed šifrirnih ključev, bo lahko dešifriral le promet, ki je bil šifriran v dani seji.

Kaj je pomembno?

- Omogočimo *Strict Transport Security*.
- Povsem onemogočimo HTTP promet in vse zahtevke brskalnikov preusmerimo na HTTPS protokol?
- Onemogočimo ponovno pogajanje za vzpostavitev SSL seje (*SSL session renegotiation*), saj omogoča MITM napad.
- Problematična sta tudi tim. *Secure* ter *Insecure Client-Initiated Renegotiation*, saj omogočata napad z onemogočanjem storitve (DoS napad).

Kaj je pomembno?

- Preverimo ali ima strežnik nastavljeno tim. netoleranco previsoke TLS različice (ang. *TLS version intolerance*), s čimer v prihodnosti preprečimo napad z vsiljevanjem šibkejše različice protokola (ang. *version rollback attack*).
- Redno posodabljajmo zaledne kriptografske knjižnice (zlasti OpenSSL :-)) ter celoten sistem.

HTTPS optimizacija?

- Vključimo nadaljevanje SSL seje (tim. *session resumption*).
  - Gre za vrsto optimizacije, ki pohitri kasnejše (ponovno) vzpostavljanje HTTPS povezave.
- Omogočimo OCSP pripenjanje (ang. *OCSP stapling*).
  - Gre za optimizacijo oz. pohitritev preverjanja veljavnosti digitalnega potrdila preko OCSP strežnikov (*Online Certificate Status Protocol* je poseben protokol, ki omogoča preverjanje veljavnosti izdanih digitalnih potrdil).

Primer: Nginx

- Namestimo najnovejšo različico.
- Različica 1.3 omogoča OCSP pripenjanje (ang. *OCSP stapling*).
- Parametri:
  - samodejno preusmeritev vseh HTTP zahtevkov na HTTPS: *rewrite*;
  - nastavitev Diffie-Hellmanovih parametrov: *ssl_dhparam*;
  - vklop nadaljevanja SSL seje: *ssl_session_cache* ter *ssl_session_timeout*;
  - nastavitev preferenc šifrirnih algoritmov: *ssl_ciphers*;
  - dodajanje STS vzglavja: *add_header*.

```
# HTTP streznik
server {
    listen 80 default_server;
    server_name moj.streznik.si;

    # Samodejna preusmeritev na HTTPS
    rewrite ^ https://moj.streznik.si$request_uri? permanent;
}

# HTTPS streznik
server {
    listen 443 ssl;
    server_name moj.streznik.si;

    ssl on;
    ssl_certificate /etc/nginx/web-server-chained.crt;
    ssl_certificate_key /etc/nginx/web-server.key;
    ssl_dhparam /etc/nginx/dhparam_4096.pem;
    # Vklop nadaljevanja SSL seje (tim. session resumption)
    ssl_session_cache    shared:SSL:10m;
    ssl_session_timeout  10m;

    # Uporabimo sodobne sifrirne protokole ter Perfect Forward
Secrecy, zascita pred BEAST napadom
    ssl_prefer_server_ciphers on;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
```

(se nadaljuje)

```
    ssl_ciphers
EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EECDH+ECDSA+SHA384:EECDH+ECDSA+S
HA256:EECDH+aRSA+SHA384:EECDH+aRSA+SHA256:EECDH+aRSA+RC4:EECDH:EDH+aR
SA:RC4-SHA:RC4:HIGH:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!PSK:!SRP:!
DSS;
    # OPOMBA:
    # - ce zelimo, lahko omogocimo RC4 samo z novejšimi brskalniki (v
tem primeru BEAST napad ni onemogočen s strani streznika (server side
mitigation)):
    # :+RC4:RC4;
    # - ce zelimo, lahko onemogocimo RC4 (v tem primeru BEAST napad
ni onemogočen s strani streznika (server side mitigation)):
    # :!RC4;

    # Lokacija spletnih datotek
    root /usr/share/nginx/www;
    index index.html index.htm;

    # Dodajanje STS vzglavja
    add_header Strict-Transport-Security max-age=31536000;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ /index.html;
    }

}
```

Primer: Apache2

- Namestimo najnovejšo različico.
- Omogočimo Apachejev SSL modul:
  - a2enmod ssl
- Naložimo modul za HTTP vzglavja (mod-headers.so):
  - cp -arp /etc/apache2/mods-available/headers.load /etc/apache2/mods-enabled/headers.load
- Za preprečitev CRIME in BREACH napada je potrebno izključiti SSL kompresijo.
  - Parameter SSLCompression je mogoče uporabiti v Apache različicah od 2.2.24 dalje.
  - starejše različice: v /etc/apache2/envvars na konec dodamo: *export OPENSSL_NO_DEFAULT_ZLIB=1*

```
<VirtualHost *:443>
 ServerAdmin admin@server.si
 ServerName  server.si
 ServerAlias www.server.si

 ...

 # Vkljucimo HTTPS
 SSLEngine On

 # Dolocimo lokacijo digitalnih potrdil, vkljucimo tudi tim. vmesno
potrdilo
 SSLCertificateFile /etc/apache2/ssl/server.si.crt
 SSLCertificateKeyFile /etc/apache2/ssl/server.si.key
 SSLCertificateChainFile /etc/apache2/ssl/sub.class1.server.ca.pem

 # Od Apache razlicice 2.4.2 dalje je omogocena uporaba DH
parametrov.
 SSLDHParametersFile /etc/apache2/ssl/dhparam_4096.pem

 # Izkljucimo manj varne protokole
 SSLProtocol All -SSLv2 -SSLv3

 # Dolocimo, da se upostevа preferencni vrstni red sifrirnih
algoritmov streznika in ne odjemalcev
 SSLHonorCipherOrder On
```

(se nadaljuje)

```
 # Za preprecitev CRIME in BREACH napada je potrebno izkljuciti SSL
kompresijo.
 # Parameter je mogoce uporabiti v Apache razlicicah od 2.2.24 dalje.
 SSLCompression off

 # Dodamo HSTS vzglavje
 Header add Strict-Transport-Security "max-age=31536000"

 # Preferencni vrstni red sifrirnih algoritmov
 SSLCipherSuite
'EDH+CAMELLIA:EDH+aRSA:EECDH+aRSA+AESGCM:EECDH+aRSA+SHA384:EECDH+aRSA
+SHA256:EECDH:+CAMELLIA256:+AES256:+CAMELLIA128:+AES128:+SSLv3:!
aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!
ECDSA:CAMELLIA256-SHA:AES256-SHA:CAMELLIA128-SHA:AES128-SHA'

 # Resevanje HTTPS tezav z brskalnikom Internet Explorer
 # Vec o tem na: https://httpd.apache.org/docs/2.2/ssl/ssl_faq.html
 SetEnvIf User-Agent ".*MSIE.*" \
   nokeepalive ssl-unclean-shutdown \
   downgrade-1.0 force-response-1.0
...
</VirtualHost>
```

HTTPS in spletni brskalnik

HTTPS varnost ima dva konca

- Starejši odjemalci (npr. Internet Explorer 6.0, itd.) se na dovolj varno nastavljeno HTTPS spletno stran morda sploh ne bodo mogli povezati.
- Uporaba TLS 1.2 (Firefox je privzeto podporo za TLS 1.2 dobil šele pred kratkim z različico 27). :-(
  - Test brskalnika: https://www.howsmyssl.com/.
- Pripenjanje digitalnih potrdil (ang. *certificate pinning*): zaupanje ne temelji več na overoviteljski verigi zaupanja, pač pa zaupamo samo točno določenim digitalnim potrdilom oziroma digitalnim potrdilom, ki jih je izdal zgolj točno določen overitelj.

HTTPS varnost ima dva konca

- Identiteto (kontrolno vsoto potrdila) brskalniku sporoči HTTPS strežnik preko posebnega HTTP vzglavja, veljavnost potrdila pa je časovno določena - če se torej potrdilo predčasno spremeni, je to pri brskalniku znak za alarm oz. sum, da je morda prišlo do zlorabe.
- TOFU/POP: TOFU (ang. *Trust-On-First-Use*) / POP (ang. *Persistence Of Pseudonym*) - odjemalec si ob prvi povezavi do HTTPS strežnika zapomni digitalni prstni odtis njegovega digitalnega potrdila. Ob ponovnem (kasnejšem) obisku nato odjemalec preveri ali je digitalni podpis enak, ali pa se je spremenil. V slednjem primeru uporabniku prikaže obvestilo oz. opozorilo.

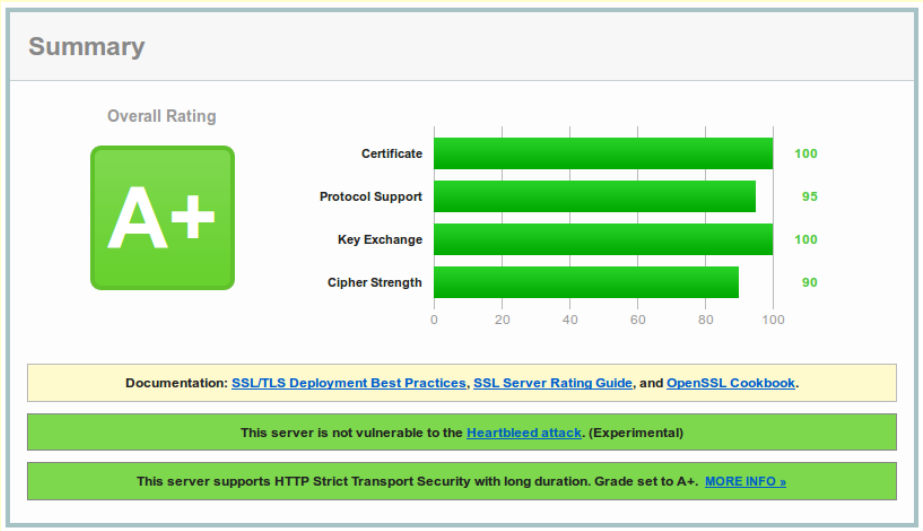
HTTPS varnost ima dva konca

- Firefox dodatek *Certificate Patrol*, *Pet Name Tool*.
- *HTTPS Everywhere* (Firefox, Chrome in Opera): samodejno preusmerjanje na HTTPS povezave.
- *SSL Observatorij*: naprednejši TOFU pristop; omogoča beleženje v katerem omrežju smo skušali dostopati do HTTPS strežnika ter katero potrdilo smo prejeli. S tem je mogoče ugotoviti katera omrežja oz. katere države izvajajo MITM napade.

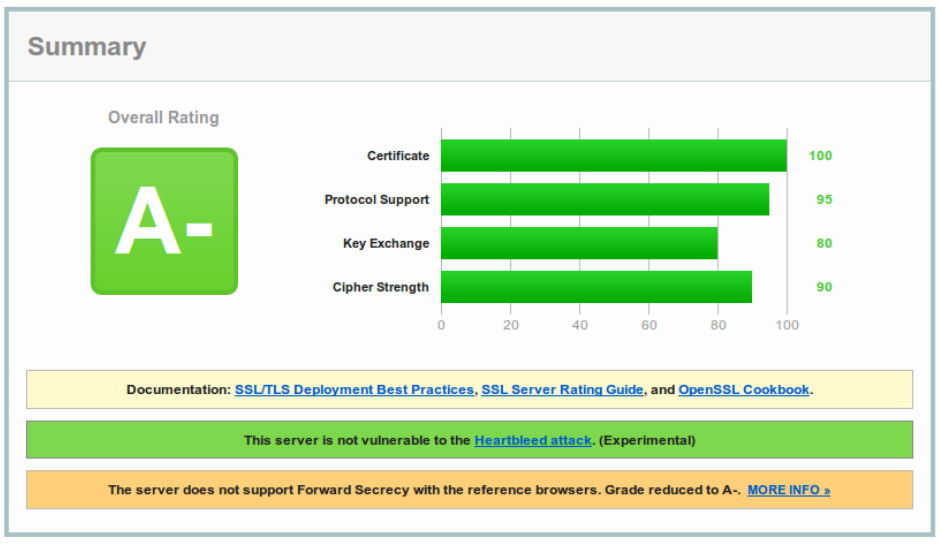
HTTPS varnost v praksi...

SSL testiranje slovenskih spletnih strani

- Qualys SSL Labs test, https://www.ssllabs.com/ssltest/



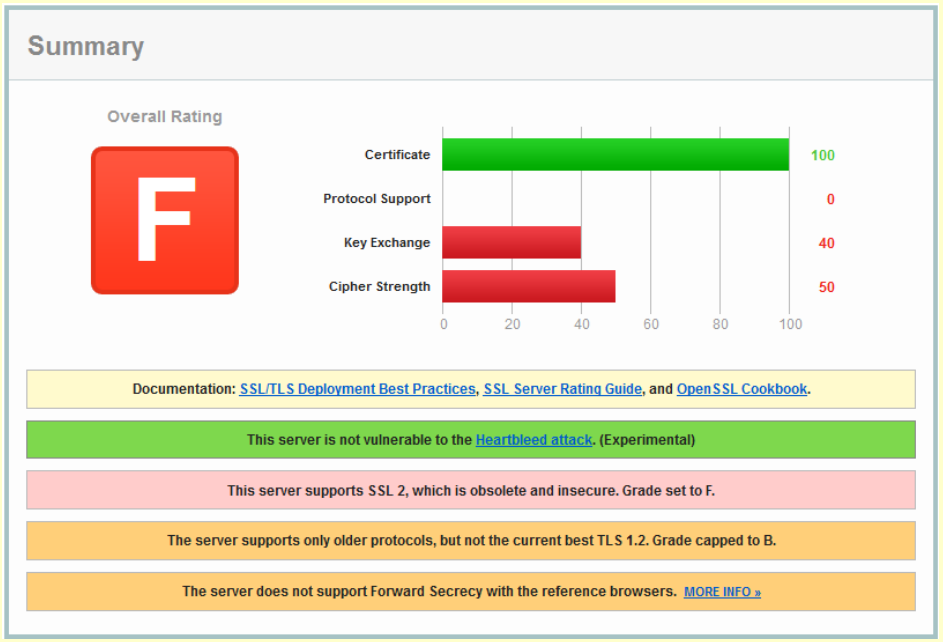
Nginx/1.4.6 (Debian7.4)



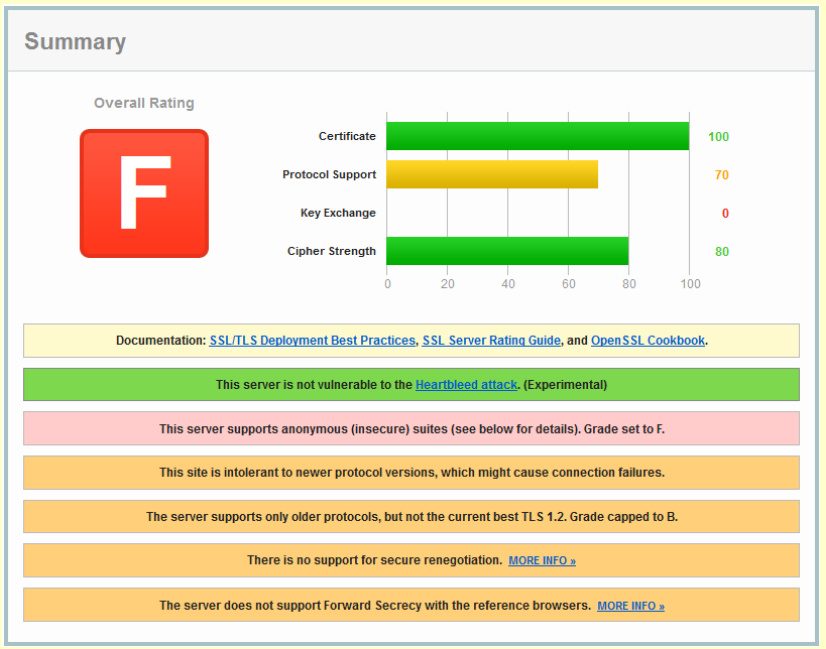
Apache/2.2.22 (Ubuntu 12.04.4 LTS)

"Spletne strani slovenskih bank so precej dobro zavarovane, večina jih tudi uporablja Microsoftove rešitve, ki niso ranljive za programsko pomanjkljivost Heartbleed."

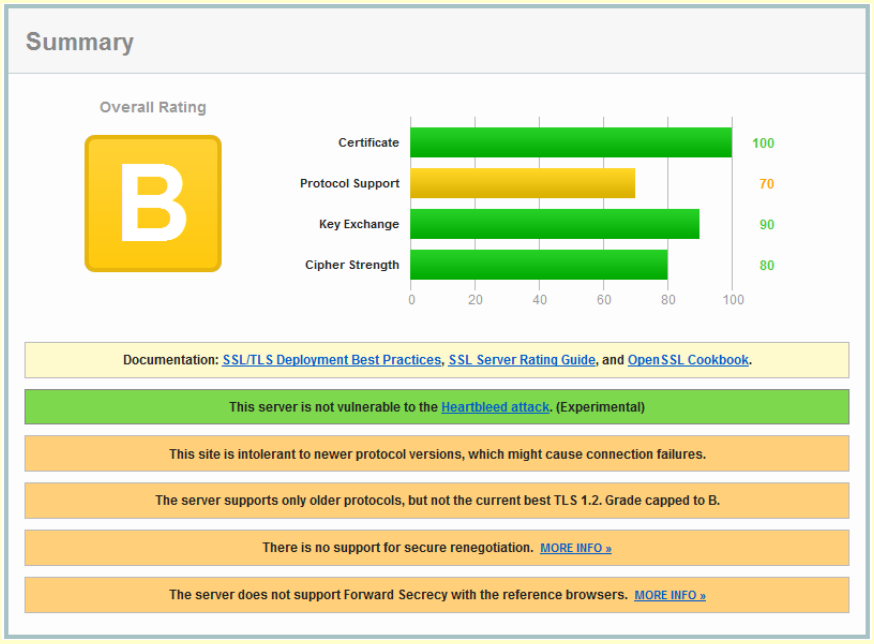
--Tadej Hren, SI-CERT, za STA



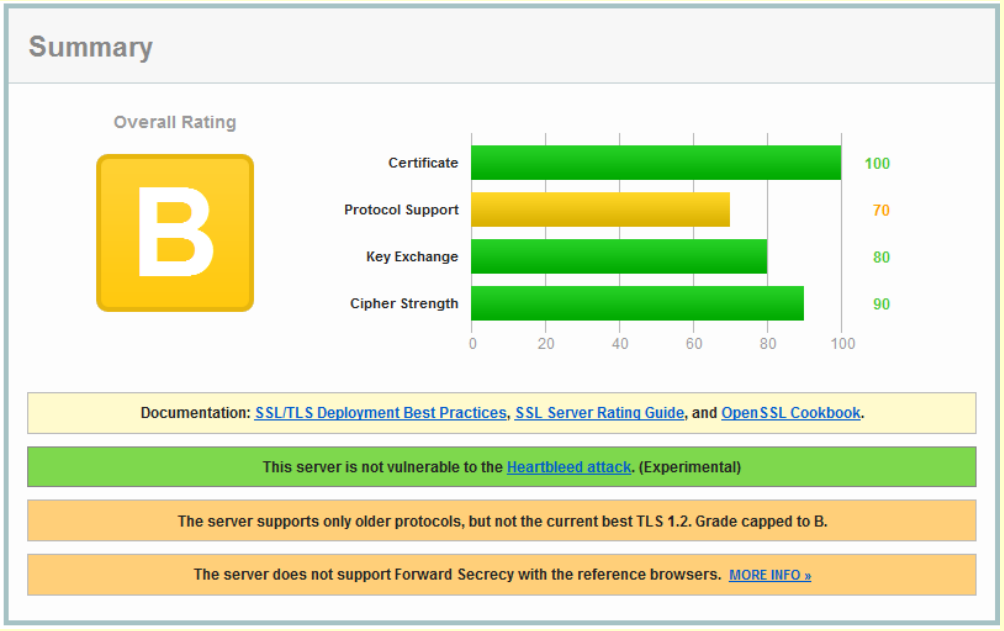
pbspikanet.pbs.si



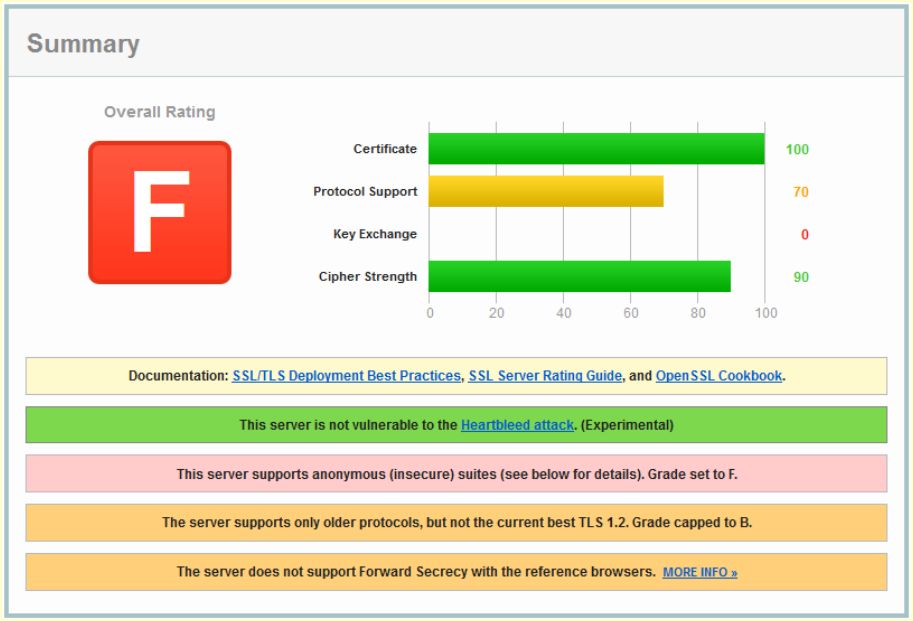
splet.probanka.si



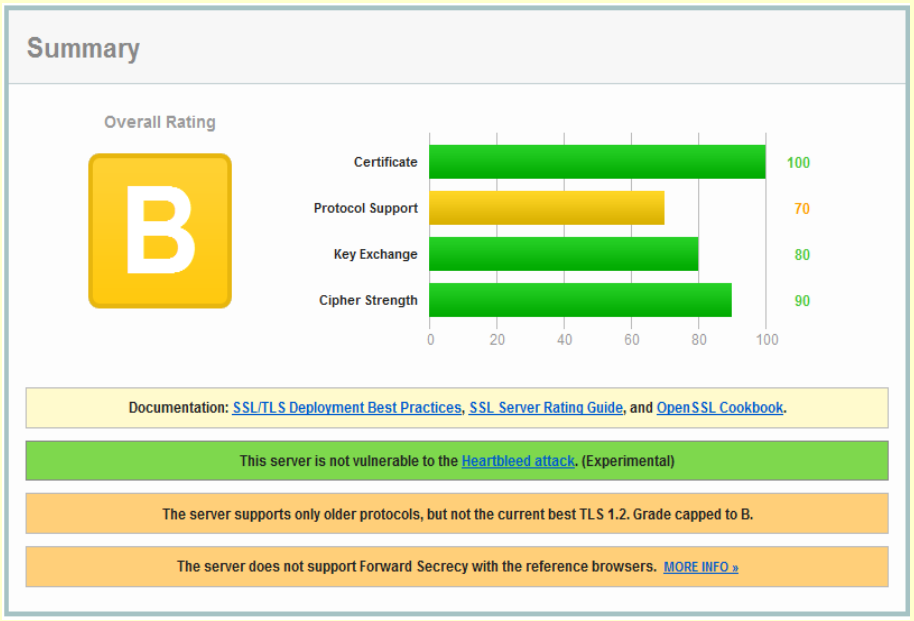
eureka.raiffeisen.si



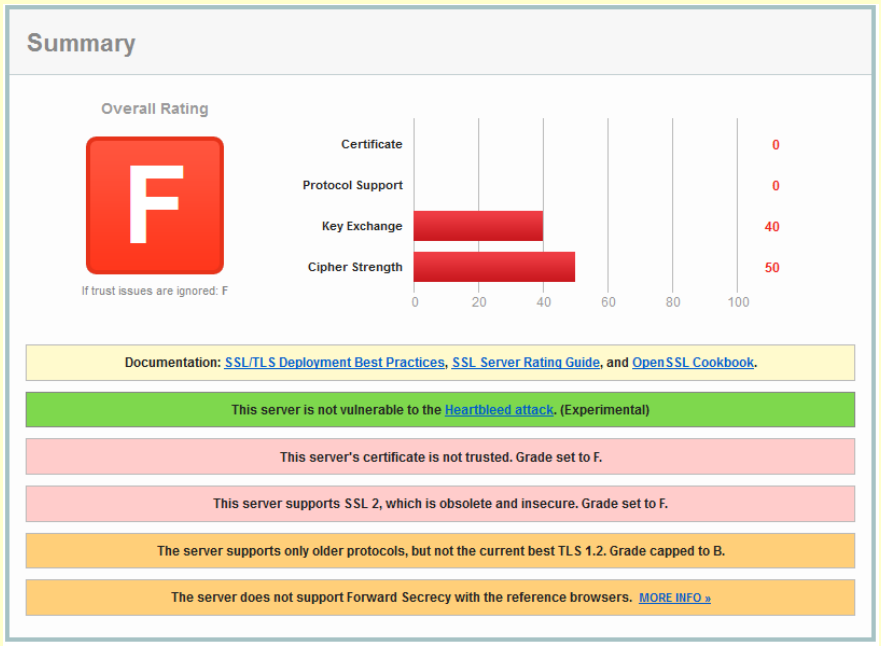
BANKANET.NKBM.SI



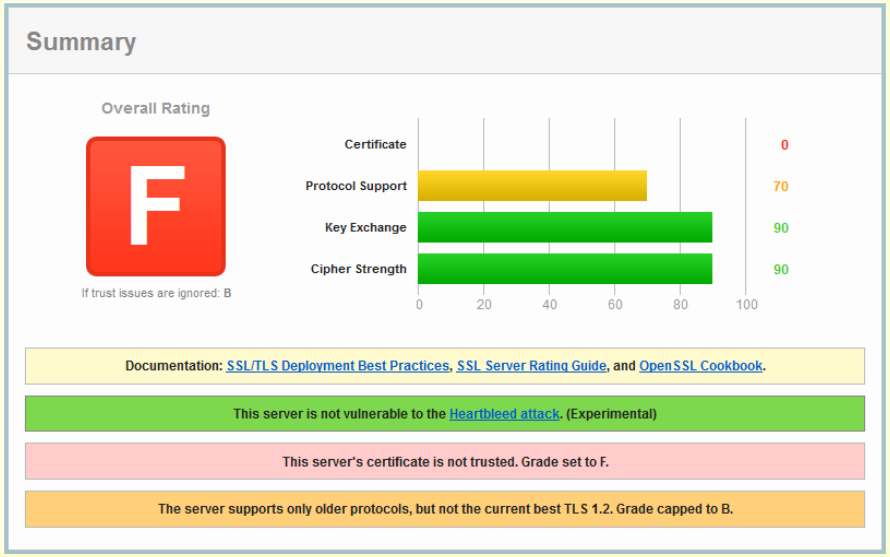
*.sid.si



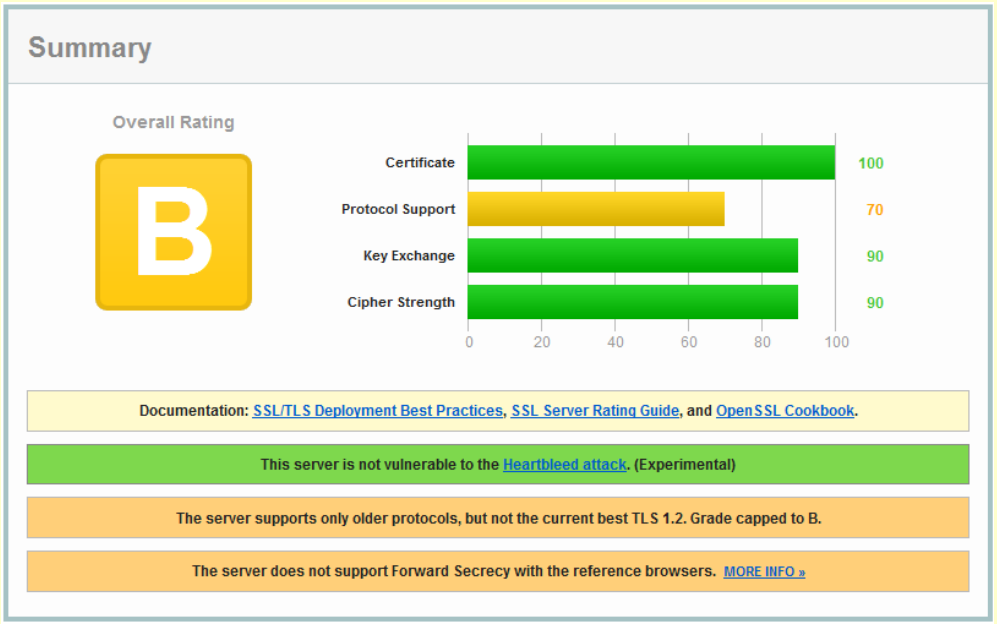
www.skb.net



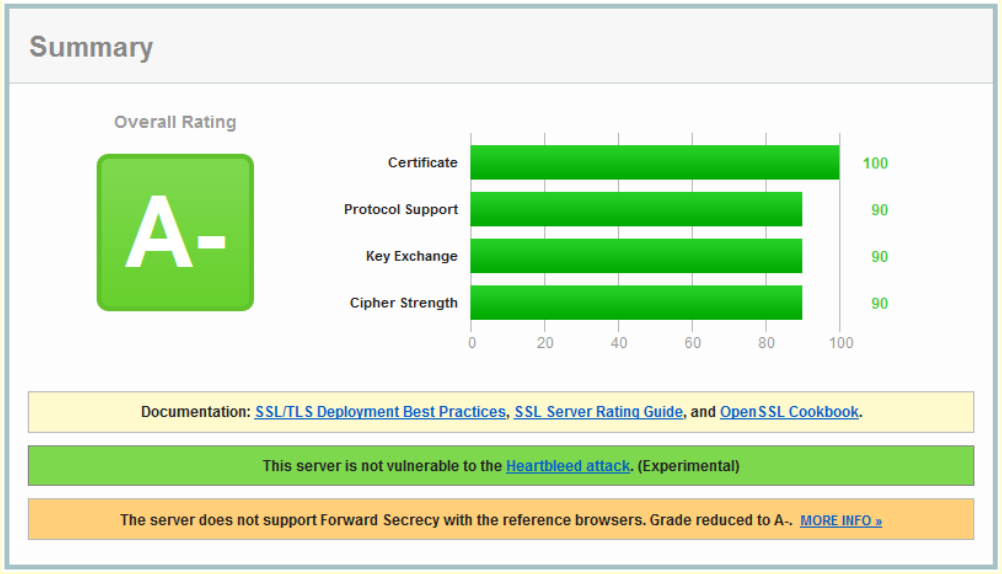
ujpnet.gov.si



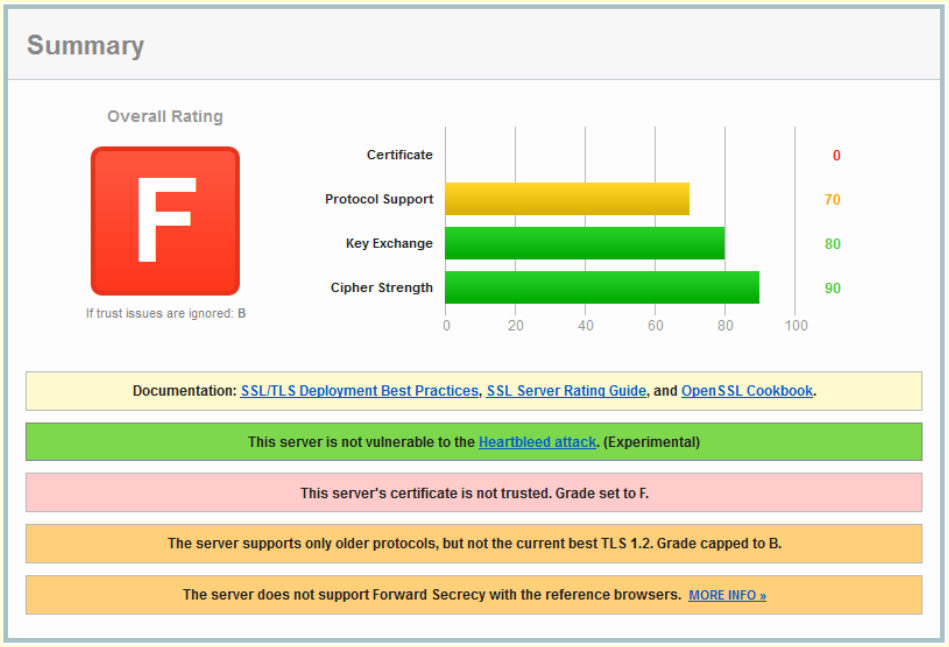
www.unicreditbank.si



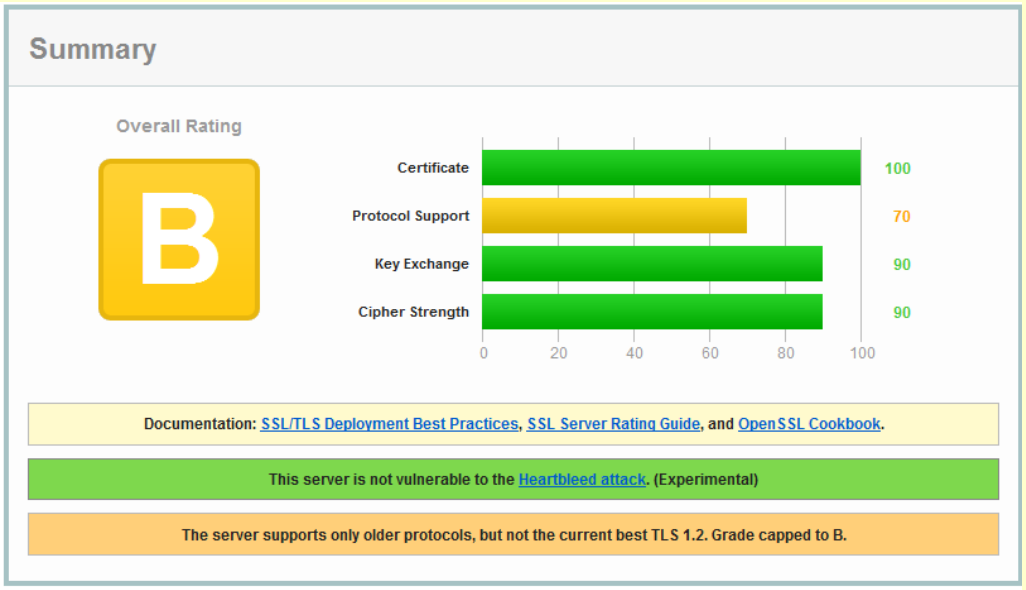
mpoti.abanka.si



netstik.sparkasse.si



klik.nlb.si

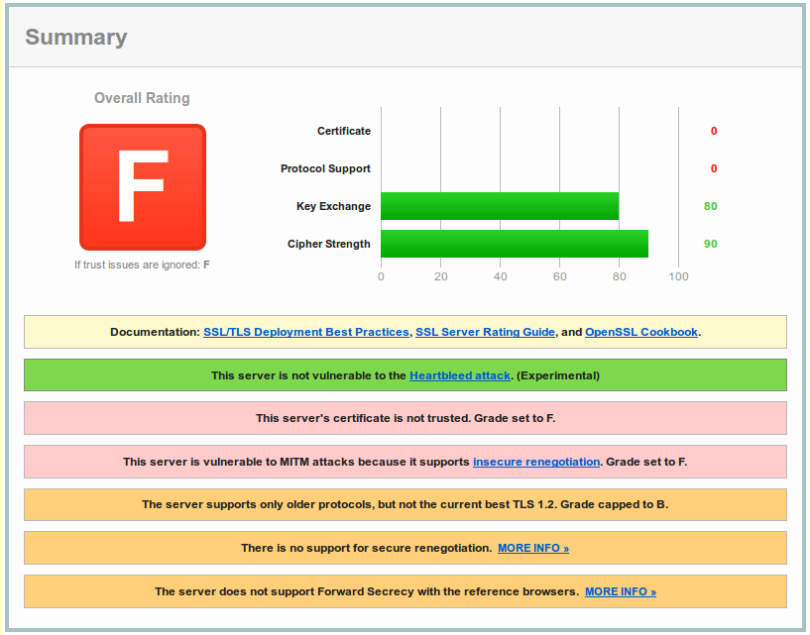


www.bsi.si

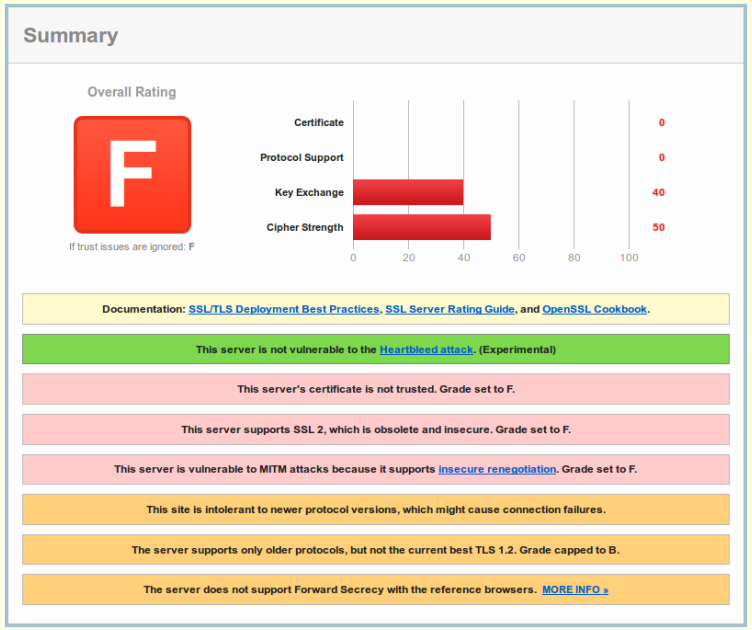
"informacijski sistemi slovenske javne uprave so varni pred morebitnimi zlorabami, saj pomanjkljivega šifrirnega mehanizma OpenSSL ne uporabljajo«
-- Ministrstvo za notranje zadeve

"varnost spletnega mesta eDavki ni ogrožena" ... "v zvezi s tem nimamo nobenih težav"

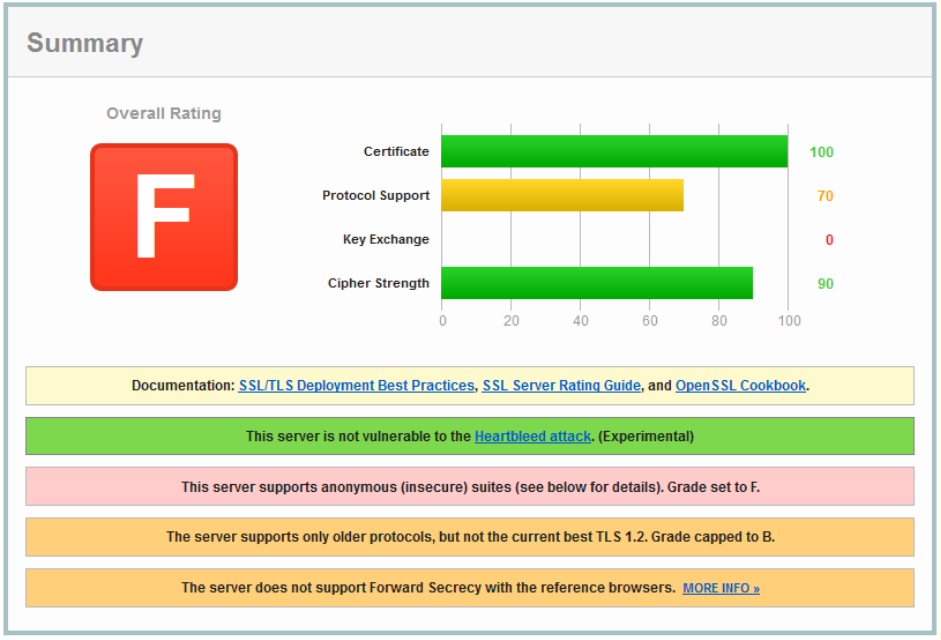
--DURS



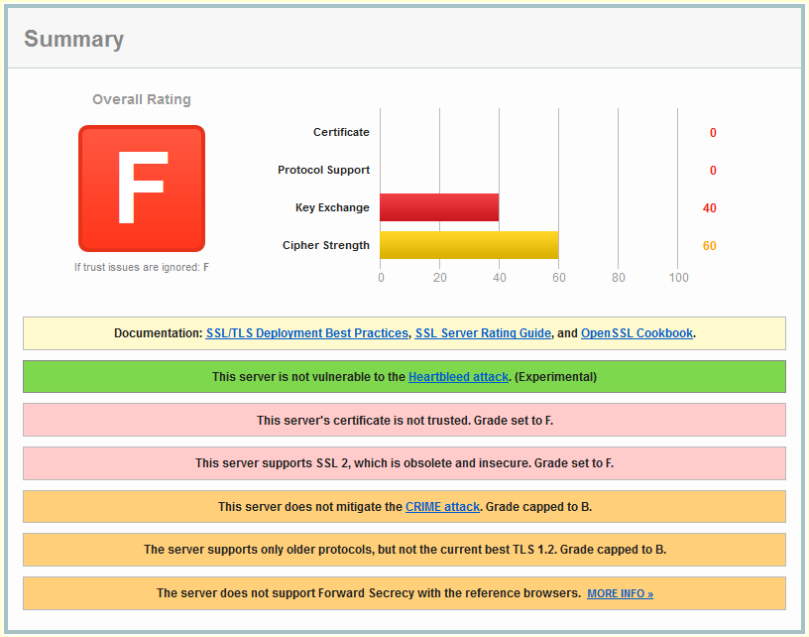
eDavki



e-Uprava



posta.oddo.gov.si



vpndist.gov.si

Vprašanja?