Summary

Overall Rating

- Certificate: 0
- Protocol Support: 95
- Key Exchange: 80
- Cipher Strength: 90

If trust issues are ignored: B

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

This server's certificate is not trusted, see below for details.

This server's certificate chain is incomplete. Grade capped to B.

This server supports TLS_FALLBACK_SCSV to prevent protocol downgrade attacks.

Authentication

Server Key and Certificate #1

- Common names: svn.cacert.org
- Alternative names: svn.cacert.org cert.svn.cacert.org nocert.svn.cacert.org
- Prefix handling: Not required for subdomains
- Valid from: Tue Apr 08 00:08:21 PDT 2014
- Valid until: Thu Apr 07 00:08:21 PDT 2016 (expires in 1 year and 3 months)
- Key: RSA 4096 bits (e 65537)
- Weak key (Debian): No
- Issuer: CAcert Class 3 Root
- Signature algorithm: SHA512withRSA
- Extended Validation: No
- Revocation information: CRL, OCSP
- Revocation status: Unchecked (only trusted certificates can be checked)
- Trusted: NO TRUSTED (Why?)

Additional Certificates (if supplied)

- Certificates provided: 2 (3562 bytes)
- Chain issues: Incomplete, Extra certs, Contains anchor
Additional Certificates (if supplied)

#2

Subject
CA Cert Signing Authority  Not in trust store
Fingerprint: 135cec36f49bd8e0b1ab270c80884676be8f33

Valid until
Tue Mar 29 05:29:49 PDT 2033 (expires in 18 years and 3 months)

Key
RSA 4096 bits (e 65537)

Issuer
CA Cert Signing Authority  Self-signed

Signature algorithm
MD5withRSA  Weak, but no impact on root certificate

Certification Paths

No trust paths available
Issuer unknown, or intermediate certificate(s) missing.

Configuration

Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS 1.2</td>
<td>Yes</td>
</tr>
<tr>
<td>TLS 1.1</td>
<td>Yes</td>
</tr>
<tr>
<td>TLS 1.0</td>
<td>Yes</td>
</tr>
<tr>
<td>SSL 3</td>
<td>No</td>
</tr>
<tr>
<td>SSL 2</td>
<td>No</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Cipher Suite</th>
<th>Protocol</th>
<th>Key Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)</td>
<td>FS</td>
<td>128</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)</td>
<td>FS</td>
<td>128</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)</td>
<td>FS</td>
<td>128</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x6b)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x88)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (0x9e)</td>
<td>FS</td>
<td>128</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (0x33)</td>
<td>FS</td>
<td>128</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)</td>
<td>FS</td>
<td>128</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA (0x35)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x84)</td>
<td>FS</td>
<td>256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256 (0x3c)</td>
<td>FS</td>
<td>128</td>
</tr>
<tr>
<td>TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x2f)</td>
<td>FS</td>
<td>128</td>
</tr>
<tr>
<td>TLS_RSA_WITH_SEED_CBC_SHA (0x96)</td>
<td>FS</td>
<td>128</td>
</tr>
</tbody>
</table>
### Cipher Suites (SSL 3+ suites in server-preferred order; deprecated and SSL 2 suites always at the end)

| TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x41) | 128 |

### Handshake Simulation

#### Android 2.3.7
- No SNI
- TLS 1.0: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) FS

#### Android 4.0.4
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Android 4.1.1
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Android 4.2.2
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Android 4.3
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Android 4.4.2
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0x8) FS

#### BingBot Dec 2013
- No SNI
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### BingPreview Jun 2014
- TLS 1.0: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) FS

#### Chrome 39 / OS X R
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Firefox 31.0 ESR / Win 7
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Firefox 34 / OS X R
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Googlebot Jun 2014
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### IE 6 / XP
- No FS
- Protocol or cipher suite mismatch

#### IE 7 / Vista
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### IE 8 / XP
- No FS
- Protocol or cipher suite mismatch

#### IE 8-10 / Win 7 R
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### IE 11 / Win 7 R
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### IE 11 / Win 10 Preview R
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0x8) FS

#### IE Mobile 10 / Win Phone 8.0
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### IE Mobile 11 / Win Phone 8.1
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Java 6u45
- No SNI
- TLS 1.0: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) FS

#### Java 7u25
- TLS 1.0: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) FS

#### Java 8u12
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### OpenSSL 0.9.8y
- TLS 1.0: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) FS

#### OpenSSL 1.0.1h
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0x8) FS

#### Safari 5.1.9 / OS X 10.6.8
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Safari 6 / OS 6.0.1 R
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Safari 7 / iOS 7.1 R
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Safari 8 / iOS 8.0 Beta R
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Safari 6.0.4 / OS X 10.8.4 R
- TLS 1.0: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Safari 7 / iOS X 10.9 R
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0x8) FS

#### Yahoo Slurp Jun 2014
- No SNI
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0x8) FS

#### YandexBot Sep 2014
- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0x8) 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 tend to 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

<table>
<thead>
<tr>
<th>Secure Renegotiation</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Client-Initiated Renegotiation</td>
<td>No</td>
</tr>
<tr>
<td>Insecure Client-Initiated Renegotiation</td>
<td>No</td>
</tr>
<tr>
<td>BEAST attack</td>
<td>Not mitigated server-side (more info) TLS 1.0: 0xc814</td>
</tr>
</tbody>
</table>

### Protocol Details

<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>POODLE (SSLv3)</td>
<td>No, SSL 3 not supported (more info)</td>
</tr>
<tr>
<td>POODLE (TLS)</td>
<td>No (more info)</td>
</tr>
<tr>
<td>Downgrade attack prevention</td>
<td>Yes, TLS_FALLBACK_SCSV supported (more info)</td>
</tr>
<tr>
<td>TLS compression</td>
<td>No</td>
</tr>
<tr>
<td>RC4</td>
<td>No</td>
</tr>
<tr>
<td>Heartbeat (extension)</td>
<td>Yes</td>
</tr>
<tr>
<td>Heartbleed (vulnerability)</td>
<td>No (more info)</td>
</tr>
<tr>
<td>OpenSSL CCS vuln. (CVE-2014-0224)</td>
<td>No (more info)</td>
</tr>
<tr>
<td>Forward Secrecy</td>
<td>Yes (with most browsers) ROBUST (more info)</td>
</tr>
<tr>
<td>Next Protocol Negotiation (NPN)</td>
<td>No</td>
</tr>
<tr>
<td>Session resumption (caching)</td>
<td>Unknown</td>
</tr>
<tr>
<td>Session resumption (tickets)</td>
<td>Yes</td>
</tr>
<tr>
<td>OCSP stapling</td>
<td>No</td>
</tr>
<tr>
<td>Strict Transport Security (HSTS)</td>
<td>Unknown</td>
</tr>
<tr>
<td>Public Key Pinning (HPKP)</td>
<td>No</td>
</tr>
<tr>
<td>Long handshake intolerance</td>
<td>No</td>
</tr>
<tr>
<td>TLS extension intolerance</td>
<td>No</td>
</tr>
<tr>
<td>TLS version intolerance</td>
<td>TLS 2.98</td>
</tr>
<tr>
<td>SSL 2 handshake compatibility</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Miscellaneous

- **Test date**: Sun Dec 14 03:13:12 PST 2014
- **Test duration**: 118.40 seconds
- **HTTP status code**: Request failed
- **HTTP server signature**: Unknown
- **Server hostname**: svn.cacert.org

### Why is my certificate not trusted?

There are many reasons why a certificate may not be trusted. The exact problem is indicated on the report card in bright red. The problems fall into three categories:

1. **Invalid certificate**
   - A certificate is invalid if:
     - It is used before its activation date
     - It is used after its expiry date
     - Certificate hostnames don’t match the site hostname
     - It has been revoked

2. **Invalid configuration**
   - In some cases, the certificate chain does not contain all the necessary certificates to connect the web server certificate to one of the root certificates in our trust store. Less commonly, one of the certificates in the chain (other than the web server certificate) will have expired, and that invalidates the chain.

3. **Unknown Certificate Authority**
   - In order for trust to be established, we must have the root certificate of the signing Certificate Authority in our trust store. SSL Labs does not maintain our own trust store; instead we use the store maintained by Mozilla.
If we mark a web site as not trusted, that means that the average web user's browser will not trust it either. For certain special groups of users, such sites can still be secure. For example, if you can securely verify that a self-signed web site is operated by a person you trust, then you can trust the self-signed web site too. Or, if you work for an organisation that manages its own trust, and you have their own root certificate already embedded in your browser. Such special cases do not work for the general public, however, and this is what we indicate on our report card.

4. Interoperability issues

In some rare cases trust cannot be established because of interoperability issues between our code and the code or configuration running on the server. We manually review such cases, but if you encounter such an issue please feel free to contact us. Such problems are very difficult to troubleshoot and you may be able to provide us with information that might help us determine the root cause.

SSL Report v1.11.1