SSL Report: cats.cacert.org (213.154.225.243)

Summary

Overall Rating

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate</td>
<td></td>
</tr>
<tr>
<td>Protocol Support</td>
<td>95</td>
</tr>
<tr>
<td>Key Exchange</td>
<td>80</td>
</tr>
<tr>
<td>Cipher Strength</td>
<td>90</td>
</tr>
</tbody>
</table>

If trust issues are ignored: A

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

This server is not vulnerable to the POODLE attack because it doesn't support SSL 3. MORE INFO »

This server supports TLS_FALLBACK_SCSV to prevent protocol downgrade attacks.

Authentication

Server Key and Certificate #1

- Common names: cats.cacert.org
- Alternative names: cats.cacert.org
- Prefix handling: Not required for subdomains
- Valid from: Tue Apr 08 20:45:59 UTC 2014
- Valid until: Thu Apr 07 20:45:59 UTC 2016 (expires in 1 year and 5 months)
- Key: RSA 4096 bits
- Weak key (Debian): No
- Issuer: CA Cert Signing Authority
- Signature algorithm: SHA512withRSA
- Extended Validation: No
- Revocation information: CRL, OCSP
- Revocation status: Unchecked (only trusted certificates can be checked)
- Trusted: No NOT TRUSTED (Why?)

Additional Certificates (if supplied)

- Certificates provided: 2 (3474 bytes)
- Chain issues: Contains anchor

#2

- Subject: CA Cert Signing Authority
- SHA1: 135cac36f49cbbe693b1ab27bd0d808b4676ce06f33
- Valid until: Tue Mar 29 12:29:49 UTC 2033 (expires in 18 years and 5 months)
Additional Certificates (if supplied)

<table>
<thead>
<tr>
<th>Key</th>
<th>RSA 4096 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer</td>
<td>CA Cert Signing Authority</td>
</tr>
<tr>
<td>Signature algorithm</td>
<td>MD5withRSA</td>
</tr>
</tbody>
</table>

Certification Paths

Path #1: Not trusted (path does not chain to a trusted anchor)

1. Sent by server: cats.cacert.org
   SHA1: 2e0b086ef148b7669d65d18fe9b320c34b1425
   RSA 4096 bits / SHA512withRSA

2. Sent by server: CA Cert Signing Authority
   SHA1: 135cec36f49cb8e93b1ab270cd5d08846760ae8f33
   RSA 4096 bits / MD5withRSA
   Weak or insecure signature, but no impact on root certificates

Configuration

Protocols

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

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

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030): ECDH 256 bits (eq. 3072 bits RSA) FS 256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028): ECDH 256 bits (eq. 3072 bits RSA) FS 256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014): ECDH 256 bits (eq. 3072 bits RSA) FS 256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA256 (0xc027): ECDH 256 bits (eq. 3072 bits RSA) FS 256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc013): ECDH 256 bits (eq. 3072 bits RSA) FS 256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f): DH 1024 bits (p: 128, g: 1, Ys: 128) FS 256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x3d): DH 1024 bits (p: 128, g: 1, Ys: 128) FS 256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x35): DH 1024 bits (p: 128, g: 1, Ys: 128) FS 256
- TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x84): DH 1024 bits (p: 128, g: 1, Ys: 128) FS 256
- TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x45): DH 1024 bits (p: 128, g: 1, Ys: 128) FS 256
- TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d): 256
- TLS_RSA_WITH_AES_256_CBC_SHA256 (0x9c): 256
- TLS_RSA_WITH_AES_256_CBC_SHA (0x33): 256
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x45): 256
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x41): 256
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x35): 256
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x33): 256
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x31): 256
Qualys SSL Labs - Projects / SSL Server Test / cats.cace...

https://www.ssllabs.com/ssltest/analyze.html?d=cats.cace...

OpenSSL CCS vuln. (CVE-2014-0224)
No (Heartbleed (vulnerability)
Yes (Heartbeat (extension)
Yes (RC4
No (POODLE attack
Yes (BEAST attack
No (POODLE attack
No (SSL 3 not supported
No (POODLE attack
Yes (with most browsers)
ROBUST

<table>
<thead>
<tr>
<th>Protocol Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Secure Renegotiation</strong></td>
</tr>
<tr>
<td>Secure Client-Initiated Renegotiation</td>
</tr>
<tr>
<td>Insecure Client-Initiated Renegotiation</td>
</tr>
<tr>
<td><strong>BEAST attack</strong></td>
</tr>
<tr>
<td><strong>POODLE attack</strong></td>
</tr>
<tr>
<td><strong>Downgrade attack prevention</strong></td>
</tr>
<tr>
<td><strong>TLS compression</strong></td>
</tr>
<tr>
<td><strong>RC4</strong></td>
</tr>
<tr>
<td><strong>Heartbeat (extension)</strong></td>
</tr>
<tr>
<td><strong>Heartbleed (vulnerability)</strong></td>
</tr>
<tr>
<td><strong>OpenSSL CCS vuln. (CVE-2014-0224)</strong></td>
</tr>
<tr>
<td><strong>Forward Secrecy</strong></td>
</tr>
</tbody>
</table>
Protocol Details

- Next Protocol Negotiation: No
- Session resumption (caching): Unknown
- Session resumption (tickets): Yes
- OCSP stapling: No
- Strict Transport Security (HSTS): Unknown
- Long handshake intolerance: No
- TLS extension intolerance: No
- TLS version intolerance: TLS 2.98
- SSL 2 handshake compatibility: Yes

Miscellaneous

- Test date: Wed Oct 22 13:08:44 UTC 2014
- Test duration: 106.570 seconds
- HTTP status code: Request failed
- HTTP server signature: Unknown
- Server hostname: cats.cacert.org
- PCI compliant: No
- FIPS-ready: No

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
2. Invalid configuration
3. Unknown Certificate Authority

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 entire 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 its 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 web 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 that 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 v1.10.36