

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > cacert.org

SSL Report: cacert.org (213.154.225.245)

Assessed on: Sat Oct 18 09:31:03 UTC 2014 | [Clear cache](#)

[Scan Another »](#)

Summary

Overall Rating

T

If trust issues are ignored: A

Certificate	0
Protocol Support	90
Key Exchange	80
Cipher Strength	90

0 20 40 60 80 100

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 is vulnerable to the POODLE attack. If possible, disable SSL 3 to mitigate. Grade capped to C. [MORE INFO »](#)

This server supports TLS_FALLBACK_SCSV to prevent protocol downgrade attacks.

Authentication



Server Key and Certificate #1

Common names	www.cacert.org
Alternative names	www.cacert.org secure.cacert.org wwwmail.cacert.org cacert.org www.cacert.net cacert.net www.cacert.com cacert.com
Prefix handling	Both (with and without WWW)
Valid from	Mon Apr 28 20:57:55 UTC 2014
Valid until	Wed Apr 27 20:57:55 UTC 2016 (expires in 1 year and 6 months)
Key	RSA 2048 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	1 (1678 bytes)
Chain issues	Incomplete



Certification Paths

No trust paths available

Certification Paths

Issuer unknown, or intermediate certificate(s) missing.

Configuration



Protocols

TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3 INSECURE	Yes
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_128_GCM_SHA256 (0xc02f)	ECDH 256 bits (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)	ECDH 256 bits (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH 256 bits (eq. 3072 bits RSA) FS	128
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 (0x6b)	DH 1024 bits (p: 128, g: 1, Ys: 128) FS	256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39)	DH 1024 bits (p: 128, g: 1, Ys: 128) FS	256
TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x88)	DH 1024 bits (p: 128, g: 1, Ys: 128) FS	256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e)	DH 1024 bits (p: 128, g: 1, Ys: 128) FS	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (0x67)	DH 1024 bits (p: 128, g: 1, Ys: 128) FS	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33)	DH 1024 bits (p: 128, g: 1, Ys: 128) FS	128
TLS_DHE_RSA_WITH_SEED_CBC_SHA (0x9a)	DH 1024 bits (p: 128, g: 1, Ys: 128) FS	128
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (0x45)	DH 1024 bits (p: 128, g: 1, Ys: 128) FS	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_256_CBC_SHA (0x35)		256
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x84)		256
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c)		128
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)		128
TLS_RSA_WITH_SEED_CBC_SHA (0x96)		128
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x41)		128
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012)	ECDH 256 bits (eq. 3072 bits RSA) FS	112
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x16)	DH 1024 bits (p: 128, g: 1, Ys: 128) FS	112
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)		112



Handshake Simulation

Android 2.3.7 <small>No SNI²</small>	TLS 1.0	TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) FS	128
Android 4.0.4	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
Android 4.1.1	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
Android 4.2.2	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
Android 4.3	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
Android 4.4.2	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) FS	256
BingBot Dec 2013 <small>No SNI²</small>	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
BingPreview Jun 2014	TLS 1.0	TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) FS	256

Handshake Simulation

Chrome 37 / OS X R	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
Firefox 24.2.0 ESR / Win 7	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
Firefox 32 / OS X R	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
Googlebot Jun 2014	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
IE 6 / XP No FS ¹ No SNI ²	SSL 3	TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	No FS	112
IE 7 / Vista	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
IE 8 / XP No FS ¹ No SNI ²	TLS 1.0	TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)	No FS	112
IE 8-10 / Win 7 R	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
IE 11 / Win 7 R	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
IE 11 / Win 8.1 R	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)	FS	256
IE Mobile 10 / Win Phone 8.0	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
IE Mobile 11 / Win Phone 8.1	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
Java 6u45 No SNI ²	TLS 1.0	TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33)	FS	128
Java 7u25	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	FS	128
Java 8b132	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	FS	128
OpenSSL 0.9.8y	TLS 1.0	TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39)	FS	256
OpenSSL 1.0.1h	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	FS	256
Safari 5.1.9 / OS X 10.6.8	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
Safari 6 / iOS 6.0.1 R	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)	FS	256
Safari 7 / iOS 7.1 R	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)	FS	256
Safari 8 / iOS 8.0 Beta R	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)	FS	256
Safari 6.0.4 / OS X 10.8.4 R	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	FS	256
Safari 7 / OS X 10.9 R	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)	FS	256
Yahoo Slurp Jun 2014 No SNI ²	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	FS	256
YandexBot Sep 2014	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	FS	256

(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.

(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

Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Not mitigated server-side (more info) SSL 3: 0xc014, TLS 1.0: 0xc014
POODLE attack	Vulnerable INSECURE (more info)
Downgrade attack prevention	Yes, TLS_FALLBACK_SCSV supported (more info)
TLS compression	No
RC4	No
Heartbeat (extension)	Yes
Heartbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
Forward Secrecy	Yes (with most browsers) ROBUST (more info)
Next Protocol Negotiation	No
Session resumption (caching)	Yes
Session resumption (tickets)	Yes
OCSP stapling	No
Strict Transport Security (HSTS)	No
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	TLS 2.98
SSL 2 handshake compatibility	Yes



Miscellaneous

Test date	Sat Oct 18 09:28:44 UTC 2014
Test duration	139.548 seconds
HTTP status code	200
HTTP server signature	Apache/2.2.22 (Debian)
Server hostname	www.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