SSL Report: wolke.4d53.de (217.235.241.249)

Assessed on: Sun, 29 May 2016 18:53:33 UTC | Clear cache

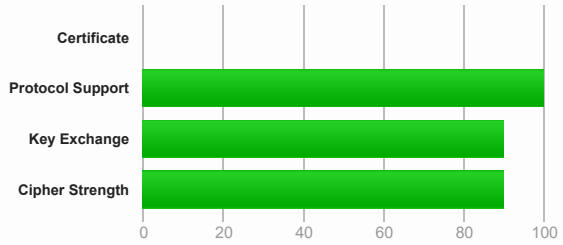
Scan Another »

Summary

Overall Rating

T

If trust issues are ignored: A



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.

HTTP Strict Transport Security (HSTS) with long duration deployed on this server. MORE INFO »

Authentication

Server Key and Certificate #1

Subject	wolke.4d53.de Fingerprint SHA1: d0f2a11928316e3d0d60cb7be9d40666e4e9688a Pin SHA256: e4iF3NQLfVWOmRjMOh2ceryEw6iPX8OGK9FqXE2lzVc=
Common names	wolke.4d53.de
Alternative names	wolke.4d53.de
Valid from	Sun, 28 Feb 2016 15:11:00 UTC
Valid until	Sat, 28 May 2016 15:11:00 UTC (expired 1 day, 3 hours ago) **EXPIRED**
Key	RSA 4096 bits (e 65537)
Weak key (Debian)	No
Issuer	Let's Encrypt Authority X1 AIA: http://cert.int-x1.letsencrypt.org/
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	No
Revocation information	OCSP OCSP: http://ocsp.int-x1.letsencrypt.org/
Revocation status	Unchecked (only trusted certificates can be checked)
Trusted	No **NOT TRUSTED** (Why?)

Additional Certificates (if supplied)

Certificates provided	2 (2734 bytes)
Chain issues	None

#2

Subject	Let's Encrypt Authority X1 Fingerprint SHA1: 3eae91937ec85d74483ff4b77b07b43e2af36bf4 Pin SHA256: YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg=
Valid until	Mon, 19 Oct 2020 22:33:36 UTC (expires in 4 years and 4 months)
Key	RSA 2048 bits (e 65537)

Issuer	DST Root CA X3
Signature algorithm	SHA256withRSA

Certification Paths

Path #1: Not trusted (timestamp check failed)

1	Sent by server	wolke.4d53.de Fingerprint SHA1: d0f2a11928316e3d0d60cb7be9d40666e4e9688a Pin SHA256: e4iF3NQLfVWOmRjMOh2ceryEw6iPX8OGK9FqXE2lzVc= RSA 4096 bits (e 65537) / SHA256withRSA Valid until: Sat, 28 May 2016 15:11:00 UTC **EXPIRED**
2	Sent by server	Let's Encrypt Authority X1 Fingerprint SHA1: 3eae91937ec85d74483ff4b77b07b43e2af36bf4 Pin SHA256: YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg= RSA 2048 bits (e 65537) / SHA256withRSA
3	In trust store	DST Root CA X3 Self-signed Fingerprint SHA1: dac9024f54d8f6df94935fb1732638ca6ad77c13 Pin SHA256: Vjs8r4z+80wjNcr1YKepWQboSIRi63WsWXhIMN+eWys= RSA 2048 bits (e 65537) / SHA1withRSA Weak or insecure signature, but no impact on root certificate

Configuration

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_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH secp256r1 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH secp256r1 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH secp256r1 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH secp256r1 (eq. 3072 bits RSA) FS	128

Handshake Simulation

Android 2.3.7 No SNI [2]	Server sent fatal alert: protocol_version			
Android 4.0.4	Server sent fatal alert: protocol_version			
Android 4.1.1	Server sent fatal alert: protocol_version			
Android 4.2.2	Server sent fatal alert: protocol_version			
Android 4.3	Server sent fatal alert: protocol_version			
Android 4.4.2	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS	
Android 5.0.0	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS	
Baidu Jan 2015	Server sent fatal alert: protocol_version			
BingPreview Jan 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS	
Chrome 48 / OS X R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS	
Firefox 31.3.0 ESR / Win 7	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS	
Firefox 42 / OS X R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS	
Firefox 44 / OS X R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS	
Googlebot Feb 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS	
IE 6 / XP No FS [1] No SNI [2]	Server closed connection			
IE 7 / Vista	Server sent fatal alert: protocol_version			
IE 8 / XP No FS [1] No SNI [2]	Server sent fatal alert: protocol_version			
IE 8-10 / Win 7 R	Server sent fatal alert: protocol_version			
IE 11 / Win 7 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS	

IE 11 / Win 8.1 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1	FS
IE 10 / Win Phone 8.0	Server sent fatal alert: protocol_version				
IE 11 / Win Phone 8.1 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	ECDH secp256r1	FS
IE 11 / Win Phone 8.1 Update R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1	FS
IE 11 / Win 10 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Edge 13 / Win 10 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Edge 13 / Win Phone 10 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Java 6u45 No SNI [2]	Server closed connection				
Java 7u25	Server sent fatal alert: protocol_version				
Java 8u31	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp256r1	FS
OpenSSL 0.9.8y	Server sent fatal alert: protocol_version				
OpenSSL 1.0.1l R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
OpenSSL 1.0.2e R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Safari 5.1.9 / OS X 10.6.8	Server sent fatal alert: protocol_version				
Safari 6 / iOS 6.0.1 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1	FS
Safari 6.0.4 / OS X 10.8.4 R	Server sent fatal alert: protocol_version				
Safari 7 / iOS 7.1 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1	FS
Safari 7 / OS X 10.9 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1	FS
Safari 8 / iOS 8.4 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1	FS
Safari 8 / OS X 10.10 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1	FS
Safari 9 / iOS 9 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Safari 9 / OS X 10.11 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Apple ATS 9 / iOS 9 R	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Yahoo Slurp Jan 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
YandexBot Jan 2015	RSA 4096 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	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	No
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)	Yes
Heartbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
Forward Secrecy	**Yes (with most browsers) ROBUST** (more info)
ALPN	No
NPN	No
Session resumption (caching)	Yes
Session resumption (tickets)	Yes
OCSP stapling	Yes (not verified because the certificate is not trusted)
Strict Transport Security (HSTS)	**Yes** max-age=15768000
HSTS Preloading	**Not in: Chrome Edge Firefox IE Tor**
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

Miscellaneous

Test date	Sun, 29 May 2016 18:51:41 UTC
Test duration	97.412 seconds
HTTP status code	200
HTTP server signature	Apache/2.4.10 (Raspbian)
Server hostname	pD9EBF1F9.dip0.t-ipconnect.de

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 v1.22.37