

OAuch

Analyzing the Security Best Practices in the OAuth 2.0 Ecosystem

Pieter Philippaerts

“Once you have implemented OAuth2, how do you know you have implemented it securely?”

SSL Server Test: www.google.com

https://www.ssllabs.com/ssltest/analyze.html?d=www.google.com&s=172.217.6.68&hideResul...

Qualys. SSL Labs

Home Projects Qualys Free Trial Contact

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > [www.google.com](#) > 172.217.6.68

SSL Report: [www.google.com](#) (172.217.6.68)

Assessed on: Mon, 20 Jul 2020 06:35:49 UTC | [HIDDEN](#) | [Clear cache](#) [Scan Another »](#)

Summary

Overall Rating

B

Category	Score
Certificate	100
Protocol Support	75
Key Exchange	100
Cipher Strength	100

Visit our [documentation page](#) for more information, configuration guides, and books. Known issues are documented [here](#).

This server supports TLS 1.0 and TLS 1.1. Grade capped to B. [MORE INFO »](#)

This server supports TLS 1.3.

Static Public Key Pinning observed for this server.

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

DNS Certification Authority Authorization (CAA) Policy found for this domain. [MORE INFO »](#)

Scan results for www.facebook.com

https://securityheaders.com/?q=www.facebook.com&follo...

Security Headers

Sponsored by [Report URI](#)

Home About Donate

Scan your site now

☐ Hide results ☒ Follow redirects

Security Report Summary

A

Site:	https://www.facebook.com/
IP Address:	2a03:2880:f131:83:face:b00c:0:25de
Report Time:	20 Jul 2020 10:51:59 UTC
Headers:	<div> <div>✓ Strict-Transport-Security</div> <div>✓ Content-Security-Policy</div> <div>✓ X-Content-Type-Options</div> <div>✓ X-Frame-Options</div> <div>✗ Referrer-Policy</div> <div>✗ Feature-Policy</div> </div>
Warning:	Grade capped at A, please see warnings below.

Supported By

Site results - OAuch

https://oauch.io/Dashboard/Results/5a74464d-73...

Dashboard Tests overview FAQ About OAuch

Site results

The site was successfully tested on June 26, 2020 at 14:45 The following issues were discovered:

- Mandatory test cases failed: **5**
- Recommended test cases failed: **3**
- Optional test cases failed: **4**

[Run a new test](#)

[What's this?](#)

Tests overview

Failed tests [All tests](#) [Full log](#) [Reporting](#) [History](#)

Http Properties

- Are deprecated TLS versions supported on the token server: **YES** [[recommended](#), [more info](#)]
The token server allows connections with deprecated versions of the TLS protocol.
- Authorization page has a content security policy: **NO** [[recommended](#), [more info](#)]
In order to prevent clickjacking, authorization servers should use Content Security Policy (CSP) level 2 or greater.

Authorization Code Flow

- Is the redirect URI checked when exchanging a code: **NO** [[mandatory](#), [more info](#)]



Site dashboard - OAuch

https://oauch.io/Dashboard

OAuth 2.0 Sites

Shared resources

- Client certificates

Site tools

- Add new site
- Export data
- Import settings

All available sites (1)

- FitBit

Dashboard Tests overview FAQ About OAuch

Site dashboard

This is an overview of the sites you have added to OAuch.

?

FitBit

Last test: *untested*

Mandatory tests failed	?
Recommended tests failed	?
Optional tests failed	?
See detailed results	
Change site settings	

Copyright © 2020 [DistriNet Research Group](#). All rights reserved.

Version 1.0

https://oauch.io/Dashboard/Settings/c5c1a1e0-d3cd-49d9-bd9b-d5016def093c

Running test - OAuch

Dashboard Tests overview FAQ About OAuch

Test progress

Stalled test

Click the 'stalled test' button if the tests is stalled (i.e. if the pop-up window shows an error page and the testing process seems to be stuck).

If you wish to cancel the entire test run, you can click the button below. There will be no partial results available.

Now running test: AuthorizationCodeFlowSupported Working...

Latest log output

- Connected to OAuch command server
- Pop-up window ready, connecting to OAuch command server
- Waiting for pop-up to open

Detected features

- Authorization Code Flow
- Implicit Flow
- Client Credentials Flow
- Hybrid Flow
- Device Flow
- Password Flow
- Access Tokens
- Refresh Tokens
- JWT Tokens
- ID Tokens
- Test URI

Abort all tests

Copyright © 2020 [DistriNet Research Group](#). All rights reserved. Version 1.0

Test Overview

App Authorization

https://www.fitbit.com/oauth2/authorize?response_t...

fitbit

OAuch test by DistriNet would like the ability to access the following data in your Fitbit account.

- ☒ Allow All
- ☒ profile

If you allow only some of this data, OAuch test may not function as intended. Learn more about these permissions [here](#).

Deny **Allow**

The data you share with OAuch test will be governed by DistriNet's [Privacy Policy](#) and [Terms of Service](#). You can revoke this consent at any time in your [Fitbit account settings](#).

Signed in as osv2020@pieterp.be [Not you?](#)

Authorization and Callback Window

Site results - OAuch

https://oauch.io/Dashboard/Results/10728fbc-1c39-44a8-be5b-74df62976bf7


DashboardTests overviewFAQAbout OAuch

Site results

The site was successfully tested on July 20, 2020 at 12:41. The following issues were discovered:

- Mandatory test cases failed: **3**
- Recommended test cases failed: **2**
- Optional test cases failed: **0**

[Run a new test](#)



[What's this?](#)

Tests overview

Failed tests

All testsFull logReportingHistory

Http Properties

- Does the token server allow multiple instances of the same parameter: **YES** [mandatory, [more info](#)]
Request and response parameters must not be included more than once.
- Does the authorization server allow multiple instances of the same parameter: **YES** [mandatory, [more info](#)]
Request and response parameters must not be included more than once.
- Are deprecated TLS versions supported on the token server: **YES** [recommended, [more info](#)]
The token server allows connections with deprecated versions of the TLS protocol.
- Are client certificates used: **NO** [recommended, [more info](#)]
Authorization and resource servers should use mechanisms for sender-constrained access tokens to prevent token replay. The use of Mutual TLS for OAuth 2.0 is recommended.

Access and Refresh Tokens

- Refresh tokens are invalidated when used multiple times: **NO** [mandatory, [more info](#)]
The authorization server must revoke the active refresh token if the previous refresh token is used multiple times.



Analyzing the OAuth 2.0 Ecosystem

What we did

- › We tested 100+ OAuth implementations
 - ›› 94 deployed and publicly available services
 - ›› 17 OIDC providers, 77 OAuth 2.0 API providers
- › We drew statistics over the sites and over the individual countermeasures

Supported Flows

API Providers

- › 94% support Authorization Code flow
- › 44% support Implicit flow
- › 30% support Client Credentials flow
- › 3% support Password flow

OIDC Providers

- › 100% support Authorization Code flow
- › 35% support Client Credentials flow
- › 24% support Implicit flow
- › 24% support Hybrid flow
- › 6% support Device flow

Failure Rates

API Providers

- › 38.0% average failure rate ($\pm 6.9\%$)
 - › 31% *must* failures
 - › 40% *should* failures
 - › 85% *may* failures

OIDC Providers

- › 28.0% average failure rate ($\pm 7.0\%$)
 - › 22% *must* failures

Client Authentication

Client Type

- › 1% support only public clients
- › 1% support confidential clients (crypto key)
- › 98% support confidential client (password)
 - › However, 12% do not use/require the password

Client Authentication

Authorization servers must support the *Authorization* header

- › Support is mandatory, but only 69% support it
- › Other sites use form POST

Proof Key for Code Exchange

Authorization servers must support PKCE

- › Only 12% of API providers support PKCE
 - ›› Mostly ignored
 - ›› Sometimes disallowed

Proof Key for Code Exchange

For the API providers supporting PKCE:

- › None required PKCE
- › 33% supported *plain* PKCE
- › 44% allowed very short verifiers
- › 56% were vulnerable to PKCE sidestep attack¹

¹ <https://mailarchive.ietf.org/arch/msg/oauth/qrLAf3nWRt8HAFkO49qGrPRuelo/>

Redirect URI Matching

Callback URIs must be precisely matched

- › Only 48% of sites do this

Token endpoint must compare the callback URI with the one received in the authorization request

- › Only 43% of sites do this

Authorization Codes

Authorization codes must only be used once

- › 76% disallow code exchange
- › 12% disallow code exchange and revoke previously granted access tokens
- › 12% allow multiple code exchanges

Access Tokens

- › Are mostly opaque (only 15% JWT)
- › Are long (85% over 128 bits of entropy)
- › Can often be used as URI query parameter (44%)

Refresh Tokens

- › Are used by 66% of sites
- › When *refresh token rotation* is used, refresh tokens must be single use
 - ›› Of these sites, only 34% prohibited exchanging the same refresh token multiple times
 - ›› Active refresh tokens were never revoked

Access Tokens and Refresh Tokens

If refresh tokens are used, access token lifetime should be short

- › < 1 hour: 36%
- › < 8 hours and > 1 hour: 27%
- › < 24 hours and > 8 hours: 10%
- › > 24 hours: 27%

Some of the other results

- › 26% allow authorization pages to be framed (*mandatory*)
- › 29% allow the caching of sensitive values (*mandatory*)
- › 70% do not suppress the referrer header (*optional*)
- › 94% do not support *form post response mode* (*optional*)
- › 85% allow parameters to be included multiple times (*mandatory*)
- › 60% of OIDC servers do not support POST authorization requests (*mandatory*)
- › 50% of OIDC servers did not require a *nonce* for the implicit flow (*mandatory*)
- › 83% do not support token revocation (*optional*)
 - » Of those that did, 42% accept revoked refresh tokens (*mandatory*)
- › ...

Work in progress...

- › These results are a work-in-progress
 - ›› The full analysis will hopefully be published soon
- › The OAuch tool will be available at <https://oauch.io/> (early September)
 - ›› Offline download by the end of the year

Conclusions

- › Having a formal verification of the OAuth2 protocol is great (and necessary)!
 - › ... but we also need tools to verify practical implementations
- › A lot of sites can benefit from implementing missing countermeasures



Thank you!

<https://distrinet.cs.kuleuven.be/>

Pieter.Philippaerts@kuleuven.be