OAuth 2.0 Device Posture Signals
draft-wdenniss-oauth-device-posture-00

Abstract

Enterprise and security focused OAuth providers typically want additional signals to confirm user presence when users return to previously authorized apps. Rather than requiring a full reauthentication, or require enrollment in a mobile device management solution, some authorization servers may be willing to accept device posture signals from the app, like the fact that device has a lock screen, as confirmation of user presence. This document details how OAuth native app clients can communicate device posture signals to OAuth providers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 12, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ........................................... 2
2. Notational Conventions ................................. 3
3. Device Posture Signal Dictionary ........................ 3
4. Authorization Request Device Posture Hint .............. 3
5. Token Endpoint Device Posture Enforcement ............ 4
6. Security Considerations ................................ 5
   6.1. Device Posture Scope ............................... 5
   6.2. Spoofed Devices .................................. 5
   6.3. App Trustworthiness ................................ 5
7. IANA Considerations .................................... 5
   7.1. OAuth Parameters Registration ........................ 5
      7.1.1. Registry Contents .............................. 6
   7.2. OAuth Extensions Error Registration ................ 6
      7.2.1. Registry Contents .............................. 6
   7.3. Device Posture Keys Registry ........................ 6
      7.3.1. Registration Template ........................... 7
      7.3.2. Initial Registry Contents ....................... 7
8. References .............................................. 8
   8.1. Normative References ................................ 8
   8.2. Informative References .............................. 8
Appendix A. Acknowledgements ............................ 8
Author’s Address .......................................... 9

1. Introduction

Users who follow strong security practices on their devices — such as configuring screen locks, and not enabling admin privileges (commonly known as "rooting" or "jailbreaking") — shouldn’t need to reauthenticate frequently to the individual apps on their device.

This specification details how apps can send device posture signals to the OAuth Token Endpoint, enabling it to enforce device policy compliance, and avoid the need for reauthentication in some cases.

It is designed to provide a mechanism for honest apps to communicate device posture. By itself it doesn’t protect against malicious users, dishonest apps, or compromised devices, but the signal format described could carry signals that do.
2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in Key words for use in RFCs to Indicate Requirement Levels [RFC2119]. If these words are used without being spelled in uppercase then they are to be interpreted with their normal natural language meanings.

3. Device Posture Signal Dictionary

The device posture is a dictionary of signals asserted by the app about the device. The structure is send as an added parameter in several places during the OAuth flow, as documented in the subsequent sections.

All device posture keys are OPTIONAL and MUST only be set when the attribute can be obtained by the app. The standard attribute keys are as follows:

- **screen_lock**
  - Boolean. True if the user has a screen lock, such as a pin, pattern biometric, etc.

- **root_privileges**
  - Boolean. True if user apps can access root device privileges. For mobile operating systems, known as "jailbreaking" on iOS and "rooting" on Android.

- **device_attestation**
  - Dictionary. An attestation from the operating system, containing a signed-statement about the device and/or the app. The format is a dictionary, the specifics of which depends on the operating system.

An example device posture dictionary:

```json
{
   "screen_lock": true,
   "root_privileges": false
}
```

4. Authorization Request Device Posture Hint

Clients MAY send the device posture signal dictionary to the authorization server in the authorization request. These signals, except for those that are signed and bound to the device are susceptible to client-side modification by end-users. While untrusted, such signals can still be used as hints by the
authorization server to present a better user experience, like informing the user they need a lock screen.

Error encountered during authorization can be displayed to the user in the browser making this a more user friendly way to instruct the user on how to move their device into conformance. The token endpoint (on which errors are less user-friendly as there’s no user agent), can then enforce the restrictions per Section 5.

The following parameters are added to the OAuth 2.0 Authorization Request:

device_posture_hint
   JSON String. URL-encoded JSON dictionary, contains the Device Posture Signals defined in Section 3.

5. Token Endpoint Device Posture Enforcement

Clients that follow this specification MUST send the device posture signals on every request to the token endpoint.

Token Endpoints SHOULD verify that the posture conforms to their requirements and act accordingly.

The following parameters are added to all requests to the Token Endpoint:

device_posture
   JSON String. URL-encoded JSON dictionary, contains the Device Posture Signals defined in Section 3.

The app MUST obtain fresh device posture information before every request to the Token Endpoint, and MUST NOT include stale information (rather, it should drop any signals it cannot freshly obtain).

For token refresh requests, where the device posture has been previously communicated, if an attribute is missing, the Token Endpoint may choose to use the previous value, based on it’s own policy and freshness requirements.

If the policy does not meet requirements, the Token Endpoint SHOULD return the following error code:

device_posture_invalid
   Error indicating that the device posture does not meet requirements. The error description SHOULD contain details on why this is is the case.
6. Security Considerations

6.1. Device Posture Scope

This specification is designed to help authorization servers enforce security policy (like requiring a lock screen) on end-users. The intent is to enforce restrictions on honest users, to force them to follow security practices set out by the authorization server. By itself, it offers no protection against malicious users, dishonest apps, or compromised devices.

Combined with other technologies like device-based attestations and token binding may enable such protection, and this specification could be used to transmit secure signals, but that topic is out of scope for this specification.

6.2. Spoofed Devices

It is possible to at a device level completely spoof the device posture. Even statements signed by the operating system are vulnerable to spoofing, as it’s possible a statement from the real device can be replayed on a spoofed device, unless such statements include a binding to the device itself. Per Section 6.1, this topic is out of scope for this specification.

6.3. App Trustworthiness

This specification is designed to allow trusted apps to report device posture to the authorization server to help the server enforce security policy on end-users. It does not by itself force apps to be honest, or genuine. Genuine apps (i.e. apps not lying about their client ID) might be dishonest about the device posture, and apps that are normally honest, could be spoofed, unless anti-spoofing countermeasures that are out of scope of this specification are employed.

7. IANA Considerations

7.1. OAuth Parameters Registration

This specification registers the following value in the IANA "OAuth Parameters" registry [IANA.OAuth.Parameters] established by [RFC6749].
7.1.1. Registry Contents

- Parameter name: device_posture_hint
  - Parameter usage location: authorization request
  - Change controller: IESG
  - Specification document(s): Section 4 of [[this specification]]

- Parameter name: device_posture
  - Parameter usage location: token request
  - Change controller: IESG
  - Specification document(s): Section 5 of [[this specification]]

7.2. OAuth Extensions Error Registration

This specification registers the following error in the IANA "OAuth Extensions Error Registry" [IANA.OAuth.Parameters] established by [RFC6749].

7.2.1. Registry Contents

- Error name: device_posture_invalid
  - Error usage location: authorization response, token error response
  - Related protocol extension: resource parameter
  - Change controller: IESG
  - Specification document(s): Section 5 of [[this specification]]

7.3. Device Posture Keys Registry

This specification establishes the IANA "Device Posture Keys" registry for Device Posture Dictionary keys. The registry records the Device Posture key and a reference to the specification that defines it. This specification registers the Device Posture keys defined in Section 3.

Keys are registered on an Expert Review [RFC5226] basis after a three-week review period on the oauth-reg-review@ietf.org mailing list, on the advice of one or more Designated Experts.

Registration requests sent to the mailing list for review should use an appropriate subject (e.g., "Request to register Device Posture Key: screen_lock").

Within the review period, the Designated Experts will either approve or deny the registration request, communicating this decision to the review list and IANA. Denials should include an explanation and, if applicable, suggestions as to how to make the request successful.

Registration requests that are undetermined for a period longer than
21 days can be brought to the IESG’s attention (using the iesg@ietf.org mailing list) for resolution.

Criteria that should be applied by the Designated Experts includes determining whether the proposed registration duplicates existing functionality, whether it is likely to be of general applicability or whether it is useful only for a single application, whether the value is actually being used, and whether the registration description is clear.

IANA must only accept registry updates from the Designated Experts and should direct all requests for registration to the review mailing list.

It is suggested that the same Designated Experts evaluate these registration requests as those who evaluate registration requests for the IANA "OAuth Parameters" registry [IANA.OAuth.Parameters].

7.3.1. Registration Template

Device Posture Signal Key:
The key name requested (e.g., "screen_lock"). Names may not match other registered names in a case-insensitive manner unless the Designated Experts state that there is a compelling reason to allow an exception.

Device Posture Signal Key Description:
Brief description of the device posture signal (e.g., "Screen lock active").

Change Controller:
For Standards Track RFCs, state "IESG". For others, give the name of the responsible party. Other details (e.g., postal address, email address, home page URI) may also be included.

Specification Document(s):
Reference to the document or documents that specify the parameter, preferably including URIs that can be used to retrieve copies of the documents. An indication of the relevant sections may also be included but is not required.

7.3.2. Initial Registry Contents

- Device Posture Signal Key: "screen_lock"
- Device Posture Signal Key Description: Boolean. ‘true’ when the device has a screen lock enabled.
- Change Controller: IESG
- Specification Document(s): Section 3 of [[ this specification ]]
- Device Posture Signal Key: "root_privileges"
Device Posture Signal Key Description: Boolean. True if user apps can access root device privileges.
Change Controller: IESG
Specification Document(s): Section 3 of [[ this specification ]]

Device Posture Signal Key: "device_attestation"
Device Posture Signal Key Description: Dictionary. An attestation from the operating system, containing a signed-statement about the device and/or the app.
Change Controller: IESG
Specification Document(s): Section 3 of [[ this specification ]]

8. References

8.1. Normative References

[IANA.OAuth.Parameters]


8.2. Informative References


Appendix A. Acknowledgements

The following individuals contributed ideas, feedback, and wording that shaped and formed the final specification:

Eric Sachs, John Bradley, and Andy Zmolek.