A specification for IdP hinting

Publication Date 2019-03-11
Authors: AARC Consortium Partners;AppInt members;Marcus Hardt (ed.)

Document Code: AARC-G049
DOI:

Grant Agreement No.: 730941
Work Package: JRA1

The research leading to these results has received funding from the European Community's Horizon2020 Programme under Grant Agreement No. 730941 (AARC2).

Abstract
This document defines a generic browser-based protocol for conveying - to services - hints about the IdPs or IdP-SP-proxies that should be used for authenticating the principal. This protocol, colloquially referred to as Identity Provider (IdP) hinting, can greatly simplify the discovery process for the end-user, by either narrowing down the number of possible/IdPs to choose from or by making the actual selection process fully transparent.

Table of Contents

Table of Contents 2
1. Introduction 3
1.1. Conventions 3
2. Context 4
3. Specification 5
References 6
Appendix A 7
A.1 Example Scenarios 7
A.2 Examples for idphints 7

1. Introduction

Authentication to a service in a multi-IdP environment requires that the service redirect the incoming user to their home identity provider (IdP). Currently this is often accomplished by discovery services (often also called “where are you from” or WAYF services), where the user chooses their home IdP.

The rise of the proxy concept introduces new IdPs that may be chosen by the service instead of sending the user directly to a home IdP. Often, users have to choose between a list of IdP-SP-Proxies. This makes it increasingly difficult for users to understand which IdP is the best choice for authentication.

In this document we focus on enabling Service Providers / OIDC-Relying-Parties / WAYF Services to obtain a hint about the IdP to which the user should be sent for authentication. We define a portable and technology-agnostic way to allow services to receive hints about which IdP to use.

This mechanism can greatly simplify the discovery process for the end-user, by either narrowing down the number of possible IdPs to choose from or by making the actual selection process fully transparent.

Furthermore, the described concept includes the possibility of chaining, so that hints can be nested. This allows creating URLs that point to an SP, with a hint trail that leads via an IdP-SP-Proxy to a given home IdP.

Finally, we want to stress that this hinting process takes place before any authentication has happened. The flow of information is therefore independent of the underlying protocol used. The hints themselves, however, may contain protocol specific information. We also stress that it is only a hint. Whether the proxy or service actually honours the hint depends on the list of locally configured trusted IdPs.

1.1. Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Context

The IdP hinting mechanism described in this document is based on the following assumptions:

- **Web**: We focus on web, but do not a priori exclude non-web scenarios.
- **Context**: IdPs may be home-IdPs or IdP-SP-Proxies.
- **Trust**: Services trust IdPs based on a trust relation that is out of scope of this document. Therefore, we use the term “hinting”, to emphasize that it is certainly possible for the SP or proxy to decide not to follow a hint.
- **AARC Blueprint Architectures (BPA):** Our definition supports, but by no means requires, that services are operated in a BPA context. I.e. in addition to the previous point, end services in a BPA context would only accept hints towards supported proxies.
- The service that obtains a hint can either process it itself, or decide to pass the hint to its WAYF for filtering the list of potential IdPs. Details for this are out of scope of this document.
- Services that want to send users to a specific home-IdP for reauthentication, will need to keep track of the necessary identifier to do so.

Multiple technologies can benefit from IdP hinting. IdP hinting should at least work for SAML2 and OAuth2/OIDC based services.

3. Specification

- The identifier of the hinted IdP MUST be passed through the “idphint” GET parameter.
- POST parameters MAY be supported in addition to the GET parameter.
- The service MUST interpret the parameter “idphint” of a request as the URL-encoded identifier of the IdP to which the creator of the url intends to send the user for authentication.
- Implementations MUST also encode slashes (‘/’).
- The hinted identifiers MUST be well-defined URIs [RFC3986]:
    - For SAML it MUST be the EntityID
    - For OAuth2.0 and OIDC it MUST be the issuer
- Multiple IdPs MAY be provided, which MUST be encoded as a comma separated list of URL-encoded identifiers.
- Case sensitivity MUST follow the underlying specification of the URL-decoded identifier.

References

[RFC2119] Key words for use in RFCs to indicate Requirement levels
https://tools.ietf.org/html/rfc2119

[RFC3986] Uniform Resource Identifier (URI): Generic Syntax
https://tools.ietf.org/html/rfc3986

Appendix A

A.1 Example Scenarios

There are several use cases motivating this recommendation. The purpose is to make the IdP selection process transparent to the end-user and to streamline and simplify the inherently error prone selection process.

1. Initiated before the end service
   - eInfra service serving multiple communities connected via a single eInfra proxy
     I. The community provides their users with links for eInfra services and include the idphint to the community AAI in the URLs.
     II. Both the service and the eInfra proxy need to consume the IdP hint.
   - Generic service connected to several proxies, home IdPs, etc.
     I. Same as above
     II. Only the generic service needs to consume the IdP hint.
2. Initiated by the end service
   - The service adds the idphint parameter when initiating the authentication request to the proxy.
   - This can also be used to request re-authentication at a specific home-Idp
   - Only the proxy needs to consume the IdP hint.

A.2 Examples for idphints

The examples shown contain the actual links to be used for hinting. For clarity, we also provide the pseudocode that generated the link.

Note that the mechanism is independent of the protocol of the hinted IdP. This is highlighted in the second example, where we first provide a hint pointing to an OAuth2 endpoint at the proxy and then to a SAML endpoint of the home-IdP identified by its entityID.

- Simple example:
  https://example.service.edu/?idphint=https%3A%2F%2Fhome-idp.org%2Fidp%2Fsaml
  <=>
  https://example.service.edu/?idphint=*urlencode*(**https://home-idp.org/idp/saml**)
  Service https://example.service.edu receives a hint to authenticate the user at https://home-idp.org/idp/saml

- Chained example:
  https://example.service.edu/?idphint=https%3A%2F%2Fidp-sp-proxy.org%2Foauth2%3Fidphint%3Dhttps%253A%252F%252Fhome-idp.org%252Fidp%saml
  <=>
  https://example.service.edu/?idphint=*urlencode*(**https://id**

p-sp-proxy.org/oauth2?idphint=*urlencode(***https://home-idp.org/idp/saml***))**
Service https://example.service.edu receives a hint to authenticate the user at https://idp-sp-proxy.org/oauth2?idphint=https%3A%2F%2Fhome-idp.org%2Fidp%2Fsaml, thereby passing the URL to the proxy with an encoded idphint parameter that points to the home-IdP.
The URL-encoded parameter in turn will hint the proxy to redirect the user to authenticate at https://home-idp.org/idp/saml

- Multiple IdP example:

  https://example.service.edu/?idphint=https%3A%2F%2Fone-proxy.org,https%3A%2F%2Fanother-proxy.org

  <=>
  https://example.service.edu/?idphint=*urlencode(***https://one-proxy.org***),urlencode(***https://another-proxy.org***)**