International
Virtual
Observatory
Alliance

IVOA Single-Sign-On Profile: Authentication Mechanisms

Version 2.0

IVOA Working Draft 20150728

Working group
http://www.ivoa.net/twiki/bin/view/IVOA/IvoaGridAndWebServices

This version
http://www.ivoa.net/documents/SSOAuthMech/20150728

Latest version
http://www.ivoa.net/documents/SSOAuthMech

Previous versions
http://www.ivoa.net/documents/cover/SSOAuthMech-20080124.html
http://www.ivoa.net/documents/cover/SSOAuthMech-20070904.html
http://www.ivoa.net/documents/cover/SSOAuthMech-20070621.html
http://www.ivoa.net/documents/cover/SSOAuthMech-20060519.html

Author(s)
Grid and Web Services Working Group

Editor(s)
Guy Rixon, Andreé Schaaff, Giuliano Taffoni

Abstract

Approved client-server authentication mechanisms are described for the IVOA single-sign-on profile: No Authentication; HTTP Basic Authentication; TLS with passwords; TLS with client certificates; Cookies; Open Authentication; Security Assertion Markup Language; OpenID. Normative rules are given for the implementation of these mechanisms, mainly by reference to pre-existing standards. The Authorization mechanisms are out of the scope of this document.

Status of This Document

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than "work in progress".

A list of current IVOA Recommendations and other technical documents can be found at http://www.ivoa.net/Documents/.

Contents

1 Introduction **4**
- 1.1 Role within the VO Architecture 4

2 Scope of this standard **4**
- 2.1 Requirements 4
- 2.2 Commentary 5

3 Approved authentication mechanisms **6**
- 3.1 Requirements 6

4 HTTP Basic Authentication **7**
- 4.1 Requirements 7
- 4.2 Commentary 7

5 Details of TLS **7**
- 5.1 Requirements 7
- 5.2 Commentary 8

6 Details of TLS-with-client-certificate **8**
- 6.1 Requirements 8
- 6.2 Commentary 8

7 Details of TLS-with-password **8**
- 7.1 Requirements 8
- 7.2 Commentary 8

8 The use of Cookies **9**
- 8.1 Requirements 9
- 8.2 Commentary 9

9 Details on SAML authentication **9**
9.1 Requirements 9
9.2 Commentary 10

10 Details on OAuth **10**
10.1 Requirements 11
10.2 Commentary 11

11 Details on OpenID **11**
11.1 Requirements 11
11.2 Commentary 11

A Changes from Previous Versions **11**
A.1 Changes from v. 1.01 11

Acknowledgments

This document derives from discussions among the Grid and Web Services working-group of IVOA. It is particularly informed by prototypes built by Matthew Graham (Caltech/US-NVO), Paul Harrison (ESO/EuroVO), David Morris (Cambridge/AstroGrid) and Raymond Plante (NCSA/US-NVO). The prior art for the use of proxy certificates comes from the Globus Alliance. This document has been developed with support from the National Science Foundation's Information Technology Research Program under Cooperative Agreement AST0122449 with The Johns Hopkins University, from the UK Particle Physics and Astronomy Research Council (PPARC) and from the Europena Commission's Sixth Framework Program via the Optical Infrared Coordination Network (OPTICON).

Conformance-related definitions

The words "MUST", "SHALL", "SHOULD", "MAY", "RECOMMENDED", and "OPTIONAL" (in upper or lower case) used in this document are to be interpreted as described in IETF standard, Bradner (1997).

The *Virtual Observatory (VO)* is general term for a collection of federated resources that can be used to conduct astronomical research, education, and outreach. The International Virtual Observatory Alliance (IVOA) is a global collaboration of separately funded projects to develop standards and infrastructure that enable VO applications.

1 Introduction

IVOA's single-sign-on architecture is a system in which users assign cryptographic credentials to user agents so that the agents may act with the user?s identity and access rights. This standard describes how agents use those credentials to authenticate the user?s identity in requests to services. This standard describes also the authentication mechanism of an application or a service making a call (on behalf of someone or something else) to an API or to another service This document is essentially a *profile* against existing security standards; that is, it describes how an existing standard should be applied in an IVOA application to support single sign-on capabilities in the IVOA. In the following sections, we make specific references to details spelled out in these standards. For the purposes of validating against this standard, those referenced documents should be consulted for a full explanation of those details. Unfortunately, a reader that is unfamiliar with these external standards might find this specification confusing. To alleviate this problem, each major section is concluded by a Commentary subsection that provides some explanations of the detailed terms and concepts being referred to. The Commentary subsection may also provide recommended scenarios for how this specification might actually be realized. Note that the statements in the Commentary subsections are non-normative and should not be considered part of precise specification; nevertheless, they are indicative of the intended spirit of this document.

1.1 Role within the VO Architecture

Fig. 1 shows the role this document plays within the IVOA architecture (Arviset et al. 2010).

2 Scope of this standard

2.1 Requirements

When a service is registered in an IVOA registry, that service?s resource document may include metadata expressing conformance to one or more of the authentication mechanisms approved in the IVOA SSO profile. Such a service must implement those mechanisms as described in this document, and clients of the service must participate in the mechanism when calling the service. Is a service does not provide any SSO specification it is assumed that no authentication is required. The registration of the service interface shall contain an XML element of type *SecurityMethod* as specified in the XML schema for VOResource [VOResource]. The value of this element distinguished the authentication mechanism using the values stated in the sections below. Services registered without the metadata alluded to above

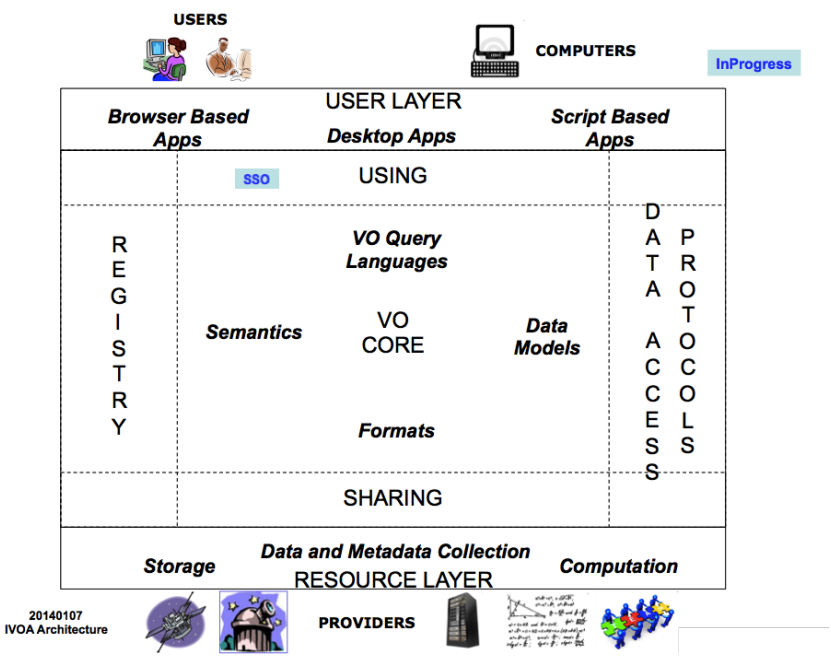

Figure 1: Architecture diagram for this document

need not support any authentication mechanism. If they do require authentication, they may use either the IVOA-standard mechanisms or others that are not IVOA standards.

2.2 Commentary

The IVOA SSO profile allows the development of a ”realm” of interoperable services and clients. Service providers opt in to this realm by implementing this current standard and by registering accordingly in the IVOA registry. This allows clients to discover a secured service through the registry and to be able to use it without being customized for the details of the specific service.

Parts of the IVOA that are not intended to be widely interoperable need not opt in to the SSO realm. In particular, ”private” services, accessed by web browsers and protected by passwords, are allowed. However, these private services should be reworked to follow the IVOA standard if they are later promoted to a wider audience.

An example of a registration for a secured interface follows.

```
<interface xmlns:vs=''ivo://www.ivoa.net/xml/VODataService/v1.0''
        xsi:type=''vs:ParamHTTP''>
        <accessURL>http://some.where/some/thing</accessURL>
        <securityMethod>ivo://ivoa.net/sso#saml2.0</securityMethod>
```

</interface>

More than one *securityMethod* can be specified:

```
<interface xmlns:vs=''ivo://www.ivoa.net/xml/VODataService/v1.0''
        xsi:type=''vs:ParamHTTP''>
        <accessURL>http://some.where/some/thing</accessURL>
        <securityMethod>ivo://ivoa.net/sso#saml2.0</securityMethod>
        <securityMethod>ivo://ivoa.net/sso#cookie</securityMethod>
        <securityMethod>ivo://ivoa.net/sso#OpenID</securityMethod>
</interface>
```

The order identify the priority of the method to use, in the example above the preferred method to access the service is *SAML*, than cookies and finally if the other are not available OpenID.

3 Approved authentication mechanisms

3.1 Requirements

The following authentication mechanisms are approved for use in the SSO profile.

- No authentication required.
- HTTP Basic Authentication
- Transport Layer Security (TLS) with passwords.
- Transport Layer Security (TLS) with client certificates.
- Cookies
- Open Authentication (OAuth)
- Security Assertion Markup Language (SAML)
- OpenID

The mechanism is associated with the interface provided by the service and registered in the IVOA registry.

Services that are registered with a IVOA registry as having a em WebService type interface [VOResource] shall support OAuth, or shall support cookies or shall support TLS with client certificates or shall require no authentication. Interfaces by which a user logs in to the SSO system shall support either TLS with client certificates, or TLS with passwords, or SAML or a combination of the them

SSO mechanism	<securityMethod>
No authentication required	none
HTTP Basic Authentication	http//www.w3.org/Protocols/HTTP/1.0/spec/html#BasicAA
TLS with password	ivo://ivoa.net/sso#tls-with-password
TLS with client certificate	ivo://ivoa.net/sso#tls-with-certificate
Cookies	ivo://ivoa.net/sso#cookie
Open Authentication	ivo://ivoa.net/sso#OAuth
SAML	ivo://ivoa.net/sso#saml2.0
OpenID	ivo://ivoa.net/sso#OpenID

Table 1: List of approved authentication mechanisms and the corresponding securityMethod

4 HTTP Basic Authentication

4.1 Requirements

Services using HTTP basic authentication shall use the authentication mechanism described in the RFC7235 (Fielding 2014) that updates RFC2617 (Franks et al. 1999). Interfaces using this mechanism shall be be registered with the security method

`http://www.w3.org/Protocols/HTTP/1.0/spec/html#BasicAA`

4.2 Commentary

HTTP provides a simple challenge-response authentication framework that can be used by a server to challenge a client request and by a client to provide authentication information. The HTTP authentication framework does not define a single mechanism for maintaining the confidentiality of credentials. HTTP depends on the security properties of the underlying transport- or session-level connection to provide confidential transmission of header fields. Connection secured with TLS are recommended prior to exchanging any credentials.

5 Details of TLS

5.1 Requirements

Services using Transport Layer Security (TLS) shall do so according to the TLS v1.2 standard RFC5246 (Dierks & Rescorla 2008).

5.2 Commentary

TLS supersedes the Secure Sockets Layer that is an outdated cryptographic protocol. TLS v1.0 was based on SSL v3.0; the actual version of TLS is V1.2 described in by Dierks & Rescorla (2008). TLS v1.2 is back-compatible with TLS v1.0, TLS v1.1 and SSL v3.0. "TLS versions 1.0, 1.1, and 1.2, and SSL 3.0 are very similar, and use compatible ClientHello messages; thus, supporting all of them is relatively easy.[...] TLS 1.2 clients that wish to support SSL 2.0 servers MUST send version 2.0 CLIENT-HELLO messages defined in [SSL2]." (Dierks & Rescorla 2008).

6 Details of TLS-with-client-certificate

6.1 Requirements

Certificates shall be transmitted and checked according to the TLS v1.2 standard [RFC5246].

Services implementing TLS must support certificate chains including proxy certificates according to RFC6818 (Yee 2013).

Interfaces using this mechanism shall be be registered with the security method `ivo://ivoa.net/sso#tls-with-client-certificate`.

6.2 Commentary

When Mutual Certificate Authentication is configured for REST services, both, the client and the service perform identity verification or authentication through X.509 certificates.

The client authenticates the service during the initial SSL handshake, when the server sends the client a certificate to authenticate itself.

7 Details of TLS-with-password

7.1 Requirements

The user-name and password shall be passed in the message protected by the TLS mechanism, not as part of the mechanism itself. The "HTTP basic authentication" should be used with particular attention.

Interfaces using this mechanism shall be be registered with the security method `ivo://ivoa.net/sso#tls-with-password`.

7.2 Commentary

"HTTP basic authentication" passes the user-name and password in the HTTP headers, assuming that the credentials are not a natural part of the

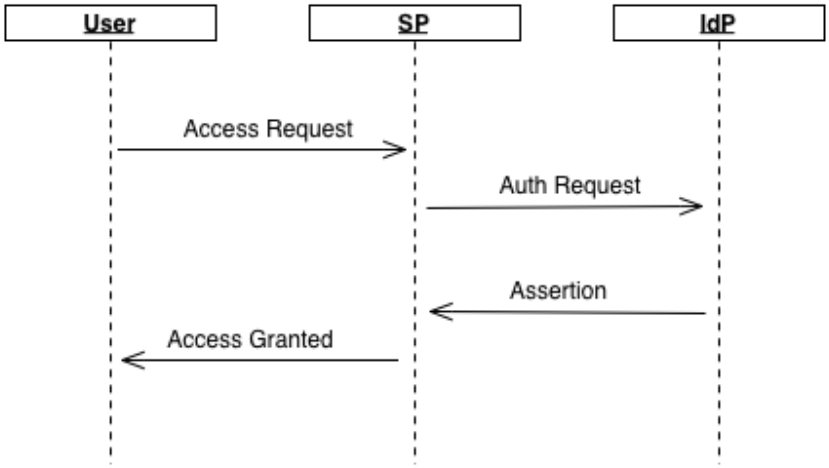

Figure 2: Simplified picture of SAML 2.0 authentication.

message body. This standard applies the TLS-with-Password mechanism only to the special case of logging in to the SSO realm. Hence, the user name and password are logically part of the message body, not the message header.

8 The use of Cookies

8.1 Requirements

Cookie-Based Authentication uses server side cookies to authenticate the user on every request. The way to manage cookies for authentication is described in RFC6265 (Barth 2013).

8.2 Commentary

RESTful web services should use session-based authentication, either by establishing a session token via a POST or by using an API key as a POST body argument or as a cookie. User names, passwords, session tokens, and API keys should not appear in the URL, as this can be captured in web server logs, which makes them intrinsically valuable.

9 Details on SAML authentication

9.1 Requirements

Services using SAML authentication mechanisms shall shall do so according to the saml-core-2.0-os OASIS standard (Cantor et al. 2005*a*). SAML

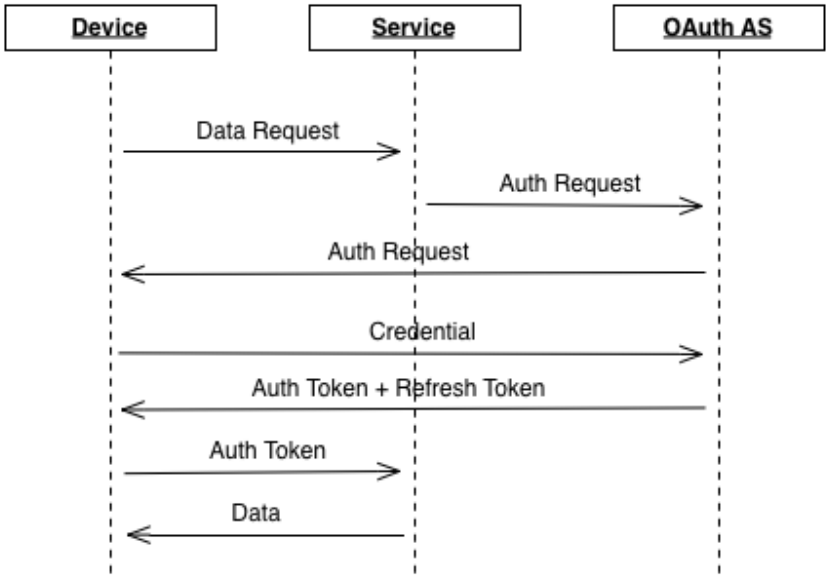

Figure 3: Simplified picture of OAuth 2.0 authentication.

includes protocols and protocol bindings and security (Cantor et al. 2005*b*).

9.2 Commentary

SAML presumes two primary roles in any transaction: the organisation where the identity is established, known as the Identity Provider ("IdP"), or Asserting Party ("AP"); and the organisation which (for this transaction) wants to use this identity, known as the Service Provider ("SP"), or Relying Party ("RP").

A user attempts to access an application with the Service Provider. The SP needs to establish the identity of this user, and so sends an authentication request to the Identity Provider.

The user authenticate with the IDP (IDP is taking care of the authentication mechanisms and protocols e.g. Kerberos, ldap etc.) so the IdP can send back an 'Assertion' to the SP. Now the SP knows who the user is, and can process that user accordingly (see Fig. 2).

SAML2.0 allow also to service discovery mechanisms

10 Details on OAuth

10.1 Requirements

Services using OAuth authentication mechanisms shall do so according to the RFC6749 (Hardt 2012).

10.2 Commentary

Open Authentication 2.0 (also in conjunction with OpenID Connect) is actually the adopted standard to handle identity in the framework of RESTful web services. OAuth is used when an application is making a request on behalf of a user.

OAuth introduces the notion of an 'authorization token', a 'refresh token' and Authorization Service (AS). The 'authorization' token states that the client application has the right to access services on the server (see Fig. 3). However, it does not supersede any access control decisions that the server-side application might make.

OAuth is related to delegate credential from an application to another.

11 Details on OpenID

11.1 Requirements

Services using OpenID authentication mechanisms shall do so according to the OpenID Foundation standards (OpenID 2007)

11.2 Commentary

OpenID is an open and decentralized authentication and identity system. OpenID relying parties do not manage end user credentials such as passwords or any other sensitive information which makes authentication and identity management much simpler and secure. In a RESTful environment OpenID Connect (Sakimura et al. 2014) is commonly adopted as authentication solution. "OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol, which allows computing clients to verify the identity of an end-user based on the authentication performed by an authorization server, as well as to obtain basic profile information about the end-user in an interoperable and REST-like manner." (OpenID 2007).

A Changes from Previous Versions

A.1 Changes from v. 1.01

- We remove all the references to SOAP as deprecated from IVOA

- We add new security methods ad relative discussion sessions: OpenID, SAML, Cookies, HTTP basic authentication

References

Arviset, C., Gaudet, S. & the IVOA Technical Coordination Group (2010), 'IVOA architecture', IVOA Note.
URL: *http://www.ivoa.net/documents/Notes/IVOAArchitecture*

Barth, A. (2013), 'Http state management mechanism', RFC 6265.
URL: *https://tools.ietf.org/rfc/rfc6265.txt*

Bradner, S. (1997), 'Key words for use in RFCs to indicate requirement levels', RFC 2119.
URL: *http://www.ietf.org/rfc/rfc2119.txt*

Cantor, S., Kemp, J., Philpott, R. & Maler, E. (2005*a*), 'Assertions and protocols for the oasis security assertion markup language (saml) v2.0', saml-core-2.0-os.
URL: *http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf*

Cantor, S., Kemp, J., Philpott, R. & Maler, E. (2005*b*), 'Bindings for the oasis security assertion markup language (saml) v2.0', saml-bindings-2.0-os.
URL: *http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf*

Dierks, T. & Rescorla, E. (2008), 'The transport layer security (tls) protocol version 1.2', RFC 5246.
URL: *https://tools.ietf.org/rfc/rfc5246.txt*

Fielding, R. (2014), 'Hypertext transfer protocol (http/1.1): Authentication', RFC 7235.
URL: *https://tools.ietf.org/rfc/rfc7235.txt*

Franks, J., Hallam-Baker, P. & Hostetler, J. (1999), 'Hypertext transfer protocol (http/1.1): Authentication', RFC 2617.
URL: *https://tools.ietf.org/rfc/rfc7235.txt*

Hardt, D. (2012), 'The oauth 2.0 authorization framework', RFC 6742.
URL: *https://tools.ietf.org/rfc/rfc6749.txt*

OpenID (2007), 'Openid authentication 2.0 final', OpenID.
URL: *http://openid.net/specs/openid-authentication-2_0.html*

Sakimura, N., Bradley, J., Jones, M., de Medeiros, B. & Mortimore, C. (2014), 'Openid authentication 2.0 final', OpenID.
URL: *http://openid.net/specs/openid-authentication-2_0.html*

Yee, P. (2013), 'Updates to the internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile', RFC 6818.
URL: *https://tools.ietf.org/rfc/rfc6818.txt*