

HTTP Header Authentication Provider - SSO

NOTE: Ubisecure product names were unified in autumn 2011. All products which started with term "Ubilogin" were renamed to start with term "Ubisecure". In documentation this name change is implemented retroactively, i.e., the new naming practice is used also when referring to old software versions which started with term "Ubilogin" at the time of their release.

About This Documentation

This documentation describes the Ubilogin HTTP Header Authentication Provider and how it is used with the Ubisecure Authentication Server to create extensible authentication systems.

HTTP Header Authentication Provider Functionality

HTTP Header Authentication Provider enables moving identity information from a 3rd party Identity Provider to UAS using HTTP headers. HTTP Header Authentication provider is installed as a Java Servlet web application and the 3rd party Identity Provider has to be set to authenticate users of this application and to set the identity information to HTTP headers. HTTP Header Authentication Provider then reads the user's identity information from HTTP headers and sends user authentication information to UAS.

The HTTP Header Authentication Provider is implemented as a standard Java Servlet web application. The web application can be deployed in any standard Java Application server that implements the Java Servlet 2.3 specification.

Certificate Authentication Provider - SSO

Ubisecure Single Sign-On

The Ubisecure Single Sign-On software product provides a web access management solution that enables access management and single sign-on user authentication using a wide selection of authentication methods, for example: username and password, One-Time Passwords, smart card (or other client certificate), or GSM short messages (plain text or signed).

The key functionality of Ubisecure Single Sign-On is to offer access management and related services for web applications with a selection of authentication methods to best serve the needs of the application or user level in question.

Ubisecure Single Sign-On Authentication Process

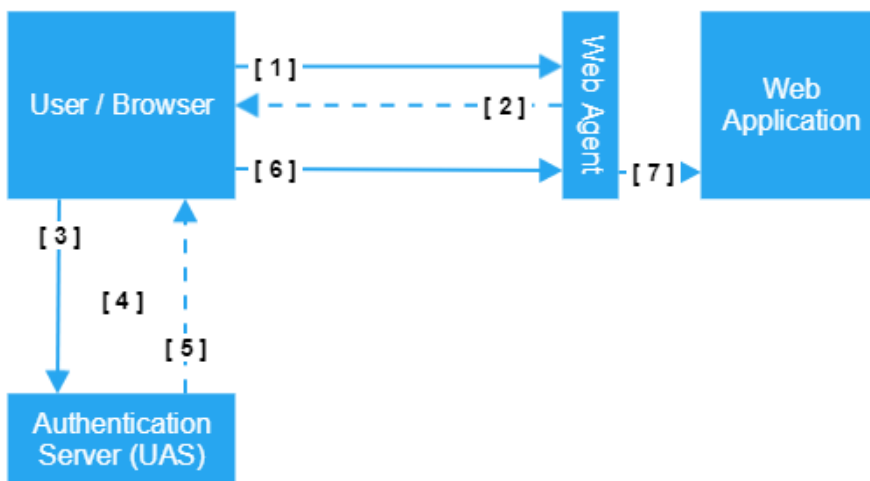


Figure 1. The Ubisecure Single Sign-On authentication and single sign-on process with Web Application

1. The user makes a request with his browser to a restricted web application.
2. The Ubisecure Web Application intercepts the request because no valid credentials are included in the request. The Web Application creates a *ticket request* and redirects the request to the Ubisecure Authentication Server (UAS).
3. The browser forwards the *ticket request* to UAS. UAS validates the *ticket request* and starts the authentication process.
4. Authentication process. **SSO functionality**: If the user has a valid existing session with UAS then the authentication process is skipped.
5. When the user has been authenticated, UAS creates a *ticket response* and redirects the browser back to the web application. UAS also creates an authentication session with the user.
6. The browser repeats the initial request to the web application, now with a *ticket response* included.
7. The Web Application validates the *ticket response* and allows the request to get through to the web application.

Ubisecure Authentication Server authenticates users, implements access control and sends authentication information in encrypted format to Ubisecure Web Applications. The Ubisecure Web Application deciphers and validates authentication information received from the Authentication Server and allows validated requests to get through to the web application. The Web Application also passes information about the authenticated identity to the web application.

Ubisecure Authentication Providers

Ubisecure Authentication Providers extend the available authentication methods available to Ubisecure Authentication Server (UAS) in cases where authentication must be performed on a different network or platform. There are four types of Ubisecure Authentication Providers:

1. Windows Authentication Provider
2. Certificate Authentication Provider
3. Http Header Authentication Provider
4. Custom Authentication Provider

Ubisecure Authentication Provider Authentication Process

Ubisecure Authentication Server and an Authentication Provider interoperate in a similar way to Ubisecure Authentication Server and a Web Application. All communication between UAS and the Authentication Provider is done through browser redirects.

Figure 1 gives an overview of the authentication process where a Web Application requests authentication services from the Ubisecure Authentication Server. With the Authentication Providers the roles are reversed:

- The Ubisecure Authentication Server takes the role of a Web Application.
- The Authentication Provider takes the role of an Ubisecure Authentication Server.

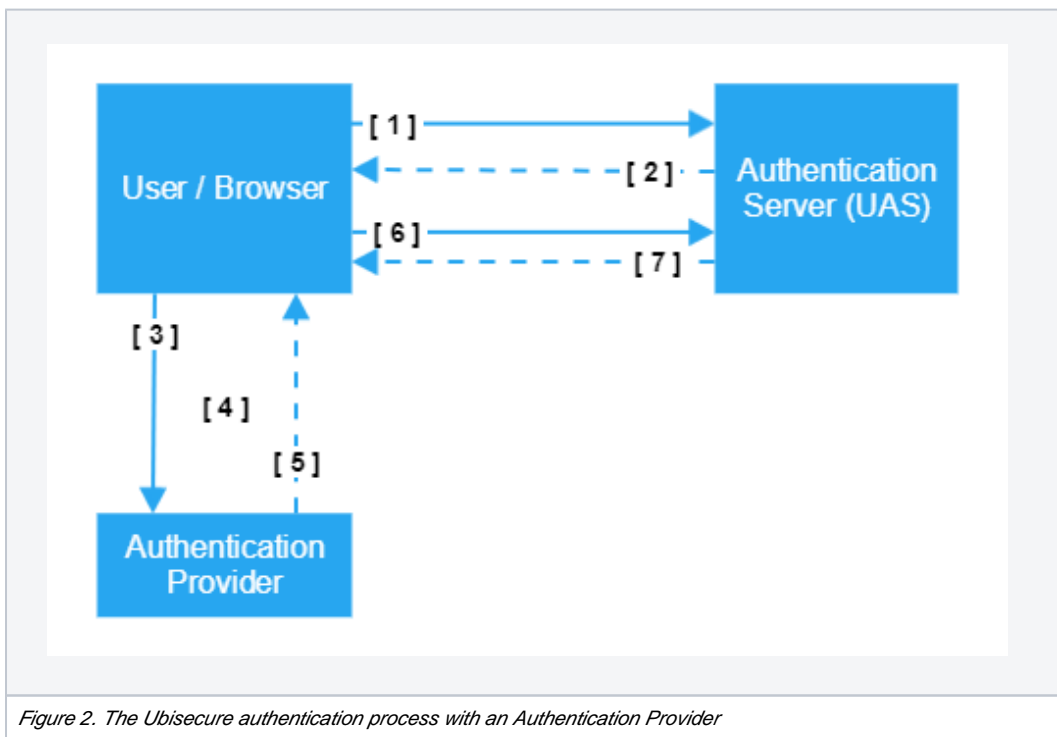


Figure 2. The Ubisecure authentication process with an Authentication Provider

1. The browser has been redirected from a Web Application to UAS with a *ticket request*.
2. The user selects an authentication method involving an Authentication Provider. The UAS creates an *authentication request* and redirects the browser to the Authentication Provider.
3. The browser forwards the *authentication request* to the Authentication Provider. The Authentication Provider validates the *authentication request* and starts the authentication process.
4. Authentication process.
5. The Authentication Provider creates an *authentication response* and redirects the browser back to UAS.
6. The browser forwards the *authentication response* to UAS.
7. UAS validates the *authentication response* and creates a *ticket response* for the Web Application that initially started the process. UAS also creates an authentication session with the user.

Although the functionality provided by the Authentication Provider is very similar to the functionality provided by UAS, there are however some key differences:

- The Authentication Provider generally does not start a session with the browser and therefore does not provide single sign-on functionality. Single sign-on is implemented by UAS.
- The Authentication Provider does not implement access control. UAS implements access control for the Web Applications.
- The Authentication Provider generally provides services for a single authentication server whereas the authentication server provides services for an unlimited number of Web Applications.

Deployment

The Authentication Provider architecture makes it possible to install UAS and the Authentication Provider on disconnected networks. Only the user's browser needs to connect to both servers. No direct connection between Ubisecure Authentication Server and the server running Ubisecure Authentication Provider is required. This possibility enables very advanced scenarios.

Windows Authentication Providers

Some authentication protocols, such as the Windows Integrated protocol with Internet Explorer or Firefox, are only enabled for Intranet use, because they require that the client computer is on the same Active Directory domain. With Ubisecure solution, it is possible to install a Windows Authentication Provider on the Intranet even if the Ubisecure Authentication Server is connected to the Internet.

Organization to Organization Authentication Providers

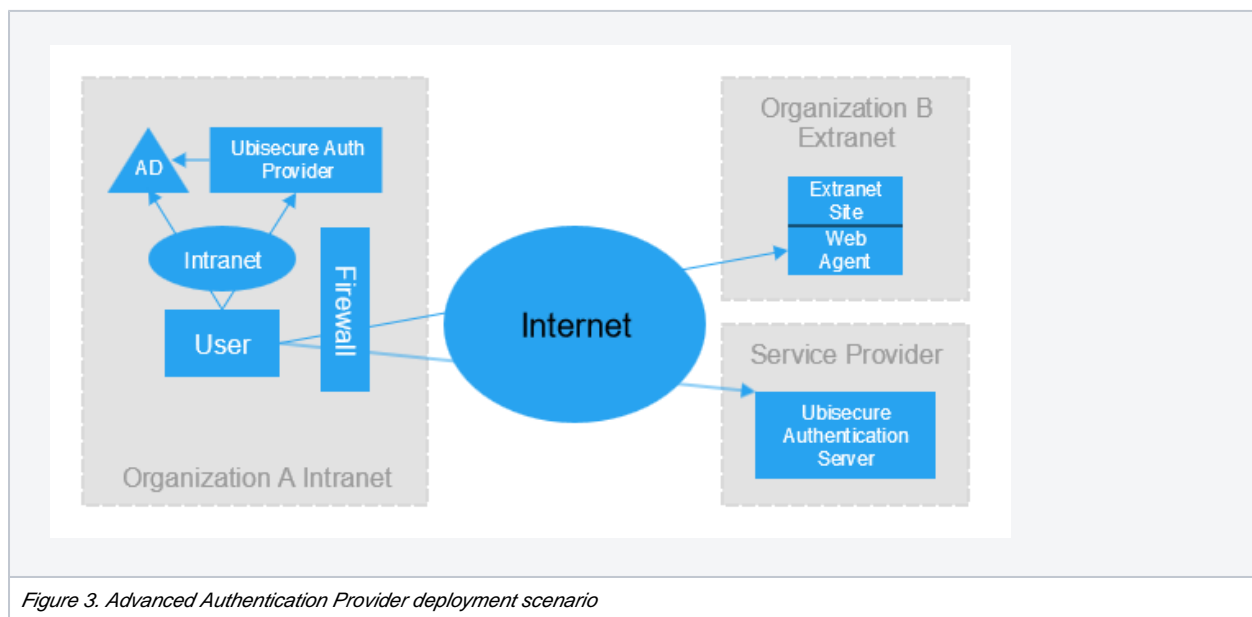


Figure 3. Advanced Authentication Provider deployment scenario

- The users at Organization A authenticate locally on the Intranet to a Windows Active Directory
- This authentication information is forwarded by the Ubisecure Authentication Provider to Ubisecure Authentication Server
- The result is that the Extranet Site at Organization B allows transparent and seamless authenticated access for users from Organization A

Identity Mappings

The Authentication Provider passes the name of the authenticated identity to UAS. UAS maps this identity to a Ubisecure identity. UAS manages separate identity mappings for each Authentication Provider. After a Ubisecure identity is established all normal access control features of Ubisecure Authentication Server are applied.