

Authportal Version 3 Overview

Abstract

The Authportal framework is a home grown single-sign-on system built at Virginia Tech. The need for this system arose when we were asked to duplicate the functionality provided by the Campus Pipeline portal product, in a home grown portal. Authportal was written to provide hooks into those applications which were deemed necessary for the portal. The current version in development (version 3) is no longer portal centric, however the name remains.

The first version of Authportal provided uni-directional single-sign-on with our Webmail system, Banner system, Filebox system, MyVT system, and CourseInfo system. That is, you could access all those system but you had to first login to the portal and then click the link to that system in the portal. This met all our requirements, however the system was not as secure or as flexible as we would have liked.

The second version of Authportal provides 3-tier single-sign-on with the portal and all the previously mentioned systems. In addition, communication between the Authportal and the systems was redesigned to provide for secure transactions. This is the system that is currently being used at Virginia Tech.

The third version of the Authportal is what this document attempts to describe. It will be backwards compatible with Authportal version 2, but our hope is to provide a true single-sign-on framework so that users can login once and access application ubiquitously.

Requirements

Problem Statement

- Web applications need a mechanism to safely transport client sessions from one application to another.

Functional Requirements

- Centralized login for web applications
- Single-sign-on implementation for web applications
- Three tiered login implementation for web applications
- Meet all WEB-ISO requirements
 - WEB-ISO shall support ending a session due to absolute timeouts
 - WEB-ISO shall support ending a session due to inactivity time-outs
 - WEB-ISO shall timestamp all authentication events, to facilitate absolute timeouts
 - WEB-ISO assertions shall timestamp last use to facilitate inactivity timeouts
 - WEB-ISO shall support user authentication sessions: both a login server authentication session for network-wide authentication, and application server authentication sessions for application authentication
 - WEB-ISO shall support ending a session (whether it be application or login) due to logout by the principal
 - WEB-ISO shall protect its data from observation by third parties or untrusted intermediaries, to protect the principal's privacy
 - WEB-ISO shall operate both in and out of an SSL environment, although SSL is required for login server interactions
 - WEB-ISO shall support "application zones" where each application community can define its own required level of authentication
 - WEB-ISO shall support application logouts, application zone logouts, and overall session logouts, where logging out of all application zones is equivalent to an overall session logout
 - WEB-ISO shall not require of the principal anything beyond a standard browser
 - WEB-ISO shall work across multiple DNS domains
- Provide an administrative interface for application configuration
- Provide a client interface for single-sign-on configuration

Nonfunctional Requirements

- Minimal impact on web application logic and development

Server Java Classes

- Servlets

Authportal Version 3 Overview

Status	Displays current information about the Authportal, such as availability and uptime.
SessionImage	Displays a GIF image which notifies the Client whether or not he is logged in.
ValidateTicket	Accepts a request message for the purpose of validating a ticket. Returns a response message with the corresponding Client identifier and action.
FlipSession	Reads the Authportal cookie and determines if it represents a valid session. If it does then that session is ended, otherwise the Client is redirected to a login page.
ValidateSession	Reads the Authportal cookie and determines if it represents a valid session. The Client is redirected based on the results.
CreateTicket	Accepts a request message for the purpose of creating a ticket for a Client. Returns a response message with the ticket.

- Singletons

ClientManager	Manages Client preference data
ServiceManager	Manages Web Service configuration data
SessionManager	Manages Client sessions
TicketManager	Manages transaction tickets
LogManager	Writes an audit trail
RealmManager	Manages Realm configuration data

- Objects

Session	Stores data for each session
---------	------------------------------

Client Java Classes

- Servlets

AuthportalStub	Accepts requests on behalf of a Web Service and communicates with the Authportal.
----------------	---

- Interfaces

ServiceInterface	Allows the AuthportalStub to communicate with a Web Service.
------------------	--

Communication Java Classes

- Objects

AuthportalKeyStore	Stores Java keystore information.
AuthportalResponse	Represents an Authportal response message.
AuthportalSignature	Represents the signature for an Authportal message.
AuthportalRequest	Represents an Authportal request message.
AuthportalTransaction	Performs transactions with an Authportal server.
AuthportalStatus	Represents an Authportal status message.
Ticket	Stores data for each ticket.

Communication

The communication between the Authportal and the Web Service Stubs contain sensitive information which should not be sent in the clear. Client identifiers, tickets, redirect URLs, and actions are all sent back and forth in communication and should all be considered private information. We recommend that implementers use SSL for this communication to keep this data private. However, the design of the Authportal does not prevent implementers from taking this risk if they so choose.

In addition to privacy, messages sent between the Authportal and the Web Service Stubs must provide authenticity. Therefore each message sent is digitally signed. This prevents a malicious host from attempting to steal the identity of a Web Service Stub or the Authportal itself and make false requests on its behalf.

Cookies

Clients are identified to the Authportal via a cookie that is set once the Client has authenticated. If the cookie is set to only be sent to the Authportal over SSL then the risk of session hijacking is minimized. In addition the data in the cookie only contains a one-time identifier for the Client.

Tickets

Tickets are used to communicate Client actions between Web Services and the Authportal. Since these tickets are transported via HTTP parameters in redirects they could be easy targets for replay attacks. There are several ways to minimize this risk. The best solution is to perform all communication over HTTPS, however this is usually not possible. Authportal tickets have a configurable timeout value associated with them, which means that from the time a ticket is issued it must be returned to the Authportal in that amount of time. Since each ticket is only good for one action an attacker would have to intercept a communication and beat the Client back to the Authportal with the ticket. Setting the timeout as low as possible makes this more difficult.

Realms

The term 'realm' in this document refers to what WEBISO calls application zones. In general a realm is a grouping of applications based on a common attribute of those applications. We believe it makes the most sense to create realms based on authentication mechanisms, however realms can be created however the Authportal administrator sees fit. Realms can also contain other realms. So it is possible to put all realms in one master realm. If you are creating authentication realms, it is then possible to create a master realm with uses a very strong authentication mechanism (such as smart card) and then place weaker authentication mechanisms inside that realm. Therefore if a Client authenticates to the master realm, he can then access all the applications in the inner realms.

Communication

This section shows the format of the messages that are sent between the Authportal and the Web Service Stubs. Communication between the Authportal and service stubs must be encrypted. (Since this communication takes place via HTTP this is done with SSL.) In addition to encrypting this channel, all messages sent between the Authportal and Web Service Stubs must be signed. The data shown here is either sample data or a description of the data that should appear in the field.

Authportal Version 3 Overview

Status Message

This message is sent by the Authportal to anyone who requests it.

Service Stubs use this message to determine if the Authportal is accepting requests.

```
<authportal-transaction>
  <response id="123456789">
    <version>3.0</version>
    <started>Tue Jun 24 02:28:21 GMT 2002</started>
    <accepting-requests>true</accepting-requests>
    <validate-
url>https://authportal.middleware.vt.edu/ValidateSession</validate-url>
  </response>
  <Signature>
    <SignatureAlgorithm>SHA1withRSA</SignatureAlgorithm>
    <SignatureData>Hex Encoded Data</SignatureData>
    <SignatureValue>Value of Encrypted Hex Encoded Data</SignatureValue>
    <SignatureCertificate>Base64 Encoded X509
Certificate</SignatureCertificate>
  </Signature>
</authportal-transaction>
```

Authportal Version 3 Overview

Request Message

This message is sent by Web Service Stubs when they are requesting an action by the Authportal.

```
<authportal-transaction>
  <request service="service1" id="123456789">
    <url>URL of the Authportal that this request should be sent to</url>
    <ticket>Authportal ticket (if one applies to this
transaction)</ticket>
    <client>Client Identifier</client>
    <redirect>URL the Client should be redirected to when this request
is
                completed (if one applies to this transaction)</redirect>
  </response>
  <Signature>
    <SignatureAlgorithm>SHA1withRSA</SignatureAlgorithm>
    <SignatureData>Hex Encoded Data</SignatureData>
    <SignatureValue>Value of Encrypted Hex Encoded Data</SignatureValue>
    <SignatureCertificate>Base64 Encoded X509
Certificate</SignatureCertificate>
  </Signature>
</authportal-transaction>
```

Authportal Version 3 Overview

Response Message

This message is sent by the Authportal in response to a Request Message.

```
<authportal-transaction>
  <response service="service1" id="123456789">
    <ticket>Authportal ticket</ticket>
    <action>Action the service should take with this Client</action>
    <client>Client Identifier</client>
    <redirect>URL the Client should be redirected to when this request
is
                completed (if one applies to this transaction)</redirect>
  </response>
  <Signature>
    <SignatureAlgorithm>SHA1withRSA</SignatureAlgorithm>
    <SignatureData>Hex Encoded Data</SignatureData>
    <SignatureValue>Value of Encrypted Hex Encoded Data</SignatureValue>
    <SignatureCertificate>Base64 Encoded X509
Certificate</SignatureCertificate>
  </Signature>
</authportal-transaction>
```

FAQ

Q. If the Authportal goes down or isn't available over the network will clients still be able to access my application?

A. Maybe, if the Authportal is your only means of authentication then your application is dependent on the Authportal. However you can configure your applications to fall back to a different authentication mechanism if the Authportal is down.

Q. Can the Authportal login page be customized for each Web Service?

A. No and we don't want it to be. Login pages are associated with the Realm that a Web Service is in, not the Web Service itself. We believe this will help users understand that they are logging into a realm of applications, and hopefully keep in mind the security implications that go along with that action.

Q. Why is the Authportal Cookie only read by the Authportal? Wouldn't it be easier to allow each Web Service Stub to decrypt the cookie?

A. While this would be easier, our implementation must deal with Web Services in different domains from that of the Authportal. These Web Services would not be sent the cookie. While the Authportal could issue the Authportal Cookie for each Web Service we felt the solution with the least number of cookies was the best solution.

Q. What data is contained in the Authportal Cookie?

A. The data in the Authportal Cookie is a random SHA-1 hash. It's only purpose is to uniquely identify the Client in the Authportal framework.

Q. When a browser is redirected from a HTTPS connection to a HTTP it will display a warning to the client. Is this problem addressed in your implementation?

A. Yes, the redirects sent by both the Authportal and Web Service Stubs are not a HTTP Status 302 redirects. They are a HTML page which uses the <meta> tag to redirect the browser. Browsers will not complain if the Client performs the redirect rather than the server.

Q. Since only the communication with the Authportal is over HTTPS aren't parts of this system vulnerable to replay attacks?

A. Yes, although the tickets sent by the Authportal contain no personal information they can be used to hijack someone's session. This can be prevented by using SSL in the Web Service.

Q. If I click 'logout' in an application, am I logging out of that application or am I logging out of the Authportal framework?

A. This is a configuration option of the Authportal. Applications must maintain their own sessions for each client, however the Authportal can request that an application end a client's session. Otherwise each application will be responsible for destroying a client's session. In this scenario logging out of one application will not stop a client from entering a different application in the Authportal framework, or the application that was just logged out of.

Authportal Version 3 Overview

Q. How does a client logout of the Authportal framework?

A. Each application in the framework must display an image which is served from the Authportal. This image is used to keep track of a client's activity, for timeout purposes. It also shows the client whether or not he is logged in. If the client is not logged in to the Authportal framework, he can click the image to begin the logon process. Conversely, if the client is already logged in, he can click the image to begin the logout process.

Q. How long does the Authportal session last?

A. When a client logs into the Authportal he must tell the Authportal if he is at a public terminal or a private terminal. Depending on the client's selection an appropriate timeout value will be selected. The value of that timeout is a configuration option.

Q. I don't want to use SSL between the Authportal and Web Service Stubs, do you support message encryption?

A. Not at this time.