Oracle Access Manager Integration

Contents

Overview

This article provides a basis for integration with Oracle Access Manager 10g.

Client code is included which you can deploy in conjunction with your Oracle Access Manager and PINsafe environment.

Prerequisites

- Oracle Access Manager 10g
- PINsafe 3.8

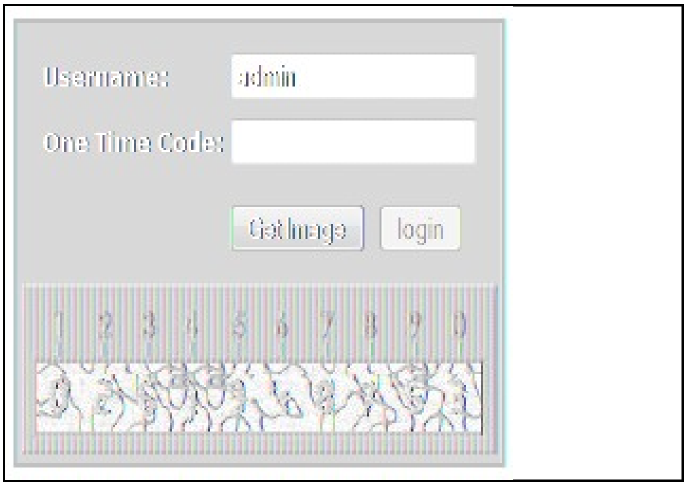
Download GenericIntegration.zip to obtain the client code for this integration. The code contains Eclipse project settings and a pre-built version of the WAR file for deployment.

Deployment

Deploy the war file into the Apache Tomcat webapps folder of your existing PINsafe 3.8 installation.

If using a PINsafe appliance, you can use WinSCP. See the WinSCP How To Guide for further information on transferring files to a PINsafe appliance.

Integration



Username: admin

One Time Code:

Image showing the Generic Integration login page
The integration requires a JSP containing username, password, OTC and the TURing image. Upon ?Logon? this is posted to a servlet which extracts the username, password and OTC from the request and calls the login (username, password, otc) of the PINsafeClient class (supplied in the jar). This method sends an HTTP request to PINsafe to allow a user to login or change their PIN.

Some pointers regarding the sample code provided:

- Marked in the code is the point in login.jsp where the creating of the OAM cookie described below should occur.
- WEB-INF/settings.xml contains the configuration to point to the PINsafe server, where to redirect upon success and whether or not to use a password.
- WEB-INF/settings2.xml contains a configuration for secondary server, for high availability purposes.

Authentication Process

During authentication, if the user is authenticated by PINsafe then depending on elements specific to your integration, you would either:

- Create an OAM cookie, which would require knowledge and availability of the OAM API;
- Redirect to the location for normal processing of the login (typically /oblix/login.cgi as detailed on the Oracle website). This would require a filter to stop anyone calling /oblix/login.cgi directly.