Oracle
Oracle PeopleSoft 9.1

RSA SecurID Ready Implementation Guide

Last Modified: July 22, 2013

Partner Information

Product Information	
Partner Name	Oracle
Web Site	http://www.oracle.com/us/products/applications/peoplesoft-enterprise/overview/index.html
Product Name	PeopleSoft
Version & Platform	9.1
Product Description	Oracle's PeopleSoft Enterprise applications are designed to address the most complex business requirements. They provide comprehensive business and industry solutions, enabling organizations to increase productivity, accelerate Business Performance and lower cost of ownership

Solution Summary

To enable RSA SecurID two-factor authentication for Oracle PeopleSoft, you must install an RSA Authentication Manager web server and configure that server as a reverse-proxy to the PeopleSoft application server. You must also implement a custom web server script/plugin/module that uses the RSA Cookie API to extract an authenticated user's username from an RSA encrypted cookie and writes it to an HTTP header variable. Finally, you must write and deploy a PeopleCode function that retrieves the username from the header variable and uses it to create a PeopleSoft session.

Supported RSA Features	
Oracle PeopleSoft 9.1	
RSA SecurID Authentication via Native RSA SecurID Protocol	Yes
RSA SecurID Authentication via RADIUS Protocol	No
On-Demand Authentication via Native SecurID Protocol	Yes
Risk-Based Authentication	Yes
Risk-Based Authentication with Single Sign-On	Yes
RSA Authentication Manager Replica Support	Yes
RSA SecurID Software Token Automation	No
RSA SecurID SD800 Token Automation	No
RSA SecurID Administrative Interface Protection	No

The table below contains the flow-of-events for a successful authentication.

Step	Component	Description
1	Browser	The user attempts to access a PeopleSoft URL on the reverse-proxy server **http://<web-server-name>/psp/ps/?cmd=start**
2	Web Server	The RSA Authentication Manager Web Server Agent intercepts the request and authenticates the user. **A custom script/module** calls the RSA Web Agent Cookie API to extract the user ID and stores it in an HTTP Response header variable. The web proxy forwards the call to the PeopleSoft Server.
3	PeopleSoft Servlet	The PeopleSoft servlet receives the HTTP request and connects to the application server using credentials set in the in the current Web Profile.
4	Application Server	The PeopleSoft application server executes a custom Signon PeopleCode function, which grabs the "real" User ID from the HTTP request and uses it to create a PeopleSoft session for the user ID.

Authentication Agent Configuration

Authentication Agents are records stored in an RSA Authentication Manager database. They contain information that allows the server to locate its clients and establish secure communication channels with them. You must use the RSA Security Console to create an agent record for each web server in your environment that acts as a reverse-proxy to PeopleSoft. You will need the following information in order to do so:

- the hostname of every web server in your environment that acts as a reverse-proxy to PeopleSoft
- IP address for all of the network interfaces on each web server in your environment that acts as a reverse-proxy to PeopleSoft

When you create an agent, set its agent type to *Standard Agent*.

Note: Each agent hostname must resolve to one or more valid IP addresses on the local network.

RSA SecurID files

RSA SecurID Authentication Files	
Files	**Location**
sdconf.rec	**Windows***: C:\Program Files\RSA Security\RSAWebAgent* **Unix***: /web_server_directory/rsawebagent*
Node Secret (*securid*)	**Windows***: C:\Program Files\RSA Security\RSAWebAgent* **Unix***: /web_server_directory/rsawebagent*
sdstatus.12	**Windows***: C:\Program Files\RSA Security\RSAWebAgent* **Unix***: /web_server_directory/rsawebagent*
sdopts.rec	**Windows***: C:\Program Files\RSA Security\RSAWebAgent* **Unix***: /web_server_directory/rsawebagent*

Note: The appendix of this document contains more detailed information regarding these files.

Partner Product Configuration

Before You Begin

This section provides instructions for enabling RSA SecurID two-factor authentication for Oracle PeopleSoft users. You should have working knowledge of PeopleSoft, PeopleTools, PeopleSoft Application Designer and RSA Authentication Manager, as well as access to the appropriate end-user and administrative documentation. Ensure that these products are running properly prior to configuring the integration. Note that this document is not intended to suggest optimum installations or configurations.

Prerequisites

You must perform the following actions before configuring the integration:

- Ensure that everyone who will be using the integration has matching usernames in PeopleSoft and RSA Authentication Manager.
- Install an RSA Authentication Manager-supported web server as a reverse-proxy to PeopleSoft.

Important: RSA does not provide support for third-party software. If you need further assistance to set up a proxy server, refer to the web server and PeopleSoft application for details.

- Install the appropriate RSA Authentication Manager Web Server Agent and the RSA Web Agent Cookie API on the reverse-proxy server.
- Write a plugin/script/ISAPI filter that uses the RSA Cookie API to read an authenticated username from the RSA Agent's encrypted cookie and write it to a custom HTTP header variable. See the **Appendix** and the *RSA Web Agent API Developer's Guide* for more information.

Important: The RSA Web Agent Cookie API provides developers with the ability to extract an authenticated username from an encrypted cookie. RSA doesn't provide any modules, scripts, etc. that perform this function. You must create one for your environment, deploy it on your web server and configure your web server proxy appropriately. Note that you must configure your proxy to run this script before it redirects to the PeopleSoft application server. See your proxy plugin's documentation for instructions.

Configure PeopleSoft

This section contains instructions for configuring PeopleSoft to enable RSA SecurID two-factor authentication. It is divided into the following subsections:

Create a PeopleSoft Default User

Log into PeopleSoft, create a user and grant the user permission to log into the PeopleSoft application. This user will invoke your Signon script. We use the following credentials in this example:

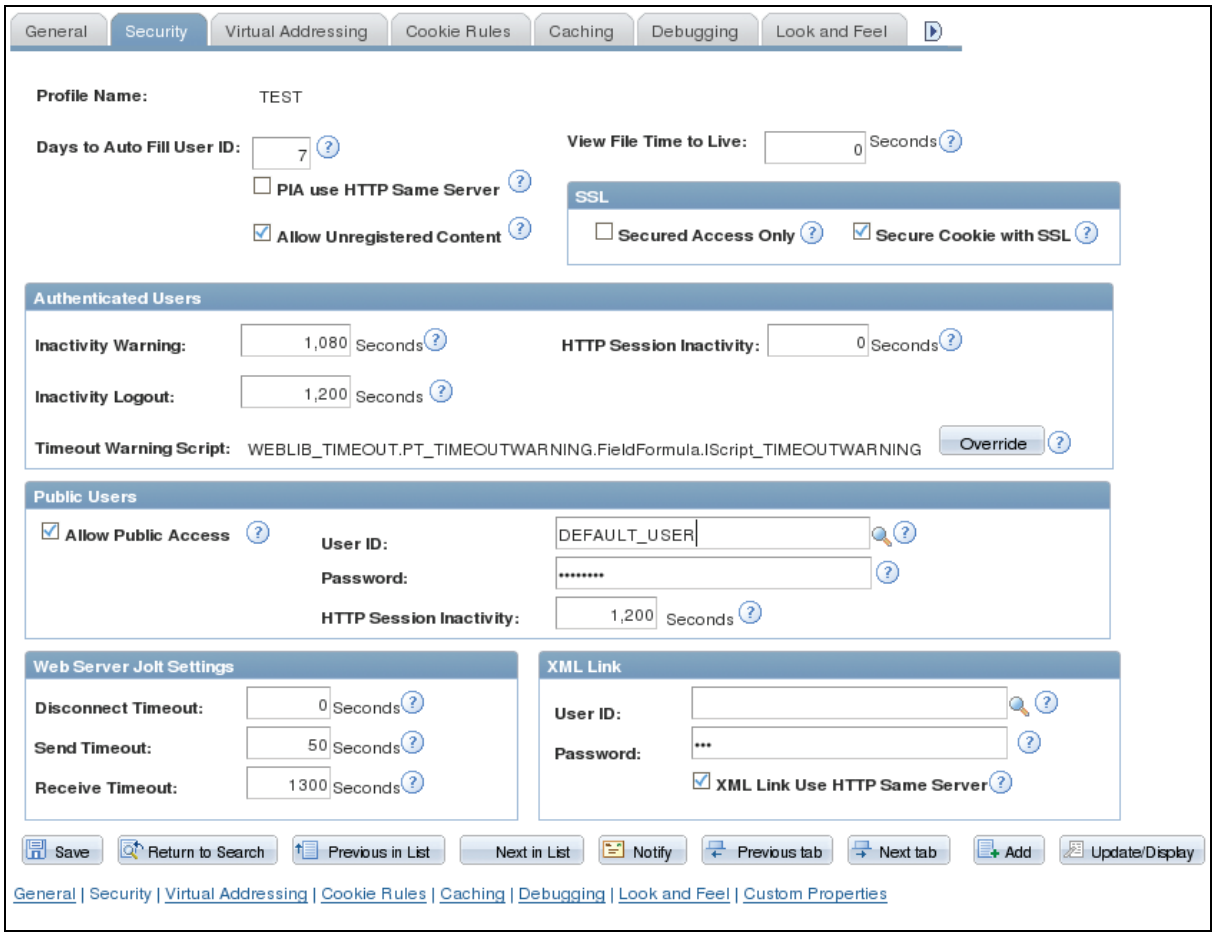
- User ID = DEFAULT_USER
- Password = PASSWORD

Note: PeopleSoft User IDs and Passwords are case sensitive.

Disable the PeopleSoft Signon Page

Since RSA Authentication Manager will be handling user authentication, you must disable the **PeopleSoft Signon** page. Follow the steps below in order to do so:

1. Log in to PeopleSoft and navigate to **PeopleTools → Web Profile → Web Profile Configuration → *<%Profile%>* → Security,** where *<%Profile%>* is the appropriate Web Profile.
2. Check the **Allow Public Access** checkbox and enter the default user's ID and password into the **User ID** and **Password** fields respectively.
3. Click the **Save** button.



Write a PeopleCode Signon Function for RSA Authentication Manager

In order for PeopleSoft to create a session for an authenticated user, it must invoke a **custom PeopleCode function** during an event in its authentication process. The function reads the authenticated username from the HTTP **header variable you chose** in your RSA Cookie API script. Once PeopleSoft executes the function, it will create a session for the authenticated user.

In the sample PeopleCode function below, the code attempts to retrieve the authenticated username from an HTTP header variable named *HTTP_AM_REMOTE_USER*. If your custom script writes the username to another variable, replace the *%Request.GetHeader* function's parameter with the variable's name.

Note: In the following example, we add the *HTTP_AM_REMOTE_USER* function to an existing record called *FUNCLIB_LDAP*. However, you may add it to a new record if you wish. See the *Enterprise PeopleTools 8.5 People Book: Security Administration* for more information.

To add the following code to the *FUNCLIB_LDAP* record:

1. Launch PeopleSoft Application Designer and login as a user who has permission to modify the *FUNCLIB_LDAP* record.
2. Open the *FUNCLIB_LDAP.LDAPAUTH* record's *FieldDefault* event's PeopleCode with the PeopleCode Editor.
3. Replace the existing code with the *RSA_AUTHENTICATION_MANAGER* function listed below.

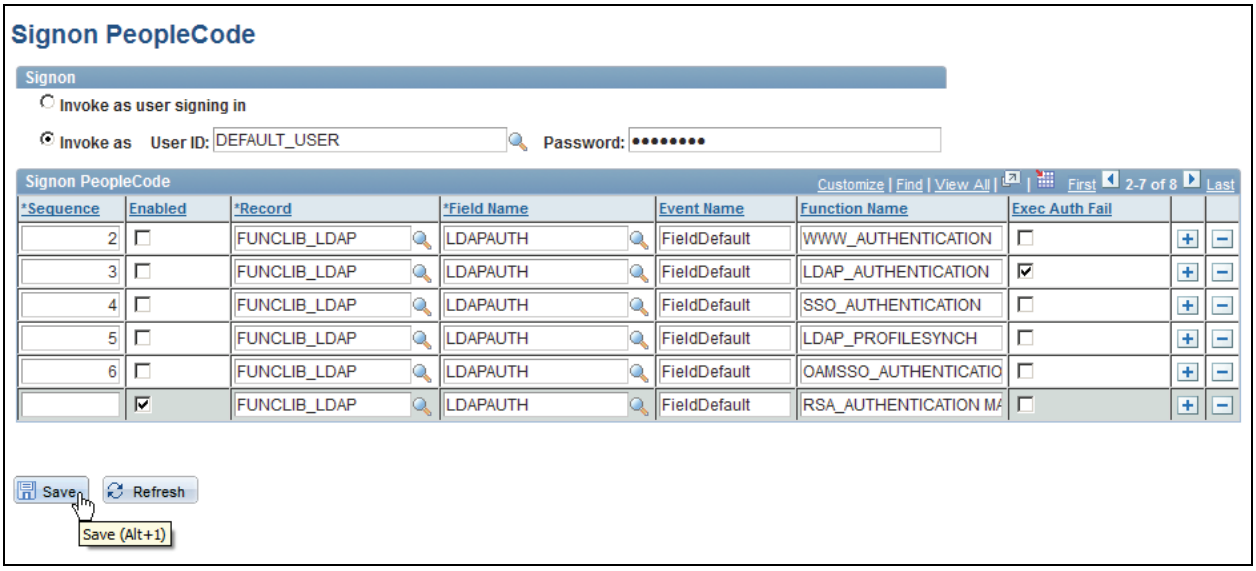
```
/*//////////////////////////////////////////////////////////////////////////////

RSA_AUTHENTICATION_MANAGER: PIA invokes this function after:

  1. RSA Authentication Manager has authenticated a user and …
  2. a custom script running on the reverse-proxy web server that hosts the RSA Web
  Agent has written the authenticated username to an HTTP Header variable.

PeopleSoft reads the username from the header and uses it to create a session.  The
code below uses the HTTP_AM_REMOTE_USER variable, but you must use the same username
you used to write the aforementioned custom script.

Note: Every RSA username must match its corresponding PeopleSoft username.

//////////////////////////////////////////////////////////////////////////////*/

Function RSA_AUTHENTICATION_MANAGER()

  If %PSAuthResult = True And &authMethod <> "WWW" And &authMethod <> "LDAP" And
         &authMethod <> "SSO" Then

       getWWWAuthConfig();

       If %SignonUserId = "DEFAULT_USER" Then
          &userID = %Request.GetHeader("HTTP_AM_REMOTE_USER");

          If &userID <> "" Then
             SetAuthenticationResult( True, Upper(&userID), "", False);
             &authMethod = "RSA";
          End-If;
       End-If;
  End-If;

End-Function;
```

Important: The header variable name you use in your PeopleCode function **must match** the one you used in your Web Agent Cookie API script/module/plugin.

Activate the PeopleCode Signon Function

To configure PeopleSoft to run the PeopleCode Signon function during the authentication process:

1. Select **PeopleTools → Security → Security Objects → Sign On PeopleCode** from the **Main Menu**, go to the last row in the **Signon PeopleCode** table and click the plus sign
2. Go to the empty row at the bottom of the table and check the **Enabled** checkbox.
3. Enter *FUNCLIB_LDAP* in the **Record** field, *LDAPAUTH* in the **Field Name** field and *FieldDefault* in the **Event Name** field.
4. Enter the name of your PeopleCode function in the **Function Name** field. (The function's name is *RSA_AUTHENTICATION MANAGER* in the sample code).
5. Leave the **Exec Auth Fail** checkbox unchecked.
6. Select the **Invoke as** radio button and enter the **default user's credentials** into the **User ID** and **Password** fields, click the **Save** button and restart the servers.



Configure the PeopleSoft Logout Link

Since RSA Authentication Manager will be handling user authentication, you must modify or disable the PeopleSoft logout link. The following example sets the logout link to point to an HTML page that closes the browser, thus destroying both PeopleSoft and RSA Authentication Manager sessions. (Consult your PeopleSoft documentation for other options.)

1. Make a backup copy of the *%PEOPLETOOLS_PORTAL_HOME%\WEB-INF\psftdocs\ps\signin.html*, open the original and replace its contents with the following:

```
<html><body>
    <SCRIPT LANGUAGE="JavaScript">
      window.opener = top;
      window.close();</SCRIPT>
  </body></html>
```

2. Restart the servers.

Configure the Authentication Agent for Risk-Based Authentication

The RSA Web Server Agent supports Risk-Based Authentication (RBA) out-of-the-box. To enable RBA for PeopleSoft users, you must generate a script that you will run from agent's default RSA SecurID logon page. The integration script redirects the user from the agent's logon page to a web application that allows RSA Authentication Manager to authenticate the user with RBA.

Follow the steps below to enable Risk-Based Authentication:

1. Open the RSA Authentication Manager Security Console for your primary RSA Authentication Manager server and select **Access → Authentication Agents → Manage Existing**.
2. Select your agent and click the **Edit** button.
3. Check the **Enable this agent for risk-based authentication** checkbox in the Risk-Based Authentication (RBA) section.
4. Select either *Password* or *SecurID* from the **Authentication Method** dropdown list. This value determines the primary authentication method that the agent will use during RBA.
5. Click the **Save Agent & Go to Download Page** button.
6. Select *RSA Authentication Agent for Web* from the **Agent Type** drop-down list and click the **Download File** button. The system will generate a JavaScript file – *am_integration.js* - for the web server agent you selected.
7. When prompted, *save am_integration.js* to your agent host machine, open the file and copy the functions named *toAbsolutePath()* and *redirectToIdP()*.
8. Navigate to the <*RSA_WEB_AGENT_TEMPLATE*> directory and open the *useridandpasscode.htm* file for editing (The file is read-only by default).

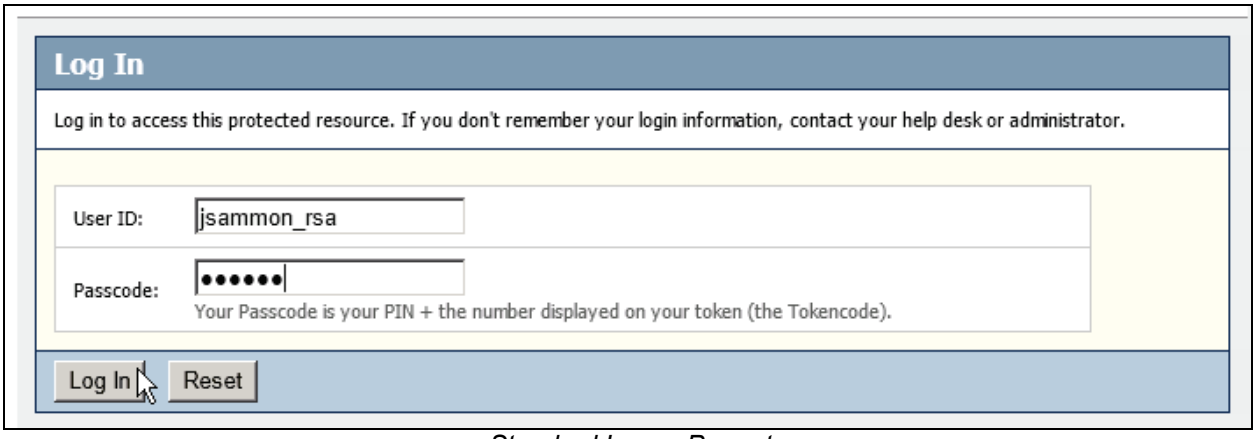
Note: By default, the <*RSA_WEB_AGENT_TEMPLATE*> directory is located at *C:/Program Files/RSA Security/RSAWebAgent/templates* on **Windows** and */usr/local/apache/rsawebagent/Templates* on **UNIX**.

9. Paste the functions you copied directly before the closing HTTP *</script>* tag in the *useridandpasscode.htm* file's header section. You'll find the </*script*> tag right before the header's closing tag - *</head>*.
10. Paste the following line directly before the document's closing *</body>* tag:

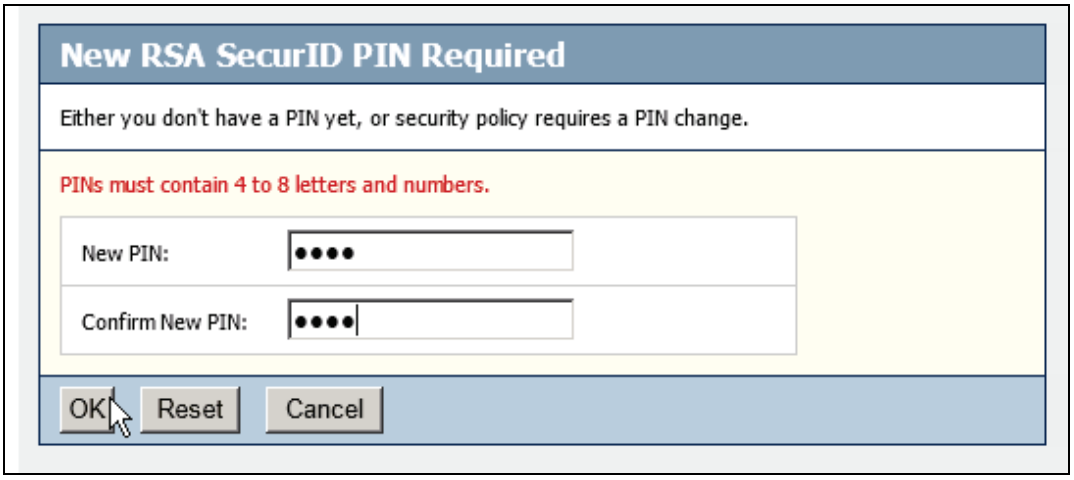
    <script language="JavaScript">window.onload=redirectToIdP()</script>

11. Save *useridandpasscode.htm* and restart the web server.

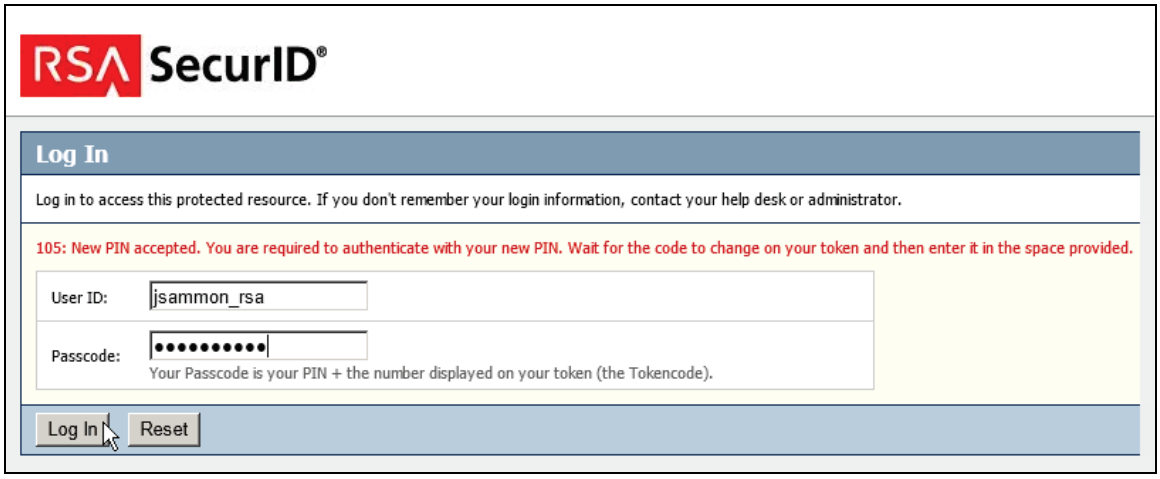
Screenshots



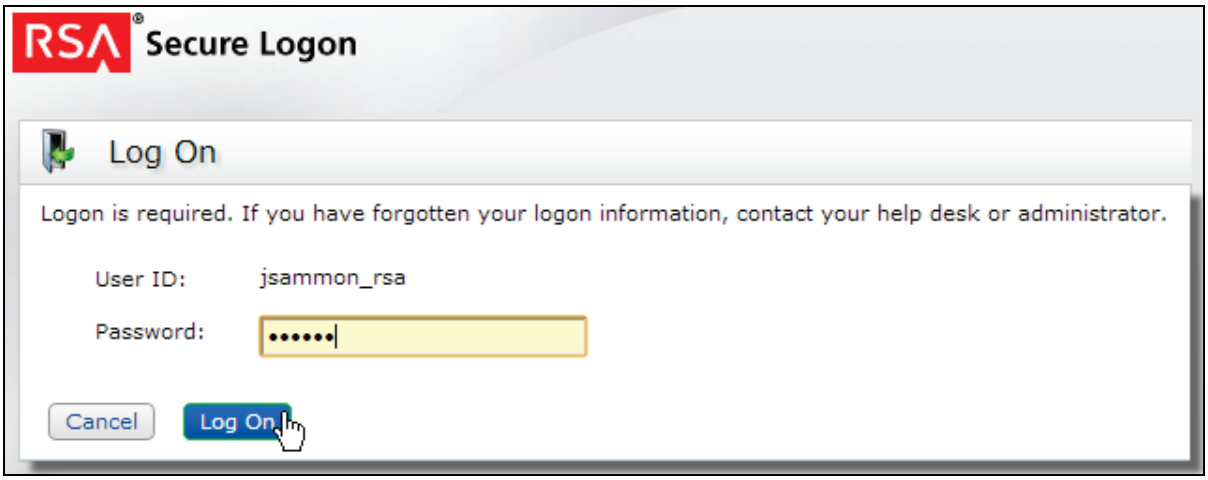
.Standard Logon Prompt



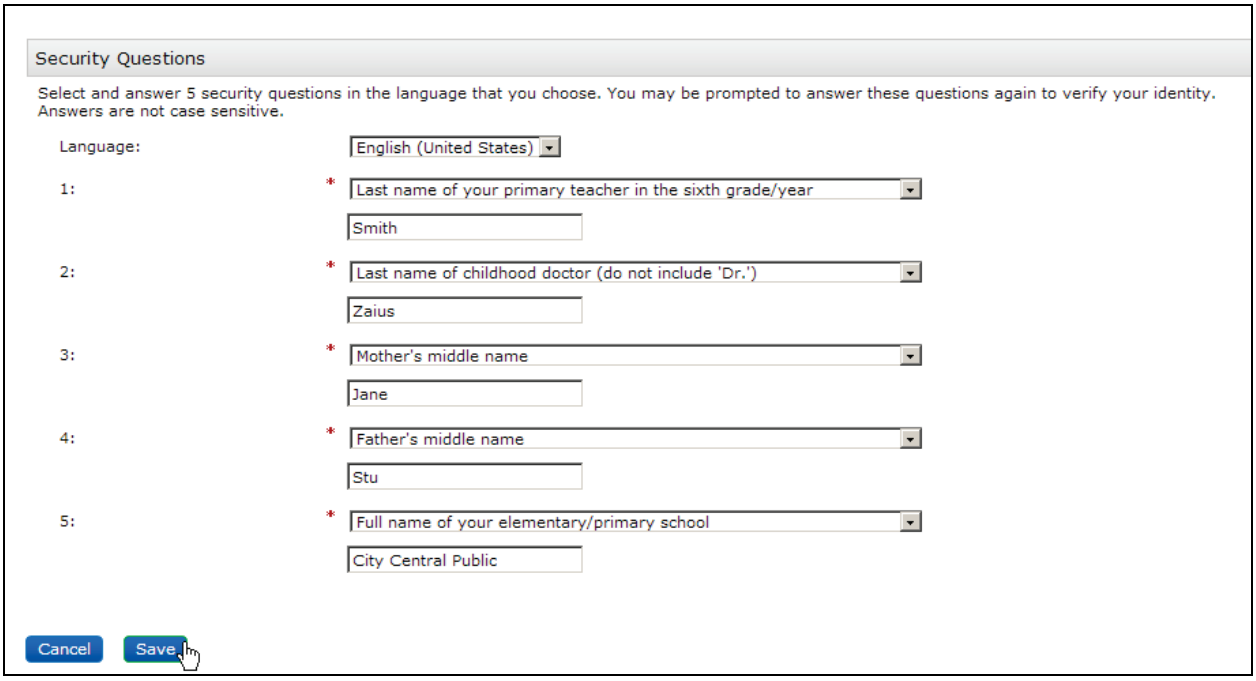
New PIN Mode Prompt



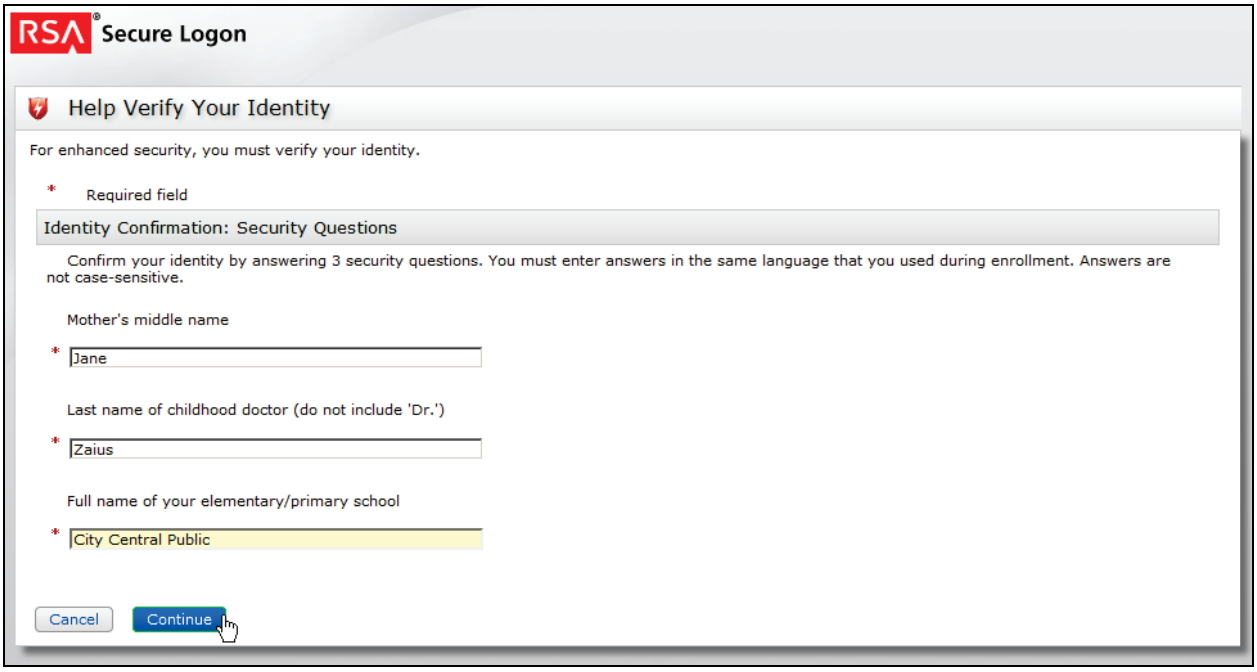
Next Tokencode Prompt



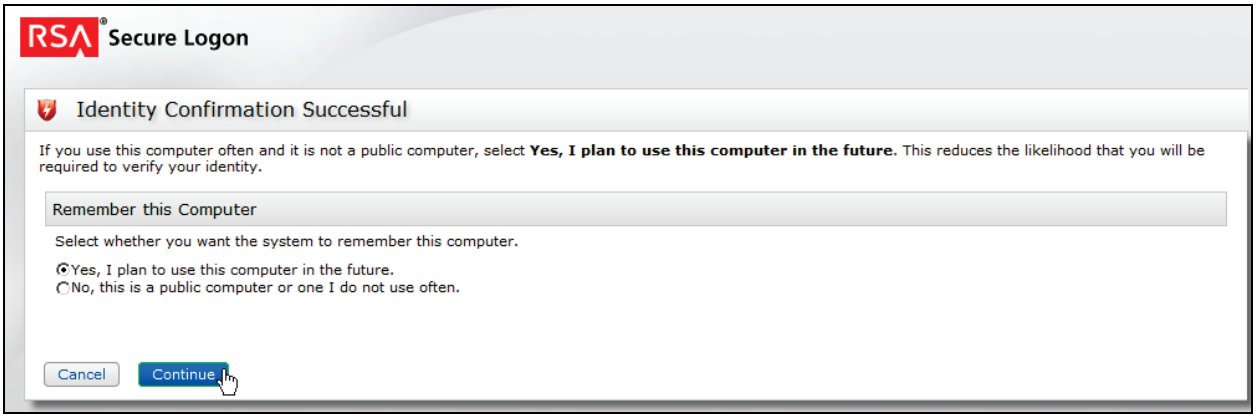
RBA Primary Authentication Prompt



RBA Security Questions Enrollment Page



RBA Security Question Authentication Prompt



RBA Device Binding Preference Prompt

Certification Checklist for RSA Authentication Manager

Date Tested: 7/16/2013

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA Authentication Manager Virtual Appliance	8.0	*N/A*
RSA Authentication Agent	7.1 for IIS	Windows 2008 R2 64-bit
Microsoft IIS Web Server	7.0	Windows 2008 R2 64-bit
Oracle PeopleSoft	9.1	Linux 2.6.18
Oracle PeopleTools	8.5	Linux 2.6.18
Oracle PeopleSoft Application Designer	8.5	Linux 2.6.18
Oracle Database	11g	Oracle Enterprise Linux 5.2.x
Oracle Tuxedo	10gR3	Linux 2.6.18
Oracle WebLogic	10.3.1	Linux 2.6.18
Oracle WebLogic Server IIS Web Server proxy plug-in (*iisproxy.dll*)	1.1	Windows 2008 R2 64-bit

Mandatory Functionality			
RSA Native Protocol		**RADIUS Protocol**	
New PIN Mode			
Force Authentication After New PIN	✓	Force Authentication After New PIN	N/A
System Generated PIN	✓	System Generated PIN	N/A
User Defined (4-8 Alphanumeric)	✓	User Defined (4-8 Alphanumeric)	N/A
User Defined (5-7 Numeric)	✓	User Defined (5-7 Numeric)	N/A
Deny 4 and 8 Digit PIN	✓	Deny 4 and 8 Digit PIN	N/A
Deny Alphanumeric PIN	✓	Deny Alphanumeric PIN	N/A
Deny Numeric PIN	✓	Deny Numeric PIN	N/A
Deny PIN Reuse	✓	Deny PIN Reuse	N/A
Passcode			
14 Digit Passcode	✓	14 Digit Passcode	N/A
4 Digit Fixed Passcode	✓	4 Digit Fixed Passcode	N/A
Next Tokencode Mode			
Next Tokencode Mode	✓	Next Tokencode Mode	N/A
On-Demand Authentication			
On-Demand Authentication	✓	On-Demand Authentication	N/A
On-Demand New PIN	✓	On-Demand New PIN	N/A
Load Balancing / Reliability Testing			
Failover (3-10 Replicas)	✓	Failover	N/A
No RSA Authentication Manager	✓	No RSA Authentication Manager	N/A

JGS

✓ = Pass ✗ = Fail N/A = Not Applicable to Integration

Date Tested: 7/19/2013

RSA Risk-Based Authentication Functionality			
RSA Native Protocol		**RADIUS Protocol**	
Risk-Based Authentication			
Risk-Based Authentication	✓	Risk-Based Authentication	N/A
Risk-Based Authentication with SSO	✓	Risk-Based Authentication with SSO	N/A

JGS

✓ = Pass ✗ = Fail N/A = Not Applicable to Integration

Appendix

Node Secret:

The node secret (*securid* is stored in the *C:\Program Files\RSA Security\RSAWebAgent* directory on **Windows** and in the */web_server_directory/rsawebagent* directory on **Unix**.

sdconf.rec:

The *sdconf.rec* file is stored in the *C:\Program Files\RSA Security\RSAWebAgent* directory on **Windows** and in the */web_server_directory/rsawebagent* directory on **Unix**.

sdopts.rec:

This *sdopts.rec* file is stored in the *C:\Program Files\RSA Security\RSAWebAgent* directory on **Windows** and in the */web_server_directory/rsawebagent* directory on **Unix**.

sdstatus.12:

The *sdstatus.12* is stored in the *C:\Program Files\RSA Security\RSAWebAgent* directory on **Windows** and in the */web_server_directory/rsawebagent* directory on **Unix**.

Developing the RSA Cookie API Web Script or Plug-in

Each time an RSA SecurID user successfully authenticates over the web, the web agent stores an encrypted authentication cookie in the user's web browser. The cookie passes the user's authentication information to the server when the user browses to a protected file or directory on that server. The RSA Web Agent Cookie API provides developers with the ability to extract an authenticated username from the cookie.

You must use the API to create a web server script or plugin (Perl CGI, JavaScript, etc.) for your environment. The script/plugin must extract the authenticated user's username from the encrypted cookie and store it in an HTTP header variable. Your **PeopleSoft Signon function** will extract the username from the variable when the web server proxy forwards the request to PeopleSoft.

> ! **Important**: Note that you must configure your proxy to run this script or plug-in before it redirects to the PeopleSoft application server.

The RSA Web Agent Cookie SDKs (Windows and UNIX) each contains a developer's guide and sample code that demonstrates how to use the API. The Windows examples include a JavaScript ASP, a VBScript ASP, an IIS ISAPI filter and a Pearl CGI. The UNIX examples include a C CGI, a Perl script and a JSP.