RSA Authentication Agent 7.1 for Web for Apache Web Server Installation and Configuration Guide

The Security Division of EMC

Contact Information

Go to the RSA corporate web site for regional Customer Support telephone and fax numbers**:** **www.rsa.com**

Trademarks

RSA and the RSA logo are registered trademarks of RSA Security Inc. in the United States and/or other countries. For the most up-to-date listing of RSA trademarks, go to **www.rsa.com/legal/trademarks_list.pdf**. EMC is a registered trademark of EMC Corporation. All other goods and/or services mentioned are trademarks of their respective companies.

License agreement

This software and the associated documentation are proprietary and confidential to RSA, are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by RSA.

Note on encryption technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Limit distribution of this document to trusted personnel.

RSA notice

The RC5™ Block Encryption Algorithm With Data-Dependent Rotations is protected by U.S. Patent #5,724,428 and #5,835,600.

April 2012

Contents

Preface

About This Guide

This guide describes how to install and configure the RSA Authentication Agent 7.1 for Web for Apache Web Server. It is intended for administrators and other trusted personnel. Do not make this guide available to the general user population.

RSA Authentication Agent 7.1 for Web for Apache Web Server Documentation

For more information about RSA Authentication Agent 7.1 for Web for Apache Web Server, see the following documentation:

Release Notes. Provides workarounds for known issues. The latest version of the *Release Notes* is available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

Installation and Configuration Guide. Describes detailed procedures on how to install and configure the Web Agent.

Developer's Guide. Provides information about developing custom programs using the Web Agent application programming interfaces (APIs). Includes an overview of the APIs.

Integrating RSA Authentication Agent for Web with RSA Authentication Manager Express Guide. Describes detailed procedures on how to install and configure the web agent to work with Authentication Manager Express(AMX).

Related Documentation

For more information about products related to RSA Authentication Agent 7.1 for Web for Apache Web Server, see the following:

RSA Authentication Manager documentation set. The full documentation set for RSA Authentication Manager 6.1.2 is included in the InstallPath\RSA Security\RSA Authentication Manager\doc directory. The updated documentation set for RSAAuthentication Manager 7.1SP4, is included in the InstallPath\doc directory.

Getting Support and Service

RSA SecurCare Online	**https://knowledge.rsasecurity.com**
Customer Support Information	**www.rsa.com/support**
RSA Secured Partner Solutions Directory	**www.rsasecured.com**

RSA SecurCare Online offers a knowledge base that contains answers to common questions and solutions to known problems. It also offers information on new releases, important technical news, and software downloads.

The RSA Secured Partner Solutions Directory provides information about third-party hardware and software products that have been certified to work with RSA products. The directory includes Implementation Guides with step-by-step instructions and other information about interoperation of RSA products with these third-party products.

Before You Call Customer Support

Make sure you have direct access to the computer running the Web Agent software.

Please have the following information available when you call:

- ❑ Your RSA Customer/License ID.
- ❑ RSA Authentication Agent 7.1 for Web for Apache Web Server software version number.
- ❑ The make and model of the machine on which the problem occurs.
- ❑ The name and version of the operating system under which the problem occurs.

1 Overview of RSA Authentication Agent 7.1 for Web for Apache Web Server

RSA Authentication Agent 7.1 for Web for Apache Web Server (Web Agent) allows you to protect all or selected web pages with RSA SecurID.

The Web Agent software, residing on a web server, intercepts all user requests for protected web pages. When a user attempts to access a URL that RSA SecurID protects, the Web Agent requests the user name and passcode and passes them to RSA Authentication Manager for authentication. If the authentication is successful, the Web Agent stores the information in a cookie in the user's browser. As long as the cookie remains valid, the user is granted access to protected web pages.

Note: Web access authentication protects http and https URLs. Web access authentication does not support gopher, news, ftp, wais, or telnet protocols.

Security Features

When combined with RSA Authentication Manager, the Web Agent enhances web server security with the strong, two-factor authentication of time-based RSA SecurID tokens. The following table describes the security features provided by the Web Agent.

Security Feature	Description
Two-factor authentication	To gain access to a protected web page, users enter their user name and a valid RSA SecurID passcode, which consists of: • A personal identification number (PIN). • The tokencode currently displayed on their RSA SecurID token.
Support for SSL	Establishes a private communication channel between the user and the web server that prevents third parties from eavesdropping.

Security Feature	Description
Tamper-evident cookies	Cookies that the Web Agent distributes to a user's browser contain: • Information indicating that the user has successfully authenticated. • An encrypted data string that is used to detect whether someone has altered the cookie contents. Any tampering is logged in the system Web Agent audit files. The Web Agent administrator can set the expiration times for the cookies during installation to help protect the URL if users walk away from their machines.
Name locking	Name locking protects against the risk that an unauthorized person might observe a user entering the passcode and submit the same passcode on a different agent host in the realm more quickly than the original user. Name locking is not needed for most customers. Name locking has no effect when the Web Agent is configured to authenticate in conjunction with RSA Authentication Manager 7.1. Name locking must be enabled for the agent host on RSA Authentication Manager 6.1.2 to gain any benefit from the feature. **Note:** The name locking feature offers security tradeoffs that may or may not be appropriate for your environment. By enabling name locking, a 30-second lock is created on RSA Authentication Manager. As with any lockout mechanism, this can be used to prevent a valid user from authenticating by continually relocking the valid user name.
Auditing	The Web Agent records: • Access attempts • Status of connections • Any instances of cookie tampering in audit logs on the agent host

Note: The security provided by the Web Agent depends on the security of the protected system. Even if the Web Agent is implemented with no vulnerabilities, the strong authentication it provides can be subverted if the underlying system is compromised. Note that, the Web Agent is intended to bolster the security of the web server and not replace it. Also, if the underlying application is insecure, the Web Agent cannot prevent those vulnerabilities from being exploited. The user is still responsible for securing the servers protected by the Web Agent.

Important: Securing servers necessarily involves securing the binaries and other files stored on the server. Vulnerabilities have been noted in web servers when symbolic links are used. RSA recommends that you avoid the use of symbolic links to confidential documents. RSA also recommends that only Administrators should be allowed access to production machines hosting web servers. You also need to ensure that sample code is not installed on production machines.

Types of User Access

Users authenticate to the Web Agent to access protected URLs. You can configure the Web Agent to:

- Protect URLs on the local server on which the Web Agent is installed
- Allow users access to URLs on other servers that the Web Agent protects in the same domain or in multiple domains

For each access type, the Web Agent distributes a cookie to the user's browser so that the user does not have to reauthenticate to each protected resource during a browser session.

The following table describes the different types of user access.

Access Type	Cookies Distributed to User's Browser Upon Successful Authentication	URLs the User Can Access	Configuration Instructions
Local	Local cookie	Protected URLs on the local web server	"Setup Menu" on page 30
Domain	Domain cookie	Protected URLs on all web servers in the domain	"Domain and Multiple Domain Menus" on page 35
Multiple domains	Domain cookies from each domain	Protected URLs on web servers in multiple domains	"Domain and Multiple Domain Menus" on page 35

2 Preparing for Installation

Hardware and Operating System Requirements

The Web Agent is supported on

Apache Web Server 2.2.* on Red Hat Enterprise Linux 4.8(32-bit)

Apache Web Server 2.0.* on Red Hat Enterprise Linux 4.8(64-bit)

Apache Web Server 2.2.* on Red Hat Enterprise Linux 5.5(32-bit and 64 bit)

Apache Web Server 2.0.* and 2.2.* on Solaris 10 with Zones Sun UltraSPARC (64-bit)

Apache Web Server 2.0.* and 2.2.* on Solaris 10 with Zones AMD(64-bit)

Apache Web Server 2.2.* on AIX 6.1(64-bit)

Apache Web Server 2.2.* on Windows 2008(32-bit and 64-bit)

Apache Web Server 2.2.* on Windows 2008R2(64-bit)

Note: Make sure that the web server machine is located in a secure area, so that, only trusted personnel can access the server.

The following table lists the operating system and disk space requirements to install the Web Agent.

Operating Systems	Red Hat Linux Enterprise 4.8(32-bit and 64-bit) Red Hat Enterprise Linux 5.5(32 -bit and 64-bit) Solaris 10 with Zone (64-bit) AIX 6.1 (64-bit) Windows 2008 SP2 (32-bit and 64-bit) Windows 2008R2 SP1 (64-bit) Solaris 10 with Zones (64-bit)
Disk Space	10 MB

Supported Browsers

Users accessing protected web pages must install one of the following web browsers on their machines:

- Microsoft Internet Explorer 7.0
- Microsoft Internet Explorer 8.0
- Microsoft Internet Explorer 9.0
- Mozilla Firefox 3.6.*

Wireless Support

RSA SecurID web authentication through wireless access protocol requires the following WAP 1.1 and WAP 1.2.1 specifications:

- Caching of cookies
- WML Document Type Definition (DTD) version 1.1

RSA SecurID users must enable the cookie acceptance feature in their browsers. They must also use web browsers that support FORMs and Persistent Client State HTTP Cookies.

Interoperability with RSA Authentication Manager and Authentication Manager Express

RSA Authentication Agent 7.1 for Web for Apache Web Server is supported on RSA Authentication Manager 6.1.x, RSA Authentication Manager 7.1.x, and Authentication Manager Express 1.0.

The Web Agent administrator must be familiar with Authentication Manager / Authentication Manager Express (AMX) and its features.

In addition, make sure that the Authentication Manager/ AMX administrator has registered users in the Authentication Manager database and has distributed tokens to the users.

Pre-Installation Tasks

You can specify the character set used by the application either at the Web Agent level or at the web site level. Character settings specified at the Web Agent level are used as default values for all protected web sites. If you specify character settings at both the Web Agent and the web site levels, the Web Agent uses the web site settings.

For servers hosting multiple character set encoding, you must specify character sets for each web site. If you do not specify the character sets correctly, the web site does not function properly and data may get corrupted.

UTF-8 is the default character set that is used when the Web Agent is installed.

During installation, you can specify the character set at the Web Agent level. If you want to override these settings, you can specify the character set for each web site individually. The character settings configured during installation are inherited by all the web sites under Apache, until overridden by the site level settings.

Before installing the Web Agent, you must complete the following tasks:

CAUTION: Make sure that no rpc servers with the name aceapi_rpc_server are already running.

1. "Enable the Apache Web Server to Work with the Web Agent."
2. "Add the Web Server to the Authentication Manager Environment."

Enable the Apache Web Server to Work with the Web Agent

Note: Before starting the Apache web server, you need to add the following directive to the **httpd.conf** file:

```
AddDefaultCharset Off
```

This ensures that the charset parameter is not sent in the HTTP header.

The Apache server binaries must have module **mod_so** and **prefork** enabled.

If your Apache web server is already installed and configured, use the following procedure to verify whether the modules are enabled.

To verify that the modules are enabled:

1. Change to the Apache web server installation directory. For example:

   ```
   cd /usr/local/apache/bin
   ```

   Note: The Apache web server installation directory for the Windows platform is located at **C:\Program Files\Apache Software Foundation\Apache 2.2\ bin**.

2. Type:

   ```
   ./httpd -l
   ```

3. Look for **mod_so.c** and **prefork.c** in the output.

   Note: In the Windows environment, look for **mod_so.c** and **prefork.c** or **mod_win32.c** in the output.

   If the correct modules are not listed, you must recompile the Apache web server binaries with the modules enabled. For instructions, see your Apache web server documentation.

Proceed to the following section, "Add the Web Server to the Authentication Manager Environment."

Add the Web Server to the Authentication Manager Environment

To add the web server to the Authentication Manager environment:

1. Register the Web Agent as an agent of Authentication Manager. The Agent type must be **Net OS Agent** for RSA Authentication Manager 6.1.2, latest service pack and **Web Agent** for RSA Authentication Manager 7.1SP4.

   Note: For more information on adding the Web Agent to RSA Authentication Manager 7.1 SP4, see the RSA Security Console Help topic "Add Authentication Agents."

2. Get the **sdconf.rec** file from your Authentication Manager administrator.

   The Web Agent software uses the **sdconf.rec** file to locate the Authentication Manager on the network.

3. Create a folder called **ace** in the **/var** directory and save the **sdconf.rec** file in the **/var/ace** directory.

   The user owning the web server must have write permissions to the directory. By default, this user is called "nobody."

   Change the owner of the **/var/ace** directory to daemon and use the **chmod** command to set appropriate permissions for the **/var/ace** directory and the **sdconf.rec** file. Type:

   ```
   chmod 755 /var/ace
   chmod 755 sdconf.rec
   ```

   Note: If you install multiple agents on the server, you must create different directories to store their respective **sdconf.rec** files.

4. Add a VAR_ACE environment variable to your web server configuration file, so that it is set whenever the web server runs.

   This environment variable identifies the location of the **sdconf.rec** file. For example:

   In bash:

   ```
   export VAR_ACE=/var/ace
   ```

   In csh:

   ```
   setenv VAR_ACE /var/ace
   ```

   Note: If you install multiple agents pointing to different Authentication Managers, you need to set the value of the VAR_ACE variable during installation to point to the different directories that you created to store the **sdconf.rec** files.

Note: For an agent host running on the Windows platform, copy the Authentication Manager sdconf.rec file to the agent host.

3 Installing RSA Authentication Agent 7.1 for Web for Apache Web Server

The Web Agent can be installed to work with compatible versions of Authentication Manager as well as Authentication Manager Express.

Install the Web Agent

Perform the following steps to install the Web Agent to work with either Authentication Manager or Authentication Manager Express.

To install the Web Agent:

1. Do one of the following:
   - "Install the Web Agent on a UNIX host."
   - "Install the Web Agent on a Windows host."
2. "Configure the Agent to Work with AMX."

   Note: This step is not required for the Web Agent to work with Authentication Manager.

3. "Perform a Test Authentication."

Install the Web Agent on a UNIX host

Before you Begin

Before you install the Web Agent, ensure that the rpc server is running. Type the following command:

```
ps -ef | grep rpc
```

The output of this command is:

```
rpc       4655     1  0 12:39 ?        00:00:00 portmap
rpcuser   4675     1  0 12:39 ?        00:00:00 rpc.statd
root      4705     1  0 12:39 ?        00:00:00 rpc.idmapd
root      6262  6235  0 13:20 pts/2    00:00:00 grep rpc
```

Important: RSA recommends that you stop the web server before installing the Web Agent.

To install the Web Agent:

1. Log on to an account that has write permissions to the web server root directory.
2. Change to the directory that you created when you downloaded the software, and extract the software files.

   Note: Perform step 3 and step 4 only if you are installing the Web Agent on a virtual machine.

3. Create a file named **sdopts.rec** and save it in the location where you saved the **sdconf.rec** file.

   Provide the required permissions to access the **sdopts.rec** file. Type:

   ```
   chmod 755 sdopts.rec
   ```

4. Use the CLIENT_IP keyword to specify an IP address override for the Web Agent host in the **sdopts.rec** file, as follows:

   ```
   CLIENT_IP= ip_address_of_your_machine
   ```

5. Run the installation script.

   Change to the directory where the installation kit is available and type:

   ```
   ./install
   ```

   Ensure that the installation script has execute permission. To set the permission, type:

   ```
   chmod u+x install
   ```

   Note: If you already have RSA Authentication Agent 7.1 for Web for Apache web server installed on your machine, and you try installing the Web Agent again, you receive a message that the Web Agent is already installed and the installation stops.

6. When prompted to specify where you obtained your Web Agent product, if you obtained it from somewhere other than the countries listed, type **n**. Otherwise, press ENTER.
7. Type **A** to accept the License Terms and Conditions.
8. If the path to the **sdconf.rec** file is correct, press ENTER.

   Note: If the path to the **sdconf.rec** file is not specified, installation will not proceed.

   The pathname entered for the VAR_ACE environment variable is displayed. If the pathname is not correct, it may not be correctly defined in the variable. For information about this setting, see "Add the Web Server to the Authentication Manager Environment" on page 14.

9. When prompted for the path to the Apache servername directory, specify the complete path to the web server, and press ENTER.
10. Verify the complete path to the Apache configuration file and Apache httpd binary file, and press ENTER.
11. Set **LD_LIBRARY_PATH** (**LIBPATH** for AIX) to rsawebagent directory.

Important: This is required to start the Apache server.

After successful installation of the Web Agent, the configuration script starts automatically. For manual configuration instructions, see "Config Utility" on page 30.

Next Steps

After installing the Web Agent, check if the rpc server and RSALogOffCookie service are running as expected.

To check if the rpc server is running:

1. Start the Apache web server. Change to the **bin** directory of the web server. Type:
   ```
   ./apachectl -k start
   ```

   Important: Ensure that you start portmap before invoking the rpc server.

2. From the **bin** directory of the web server, type:
   ```
   ps -ef | grep rpc
   ```
   The output of the above command is:
   ```
   rpc 4655     1  0 12:39 ? 00:00:00 portmap
   rpcuser 4675     1  0 12:39 ? 00:00:00 rpc.statd
   root 4705     1  0 12:39 ? 0:00:00 rpc.idmapd
   nobody 6798     1  0 13:23 ? 00:00:00 aceapi_rpc_server
   nobody 6804  6798  0 13:23 ? 00:00:00 aceapi_rpc_server
   root 6914  6897  0 13:36 pts/2    00:00:00 grep rpc
   ```
   This command checks if the rpc server is running. If you see aceapi_rpc_server in the output of the ps command, the rpc server is running. Else, restart the Apache web server to start the rpc server.
3. Check if the RSALogOffCookie service is running as expected.

Install the Web Agent on a Windows host

To install the Web Agent:

1. Log on to the machine as a local administrator.
2. Browse to the directory you created when you downloaded the software, and double-click **setup.exe**.

   Note: There are separate installers for the 32-bit and 64-bit versions of Microsoft Windows 2008 SP2 and Windows 2008R2. Select the installer that is appropriate for your operating system.

3. Follow the prompts until the **sdconf.rec Location** dialog box opens. Specify the location of the configuration file.
4. Specify the location of the Apache folder.
5. Specify the location of the httpd.conf file.
6. Specify the location to install the RSA Authentication Agent for Web for Apache.
7. Follow the prompts to complete installation.
8. Restart the Apache web server.

After successful installation of the Web Agent, the configuration script starts automatically. For manual configuration instructions, see "Config Utility" on page 30.

Configure the Agent to Work with AMX

The Web Agent has to be configured for it to work with the Risk Based Authentication feature of Authentication Manager Express.

For more information on the configuration steps for the Agent to work with AMX, refer to the *Integrating RSA Authentication Agent for Web with RSA Authentication Manager Express Guide*.

Note: The following features are unavailable in a Web Agent configured for AMX:
Use separate user name and PASSCODE pages
Use JavaScript pop-up window to authenticate in frames

Perform a Test Authentication

You test authentication to:

- Verify the authentication environment
- Create a node secret for the Web Agent

The node secret is a symmetric encryption key that Authentication Manager and the Web Agent use to encrypt and decrypt packets of data as they travel across the network. The first time a user successfully authenticates or tests authentication from a Web Agent host, Authentication Manager creates a node secret for that Web Agent host and stores it in the Authentication Manager database. A copy of the node secret is encrypted and sent to the Web Agent. The node secret is stored in a file on the Web Agent host.

If the node secret on the Web Agent host is corrupted or does not match the node secret in the Authentication Manager database, encrypted communications between the Web Agent and Authentication Manager cannot work. If this happens, Authentication Manager logs a node verification failure message in the Authentication Manager Activity Monitor.

To test authentication for UNIX:

1. Change to the Web agent directory (*/web_server_directory*/**rsawebagent** is the default).
2. Type:
   ./acetest
3. Enter a valid user name and passcode.
   If you are repeatedly denied access, test the Authentication Manager status. For more information, see "acestatus" on page 53, or contact your Authentication Manager administrator.

To test authentication for Windows:

1. Access the directory where the WebAgent is installed.
2. Click to run **sdtest.exe**.
3. In the RSA SecurID Authentication Information dialog box, click **RSA ACE/Server Status**, verify the information, and then click **OK**.
   For more information about the server status, see the Help topic "Verifying the Status of Your Environment."n and Configuration Guide
4. Click **RSA ACE/Server Test Directly**.
5. In the **Token type** list, select **Key fob or metal card**.
6. In the **User name** field, enter the user name of the registered RSA SecurID user assigned to the token.
7. In the **Passcode** field, enter the passcode, which is the PIN followed by the tokencode generated by your RSA SecurID token.
8. Click **OK**.

Migrate Web Agent Configuration Information

Note: This section is only applicable to the UNIX platform.

If you already have RSA Authentication Agent 5.3 or 7.0 for Web for Apache web server installed on your machine, you can migrate to RSA Authentication Agent 7.1 for Web for Apache web server.

Note: If the web site you are migrating used the RSA logoff URL in Web Agent 5.3 or 7.0 , it may not work with the current version of the Web Agent, when you enable the **Use RSA Token for Cross-Site Request Forgery Protection** option in the Configuration menu. For more information on the logoff URL in Web Agent 7.1, see "Invalidate Web Access Authentication Cookies Using the Logoff URL" on page 27. By turning off the **Use RSA Token for Cross-Site Request Forgery Protection** option, you can retain the old functionality of the logoff URL. But, you might lose protection against cross-site request forgery attacks on pages that are dependent on templates provided by RSA. RSA recommends that you modify the application suitably to take advantage of this protection for your web site.

If you have customized the web site template in Web Agent 5.3 or 7.0 to a language other than English, you must configure the web site level settings in Web Agent 7.1 to the appropriate character set while migrating.

If a web site protected by Web Agent 5.3 or 7.0 does not use the UTF-8 character set, configure the web site level settings in Web Agent 7.1 with the appropriate character set while migrating. For more information, see "Setup Menu" on page 30.

To migrate the Web Agent:

1. Run the installation script.
   Change to the directory where the installation kit is available and type:
   ```
   ./install
   ```
2. Copy the **RSAWebAgent.INI** file from the **rsawebagent** directory of the older version of the Web Agent to the current **rsawebagent** directory using the following command:
   ```
   cp /directory of old rsawebagent/RSAWebAgent.INI
   /directory of new rsawebagent/
   ```
   This command overwrites the **RSAWebAgent.INI** file in the current **rsawebagent** directory.
3. Run the **./config** script from the current **rsawebagent** directory.

Upgrade to Web Agent 7.1

You can upgrade to Web Agent 7.1 from Web Agent 7.0 on the following:

- Apache 2.2.x on RedHat Enterprise Linux 5.1(32/64-bit)
- Apache 2.2.x on RedHat Enterprise Linux 4(32-bit)

To upgrade to Web Agent 7.1:

1. Stop the Apache Web Server.
2. Run the installation script.
   Change to the directory where the installation kit is available and type:

```
./install
```

Ensure that the installation script has execute permission. To set the permission, type:

```
chmod u+x install
```

Note: If you already have RSA Authentication Agent 7.1 for Web for Apache web server installed on your machine, and you try installing the Web Agent again, you receive a message that the Web Agent is already installed and the installation stops.

3. When prompted to specify where you obtained your Web Agent product, if you obtained it from somewhere other than the countries listed, type **n**. Otherwise, press ENTER.
4. Type **A** to accept the License Terms and Conditions.
5. If the path to the **sdconf.rec** file is correct, press ENTER.

Note: If the path to the **sdconf.rec** file is not specified, installation will not proceed.

The pathname entered for the VAR_ACE environment variable is displayed. If the pathname is not correct, it may not be correctly defined in the variable. For information about this setting, see "Add the Web Server to the Authentication Manager Environment" on page 14.

6. When prompted for the path to the Apache servername directory, specify the complete path to the web server, and press ENTER.
7. Verify the complete path to the Apache configuration file and Apache httpd binary file, and press ENTER.
8. When prompted to upgrade the Web Agent, click **Yes**.
   The upgradation will overwrite the old files.
9. When prompted to backup the current rsawebagent to rsawebagent.old, click **Yes**.
   After successful installation of the Web Agent, the configuration script starts automatically. For manual configuration instructions, see "Config Utility" on page 30.
10. Stop the Apache web server.
11. After upgrading to Web Agent 7.1, follow the steps below to convert the node secret from the old format to the new format:
    a. Download the Authentication SDK from **https://knowledge.rsasecurity.com/scolcms/set.aspx?id=8635**. The agent_nsload utility that is used to convert the node secret from the old format to the new format, is available with the Authentication SDK.
    b. Refer to the documentation available with the SDK to find how to use the utility.
    c. Use the utility to convert the node secret to the new format.

d. Copy the node secret to the VAR_ACE location and give required permissions.

12. Restart the Apache web server.

Next Steps

After installing the Web Agent, check if the rpc server and RSALogOffCookie service are running as expected.

To check if the rpc server is running:

1. Start the Apache web server. Change to the **bin** directory of the web server. Type:
   ```
   ./apachectl -k start
   ```

   Important: Ensure that you start portmap before invoking the rpc server.

2. From the **bin** directory of the web server, type:
   ```
   ps -ef | grep rpc
   ```
   The output of the above command is:
   ```
   rpc 4655     1  0 12:39 ? 00:00:00 portmap
   rpcuser 4675     1  0 12:39 ? 00:00:00 rpc.statd
   root 4705     1  0 12:39 ? 0:00:00 rpc.idmapd
   nobody 6798     1  0 13:23 ? 00:00:00 aceapi_rpc_server
   nobody 6804  6798  0 13:23 ? 00:00:00 aceapi_rpc_server
   root 6914  6897  0 13:36 pts/2    00:00:00 grep rpc
   ```
   This command checks if the rpc server is running. If you see aceapi_rpc_server in the output of the ps command, the rpc server is running. Else, restart the Apache web server to start the rpc server.

3. Check if the RSALogOffCookie service is running as expected.

Uninstall the Web Agent

The following procedure explains how to uninstall the Web Agent.

Note: RSA recommends that you stop your web server before uninstalling the Web Agent.

To uninstall the Web Agent:

1. Change to the **rsawebagent** directory.
2. Run the uninstallation script from the **rsawebagent** directory. Type:
   ```
   ./uninstall
   ```

To uninstall the Web Agent (Windows):

1. Click **Start > Settings > Control Panel > Programs and Features**.
2. Scroll down the list of programs installed, and then click **RSA Authentication Agent for Web for Apache**.

3. Click **Uninstall**.

4 Configuring Web Access Authentication Settings

Administer the Web Access Authentication Settings

You administer the web access authentication settings of your web servers using the various utilities. You can quickly add, remove, and view URLs from the protected resource list without having to directly access all of the configuration settings.

Using these utilities you can perform the following tasks :

- "Change Configuration Settings."
- "Manage URLs."
- "Edit Virtual Web Servers."

Important: By default, the Web Agent sets the ownership and permission to all the files and directories it uses. Changing these permissions or ownership properties could create a security hole in the system.

Change Configuration Settings

The initial configuration sets default attribute values in the Web Agent configuration file. Once this configuration is complete, run the configuration script again if you want to make changes to individual virtual servers set up on this web server. For example, you may find that you need a longer cookie expiration time.

To change configuration settings:

1. Run the configuration script in the Web Agent installation directory. Type:

   ```
   ./Config
   ```

   A list of the current web server and any virtual servers you have set up in the web server configuration file appears. For Windows Apache type: **Config.exe**

   Note: If you have not set up a virtual server, only the current web server is displayed in the list. For instructions on adding a virtual server, see "Edit Virtual Web Servers" on page 27.

2. Choose the server you want to configure. You can make changes to the default settings applied to all servers, or you can make changes to an individual server.

For details about the different configuration menus, see "Config Utility" on page 30.

Manage URLs

By default, the Web Agent protects all URLs on the web server on which the Web Agent is installed. The protectURL utility is an interactive menu from which you can protect, remove, or unprotect individual URLs. The protectURL utility is located in the default Web Agent directory. The protectURL utility only accepts relative paths for protecting and unprotecting a URL.

Note: The command line options are applicable only for UNIX.

Type:

```
./protectURL
```

Note: By default, the root "/" is protected. To remove protection for the root, you must use the option "Unprotect a URL." Removing the root using the option "Remove a URL" does not unprotect the root.

You can also manage the protected resource list by importing a list of URLs from a file.

To add URLs to the protected resource list:

Type:

```
./protectURL -a -f listURL
```

where *listURL* is a text file that contains a list of URLs, with one URL per line, that you want to add to the resource list.

Note: Ensure that you save the listURL file under the **rsawebagent** directory. If you save the file in a different location, specify the complete path in the ./protectURL command.

To remove protected URLs from the resource list

- Type:

  ```
  ./protectURL -d -f listURL
  ```

  All of the URLs listed in the file are removed from the protected resource list.

  Important: When you unprotect a URL, all URLs under it are also unprotected.

Administrators can manage the protected resource list using command line-only operations.

- For a list of options and syntax, type:

  ```
  ./protectURL -h
  ```

Edit Virtual Web Servers

You can edit the virtual web servers by performing any of the following tasks:

- "Add a Virtual Server"
- "Remove a Virtual Server"

Add a Virtual Server

To add additional virtual servers to the Web Agent configuration:

1. Run the configuration script with the name of the virtual web server. Type:

   ```
   ./config server.domain.com
   ```

2. Verify that you want to create the new server.
   The Setup menu is displayed.

For details about the different configuration menus, see "Config Utility" on page 30.

You can add as many virtual servers as you want. However, if you want access to protected URLs to function the same way on all virtual web servers, you need to make changes to your default web server rather than individual virtual servers.

Remove a Virtual Server

To remove a virtual server from the Web Agent configuration file:

Use the -d option. Type:

```
./config -d server.domain.com
```

Note: Removing a virtual server from the configuration file does not remove or disable the web server or the Web Agent.

Invalidate Web Access Authentication Cookies Using the Logoff URL

Using the logoff URL, you can set up a link on a web page that automatically invalidates users' web access authentication cookies and prompts users to authenticate.

To set up the logoff URL, add the following URL to a link on your web pages:

```
http://www.server.domain.com/webauthentication?logoff?
referrer=/sample.html
```

where:

- *server* is the name of your server.
- *domain* is the name of your domain.
- *sample*.html is the web page.

Note: In this URL, ***webauthentication*** is the WebID URL. If this URL changes in the configuration script, modify the logoff URL accordingly.

Important: If you do not provide an argument to **referrer=**, users are sent to the root directory on the virtual Web server.

This logoff URL works only if you have not selected the **Use RSA Token for Cross-Site Request Forgery Protection** option in the web access authentication properties sheet. If you select this option, the web page containing the link to the logoff URL sets the RSA token as a query parameter in the logoff URL. The logoff URL is generated dynamically, as shown in this example:

```
http://www.server.domain.com/Webauthentication?logoff?RSArand=<RSA token>&referrer=/sample.html
```

You can retrieve the RSA token from the web access authentication cookie using the RSA Web Authentication API. For more information on how to use the RSA Web Authentication API to add the RSA token in the logoff URL, see the sample programs provided with the Web Agent installer.

Note: If you do not provide an argument to **referrer=** in the logoff URL, users are sent to the root directory on the virtual web server.

Important: If you do not use a relative path to set up the logoff URL, the URL logs off the user and fails to connect to the referrer web site. The user is not prompted to reauthenticate. If you use an absolute path to set up the logoff URL, you must add an auto-redirect script to enforce RSA SecurID authentication. For information about auto-redirect scripts, see the following section, "Enforce RSA SecurID Authentication Using Auto-Redirect Scripts."

To ensure that the copied cookies are not reused after the user performs a WebAgent LogOff(), the logged off cookies are stored in a cookie cache until the cookies expire.This feature is enabled by default.

Note: To disable this feature, set the **RSA_NO_LOGOFF_COOKIE_CHECKING** environment variable.

Enforce RSA SecurID Authentication Using Auto-Redirect Scripts

The Web Agent includes an auto-redirect script that enables you to require users to authenticate before accessing a URL that is not formally protected by RSA SecurID. The URL does not have to be hosted on the same server or be within the same domain as the server on which the Web Agent is installed.

You use the customized redirect URL from the script as the hyperlink to the unprotected site. When a user clicks the HTML link to the URL that you want to protect, the script is invoked, and the user is forced to authenticate before gaining access to the site.

The Perl script included with the Web Agent is a sample script only. To use it, you must first customize it with your own code.

To customize an auto-redirect script:

1. Copy the Perl sample script (**PerlScriptRedirect.pl**) from the **/cgi_scripts** directory of your Web Agent installation, and store it in the web server's **/cgi-bin** directory.
2. Customize the script with your own code.

   Important: RSA strongly recommends that your script contains a list of URLs that users are allowed to access using the redirect URL. Compare the input argument of the script with the list of allowed URLs before any redirect takes place. Any user who attempts to access the redirect hyperlink can see the link definition and can potentially use the redirect script to access the authentication cookie. Implementing a URL comparison list minimizes security risk.

3. Use the customized redirect URL from the script as the hyperlink to the unprotected site.

An example redirect URL looks like this:

```
http://protectedHostname/webauthentication?referrer=/cgi-bin
/PerlScriptRedirect.pl?target=http://unprotectedHostname/new
_application.jsp
```

In this example:

- /webauthentication/ is the virtual Web Agent reference. It ensures that a user attempting to access the unprotected URL is prompted to authenticate.
- /cgi-bin/PerlScriptRedirect.pl is the script that performs the redirect to the input argument.
- http://*unprotectedHostname*/new_application.jsp is the input argument, or unprotected URL.

For more information about customizing auto-redirect scripts, see the instructions included in each script.

Configure the Web Agent for Proxy Servers

To authenticate through a proxy server, change the value of WebID_URL on the remote Agent-protected web server from the default value of /webauthentication to:

```
http://proxyserver.domain.com/xxx/webauthentication
```

where http://*proxyserver.domain.com*/xxx/ points to the root directory of the remote Agent-protected web server.

If SSL is enabled, the WebID URL is:

```
https://proxyserver.domain.com/xxx/webauthentication
```

Note: This change is required only if you are configuring the reverse proxy.

To make the change, run the Web Agent configuration script (**config**) on the remote Agent-protected web server. The **config** script is in the Web Agent installation directory. The WebID URL option is in the Setup menu of the configuration program.

Phantom Entries

Phantom Entries are the directory or file paths that have been marked as protected or unprotected in the configuration, but they no longer exist in the system.

Phantom entries should be removed in RSAWebagent.ini before protecting a site. The phantom entries can be identified by restarting the Apache web server.

Checking for phantom entries will only work, if the specified locations in the protected/unprotected list, are present in the document root of the Apache server. For example, the check for Phantom entries will always fail if alias locations are present in the protected/unprotected list.

Note: To disable the phantom entries check:
For Windows:
under HKLM\SOFTWARE\SDTI\RSAWebAgent create a DWORD Value 'RSA_CHECK_PHANTOM_ENTRIES' and set its value to 0.
For Unix:
set the 'RSA_CHECK_PHANTOM_ENTRIES' environment variable value to 0.

Config Utility

The configuration program is grouped in the following menus:

Menu	Reference
Setup Menu	For information, see "Setup Menu" on page 30
Configuration Menu	For information, see "Configuration Menu" on page 32
Domain and Multiple Domain Menu	For information, see "Domain and Multiple Domain Menus" on page 35

Setup Menu

The Setup menu configures how the Web Agent interacts with the browser. It includes:

- Adjusting cookie validity time
- Changing the SSL port number
- Changing the WebID URL

- Changing the location of the templates
- Changing the character set

The Setup menu displays automatically after a successful installation. To accept the defaults, press ENTER. Otherwise, type the line number of the option you want to change.

The following table lists the Setup Menu options.

Line Option	Description
1. Expiration time for idle cookie in minutes	Time in minutes for which an idle cookie is valid. When the cookie expires, the user must reauthenticate. Setting a value that is greater than the cookie expiration value deactivates this feature.
2. Expiration time for cookie in minutes	Time in minutes for which an active cookie is valid. When the cookie expires, the user must reauthenticate to get a new cookie.
3. SSL port number to be used	SSL port number to be used for secure data transfer. **Note:** Unlike the UNIX agent, where you choose the SSL port numbers from a list, for the Windows agent, you must know the SSL port numbers that are enabled with the web server.
4. WebID URL/URI	Accept the default name, unless you have an existing URL with the same name.
5. Directory for web authentication Templates	Accept the default. After the initial installation and configuration, you may customize the templates. Once you do so, run the configuration script again to designate the new location of your customized templates.
6. Characterset [UTF-8]	Specify the web site character set to set this configuration item. By default, this setting is inherited from the Web Agent level default setting specified during installation. If you want to override the Web Agent level setting, you can configure it at the web site level. For example, if your web site is designed to support the UTF-8 character set, you must set the web site level character setting to UTF-8. If you do not specify the character settings correctly, the web site might not function as expected and data might get corrupted. You can use the following command to list the character sets supported on your machine: `iconv --list` From this list, choose the character set used by your web site.

Configuration Menu

The Configuration menu configures access to protected URLs. It includes:

- Redirecting URLs to secure ports
- Using separate pages for user name and passcode
- Using the name locking feature

The Configuration menu appears automatically after you complete the Setup menu. To accept the defaults, press ENTER. Otherwise, type the line number of the option you want to change.

The following table lists the Configuration Menu options.

Line Option	Description
1. Agent protection of this web server	Accept the default. **Note:** Disable the Web Agent only when it is absolutely necessary to temporarily halt protection of all URLs on this web server for troubleshooting purposes. When the Web Agent is disabled, your data is unprotected.
2. Use RSA Authentication Manager name locking feature	Name locking protects against the risk that an unauthorized person might observe a user entering the passcode and submit the same passcode on a different agent host in the realm more quickly than the original user. With name lock, the agent host sends the user's logon name and passcode to the Authentication Manager separately. If someone attempts to use the same user name and passcode, the Authentication Manager refuses the authentication request. Name locking is not needed for most customers. Name locking has no effect when the Web Agent is configured to authenticate in conjunction with RSA Authentication Manager 7.1. Name locking must be enabled for the agent host on Authentication Manager 6.1 to gain any benefit from the feature. **Note:** The name locking feature offers security tradeoffs that may or may not be appropriate for your environment. By enabling name locking, a 30-second lock is created on RSA Authentication Manager. As with any lockout mechanism, this can be used to prevent a valid user from authenticating by continually relocking the valid user name.

Line Option	Description
3. Use separate user name and PASSCODE pages	The Web Agent uses separate HTML or WML pages to request the user name and passcode. If you disable this feature, the user name and passcode are sent across the Internet together. **Note:** Displaying the user name and passcode prompts as separate pages is necessary to fully use the security offered by name locking. But name locking comes with security tradeoffs that may or may not be appropriate for your environment. When the prompts are separated onto different pages, the Web Agent creates new sessions while submitting the user names. As with most session management systems, this creates the possibility that all sessions will be reserved, and new authentication attempts will be rejected until old sessions complete.
4. Require secure connection to access protected resource	The Web Agent connects to protected URLs through an SSL port. If you disable this feature, data transmitted over the Internet is unprotected, meaning cookies can be seen in plain text. **Note:** If you do not have an SSL connection, you must disable this feature.
5. Redirect to SSL when accessing protected resource	When a user attempts to access a protected URL through HTTP, the Web Agent redirects the user to a page where the user can log on. After successful logon, the user is redirected to the authentication page through HTTPS. **Note:** This option does not appear if you disable option 4 (Require SSL Connection).
6. Prevent caching of protected pages on clients	The Web Agent prevents the browser from caching protected pages on the local machine. If you disable this feature, protected pages may be cached on the local hard drive.
7. Auto-Submit (avoid having to click Continue after successful authentication)	After the user enters authentication information on the web page, the Web Agent automatically redirects the user to the requested page without having to click **CONTINUE**.

Line Option	Description
8. Use JavaScript pop-up window to authenticate in frames	If the protected web site uses HTML frames, sometimes the passcode prompt is too small to read clearly. To avoid this problem, display the passcode prompt in a JavaScript pop-up window. **Note:** This feature will not work if cross frame scripting is prevented by WebAgent, which is a security feature enabled by default. If the user wants to use the 'JavaScript popup' feature cross frame busting has to be disabled by setting the environment variable RSA_NO_FRAME_BUSTING=1 in the WebAgent machine. In general, it is recommended to protect the main page, instead of protecting individual frames in the page.
9. Ignore browser IP address for cookie validation	By default, this feature is disabled so that the Web Agent uses the browser IP address to sign the cookie. However, if there is a proxy or a firewall between the browser and the Web Agent, the IP address used may be the same. If you have web sites that are accessed through load balanced proxy servers, which means that the browser IP addresses may change, you may want to enable this feature. Otherwise, the user may have to authenticate quite frequently.
10. Cookie valid for the current domain	Once a user is authenticated, the user can access URLs on any of the web servers in the current protected domain. If you disable this feature, the user is asked to authenticate each time a protected URL is accessed on a different web server.
11. Cookie valid across multiple domains	Once a user is authenticated, the user can access URLs on any web server in the multiple domain list. If you disable this feature, the user is asked to authenticate each time a protected URL is accessed on a web server that is outside the current domain.

Line Option	Description
12. Use RSA Token for Cross-Site Request Forgery Protection	Enabling this option protects RSA SecurID Authentication web pages from cross-site request forgery attacks. This feature works by adding a random number, referred to as RSA token, as a hidden parameter in the forms and pages, which are based on templates provided by RSA. The RSA Web Authentication API provides functions to get the RSA token from the web access authentication cookie. A request is allowed only if the RSA token is found to be valid, as verified by the Web Agent. For the logoff URL, the web page containing the link to the RSA logoff URL uses this API to retrieve the RSA token and set it in a hidden field. This token is sent along with the logoff request. If this option is enabled, the Web Agent verifies the RSA token and accepts the request only if the token in the request is valid. To learn more about how to use the RSA Web Authentication API to add the RSA token in the logout URL, refer to the sample programs provided with the Web Agent installer. The Web Agent also sets a pre-logon cookie containing an RSA token in all the RSA web pages, such as the Logon page and the New PIN page, which is verified when you submit these pages.

Note: After making changes from the Configuration menu, if the Apache web server is running, you will be prompted to reload the **httpd.conf** file for the changes to take effect.

Domain and Multiple Domain Menus

The Domain and Multiple Domain Menu configures the domain (one domain in case of Current Domain Access and multiple domains in case of Multiple Domain Access) for which an authentication cookie is valid, and generates a new domain secret for use on other Web Agents.

If you have enabled line option number 10 (Current Domain Access) or 11 (Multiple Domain Access) in the Configuration menu, the Domain and Multiple Domain Configuration menus are displayed automatically.

The following table describes the Domain and Multiple Domain Configuration menu options.

Line Option	Description
1. Generate new domain secret for this server	A domain secret was automatically generated when you installed the Web Agent. Use this option to generate a new domain secret.

Line Option	Description
2. Generate and export new domain secret	If you have multiple web servers on which users will be able to access protected URLs, each web server within the domain must have the same domain secret. Use this option to generate and export the domain secret to a file so that you can import it to all other web servers at your site that will issue and accept domain cookies. You must name and create a password for the export file. The file is then stored in the Web Agent directory (the default directory is **rsawebagent**).
3. Import domain secret from another server	If you are configuring protected URL access in a domain environment, use this option to import the domain secret from other Agent-protected web servers. You are asked for the filename and file password that you set up in option 2 (Generate and Export Domain Secret).
Current Domain Options	**The following options appear only if you chose number 10 (Current Domain Access) in the Configuration menu.**
4. Domain name	Use this option to create subdomains. For example, suppose you have http://server1.domain1.domain.com http://server2.domain1.domain.com http://server3.domain2.domain.com http://server4.domain2.domain.com and you want to protect URLs on all of these servers. By entering **domain.com** as the Domain Name, you create a subdomain which includes all of the preceding web servers . You must enter a domain name.
5. Name of the cookie	Use this option to change the default cookie name (rsacookie). Maximum name length is 30 characters.
Multiple Domain Options	**The following options appear only if you chose number 11 (Multiple Domain Access) in the Configuration menu.** **Note:** All web servers protected using the Multiple Domain Option should be configured to use only SSL (not plain http).
6. Add Agent-protected web server to multiple domain list	Enter the Agent-protected web servers on which you want all users to access protected URLs once they have authenticated. Use the format http://*server1.domain1.com*. You must enter a domain name.

Line Option	Description
7. Remove Agent-protected web server from multiple domain list	The multiple domain list of Agent-protected web servers displays. Choose the number of the web server you want to remove from the list. (This option does not appear if there are no hosts in the multiple domain list.)
8. View the list of Agent-protected web servers in the multiple domain list	View the list of Agent-protected web servers you entered with option 6 for the multiple domain list. (This option does not appear if there are no web servers in the multiple domain list.)

CAUTION: If you have separate web servers that authenticate users to separate Authentication Manager databases, specify different domain secrets for the different domain cookies. Otherwise, users might gain unauthorized access to protected URLs.

Note: Ensure that you test the multiple server and multiple domain authentication features from the client machine and not from the machine on which the Web Agent is installed. For multiple domain authentication to work, you need to allow access to third-party cookies in the web browser.

After configuring the Web Agent for the first time following installation, the product registration web page displays. If you choose not to register now, you can access the page at your convenience, or you can run the registration script (**./registerWA**) from the Web Agent installation directory.

Important: RSA recommends that you register the software to ensure that you receive security patches as they become available.

WebAgent Logging

You can use the logging or tracing option to troubleshoot WebAgent problems. To enable logging, you must select one or more tracing level settings and one or more tracing destinations.

To enable event logging in Windows:

1. Under **HKLM\SOFTWARE**, create a registry key SDTI.
2. Under **SDTI**, create a registry key **ACECLIENT**.
3. Create a DWORD value TraceLevel with data as listed in the following table.

Value	Description
0x00000000	Disables logging
0x00000001	Logs regular messages
0x00000002	Logs function entry points
0x00000004	Logs function exit points
0x00000008	All logic flow controls use this (ifs)

Note: For combinations, add the corresponding values. For example, to log all types of messages, you must set the TraceLevel value to 0x0000000 for decimal 15.

4. Create a DWORD Value TraceDest under ACECLIENT, with data as listed in the following table.

Location	Value	Description
Event Logger	0x00000001	Logs are sent to the event log
Console	0x00000002	Logs are sent to the console
Log File	0x00000004	Logs are saved in log files
Debugger	0x00000008	Logs are saved as debugger output
No File Line Info	0x80000000	No file and line information saved

Note: For combinations, add the corresponding values. For example, to save the logs in log files as well as the debugger, you must set the value to 0x0000000c or decimal 12.

5. Create a string value TraceFile and provide the required details. By default, the string value goes to %WINDIR%ACECLIENT.LOG.

Note: The string value TraceFile is required only if you have selected Log File in the TraceDest value.

To enable logging in LINUX, SOLARIS and AIX

To enable logging, you must set the following environment (ENV) as below.

- export RSATRACELEVEL=<value>

Note: For combinations, add the corresponding values. For example, to log regular messages and function entry points, set the value to 3.

- export RSATRACEDEST=<filepath>

Specify the file path where the logs must be redirected. If you do not set this environment variable, by default the logs go to standard error.

5 Customizing Templates and Message Strings

Customized Templates

When users authenticate successfully to the Web Agent using a standard browser, the system returns a message informing them about the success of the authentication attempt through an HTML page. For wireless device micro browsers, the system returns messages in WML format.

The Web Agent provides default versions of HTML and WML templates and messages that you can customize to reflect your company's image and administrative needs. You can:

- Add a custom greeting message.
- Add your own custom graphics.
- Change standard buttons to custom graphics.
- Display web access authentication prompts in a language other than English.
- Customize the web access authentication messages.

The WebAgent templates can be classified into HTML templates and WML templates.

- **HTML Templates**
  - **Manual Authentication templates**: These templates are displayed when a user accesses the protected page from a client machine which does not have a RSA SecurID Software Token installed in it. The user has to manually enter the username and passcode.
  - **WebID Plugin templates**: These templates provide compatibility with the RSA SecurID Software Token for Windows WebID components. These components integrate the RSA SecurID software token application with Internet Explorer and Mozilla Firefox on Windows. When users navigate to a site that is protected by the RSA Authentication Agent for Web, the Web ID authentication page is displayed, which allows them to select their software token and authenticate with their user name and pin. They do not have to enter their user name and passcode, as is required in the manual authentication page.

- **WML templates**: If the user accesses the protected resource using wireless device micro browsers, the system returns the authentication pages in WML format.

Default Templates

The following table describes the default templates.

Note: If you are using RSA SecurID PINPads instead of tokens, you need to change the **passcode** and/or **useridandpasscode** templates to display the correct message to your users. The correct message to display is included in the templates in a comment section.

Template	Description
Errors	
error.htm **error.wml**	The page that RSA SecurID users see when a fatal error occurs during authentication. The **@@sub** macro in the template substitutes the error message passed from the system or from the **strings.txt** file.
forbidden.htm **forbidden.wml**	The page that RSA SecurID users see in response to requesting a forbidden URL.
Authentication Templates	
newpin.htm **newpin.wml**	The New PIN page is displayed when users are in the new pin mode or are authenticating with their token for the first time. From this page, users create their own PINs. This then loads either the WebID NewPin page (newpinplugin.htm) or the manual authentication page (newpinmanual.htm)
newpinplugin.htm	This page is displayed if the user's computer has the Software Token WebID plug-in installed. The user enters the PIN to authenticate.
newpinmanual.htm	This page is displayed if the user's computer does not have the Software Token WebID plug-in installed. The user enters passcode.
newpin1.htm **newpin1.wml**	This is the landing page to receive a system-generated PIN, which then loads either the WebID page (newpin1plugin.htm) or the manual authentication page (newpin1manual.htm). This functionality is determined in Authentication Manager.
newpin1plugin.htm	This page is if the user's computer has the Software Token WebID plug-in installed.
newpin1manual.htm	This page is displayed if the user's computer does not have the Software Token WebID plug-in installed.

Template	Description
newpin2.htm **newpin2.wml**	The New PIN page is displayed when a user is given the choice of whether to create a PIN or receive a system-generated PIN. This page then loads either the WebID page (newpin2plugin.htm) or the manual authentication page (newpin2manual.htm). This functionality is determined in Authentication Manager.
newpin2plugin.htm	This page is displayed if the user's computer has the Software Token WebID plug-in installed.
newpin2manual.htm	This page is displayed if the user's computer does not have the Software Token WebID plug-in installed.
nextprn.htm **nextprn.wml**	The page is displayed when a token is in Next Tokencode mode. This happens when a user enters a series of incorrect passcodes during authentication. After the user finally enters a correct tokencode, the user is prompted for another correct tokencode before being allowed access. This then loads either the WebID page (nextprnplugin.htm) or the manual authentication page (nextprnmanual.htm).
nextprnplugin.htm	This page is loaded if the user's computer has the Software Token WebID plug-in installed.
nextprnmanual.htm	This page is loaded if the user's computer does not have the Software Token WebID plug-in installed.
sslredir.htm **sslredir.wml** **sslredir-post.htm**	The page users might see momentarily with some browsers when they must use a secure channel to access protected pages. In some cases, users must click a link on the **sslredir** or **sslredir-post.htm** page to continue.
redirect.htm **/redirect-get.htm** **redirect.wml**	The page is displayed when users complete the authorization process or when they log off. **Note:** If you customize **redirect.htm**, you must customize **redirect-get.htm** to look the same.
redirectmanual.wml	This page is displayed to cell phone users when the cell phone does not support automatic redirection to a protected URL. The user is provided with a list of secure URLs and must manually choose one.
cancel.htm/cancel-get.htm **cancel.wml**	The page is displayed to users when they cancel the authorization process. **Note:** If you customize **cancel.htm**, you must customize **cancel-get.htm** to look the same.

Template	Description
showsys.htm **showsys.wml**	The page is displayed to users for ten seconds while the system generates an RSA SecurID PIN for them.
multidom.htm/ **multidom-get.htm** **multidom.wml**	The page is displayed when users are authenticating across multiple domains. **Note:** If you customize **multidom.htm**, you must customize **multidom-get.htm** to look the same.
userid.htm **userid.wml**	If you chose to present separate web pages to users to input the user name and passcode, this template is used for the user name. If you did not choose to present separate pages, the **useridandpasscode** template is used. This page then loads either the WebID authentication page (useridplugin.htm) or the manual authentication page (useridmanual.htm)
useridplugin.htm	This page is loaded if the user's computer has the Software Token WebID plug-in installed. The user enters the PIN to authenticate.
useridmanual.htm	This page is loaded if the user's computer does not have the Software Token WebID plug-in installed. The user enters passcode.
passcode.htm **passcode.wml**	If you chose to present separate web pages to users to input the user name and passcode, this template is used for the passcode. If you did not choose to present separate pages, the **useridandpasscode** template is used. This then loads either the WebID authentication page (passcodeplugin.htm) or the manual authentication page (passcodemanual.htm).
passcodeplugin.htm	This page is loaded if the user's computer has the Software Token WebID plug-in installed. The user enters the PIN to authenticate.
passcodemanual.htm	This page is loaded if the user's computer does not have the Software Token WebID plug-in installed. The user enters passcode.
useridandpasscode.htm **useridandpasscode.wml**	If you chose to present one web page to users to input both the user name and passcode, this template is used. If you chose to present separate web pages to input the user name and passcode, the userid and passcode templates are used. This then loads either the WebID authentication page (useridandpasscodeplugin.htm) or the manual authentication page (useridandpasscodemanual.htm).

Template	Description
useridandpasscodeplugin.htm	This page is loaded if the user's computer has the Software Token WebID plug-in installed. The user enters the PIN to authenticate.
useridandpasscodemanual.htm	This page is loaded if the user's computer does not have the Software Token WebID plug-in installed. The user enters passcode.

The HTML and WML templates use the following files, which are also installed in the **/Templates** directory.

Template	Description
Bitmaps	
denied.jpg **denied.wbmp**	If you have configured the Web Agent to allow multiple domain authentications, the word "Denied" displays if a user's authentication request to a virtual web server does not succeed.
ok.jpg **ok.wbmp**	If you have configured the Web Agent to allow multiple domain authentications, the word "OK" displays if a user's authentication request to a virtual web server succeeds.
rsalogo.jpg	This is the background graphic used on the authentication pages.
securid_banner.jpg	This graphic displays the RSA SecurID banner on the authentication pages.
Other Files	
strings.txt	This file contains text strings that display various messages while users interact with the web access authentication prompt pages.
style.css	The cascading style sheet used for the web pages.

Templates Management

During the Web Agent installation, the default templates are copied into the **/Templates** directory of your Web Agent installation. If you decide to use customized templates, you must store them in a different directory.

To access the templates and text strings, log on as a web server user as defined in the web server configuration file. To specify the location of a virtual server's customized templates, run the Web Agent Setup configuration script. For instructions on using the Setup menu, see "Setup Menu" on page 30.

Guidelines for Using Templates

To ensure that the templates function properly after you have made changes, follow these guidelines:

- Copy the templates into a new directory before making changes to them.
- Use a text editor to make changes.

Note: HTML editors add unnecessary additional HTML/WML tags to templates and may alter the substitution strings that are necessary in the templates.

- After you have completed your changes, test the templates to make sure that they are functioning properly. For information on the utilities that you can use to troubleshoot problems, see "Troubleshooting" on page 53.
- Ensure that the owner of the templates directory is "daemon" with write privileges. The web server may not be able to read the templates if you change the privileges.
- Do not alter any of the substitution strings in the templates or message text files (**webagent.msg** and **strings.txt**).

  Substitution strings are used to include error messages and text from Authentication Manager and provide placeholders for graphics and message strings.These strings begin with two "at" signs (@@).

Modify Static Text

You can change the static text in the default templates, or you can add your own static text.

To modify the text in a default template:

1. Using a text editor, open one of the templates in the directory. The templates are listed in "Templates Management" on page 47.

   Important: When editing templates, avoid altering the contents of substitution strings. These strings begin with two "at" signs (@@).

2. Delete the static text you want to change, and add the new text.

   For example, the tag **<H1>Welcome to ABC, Inc.</H1>**, when placed in the **passcode.htm** or **passcode.wml** file, changes the text of the first heading in that page from "RSA SecurID Passcode Request" to "Welcome to ABC, Inc."

3. Save and close the file.

Add Custom Graphics

You can add one or more custom graphics to the default templates.

Note: WAP or WML devices usually have limited display space for graphics. Be sure the use of graphics is appropriate for your WAP devices before using them.

To add a custom graphic to a default template:

1. Using a text editor, open one of the templates in the directory. The templates are listed in "Templates Management" on page 47.
2. Decide where you want the image to be placed on the page, and then insert the appropriate tag in the HTML or WML markup pointing to the image file. Use one of the following methods for naming graphic files:

- A substitution macro (@@URL?GetPic?image=) works with HTML and WML. With HTML, the image types must be .jpg. With WML, the image types must be .wbmp. Substitution macros cannot have absolute paths. The images must be in the same directory as the templates, and you must omit the filename extension from the file specification, as in the following example:

  ```
  <IMG src="@@URL?GetPic?image=logo" ALIGN="left">
  ```

- You can use HTTP URLs instead of substitutions if the image files reside in an area of the server that is unprotected by RSA SecurID authentication, or on a separate server hosting the URL. HTTP URLs are always absolute. Relative URLs cannot be used in templates. The image types for HTTP URLs can be .jpg, .gif, or .wbmp. For example:

  ```
  <IMG src="http://server.domain.com/img/logo.jpg"
  ALIGN="left">
  ```

Note: When using HTTP URLs, ensure that the image file you point to in the **src** path is in a directory that is not protected by RSA SecurID and that you always specify a fully qualified path to the image file.

3. Save and close the file.
4. Stop and restart the web server for the changes to take effect.

   The web authentication prompt displays the new graphic.

Changing the Buttons (HTML Only)

You can replace the standard **Reset**, and **Cancel** buttons that are displayed in the HTML templates with custom graphics.

Note: Make sure that the image file you point to in the **src** path is in a directory that is not protected by RSA SecurID and that you always specify a fully qualified path to the image file.

To change the buttons in a default template:

1. Using a text editor, open one of the HTML templates in the directory. The templates are listed in "Default Templates" on page 43.
2. Do one or all of the following:
   - To replace the **Reset** button, replace the line

     ```
     <INPUT TYPE=RESET VALUE="Reset">
     ```

     with

     ```
     <A HREF="JavaScript:document.forms[0].reset()"><IMG
     SRC="path to your image" BORDER="0"></A>
     ```

     where *path to your image* is a fully qualified path to an image file.
   - To replace the **Cancel** button, replace the line

     ```
     <INPUT TYPE=CANCEL VALUE="Cancel">
     ```

     with

```
<A HREF="JavaScript:document.forms[0].cancel()"><IMG
SRC="path to your image" BORDER="0"></A>
```

where *path to your image* is a fully qualified path to an image file.

3. Save and close the file.
4. Stop and restart the web server for the changes to take effect.

Customize Templates for Another Language

To customize the templates for a language other than English:

1. Set the browser language preference to use the appropriate language code.

   The code must correspond to your language-customized template directory name. The new language preference must appear at the top of the web browser's list of language preferences.

   Note: If the preference settings are incorrect, language-customized templates do not exist, or the Web Agent cannot find the specified templates for a virtual web server, the browser displays the default English version of the templates.

2. Store the templates in a language-specific directory under the Web Agent **/Templates** directory.

   The default directory for language-specific templates is */Web_Agent_installation_directory*/**Templates/nls/<***language_code***>** where *language_code* is the language preference code used by web browsers.

   Note: To find the correct language code, see the language preferences list of codes in the Internet Explorer or Firefox web browser. For more information about using international character sets in HTML documents, consult an HTML reference book, or go to **www.w3.org/pub/WWW/International**.

To translate HTML and WML Templates for a non-English language:

1. Create a language-specific subdirectory in the templates directory of the Web Agent.

   For example:

   ```
   ./web_server_directory/rsawebagent/Templates/nls/fr
   ```

   where fr is the language preference code for French.

2. Copy the templates to the directory that you created in step 1.
3. Customize the text strings within the templates.

   Note: Do not remove the substitution macros. These macros begin with @@. The macros are replaced with actual values when the text is displayed.

4. Run the Web Agent configuration script, so that the **Template** path in the Setup menu points to the language specific templates.

Note: The character encoding of the language being customized should be the same as the character set configured for the Web Agent. After editing the template files, you must save them using the same configured character encoding. Otherwise, the templates will not work properly in the Web Agent authentication pages. For more information, see "Pre-Installation Tasks" on page 12.

Customize Message Strings in Templates

You can customize certain messages that are displayed while users interact with the web access authentication prompt pages that are produced from the templates. The message strings are contained in a file named **strings.txt** located in the */Web_Agent_installation_directory*/**Templates** directory.

For example, **strings.txt** contains passcode page errors like:

```
[Messages]
; PASSCODE page errors and messages.
1="100: Access denied. The RSA Authentication Manager
rejected the PASSCODE you supplied. Please try again."
2="101: Access denied. Unexpected RSA Authentication
AgentError %d. Please try again."
3="102: You must enter a valid PASSCODE. Please try again."
```

Important: If you modify the message strings, make certain that you do not remove or alter the position of the variable strings (**@@SUB1**, **@@SUB2**, and so on) contained in the message text. The strings are replaced by actual values when the messages are displayed.

To customize the text displayed by the **multidom.htm** or **multidom.wml** template, search for the following section in the **strings.txt** file:

```
; multiple domain authentication string
; This is HTML only
22="<strong>Requesting authentication from server
@@SUB1</strong> <img alt=failed src="@@SUB2"><br>"
; This is for WML with image tag support
23="<strong>Server @@SUB1 <img alt="failed"
src="@@SUB2"/></strong><br/>"
```

Note: If you translate the text messages in **strings.txt** into a language other than English, you must store the translated file in the same language-specific directory where other translated templates are stored. For more information, see "Customize Templates for Another Language" on page 50.

The character encoding of the language being customized should be the same as the character set configured for the Web Agent. After editing the **strings.txt** file, you must save it using the same configured character encoding. Otherwise, the templates will not work properly in the Web Agent authentication pages. For more information, see "Pre-Installation Tasks" on page 12.

6 Troubleshooting

RSA Authentication Manager Utilities

Use the following utilities to determine communication between the Web Agent and the Authentication Manager.

These utilities reside in the Web Agent directory (/***web_server_directory***/**rsawebagent** is the default).

Unix Utilities

acestatus

This utility provides information about the Authentication Manager, such as the configuration version, the server name and address, the number of client retries, and the client time-out period. This utility also provides information about the replica instance, if it is set up.

acetest

This utility enables you to authenticate to the Authentication Manager from the command line rather than going through authentication web pages in your browser. This helps you determine whether a problem lies with the templates or with the authentication process.

Important: Make sure that you run **acetest** as the user who owns the web server. Otherwise, ownership for the files under the $VAR_ACE environment variable may change and cause RSA SecurID authentication to fail.

Windows Utility

sdtest

The sdtest utility provides information about RSA Authentication Manager, such as the configuration version, the server name and address, the number of client retries, and the client time-out period. In addition, this utility allows you to test authentication with the Authentication Manager.

Character Set Issues

If the character setting that you configure for a web site and the default character set of the web site do not match, data loss or data corruption might occur. If data loss occurs, you must reconfigure the web site level character settings to match those of the web site. For more information, see "Pre-Installation Tasks" on page 12.

Log Authentication Attempts

Authentication attempts are logged in **/*web_server_directory*/logs/error_log**.

Note: The different types of error messages logged can be found in the **webagent.msg** file located in the Web Agent directory (**/*web_server_directory*/rsawebagent** is the default).

The following table lists error messages, their causes, and possible solutions.

Error Message	Possible Cause and Solution
File /usr/local/web_server_directory/conf/*file*.conf isn't writable. (Unix)	The user account with which you logged on does not have write permissions. Log on with a web server user account that has write permissions to the web server root directory.
File C:\Program Files\web_server_directory\conf/*file*.conf isn't writable.(Windows)	The user account with which you logged on does not have write permissions. Log on with a web server user account that has write permissions to the web server root directory.

Error Message	Possible Cause and Solution
100:Access denied. The RSA Authentication Manager rejected the passcode you supplied. Please try again.	The first time an authentication occurs after the Web Agent has been installed on the web server, the Authentication Manager generates a node secret and sends it to the web server. If the node secret file is missing, or the node secret on the Authentication Manager and the web server do not match, users are denied access. Contact your Authentication Manager administrator. If the problem persists, verify that the hostname of the Web Agent resolves to the same IP address throughout the network. Contact your network administrator for assistance. If you clear the node secret on the Web Agent and Authentication Manager, you are denied access. Restart the Apache web server to resolve this issue. Frames are not supported if you enable the option **Use RSA Token for Cross-Site Request Forgery Protection**. If you want to use frames, and you have enabled the **Use RSA Token for Cross-Site Request Forgery Protection** option, you must also enable **Use JavaScript pop-up window to authenticate in frames**.
Unexpected RSA Authentication Agent error 103. Please try again.	This error is received when there are network problems. Contact your Authentication Manager administrator.

Error Message	Possible Cause and Solution

AceInitialize Failed during acetest authentication. (Unix)	• The **sdconf.rec** file is missing. Obtain an **sdconf.rec** file from your Authentication Manager administrator. Place the file in a directory that is accessible to the web server and Web Agent software. Provide the necessary permissions to access the **sdconf.rec** file. Type:
	`chmod 755 sdconf.rec`
	Restart the web server.
	• Verify that 5500 UDP traffic is not blocked. If it is blocked, the Web Agent does not have a valid route to RSA Authentication Agent.
	• Verify that the Authentication Manager is running.
	• The **sdopts.rec** file is missing. Create the **sdopts.rec** file in the same directory as **sdconf.rec**, and specify the IP address override in the **sdopts.rec** file. Provide the necessary permissions to access the **sdopts.rec** file. Type:
	`chmod 755 sdopts.rec`
	• The **/var/ace** directory does not have the correct access permissions. Change the owner of the **/var/ace** directory to daemon. Provide the necessary permissions to access the **/var/ace** directory. Type:
	`chmod 755 /var/ace`

AceInitialize Failed during acetest authentication. (Windows)	The **sdconf.rec** file is missing. Obtain an **sdconf.rec** file from your Authentication Manager administrator. Place the file in a directory that is accessible to the web server and Web Agent software. Restart the web server.
The page cannot be found.	The requested page may not be present.
RSA Securid Error. 106: Web server too busy. Please try again later.	This error may occur when communication to the Authentication Manager is down or the **sdconf.rec** file is missing. Contact your Authentication Manager administrator.
Unexpected authentication error.	This error may occur when authenticating using the **acetest** utility. Communication to the Authentication Manager is down. Contact your Authentication Manager administrator.

Error Message	Possible Cause and Solution
The Page cannot be displayed.	• Communication to the web server is down. • The web server was started without SSL. Therefore, the Redirect Secure feature in the Web Agent is disabled. The best solution is to restart the web server with SSL. You could also have users access the page with an **https** request.
RSA Web Access Authentication Extension Error. RSA Web Access Authentication: Internal server configuration error.	The path to the templates is invalid. Verify the correct path in the Web Agent configuration.
For Multi-Domain Authentication: Requesting authentication from server http://server Denied.	Make sure that the same domain secret exists on each web server within the multiple domain area.

Error Messages

The Web Agent logs events in the web server error log file.

This section lists all error and event messages alphabetically.

ACECheck processing error for userid *username*

If the **ACECheck** function returns an error, an Authentication Manager time-out or some other communications error has occurred.

ACEClose processing error *errornumber*

If the **ACEClose** function returns an error, an Authentication Manager time-out or some other communications error has occurred.

ACENext processing error for userid *username*

If the **ACENext** function returns an error, an Authentication Manager time-out or some other communications error has occurred.

ACEPin processing error for userid *username*

If the **ACEPin** function returns an error, an Authentication Manager time-out or some other communications error has occurred.

Authentication Manager: Access Denied.

The user did not enter a valid RSA SecurID passcode.

Authentication Manager: Invalid Authentication Manager configuration. User *username*.

The **sdconf.rec** file is not valid. The file is either corrupted, has been moved to another directory, or has been deleted from the system.

To correct the problem, get a new copy of **sdconf.rec** from your Authentication Manager administrator.

Authentication Manager: New PIN Accepted. User *username*.

The user successfully associated a new PIN with his or her token.

Authentication Manager: New PIN Rejected. User *username*.

The user did not successfully associate a new PIN with his or her token. If the user is attempting to create his or her own PIN, make sure that the user understands the PIN length and syntax parameter settings for your Authentication Manager.

Authentication Manager: Next Tokencode Accepted. User *username*.

After entering a series of bad passcodes, the user was prompted to enter the next tokencode from his or her token. The next tokencode was valid and the user was authenticated successfully.

Authentication Manager: User Canceled New PIN Mode. User *username*.

The user was prompted to associate a new PIN with his or her token, but the user did not complete the new PIN procedure. Make sure that the user understands how to use his or her token in New PIN mode.

Authentication Manager: User Canceled Transaction. User *username*.

The user was prompted to authenticate, but then canceled out of the Enter passcode dialog box. This is a purely informational message.

Authentication Manager: User I/O Timeout. User *username*.

Because the user waited too long at the **Enter passcode** prompt, the Authentication Agent canceled the transaction.

Cookie rejected. Cached client info does not match.

If a user is using more than one workstation, this message appears each time the user switches from one workstation to another.

Cookie rejected. Cookie failed MD5 test.

An unauthorized user has attempted to access the web server with an invalid web access authentication cookie.

Cookie rejected. Expired cookie. Username *username*

A web access authentication cookie has expired in response to the time-out values defined in the web properties sheet.

Could not initialize Authentication Agent

Will be preceded by a number of Authentication Agent error messages, such as "Cannot find sdconf.rec." Try reinstalling the **sdconf.rec** file.

Could not initialize Cookie Cache

A memory error has occurred within an internal function. Your web server may be overloaded; you may need more physical memory.

Could not open HTML template *filename*

The HTML template file is missing.

Also check the security settings for the file. Make sure the account that the web server is running has Full Access privileges to the HTML file.

Could not query value *valuename*

If you have enabled the Domain Cookies feature without setting a domain secret, you might get a **valuename DomainData** message, followed by a **Domain cookies are disabled** message.

Could not read HTML template *filename*

The HTML template file is missing.

Could not resolve hostname *hostname*

The DNS function of the web server is configured incorrectly. Domain cookies cannot be used until the configuration is corrected.

Failed authentication for userid *username*.

The Authentication Manager did not grant the user access.The most common causes for this are an incorrect user name or an invalid passcode.

Failed to create service thread, aborting.

There were too many other processes running, so the service did not start.

File incorrect size: sdconf.rec.

It is likely that the **sdconf.rec** file was not copied in binary or ftp mode. Ask the Authentication Manager administrator for a new copy of **sdconf.rec**.

File not found: sdconf.rec.

The **sdconf.rec** file was either removed or never copied from the Authentication Manager. Ask the Authentication Manager administrator for a new copy of **sdconf.rec**.

New PIN accepted for userid *username*.

The Authentication Manager verified the RSA SecurID user's new PIN.

New PIN rejected for userid *username*.

The PIN was rejected by the Authentication Manager. The user must reauthenticate to set the PIN. Check the Activity Log on the Authentication Manager.

New PIN requested from userid *username*.

The Authentication Manager has prompted the RSA SecurID user to create his or her own PIN or receive a system-generated PIN.

Next code accepted for userid *username*.

The Next Tokencode was accepted by the Authentication Manager and access was granted.

Next code rejected for userid *username*.

The user must reauthenticate.

Next code requested from userid *username*.

The user's token was in Next Tokencode mode and the Authentication Manager asked for the second tokencode.

No cookie or corrupted information.

This message appears each time a new user logs on to the web server.

Out of memory in *functionname*.

A memory error has occurred within an internal function. Your web server may be overloaded or you may need more physical memory.

Remote authentication denied for userid *username*.

Another web sever within the DNS domain has requested authentication of user *username* with a domain cookie and was not given access.

Check the security settings for the file. Make sure the account that the web server is running has Full Access privileges to the HTML file.

Remote authentication given for userid *username*.

Another web server within the DNS domain has requested authentication of user *username* with a domain cookie and was given access.

Remote authentication received deny for userid *username*.

A web server requesting authentication of a domain cookie was rejected.

Remote cookie rejected. Cookie failed MD5 test.

An unauthorized user has attempted to access the web server with an invalid web access authentication domain cookie.

Authentication Agent initialization failed

The Agent cannot make the connection to the Authentication Manager. Make sure that the Authentication Manager and the network are operational and that all network interface cards and cables are properly installed and in good condition.

Authentication Manager is not responding

There is a network communications problem between the Authentication Manager and the Web Agent, the server cannot be found (because the IP address is wrong, for example), or the Authentication Manager daemon is not running.

Authentication Manager is not responding. Run CLNTCHK to verify port and IP address of Authentication Manager.

There is a network communications problem between the Authentication Manager and the Authentication Agent, the Authentication Manager cannot be found (because the IP address is wrong, for example), or the Authentication Manager daemon is not running.

Session Manager: Failed to Create Server Thread.

There are too many server threads running (too many users connecting at once). Try widening the intervals at which users attempt to log on.

Session Manager: Failed to Resolve Hostname.

Most likely a configuration error. The machine that is connecting has no DNS or NetBIOS name, or has an invalid IP address. Make sure that your network is configured properly and that your host file entries are correct.

Session Manager: Not Enough Memory.

The system does not have enough physical RAM, or there were too many other processes running in memory. If you receive this message often, add more physical memory to the machine.

The security descriptor could not be found. The file may not exist: *filename*.

A user requested a URL that does not resolve to a file on the machine. Make sure that the user is entering the URL correctly.

The user *server/username* disconnected from port *portnumber.*

The user closed the connection on the specified port.

The user *server/username* connected on port *portnumber* on date at time and disconnected on date at time. . .

A normal Authentication Agent disconnection has occurred.

The user *username* has connected and been authenticated on port *portnumber*.

A normal (authenticated) Authentication Agent-Server connection has occurred.

Unexpected error from Authentication Agent.

The value returned by the Authentication Manager is not valid.

User <blank> canceled out of RSA SecurID Authentication routine.

The user canceled without entering a user name.

User I/O Timeout-User took too long to respond.

The system timed out after waiting for a response from the user.

User *username* canceled out of New PIN routine.

The user canceled the authentication attempt.

User *username*: ACCESS DENIED. ATTEMPT 1.

The user was denied access. Check the Authentication Manager Activity Log for the specific reason.

User *username*: Access denied. Attempt to use invalid handle. Closing connection.

An internal error occurred. If the message recurs, call the RSA Customer Support Center.

User *username*: ACCESS DENIED. Next Tokencode failed.

The user must reauthenticate.

User *username*: ACCESS DENIED. Server signature invalid.

This message indicates that the identity of the Authentication Manager could not be verified by the client. If you see this message, call the RSA Customer Support Center.

User *username*: ACE Check Error: Invalid group SID. Passcode required.

The user's group SID did not contain a valid group name. The user was challenged for an RSA SecurID passcode.

User *username*: canceled out of Next Tokencode routine.

The user canceled out of the Next Tokencode process.

User *username*: canceled out of RSA SecurID Authentication routine.

The user canceled after entering a user name.

User *username*: Domain not found. User challenged for passcode.

The user may have entered the domain name incorrectly and will be challenged for a passcode.

User *username*: New PIN accepted.

The user's New PIN was verified.

User *username*: New PIN rejected.

The PIN was rejected by the Authentication Manager. The user needs to reauthenticate to set the PIN. Check the Authentication Manager Activity Log.

User *username*: Not found. User challenged for passcode.

The user is unknown to the system, but the system still challenges the user for a passcode.

User *username*: Successfully logged on with Next Tokencode.

The Next Tokencode was accepted by Authentication Manager and access was granted to the user.

Known Issues Using Third-Party Software

Browser Issues

Both Internet Explorer and Firefox maintain a single browser session across multiple instances of the browser. If a user has successfully authenticated to a protected resource in one instance of the browser, as long as that instance remains open, all other instances of the browser share the same authentication cookie. Therefore, the user does not have to reauthenticate in any other instances of the Firefox browser to access protected resources.

To exit the browser session, users must close all instances of the browser.

Blackberry WAP browser issues

While using the Apache webagent in conjunction with a Blackberry WAP browser, the redirection after authentication fails with a 404 error. This is due to the content location being included.

This is resolved by setting the following field in the RSAWebAgent.ini;

DisableContentLocationForWAP=1

Wireless Devices

A user could experience the following scenarios when using a cellular phone equipped with a microbrowser to access protected URLs:

- If your environment includes a GSM network, your WAP connection needs to be in connection mode. Multiple domain environments require that handset devices and gateways support the receipt of cookies from multiple domains.

- Requiring an SSL connection to protected URLs creates a more secure environment. For ease of use, you can configure the Web Agent to automatically redirect the URL request to a secure connection.

  However, if your microbrowser does not support automatic redirection, you must disable the redirect option. Instead of automatic redirection, a web page opens that contains a link to the secure connection.

- When the Web Agent is configured to use a single web page for entering the user name and passcode, the LCD on certain devices may appear to be using separate pages, one for entering the user name and a second page for entering the passcode. However, the microbrowser on the device is sending the data all at once, unless you have specifically enabled the **Use Separate username and Passcode Pages** option in the Web Agent.

- When **Name Locking** and **Use Separate Username and Passcode Pages** are enabled in the Web Agent, and the carrier signal is lost after transmitting the user name, the user name is locked in the Web Agent database until the Name Lock time-out expires. Instruct the user to authenticate again after the Name Lock expiration time.

Note: The name locking feature offers security tradeoffs that may or may not be appropriate for your environment. By enabling name locking, a 30-second lock is created on RSA Authentication Manager 6.1. As with any lockout mechanism, this can be used to prevent a valid user from authenticating by continually relocking the valid user name.

Displaying the user name and passcode prompts as separate pages is necessary to fully use the security offered by name locking. But name locking comes with security tradeoffs that may or may not be appropriate for your environment. When the prompts are separated onto different pages, the Web Agent creates new sessions while submitting the user names. As with most session management systems, this creates the possibility that all sessions will be reserved, and new authentication attempts will be rejected until old sessions complete.

- It can be difficult for users to enter the PIN and tokencode within the designated time limit (typically 60 seconds) before the tokencode changes again. Most WAP devices by default are set up for alphanumeric entries. That means the user must scroll through the letters assigned to a button before reaching the numbers. Because tokencodes are always numeric, instruct users to switch their phone to numeric entry, if their phone allows this, only after entering the PIN.
- Some gateways have very specific size limitations for WML templates. You may need to reduce the amount of information provided in the templates.
- To enable the **Redirect HTTP Connections to Secure Server** option, the cellular device and its gateway must allow for SSL redirection. RSA recommends that you instruct the user to refer to the documentation provided with his or her cellular device.
- Devices that allow for an image display may, during the course of an authentication, display the status "Failed" for several seconds (depending on the speed of the microbrowser) until an image is shown on the LCD that indicates success. In these instances, the user must wait for several seconds until the success image appears. If, however, the "Failed" status message is displayed for a substantial amount of time, it is most likely valid, and the user should reauthenticate.
- For increased security on WAP browsers, RSA recommends setting the cookie expiration times to less than the defaults of 15 minutes for idle cookies and 60 minutes for all cookies.

Issues in the protectURL script

Assume that you have configured a server with the host name using the config script, and added the URLs that you want to protect or unprotect on the server using the protectURL script. In this case, the URLs that you add to the configured server will be protected or unprotected only if you access the server using the host name and not the IP address.

If you have protected or unprotected the URLs on the server and want to access the server using the IP address, you must create separate entries for both the host name and IP address using the config script. Then, you must use the protectURL script to add the URLs in both the IP and Hostname entries.

Similarly, if you have configured the server using only the IP address, you cannot protect or unprotect the URLs added using the protectURL script if you try to access the server using the host name. You must create separate entries for both IP address and host name.

Multiple Domain Issues

When connecting to multiple domains, a web page is displayed showing the domain URL and the success or failure of the connection. In some environments, the appropriate images do not appear in the web page. This problem occurs only when there is no valid certificate on the web server. If this occurs, use http instead of https when you input domains in your multiple domain list.

Important: All web servers protected using the multiple domain option should be configured to use only SSL(not plain http).

Note: All servers protected using multiple domain authentication should have the same webID URL. If you have changed the default webID URL value in any of these servers (default value is **/webauthentication**) then you must change it in all the servers, else the feature will not work.

If you have configured some URLs protected by the Web Agent for multiple domain single sign-on access, single sign-on will not work with Internet Explorer browsers, even if you have added the URLs to the trusted zone in Internet Explorer. When you access one URL and successfully authenticate, you will still be challenged when accessing the other URL configured for SSO.

To avoid this problem, in addition to allowing third-party cookies, you must also configure the following settings in the Internet Explorer browser:

1. Click **Tools** > **Internet Options**.
2. In the Internet Options dialog box, click the **Privacy** tab.
3. Click **Sites**.
4. In the Per Site Privacy Actions dialog box, type the URL that you want to configure for multiple domain single sign-on access in the **Address of Web site** text box, and click **Allow**.
5. Repeat step 4 for all the URLs participating in single sign-on.
6. Click **OK** in the Per Site Privacy Actions dialog box.
7. Click **Apply** > **OK**.

The following issues may occur when using multiple domain access on wireless devices:

- When multiple domain access is enabled in the Web Agent, a list of URLs for the domains is displayed. WAP devices that allow for an image display may, during the course of an authentication, display the "Failed" status for several seconds until an image is shown on the LCD that indicates success. In these instances, the user should wait for several seconds until the success image is shown. However, if the "Failed" status message remains for a substantial amount of time, it is most likely valid, and the user should reauthenticate.
- When multiple domain access is enabled, the Web Agent attempts to get an image from each of the domains to verify the connection. With some cell phones, the image is displayed even though the connection was never actually made. The user is forced to reauthenticate each time he or she attempts to access a URL in another domain.

  To work around this issue, set the variable **UseTextWML=1** in the **RSAWebAgent.INI** file located in the Web Agent installation directory (the default is **rsawebagent**). This forces the user to manually click a text link for each domain instead of attempting to automatically make the connection using images.

Index