RSA ACE/Agent 5.0 for Web
Installation and Administration Guide

Contact Information

See our Web sites for regional Customer Support telephone and fax numbers.

RSA Security Inc.
www.rsasecurity.com

RSA Security Ireland Limited
www.rsasecurity.ie

Trademarks

License agreement

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 exporting this product.

Distribution

Limit distribution of this document to trusted personnel.

First printing: April, 2002

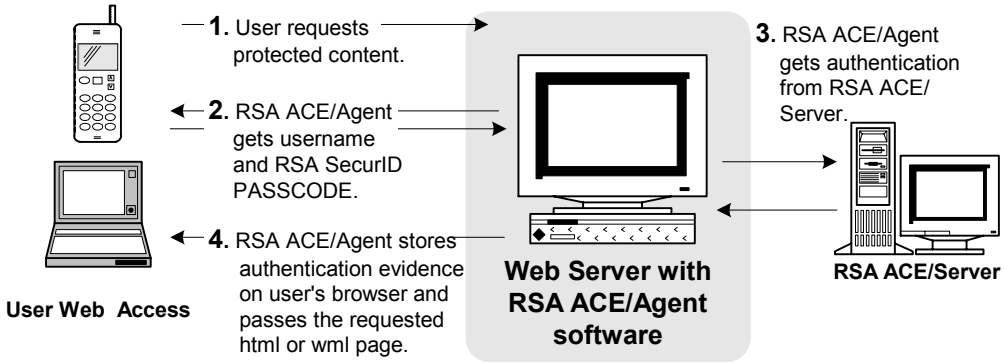
Contents

1 The RSA ACE/Agent for Web

The RSA ACE/Agent 5.0 for Web software brings two-factor RSA SecurID authentication to your Apache or Stronghold Web server running on a Solaris or Red Hat Linux system. The Web Agent software enforces user authentication to an RSA ACE/Server before serving protected Web pages to users.

A sample RSA SecurID solution is illustrated below. The RSA ACE/Agent for Web software, residing on a Web server, intercepts all requests for Web (HTML or WML) pages. If the Web Agent software determines that a user is not authenticated, it requests the user's name and PASSCODE and passes these to the RSA ACE/Server for authentication. On successful authentication, the Web Agent software stores this information as a tamper-evident cookie on the user's system. As long as the cookie remains valid, the user is granted access to Web pages.

An RSA SecurID Solution



Administrators use the Web Agent software to maintain a list of protected URLs. Administrators can protect all URLs or selected URLs as needed. This lets you use your Web site as both a public resource available to all users and as a highly secure area for posting confidential information to trusted users.

Security Features

The RSA ACE/Agent 5.0 for Web software provides the following security features to protect your Web resources.

Two-factor authentication. When users attempt to access protected Web pages, the agent software enforces RSA SecurID authentication to an RSA ACE/Server before serving protected Web pages to users. RSA SecurID authentication is stronger than traditional password protection because it requires two factors instead of one: something the user knows (PIN) and something the user has (SecurID token). Users must enter a valid RSA SecurID PASSCODE instead of a password. The PASSCODE consists of

- A secret memorized personal identification number (PIN).
- The current tokencode generated by an RSA SecurID token or software program. This tokencode changes every 60 seconds to an unpredictable new tokencode to prevent intruders from guessing it.

Support for SSL. This feature establishes a private communication channel between the user and the Web server, preventing third parties from eavesdropping.

Name Locking. This feature prevents an intruder from detecting a legitimate login attempt and then intercepting and using the legitimate user's PASSCODE to log in. With name locking, a user's first login screen accepts only the user's name. The system then prevents anyone else from using that name to log in until the user successfully logs in or fails to log in.

Tamper-evident cookies. When a user logs in, the RSA ACE/Agent 5.0 for Web software generates information indicating the user has successfully authenticated, storing it in a cookie in the user's browser. Users with valid cookies are permitted access without having to authenticate again. To prevent someone from altering or forging a cookie to gain access, the Web Agent stores an encrypted data string within the cookie. The agent uses this encrypted string to detect whether someone has altered the cookie contents. Any tampering is logged in the system Web Agent audit files.

Auditing. The Web Agent records access attempts and the status of connections in audit logs on the Agent Host.

Understanding Agent and Browser Interaction

A Web access authentication cookie is valid only during the browsing session for which it was created. If the user closes the Web browser, the user must reauthenticate to get a new cookie. However, you may also want to set restrictions on the length of time an authentication cookie is valid to help protect the URL if the user walks away from his or her computer.

The Web Agent has default expiration times set for authentication cookies to increase protection of your URLs. Setting a short idle cookie expiration time can help reduce the likelihood that an unauthorized user will gain access to protected pages through a Web browser when there is too little activity or the user has left the computer unattended. An authentication cookie expires during an active browsing session after sixty minutes. An idle authentication cookie expires after fifteen minutes.

To increase the effectiveness of idle cookies, RSA SecurID requires users to enable the option in their browsers that always forces the updating of pages. Doing so ensures that the cookies are also refreshed. If the cookies are not periodically refreshed and you have set time restrictions, users will be prompted to reauthenticate when the cookies' time-out periods expire.

Authenticating Through a Firewall or Proxy

The Web Agent includes the browser IP address when validating the authentication cookie. This way, if the cookie is stolen and the hacker tries to use it on another browser, access is denied because the IP addresses don't match. However, if you have a firewall or proxy, some users may be prompted to authenticate every time they attempt to access a protected URL. To avoid this, make sure you enable the option to ignore the browser address in the Configuration menu section of the Web Agent configuration program. Since this method provides less security than including the browser IP address, RSA Security strongly recommends that you use SSL connections to protect data passing between the browser and the Web server.

Preventing Caching of Protected Pages

When a user browses a Web site, the browser stores a copy of each visited page in disk or memory cache. Each time the user clicks **Back** or **Forward**, the browser loads the saved copy of the page, thus eliminating the time it takes to contact the server and reload the entire page.

Memory cache is deleted when the browser is closed. Disk cache is not necessarily deleted when the browser is closed. Therefore, an unauthorized user can view the pages that were stored in the disk cache long after the authorized user has quit the browsing session. The Web Agent attempts to prevent the user's browser from caching protected pages on the local desktop. The Web Agent cannot prevent the browser from copying pages to the memory cache.

Using SSL Connections

RSA Security strongly recommends that you use SSL (Secure Socket Layer). The Web Agent is configured to use a default SSL port number of 443. When configuring the Agent, enter the appropriate SSL port number for your environment. By default, the Web Agent requires that connection to protected URLs occurs over SSL.

If you are not using SSL on your Web server, you are leaving protected pages open to replay attacks. This means the data being transferred between the server and the browser is not secure. An unauthorized person can monitor an unsecured connection, intercept a user's RSA SecurID PASSCODE or authentication cookie, and then use the stolen object to gain access to protected pages.

Controlling Redirection of Browser Connections

When access to protected URLs is configured to require an SSL connection, clients that attempt to access a protected URL through an unsecure connection (standard HTTP), are redirected to a page with a link to the secure server (using HTTPS). This link uses the SSL port defined in the Web Agent.

For example, if a user attempts to access a protected URL at **http://www.exampledomain.com/sales_figures/**, the user's request is redirected automatically to **https://www.exampledomain.com/sales_figures/** (note use of the **HTTPS** protocol). The user is then asked to authenticate before the protected Web page is displayed.

If you choose not to redirect unsecured connections to a protected resource, when the user attempts to access the protected resource through an unsecured connection (for example, **http://www.exampledomain.com/sales_figures/**), the user will receive an RSA Security alert message with a link to redirect to a secure connection.

Authenticating Users

You will need to decide how to handle the authentication pages presented to your users and how the data will be transmitted.

Name Locking

A name lock protects against the danger that someone might observe the user entering his or her username and PASSCODE and present the same information on a different Web server. The Web Agent uses the name lock feature to help secure against this. After the user enters his or her username and PASSCODE, the Web server sends the authentication requests to the RSA ACE/Server (the Server) in two steps. First, the Web Agent sends the username. The Server then locks the user's token record. Then the Server asks for the PASSCODE. The Web Agent sends that data and the Server attempts to authenticate it. Upon success, the user is presented with the protected URL.

If the Server finds the user's token record is already locked, the Server refuses both the username and PASSCODE authentication requests. Versions of RSA ACE/Server earlier than 5.0 do not support the name locking feature.

CAUTION: If the RSA ACE/Server requires name locking, you must also use the name locking feature in the Web Agent. Otherwise, all authentication requests to access protected URLs will be denied. If the Server does not require name locking, you can choose to disable this feature in the Web Agent.

Using Separate Authentication Pages

To increase the level of security when passing the username and PASSCODE data from the browser to the Web Agent, the Agent presents separate authentication pages to the user. First the user is prompted for his or her username. The Web Agent then sends this data to the RSA ACE/Server. Once the Server receives this data, the user is prompted for the PASSCODE. The Web Agent then sends that data to the Server. This makes it more difficult for an eavesdropper to monitor the connection for both the username and PASSCODE. This is an especially useful feature when combined with the name locking feature. Once the RSA ACE/Server receives the username data, the user's token record is locked in the database. The locked username will not be accepted on any other protected Web server.

Presenting Authentication Pages to Users

The Web Agent has provided standard HTML and WML templates for your use. Use the templates as is, or make a copy and modify them to meet your company environment. You can change the messages displayed to your users, use your own company graphics, and translate the messages to display into your native language. Additional information on how to customize the templates is provided in "Customizing HTML and WML Forms" on page 21.

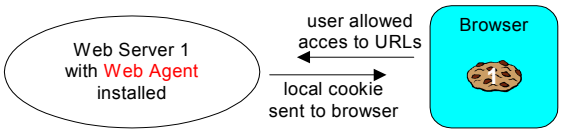
After a user enters information on the Web authentication page and presses ENTER, the Web Agent automatically continues and presents the next page. If you prefer, you can choose to have the user click CONTINUE to proceed.

As users browse a protected Web site that uses HTML frames, sometimes the PASSCODE prompt is displayed in a very small frame making it difficult to clearly read. To avoid this problem, The Web Agent allows the PASSCODE prompt to be displayed in a JavaScript popup window.

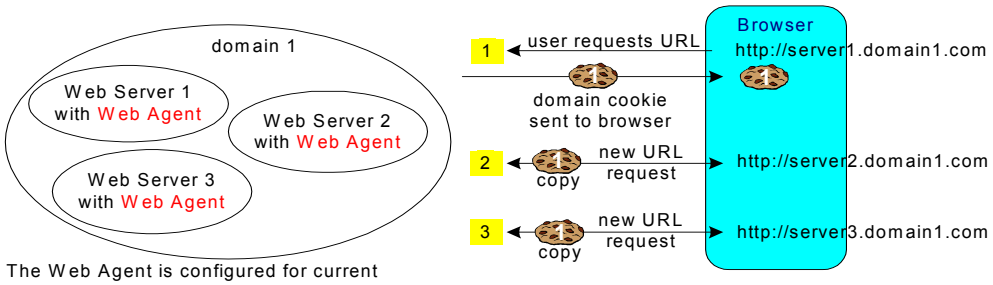
Note: To use JavaScript popup windows, users must enable Java script support in their Web browsers.

Single Sign-on to URLs

The Web Agent protects URLs on the Web server on which it is installed. This is considered local protection. When the user authenticates, a cookie is sent to the user's browser. The cookie allows the user to access the protected URLs for that Web server.



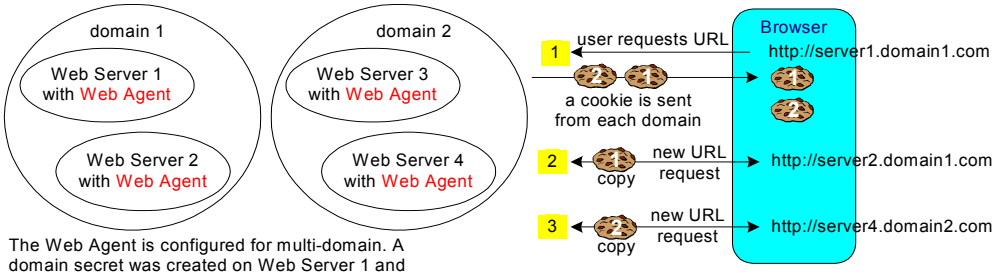
To obtain the most security, you should protect all of your company-sensitive Web data. Therefore, you should install the Web Agent on each Web server that contains valuable data. To allow users to then work in the most efficient manner, the Web Agent can be configured to allow access to URLs on all or any Agent-protected Web servers without having to authenticate themselves to each Web server. This is called **single sign-on**. This example explains single sign-on using the current domain feature.



The Web Agent is configured for current domain. A domain secret was created on Web Server 1. Domain secret from Web Server 1 is imported to Web Servers 2 and 3.

The user attempts to access a protected URL on Web Server 1. The user is prompted to authenticate. Upon successful completion, Web Server 1 sends a domain cookie to the user's browser (1) and displays the requested URL. (Notice the difference is that instead of a local cookie, a domain cookie is being sent to the browser.) The user browses the page, then clicks on a link in the Web page that calls a URL on Web Server 2 (2). The Web Agent on Web Server 2 notices there is an existing cookie and requests a copy. The Agent verifies that cookie 1 is valid for the domain and displays the Web page without requiring reauthentication. Now the user types in a completely different URL that resides on Web Server 3. This URL is not linked in any way to the previous two URLs (3). Again, the Web Agent notices there is an existing cookie, requests a copy and verifies that it is valid for the domain, and displays the Web page without requiring reauthentication. Through single sign-on (one authentication session) the user is allowed access to any URL on any server in the domain.

Many company environments have multiple domains. Of course, you want the same single sign-on capabilities in all domains in which users need to access protected URLs. This example explains single sign-on when the multi-domain feature is used along with the current domain feature.



The Web Agent is configured for multi-domain. A domain secret was created on Web Server 1 and imported to Web Servers 2, 3, and 4.

Single sign-on in this environment works much the same way as in the previous example. The difference is that when the user initially requests a protected URL and is authenticated, the Web Agent sends a cookie from each domain (1). As before, when a URL is requested from Web Server 2, the Web Agent requests a copy of cookie 1, verifies it is valid for domain 1, and displays the Web page (2). Now, the user requests another Web page. This URL happens to reside on Web Server 4. The Web Agent requests a copy of cookie 2 because it is the domain cookie for Web Servers 3 and 4. The Agent verifies domain cookie 2 and displays the Web page without requiring reauthentication. Again, through single sign-on (one authentication session) the user is allowed access to protected URLs on any server in the current domain (in this example, domain 1) as well as any server designated in the multi-domain list (in this example, Web servers 3 and 4 in domain 2).

To enable single sign-in multi-domains, you could also create a subdomain by entering *domain.com* as the Domain Name in the Domain and Multi-Domain Configuration menu. Then, users would be allowed single sign-on to any Agent-protected Web server in any domain ending with *domain.com*. See “Configuring the Software” on page 15 for more information.

There are many ways you can combine local, domain, and multi-domain access. For maximum security, RSA Security recommends that you protect all URLs containing company-sensitive data.

Domain Secret

A domain secret is a random number created by the Web Agent. The domain secret is then used when creating domain cookies on a Web server. To enable access to multiple servers, the other servers must also have the domain secret. You will need to generate and export a domain secret so you can import it to all other Web servers at your site that will issue and accept domain cookies. You can also import domain secrets from other Web servers. This is done through the Agent configuration program and is explained in the next chapter.

CAUTION: Separate Web servers that each use a separate RSA ACE/Server database to authenticate users should specify a separate domain secret for their domain cookies. Otherwise, users with an authentication cookie issued from one server might gain access to information on other servers that they are not authorized to view.

When you export the domain secret to a file, RSA Security recommends that you store the file on removable media and then securely store the file so it is not accessible to unauthorized persons.

Managing URLs

The most common administrative task you will need to perform on the Web Agent is maintaining which URLs to protect. There are a couple of ways to manage your list of protected URLs. You can add or delete individual URLs from the protected resource list or you can maintain a list of URLs in a file. The URLs listed in the file are then imported into the protected resource list.

By default, the Web Agent protects all the URLs on the listed Web server. The protectURL utility, located in the Web Agent directory, provides you with quick access to maintaining the protected resource list. See “URL Management” on page 27 for information on how to add and delete URLs from the protected resource list.

2 Installation and Configuration

If you are unfamiliar with the area of protecting URLs, please read the previous overview chapter, “The RSA ACE/Agent for Web”, before installing your Web Agent so you will better understand the configuration choices you need to make.

This chapter contains the following instructions:

- Preparing the Web server and RSA ACE/Server environments to work with the Agent
- Installing the RSA ACE/Agent 5.0 for Web software
- Configuring the RSA ACE/Agent 5.0 for Web software to authenticate users
- Customizing authentication messages for your end users

Installation Requirements

Before installing the Web Agent, you need to make sure your Web server and RSA ACE/Server environments are configured properly to work with the Agent.

The Agent software requires approximately 10 MB of free disk space.

Enabling the Web Server to Work with the Agent

The Apache server binaries must have module **mod_so** enabled. This module supports loading shared objects into the server at start-up or restart time.

If your Apache Web server is already installed and configured, you can check whether this module is enabled as follows:

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

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

2. Type this command to list the Apache Web server modules that are enabled.

   ```
   ./httpd -l
   ```

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

   If **mod_so** is not listed, you must recompile the Apache Web server binaries with this module enabled. Refer to your Apache documentation for instructions.

Adding the Web Server to the RSA ACE/Server Environment

Successful user authentication requires the Web server machine to be an Open Agent Host of your RSA ACE/Server. (**Tip:** Agent Hosts are called Clients in RSA ACE/Server version 4.1 and earlier.)

1. Use the RSA ACE/Server to configure the Web server machine as an Open Agent Host of your RSA ACE/Server. Ask your RSA ACE/Server administrator for assistance.
2. Obtain an **sdconf.rec** file from your RSA ACE/Server administrator and place the file in a directory that is accessible to the Web server and RSA ACE/Agent for Web software (for example: var/ace). The Web Agent software uses this file to locate the RSA ACE/Server on the network. Ask your RSA ACE/Server administrator for assistance.
3. Log into an account on the Web server machine that has write permissions to the Web server root directory. This is the Apache user account designated in your Apache configuration file.
4. Add a VAR_ACE environment variable to your Web server configuration file so it is set whenever the Web server runs. This environment variable identifies the location of the **sdconf.rec** file. For example:

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

Installing the RSA ACE/Agent Software

To install the RSA ACE/Agent 5.0 for Web software, follow these steps:

1. Log into an account that has write permissions to the Apache server root directory.

   RSA Security recommends that you stop your Web server before installing the Web Agent. However, if you have Web sites that cannot be taken out of service, the Agent will install with http services running.
2. Run the install script.
   - If you downloaded the software, change to the directory you created when you downloaded the software and type

     ```
     ./install
     ```
   - If the software is on a CD, start the installation as follows:

     For Red Hat Linux:

     ```
     /mnt/cdrom/install
     ```

     For Solaris:

     ```
     /cdrom/apachecd/install
     ```
3. You are asked from where you obtained your Web Agent product. If you obtained this product from somewhere other than the countries listed, type **n** to display an alternate license agreement; otherwise press ENTER to continue.

4. After carefully reviewing the license text, type **A** to accept the License Terms and Conditions that are displayed and continue installing the software. If you do not accept, the installation aborts.

5. Press ENTER to accept the **sdconf.rec** path if it is correct.

   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. See the preceding section, "Adding the Web Server to the RSA ACE/Server Environment", for information about this setting.

6. For each of the remaining installation prompts, press ENTER to accept the default value, or type in a different path.

The Agent program files are now installed.

The install script created a backup of **/usr/local/apache/conf/httpd.conf** under the file name **/usr/local/apache/conf/httpd.conf.*date***, where *date* is the date and time when the backup was created.

The configuration script is automatically started.

Configuring the Software

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. See "Making Configuration Adjustments" on page 28 for information on running the configuration script again.

1. The Setup menu displays. Press ENTER to accept the defaults and continue with configuration, or type in the line number of the option you want to change. The options are described in the following table:

Line Option	Description
1 Idle Cookie Expiration	Set the amount of time, in minutes, an idle cookie is valid. When the cookie expires, the user will have to authenticate again. Setting a value that is greater than the Cookie Expiration value will make this feature inactive.
2 Cookie Expiration	Set the length of time, in minutes, that an active cookie is valid. If the user closes the Web browser or when the cookie expires, the user must authenticate again to get a new cookie.
3 SSL Port Number	Type in the SSL port number to be used for secure data transfer.
4 WebID URL	Accept the default name, unless you have an existing URL with the same name.

Line Option	Description
5 HTML/WML Templates	Accept the default. After the initial installation and configuration, you may decide to customize the templates. Once that process is complete, run the configuration script again to designate the new location of your customized templates. See "Customizing HTML and WML Forms" on page 21 for instructions on customizing templates.

2. The Configuration menu displays. Press ENTER to accept the defaults and continue with the configuration, or type in the line number of the option you want to change. The options are described in the following table:

Line Option	Description
1 Agent Protection	Accept the default. **Caution:** Make sure you only disable the Web Agent 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 valuable data is unprotected.
2 Name Locking	The Web Agent attempts to lock the user's name while waiting for the PASSCODE so the name cannot be used elsewhere during the authentication process. (Note: If your RSA ACE/Server is earlier than version 5.0, this feature is not supported. You need to disable it.)
3 Separate Pages	The Web Agent uses separate HTML or WML pages to request the user's name and PASSCODE, for added security. If you disable this feature, the username and PASSCODE will be sent across the Internet together.
4 Require SSL Connection	The Web Agent requires connection to protected URLs through an SSL port. If you disable this feature, data passed over the Internet will be unprotected.
5 Redirect	When a user attempts to access a protected URL through an unprotected page, the Web Agent redirects the user to an authentication page. If you disable this feature, the user will receive an RSA Security message and a link to a secure connection. **Note:** This option does not appear if option 4, Require SSL Connection, is disabled.
6 Caching Pages	The Web Agent attempts to keep the browser from caching protected URLs on the local PC. If you disable this feature, protected URLs may be cached on the local hard drive.
7 Auto Submit	After the user enters authentication information on the Web page, the Agent automatically presents the next screen so the user does not have to click CONTINUE.

Line Option	Description
8 Java Script Popup	By default, this feature is disabled. The Web Agent allows the use of Java Script Popup Windows in case Web pages are designed using frames.
9 Ignore Browser Address	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 Agent, the IP address used may be the same. If you have Web sites that are accessed through load balanced proxy servers, meaning that the browser IP addresses may change, you may want to enable this feature. Otherwise, the user may have to authenticate quite frequently.
10 Current Domain Access	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 will be asked to reauthenticate each time a protected URL is accessed on a different Web server.
11 Multi-Domain Access	Once a user is authenticated, the user can access URLs on any Web server in the multi-domain list. If you disable this feature, the user will be asked to reauthenticate each time a protected URL is accessed on a Web server that is outside the current domain.

3. If you chose to enable number 10 or 11 in the Configuration menu, the Domain and Multi-Domain Configuration menu displays. The options are described in the following table:

Line Option	Description
1 Generate Domain Secret	A domain secret was automatically generated when you installed the Web Agent. Use this option to generate a new domain secret for security reasons, perhaps to be in accordance with company security procedures.
2 Generate and Export Domain Secret	If you have multiple Web servers for which users will be able to access protected URLs (single sign-on), 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 you can import it to all other trusted Web servers at your site that will issue and accept domain cookies. You are asked to name the export file and create a password for the file. The file is then stored in the Web Agent directory (default: rsawebagent). See “Single Sign-on to URLs” on page 9 for more information on setting up domain and multi-domain configurations.

Line Option	Description
3 Import Domain Secret	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 file name and file password. See “Single Sign-on to URLs” on page 9 for more information on setting up domain and multi-domain configurations.
Current Domain Options	**The options below only appear 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 Web servers above.
5 Cookie Name	Use this option to change the default cookie name (rsacookie). Maximum name length is 30 characters.
Multi-Domain Options	**The options below only appear if you chose number 11, Multi-Domain Access, in the Configuration menu.**
6 Add to Multi-Domain List	Enter the Agent-protected Web servers for which you want all users to access protected URLs once they have authenticated. Use the format http://*server1.domain1.com*.
7 Remove from Multi-Domain List	The Multi-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 Multi-Domain List.)
8 View Multi-Domain List	View the list of Agent-protected Web servers you entered with option 6 for the Multi-Domain List. (This option does not appear if there are no Web servers in the Multi-Domain List.)

CAUTION: Separate Web servers that authenticate users to separate RSA ACE/Server databases should make sure a different domain secret is specified for the different domain cookies. Otherwise, users might gain access to protected URLs for which they should be unauthorized.

By default, the Web Agent protects all URLs on the Web server(s) listed. You can also choose to protect only some of the URLs. You can add individual URLs to the protected resource list, or you can list all the URLs you want protected in a file. See “URL Management” on page 27 for information on how to manage the protected resource list of URLs.

After you have completed configuring the Agent, the product registration Web page opens. If you choose not to register now, you can access the page at your convenience **www.rsasecurity.com/go/register/agent.asp**, or you can run the registration script

```
./registerWA
```

from the Web Agent installation directory.

Testing the Installation

You can test your installation by using a browser to access a protected Web server page. The server sends a page requesting authentication information (username and PASSCODE). If you have a valid RSA ACE/Server user account and the corresponding RSA SecurID token, you should be able to authenticate to the server. On successful authentication, the Web server sends the requested page.

If you are having difficulty authenticating, the “Troubleshooting”chapter contains information about tools you can use to test communication between the Agent and the RSA ACE/Server, to test the authentication process, and to test the status of the Server.

Customizing Authentication Progress Messages

When end users access your HTML or WML pages, the server prompts them to enter their RSA SecurID PASSCODE. The prompts and other authentication progress messages are sent to end users as HTML pages for browser users or WML pages for wireless (micro-browser) users. While the default progress messages are suitable for many circumstances, you can customize them if necessary. The “Customizing User Authentication Messages”chapter provides information on how to customize the HTML and WML pages included with this Web Agent product.

3 Customizing User Authentication Messages

When users authenticate using a Web browser or a wireless device microbrowser, the RSA ACE/Agent 5.0 for Web software prompts users for their username and RSA SecurID PASSCODE, finally informing them whether they have successfully authenticated.

For standard browsers, the system returns these messages as HTML pages. For wireless device microbrowsers, the system returns messages in WML format. The Web Agent software provides default versions of HTML and WML messages. You can also customize these messages to meet the specific needs of your organization. For example you might want to put your company name or logo on the pages, or include a telephone number or an e-mail address to contact for users who receive authentication failure messages.

This chapter provides information on what the default templates are and how to change some of the text messages displayed to users while authenticating.

Customizing HTML and WML Forms

The HTML and WML templates can be customized to reflect your company's image and administrative needs. You can edit the HTML and WML forms and supporting files to

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

The following table summarizes the purpose of each template. The default location of the HTML and WML templates is **/usr/local/apache/rsawebagent/Templates**. However, you may have designated a different location during installation of the Agent software. Review "Rules for Editing Templates and Message Strings" on page 24 before making any changes.

Note: If you are using RSA SecurID PINPADs instead of tokens, you will 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	Purpose
Errors	
error.htm **error.wml**	The page displayed when a fatal error occurs while authenticating a user. The **@@sub** macro in the template substitutes the error message passed from the system or from the **strings.txt** file.
Authentication Templates	
newpin.htm **newpin.wml**	The New PIN page displayed when users are authenticating with their token for the first time. From this page, users create their own PINs.
newpin1.htm **newpin1.wml**	The New PIN page displayed to a user that will receive a system-generated PIN. This functionality is determined in the RSA ACE/Server.
newpin2.htm **newpin2.wml**	The New PIN page displayed when a user is given the choice of whether to create their own PIN or receive a system-generated PIN. This functionality is determined in the RSA ACE/Server.
nextprn.htm **nextprn.wml**	The page 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.
sslredir.htm **sslredir.wml**	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** (**.htm** or **.wml**) page to continue.
redirect.htm **redirect.wml**	The page displayed to users when they complete the authorization process or when they log off.
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.wml**	The page displayed to users when they cancel out of the authorization process.

Template	Purpose
showsys.htm **showsys.wml**	The page displayed to users for ten seconds when the system generates an RSA SecurID PIN for them.
multidom.htm **multidom.wml**	The page displayed when users are authenticating across multiple domains.
userid.htm **userid.wml**	If you chose to present separate Web pages to users to input the username and PASSCODE, this template is used for the username. If you did not chose to present separate pages, the useridandpasscode template is used.
passcode.htm **passcode.wml**	If you chose to present separate Web pages to users to input the username and PASSCODE, this template is used for the PASSCODE. If you did not chose to present separate pages, the useridandpasscode template is used.
useridandpasscode.htm **useridandpasscode.wml**	If you chose to present one Web page to users to input both the username and PASSCODE, this template is used. If you chose to present separate Web pages to input the username and PASSCODE, the userid and passcode templates are used.

The HTML and WML forms are supported by the following files, which are also installed into the Templates directory:

Template	Purpose
Web Bitmaps	
denied.jpg	If you have configured the Web Agent to allow multiple domain authentications, the word “Denied” is displayed if a user’s authentication request to a virtual Web server does not succeed.
ok.jpg	If you have configured the Web Agent to allow multiple domain authentications, the word “SUCCESS” is displayed 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.
WAP Bitmaps	
denied.wbmp	If you have configured the Web Agent to allow multiple domain authentications, the word “Denied” is displayed if a user’s authentication request to a virtual Web server does not succeed.
ok.wbmp	If you have configured the Web Agent to allow multiple domain authentications, the word “OK” is displayed if a user’s authentication request to a virtual Web server succeeds.

Template	Purpose
Other Files	
strings.txt	This file contains text strings that are used to display various messages while users interact with the Web access authentication prompt pages that are produced from the HTML or WML templates.
style.css	The cascading style sheet used for the Web pages.

Rules for Editing Templates and Message Strings

To access the templates and text strings, log in as an Apache user as defined in the Apache configuration file. To ensure that the templates will still function properly after you have made changes, adhere to the following rules:

- Copy the templates into a new directory before making changes to them. If any templates are missing from this new directory, the Web Agent will automatically default back to the original templates.
- Only use a text editor to make changes. Programs such as FrontPage and HomeSite tend to add a lot of additional HTML tags to templates. There is also a possibility that these programs may alter the substitution strings that are necessary in the templates.
- **Important:** Do not alter any of the substitution strings in the templates or message text files (webagent.msg and strings.txt). These strings begin with two "at" signs (@@). The substitution strings are used to include error messages and text from the RSA ACE/Server and provide place holders for graphics and message strings.
- When using your own graphics, two methods are available for naming files.
  - A substitution macro (@@URL?GetPic?image=) works with HTML and WML. With WML, the images must be in WBMP format. With HTML, the images must be in JPG format. 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 Web server that is unprotected by RSA SecurID authentication, or on a separate Web 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 configuration as in the following example:

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

- You can replace the standard Send and Reset buttons that appear in the HTML templates with custom graphics. This approach is not supported by WML. Make sure the image file you point to in the **src** path is in a directory that is not RSA SecurID-protected and that you always specify a fully-qualified path to the image file.
- Test the templates after you've completed your changes to make sure they are still functioning properly. See "Troubleshooting"for information on utilities you can use to troubleshoot problems.
- Run the Web Agent configuration script to point to the new templates directory location containing the customized templates (located in the Setup configuration menu).

Customizing Templates for Another Language

If you need to customize the templates for a language other than English, you must store them in a language-specific directory under the Web Agent templates directory. The default directory for language specific templates is **/usr/local/apache/rsawebagent/Templates/nls/*language_code*** where *language_code* is the language preference code used by Web browsers.

To use international characters in the templates, consult an HTML reference book or visit the World Wide Web Consortium's Web site at **www.w3.org/pub/WWW/International** for more information about using international character sets in HTML documents.

Note: Templates that support the RSA SecurID Software Plugin are stored in the default directory **/usr/local/apache/rsawebagent/Templates/nls/en-securid**. These English language templates contain JavaScript for managing the RSA SecurID Software Plugin and are not presently easily customized for non-English languages. Do not confuse the templates in this directory (ending in **en-securid**) with other language-customized templates.

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

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

   For example, **/usr/local/apache/rsawebagent/Templates/nls/fr** where **fr** is the language preference code for French.

   Note: To find the correct language code, refer to the language preferences list of codes in the Internet Explorer or Netscape Navigator Web browsers.

2. Copy the templates to the language-specific subdirectory that you have just created.
3. Translate the text strings, graphics, and buttons, making sure not to remove or alter the substitution strings (@@ in the templates). The variables will be replaced with actual values when the text is displayed.

4. Test the new templates to make sure they function properly.
5. Run the Web Agent configuration script and update the Template path in the Setup menu to point to the language specific templates.
6. Make sure your end users have their browser language preference set 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 their Web browser's list of language preferences. If the preference settings are not set correctly or language-customized templates do not exist, the browser displays the default English version of the templates.

4 Administration

The main administration task you will perform with the Agent for Web is managing the URLs you want protected. A utility is provided that allows you to quickly add, remove, and view URLs from the protected resource list, without having to access all of the configuration settings.

Once your Web Agent software has been operational for a while, you may find that you need to make adjustments to original configuration settings. For example, you may find that you need a longer validity period before an active cookie expires or that you want users to have access to Web servers in other domains without having to reauthenticate themselves.

This chapter provides information on the following:

- Using the protectURL utility to manage URLs
- Running the configuration script to make adjustments
- Uninstalling the Web Agent software

URL Management

By default, the Web Agent protects all URLs on the Web server, along with any virtual servers, on which the Agent is installed. If you do not want all URLs protected, you can unprotect specific URLs. It is important to remember that when you unprotect a URL that occurs high in the directory level, all subsequent URLs under it are also unprotected.

The **protectURL** utility provides you with an interactive menu from which to manage URLs you want protected. You can add, remove, or temporarily unprotect individual URLs in the protected resource list. The protectURL utility is located in the Web Agent directory (default: rsawebagent). Run

```
./protectURL
```

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, run

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

where *listURL* is a plain text file that you have named which contains a list of URLs, one URL per line, that you wanted added to the resource list.

To remove protected URLs from the resource list, run

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

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

Advanced UNIX administrators can manage the protected resource list using command line only operations. Run

```
./protectURL -h
```

for a list of options and syntax.

Making Configuration Adjustments

You may need to make adjustments to the default configurations for the Web Agent. For example, you may find that you need a longer cookie expiration time. You may also want to make adjustments to individual virtual servers. You make configuration adjustments by simply running the configuration program again.

Run the configuration script in the Web Agent installation directory.

```
/usr/local/apache/rsawebagent/config
```

A list will display the current Web server and any virtual servers you have set up in the Apache **httpd.conf** file. You may choose to make changes to the default settings applied to all servers, or you may choose to make changes to an individual server. Once you have chosen the server to configure, the configuration program is grouped into three menus.

Setup. Configure how the Agent will interact with the browser. Adjust cookie validity time, change the SSL port number, WebID URL, or the location of the directory for the HTML and WML templates that are used to display the user authentication pages.

Configuration. Configure access to protected URLs, such as redirection to secure ports, using separate pages for username and PASSCODE, using the name locking feature, etc.

Domain and Multi-Domain. Configure for which domain(s) an authentication cookie is valid and generate a new domain secret for use on other Web Agents.

Virtual Web Servers

When you configure the Web Agent, the Agent checks the Apache **http.conf** file for any virtual Web servers. The different virtual server names are added to the Web Agent configuration file and are then displayed when the configuration script is run. You can add or remove virtual servers from the list.

Adding

To add additional virtual servers to the Web Agent configuration, run the configuration script with the name of the virtual Web server

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

Verify that you want to create the new server. You will now walk through the three configuration menus. Either make changes that will affect only this virtual server, or press ENTER through each configuration menu to accept the defaults.

You can add as many virtual servers as you like. However, it is important to remember that if you want access to protected URLs to behave the same on all virtual Web servers, you need to make changes to your default Web server rather than individual virtual servers.

Removing

To remove a virtual server from the Web Agent configuration file, use the -d option.

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

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

Uninstalling the Agent

To uninstall the Web Agent:

1. Change to the Apache directory.

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

   RSA Security recommends that you stop your Web server before uninstalling the Web Agent. However, if you have Web sites that cannot be taken out of service, the Agent will uninstall with http services running.

2. Run the uninstall script from the Apache directory.

   ```
   rsawebagent/uninstall
   ```

The Agent software and the Web Agent installation directory are removed.

5 Troubleshooting

There are two utilities that test interaction with the RSA ACE/Server. The Web Agent also logs authentication attempts in the Apache error log.

acestatus. Provides information about the RSA ACE/Server for which the Web Agent is configured to communicate.

acetest. Tests user authentication.

error_log. Authentication attempts are written to the Apache error log file.

RSA ACE/Server Utilities

Use these utilities to determine communication between the Web Agent and the RSA ACE/Server. These utilities reside in the Web Agent directory (**/usr/local/apache/rsawebagent** is the default).

acestatus

This utility provides information about the RSA ACE/Server such as the configuration version, the server name and address, the number of client retries, and the client time-out period.

acetest

This utility allows you to authenticate to the RSA ACE/Server from the command line rather than going through authentication Web pages in your browser. This will help you determine whether a problem lies with the templates or with the authentication process itself.

Logging Authentication Attempts

Authentication attempts are logged in the Apache **error_log** file (**/usr/local/apache/logs** is the default path). The different types of error messages logged can be found in the **webagent.msg** file located in the Web Agent directory (**/usr/local/apache/rsawebagent** is the default).

The following table provides a list of possible error messages and their cause:

Error Message	Possible Cause and Solution
File /usr/local/apache/conf/httpd.conf isn't writable.	The user account with which you are logged in does not have write permissions. Log in 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 ACE/Server 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, a node secret is generated by the RSA ACE/Server and sent to the Web server. This error is also received if the node secret file is missing or the node secret on the RSA ACE/Server and Web server do not match. Contact your RSA ACE/Server administrator.
Unexpected RSA ACE/Agent error 103. Please try again.	This error is received when there are network problems. Contact your RSA ACE/Server administrator.
AceInitialize Failed during acetest authentication.	The **sdconf.rec** file is missing in the VAR_ACE directory. Obtain an **sdconf.rec** file from your RSA ACE/Server administrator. Place the file in a directory that is accessible to the Web server and RSA ACE/Agent for Web software. Restart the Web server.
The page cannot be found.	The requested page may not present in the **./htdocs** directory (a standard Web server directory in which Web pages are stored).
RSA Securid Error. 106: Web server too busy. Please try again later.	This error may occur when communication to the RSA ACE/Server is down or the sdconf.rec file is missing from the **var/ace** directory. Contact your Server administrator.
Unexpected authentication error.	This error may occur when authenticating using the acetest utility. Communication to the RSA ACE/Server is down. Contact your Server administrator.
The Page cannot be displayed.	There are two possible causes for this error message. • 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 HTML and WML templates is invalid. Verify the correct path in the Web Agent configuration (this is an option in the Setup Configuration menu when you run **./config** from the Web Agent directory).
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 multi-domain area. See “Single Sign-on to URLs” on page 9 for an overview of how multi-domain configuration works.

Known Problems of Third-Party Software

Browser Issues

When using the browser plugin (**sdplugin.exe**) that comes with the RSA SecurID Software Token installation package, there is a configuration issue when using Netscape as the browser. The main issue is described in the Troubleshooting section of the RSA SecurID Software Token User's Guide and instructions are provided for correctly configuring Netscape 4.78 to work with the plugin.

Unfortunately, to date, the workaround does not solve the issue if you are using the plugin with Netscape 6.*x*.

If you are using the Quick Launch feature in Netscape 6.2, the browser cookie will remain valid as long as the Netscape icon remains in the system tray, even after you close the browser.

Wireless Devices

RSA Security notes that the level of compliance with WAP 1.1 and 1.2.1 specifications may vary depending on the handset device and/or the service provider being used. Some service providers and handset device manufacturers may have implemented only the minimum number of WAP 1.1 and 1.2.1 requirements. Due to these circumstances, users may experience different results depending on the device, service provider, and configuration that is being implemented.

RSA SecurID Web authentication using wireless access protocol requires the following WAP 1.1 and 1.2.1 specifications:

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

In addition, 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.

RSA Security recommends that you instruct users to consult the manufacturer of the device being used, as well as the service provider, to ensure that these features have been implemented.

RSA ACE/Agent and RSA ACE/Server administrators should be aware of the following items pertaining to RSA SecurID Web authentication. These have been determined by RSA Security to be potentially common scenarios that could be experienced when using a cellular phone equipped with a microbrowser to access RSA SecurID-protected URLs.

- 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, not all microbrowsers support automatic redirection. In this case you will need to disable the redirect option. A Web page is then presented with a link to the secure connection that users will have to manually click.

- When the Web Agent is configured to use a single Web page for entering the username and PASSCODE, the LCD on certain devices may appear to be using separate pages: one for entering the username 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 Page for Username and PASSCODE** option in the Web Agent.
- When Name Locking and Use Separate Page for Username and PASSCODE are enabled in the Web Agent, and the carrier signal is lost after transmitting the username, the username is locked in the RSA ACE/Agent database until the Name Lock timeout expires. Instruct the user to authenticate again after the Name Lock expiration time.
- 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. Since tokencodes are always numeric, instruct users to switch their phone to numeric entry, if their phone allows this, only after entering the PIN (remember, PINs can be alphanumeric). This will make it easier to enter the tokencode.
- Some gateways have very specific size limitations for WML templates. You may need to reduce the amount of information provided in the templates.

Multi-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 GIFs used to show "Success" or "Failed" do not appear in the Web page. If this occurs, do not use **https** when you input domains in your multi-domain list, just use **http**. As far as RSA has been able to determine, this problem only occurs when there is no valid certificate on the Web server and using some versions of Internet Explorer. Therefore, this problem will basically only occur in a test environment.

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

- When Multi-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 (depending on the speed of the microbrowser) 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 attempt to authenticate again.
- When multi-domain is enabled, the Web Agent attempts to get an image from each of the domains to see if it has connected. With some cell phones, the image is displayed, but the connection was never actually made. So, once the user has authenticated once in a multi-domain environment and then attempts to access a URL in another domain, the user is asked to authenticate again rather than having single sign-on.

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 will force the user to manually click on a text link for each domain instead of attempting to automatically make the connection using images.

Proxy Servers

To be able 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

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

where **https://*proxyserver.domain.com*/xxx/** is the path to the root directory of the remote Agent-protected Web server. To make the change, run the Web Agent configuration script on the remote Agent-protected Web server. The WebID URL option is in the Setup menu of the configuration program.

The proxy server requires this configuration

ProxyPass /xxx https://*remoteserver.domain.com*
ProxyPassReverse /xxx https://*remoteserver.domain.com*

Getting Support and Service

RSA SecurCare® Online	**www.rsasecurity.com/support/securcare**
Technical Support Information	**www.rsasecurity.com/support**

Note: There is no provision for technical support during the warranty period unless a valid Software Service Contract is in force.

Make sure that you have direct access to the computer running the RSA ACE/Agent 5.0 for Web software.

Please have the following information available when you call:

- ❑ RSA ACE/Server version number.

  _____

- ❑ The make, model number, and operating system of the computer on which the problem occurs.

  _____

- ❑ Web server and version number being used.

  _____

Index

T

U

V

W