RSA® Access Manager 6.2 SP2
Security Configuration Guide

Contact Information

Go to the RSA corporate website for regional Customer Support telephone and fax numbers:
www.emc.com/domains/rsa/index.htm

Trademarks

RSA, the RSA Logo and EMC are either registered trademarks or trademarks of EMC Corporation in the United States and/or other countries. All other trademarks used herein are the property of their respective owners. For a list of EMC trademarks, go to **www.emc.com/legal/emc-corporation-trademarks.htm#rsa**.

License Agreement

This software and the associated documentation are proprietary and confidential to EMC, 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 EMC.

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

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

September 2014

Contents

Preface

About This Guide

This guide provides an overview of the settings available in RSA® Access Manager Servers and compatible Agents to help ensure secure operation of the product. It is intended for administrators and other trusted personnel. Do not make this guide available to the general user population.

RSA Access Manager Documentation

For more information about RSA Access Manager, see the following documentation:

Release Notes. Provides information about what is new and changed in this release, as well as workarounds for known issues. The latest version of the Release Notes is available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

Getting Started. Lists what the kit includes (package, licenses and documentation), specifies the location of documentation, and lists RSA Customer Support web sites.

Planning Guide. Provides a general understanding of RSA Access Manager, its high-level architecture, its features, and deployment information.

Servers Installation and Configuration Guide. Provides instructions for installing and configuring the RSA Access Manager Servers and additional components. This guide also contains descriptions of the different configuration options, features, and production environment considerations.

Administrator's Guide. Provides information for security administrators about using the RSA Administrative Console to administer users, resources, and security policy in RSA Access Manager.

Developer's Guide. Provides information about developing custom programs using application programming interfaces (APIs) included with the RSA Access Manager Servers.

API Delta Document. Provides information about the differences between previous and current versions of the APIs included with the RSA Access Manager Servers.

Upgrade Guide. Provides information about how to upgrade previous versions of RSA Access Manager Servers, data store schema, and data to the current version.

RSA Administrative Console Help. Provides instructions on how to perform specific administrative tasks. To view Help, click the **Help** tab on the RSA Administrative Console screen.

RSA Access Manager User Self- Service Console Help. Describes day-to-day user tasks performed in the User Self-Service Console. To view Help, click the **Help** tab on the RSA User Self-Service Console.

Related Documentation

For more information about products related to RSA Access Manager, see the following:

RSA Access Manager Agents documentation set. The documentation related to Agents is available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

RSA Adaptive Authentication documentation set. The documentation related to RSA Adaptive Authentication is available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

RSA Envision documentation set. The documentation related to RSA Envision is available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

Getting Support and Service

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

RSA SecurCare Online offers a knowledgebase 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 Secured by RSA 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 that you have direct access to the computer running the RSA Access Manager software.

Please have the following information available when you call:

- ❑ Your RSA Security Customer/License ID.

  This is a paper license. You can find this number only on the license distribution medium. If you do not have access to the paper-based RSA Customer/License ID, contact RSA Customer Support.
- ❑ RSA Access Manager software version number and patch level.
- ❑ The make and model of the machine on which the problem occurs.
- ❑ The name, version, and patch level of the operating system under which the problem occurs.

1 Security Configuration Settings on RSA Access Manager Servers

Security Configuration Settings for Servers

This section provides an overview of the settings available in RSA Access Manager Servers to help ensure secure operation. Security settings are divided into the following categories:

Access Control Settings for User Authentication and Authorization. Describes settings that limit access by end users, RSA Access Manager Servers, and external components.

Log Settings for Error and Debug Logs. Describes settings related to event logging.

Intercomponent Security Settings. Describes security settings related to RSA Access Manager network communications.

Data Security Settings for Data at Rest. Describes settings that help ensure protection of the data that is handled by RSA Access Manager Servers.

Other Security Considerations. Describes additional security settings.

Access Control Settings for User Authentication and Authorization

Access control settings help in protecting the resources against unauthorized access.

Authorization Server Mode

Setting	`cleartrust.aserver.authorization_mode`
Configuration File Location	*AXM_HOME*/conf/aserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path

Description	Controls access to unprotected resources, and works in conjunction with the RSA Access Manager Agent to determine whether an URL is protected. Allowed values are `active` and `passive`. In passive mode, all resources on an RSA Access Manager-protected web server are protected by default. For active mode, review the exclusion lists in the Agents configuration. See Chapter 2, Security Configuration Settings on RSA Agents.
RSA Recommendations	To secure all resources with or without an access policy, set this parameter to `passive`. **Note:** This change can disrupt existing deployments because an explicit "allow" access policy is required for a user to access the resource.

Handle Invalid User

Setting	`cleartrust.aserver.handle_invalid_user`
Configuration File Location	*AXM_HOME*/conf/aserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Controls the workflow of redirecting the user to the password screen for an invalid user ID instead of displaying the login failed error message.

Lockout Mode

Setting	`cleartrust.aserver.lockout_mode`
Configuration File Location	*AXM_HOME*/conf/aserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Controls the logic of returning ADMIN_LOCKOUT when the user is locked out irrespective of his credentials.
RSA Recommendations	It is recommended to set this parameter value to 2.

Key Server DNS Check

Setting	`cleartrust.keyserver.session_key_dns_check`

Configuration File Location	*AXM_HOME*/conf/keyserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Enables the Key Server to do a DNS check on the IP address of the client connecting to it. This is important because while generating the shared secret key, both the client name and the DNS are considered. Use `True` or `False`.
RSA Recommendations	Enable this setting to help ensure that the DNS in the environment is secure: `cleartrust.keyserver.session_key_dns_check=True`

Key Server Token Lifetime

Setting	`cleartrust.keyserver.token_lifetime`
Configuration File Location	*AXM_HOME*/conf/keyserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Sets the allowed idle time for single sign-on (SSO) tokens. Determines how long the Key Server must retain keys that are no longer used for encryption but are still valid for decryption. Use an integer, a space, and one of the following time identifiers: `hour | mins | secs`
RSA Recommendations	This value: • Should be greater than the sum of `idle_timeout` and `post_url_idle_timeout` parameters in the **webagent.conf** file of RSA Access Manager Agents. • Must be at least twice the value of `session_key_life` to prevent token decryption failure.

Key Server Session Key Life

Setting	`cleartrust.keyserver.session_key_life`
Configuration File Location	*AXM_HOME*/conf/keyserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies how long a session key is valid for encrypting new single sign-on (SSO) tokens. The default value is `30 mins`. Use an integer, a space, and one of the following time identifiers: `hour | mins | secs`

RSA Recommendations	Use the lowest possible value based on the user's idle time with the system.

Unique User Sessions

RSA Access Manager provides an option to disable concurrent user sessions per IP address. By default, there are no restrictions on the number of sessions for a user from a particular IP address. Enabling this option helps prevent the user from creating concurrent sessions from the same client machine.

To help increased security, RSA recommends disabling concurrent user sessions per IP address. For more information, go to the *Servers Installation and Configuration Guide* locate the "Enhanced Functionality" chapter, and see the "Configuring Unique User Session" section.

Log Settings for Error and Debug Logs

The default location of RSA Access Manager Server logs is: ***AxM_HOME*/ logs/** where ***AxM_HOME*** is the RSA Access Manager Server installation path.

Logging Levels

The following items are logged by RSA Access Manager, depending on the levels of logging configured.

- server start/stop messages
- error messages
- user authentication requests
- resource authorization requests
- administrative API transactions
- Authorization Server registration information

Note: Do not set the log level above 20 for production environments. A log level higher than 20 impacts system performance.

For more information, go to the *RSA Access Manager Troubleshooting Guide*, and see the "Log Settings for Error and Debug Logs" section.

Logs Directory Permissions

Log files contain sensitive information. For example, Authorization Server logs identify which users have access to which resources. To help secure Authorization Server log files, RSA recommends that you grant log file access only to the most trusted administrators.

For more information, go to the *Servers Installation and Configuration Guide*, locate the "Implementing Security Features" chapter, and see the "Protecting the RSA Access Manager Directory" section.

Intercomponent Security Settings

Intercomponent security settings are designed to secure communication channels between RSA Access Manager Servers and Agent, as well as between the RSA Access Manager web application and external systems or components.

Additionally, these security settings help RSA Access Manager Server components, specifically the Dispatcher, Authorization Server, and Entitlements Server, to communicate securely among themselves.

SSL between RSA Access Manager Servers and Agents

Use SSL encryption to help secure communications between RSA Access Manager Servers and Agents.

Mutually authenticated SSL mode

Setting	`cleartrust.net.ssl.use`
Configuration File Location	*AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the communications mode used between RSA Access Manager Servers and Agents. The server can be configured to use any of the following: • `Clear` - Clear text (no encryption) • `Anon` (default) - Anonymous SSL (SSL encryption with no certificate authentication) • `Auth` - Mutually authenticated SSL (SSL encryption with PKI certificate authentication) For more information on setting up mutually authenticated SSL between Servers and Agents, go to the *Servers Installation and Configuration Guide*, locate the "Implementing Security Features" chapter, and see the "Configuring Mutually Authenticated SSL" section.
RSA Recommendations	For stronger security, use `Auth`.

CA Keystore File

Setting	`cleartrust.net.ssl.ca.keystore_file`

Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the name of the CA keystore file. This file is used to validate the certificate chain of clients and servers. For additional information on using this parameter, consult the configuration file.

CA Keystore Type

Setting	`cleartrust.net.ssl.ca.keystore_type`
Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the type of CA keystore. For additional information on using this parameter, consult the configuration file.

CA Keystore Provider

Setting	`cleartrust.net.ssl.ca.keystore_provider`
Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the provider of the keystore algorithm used for unlocking and using the CA keystore. For additional information on using this parameter, consult the configuration file.

CA Keystore Passphrase

Setting	`cleartrust.net.ssl.ca.keystore_passphrase`
Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the password required to unlock the CA keystore. For additional information on using this parameter, consult the configuration file.
RSA Recommendations	Encrypt this parameter. For more information, see "Encrypting Configuration File Parameters" on page 23.

Private Keystore File

Setting	`cleartrust.net.ssl.private.keystore_file`
Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the keystore file where the private key of the server is stored. For additional information on using this parameter, consult the configuration file.

Private Keystore Type

Setting	`cleartrust.net.ssl.private.keystore_type`
Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path

Description	Specifies the type of keystore where the private key is stored. For additional information on using this parameter, consult the configuration file.

Private Keystore Provider

Setting	`cleartrust.net.ssl.private.keystore_provider`
Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the keystore algorithm used for unlocking and using the private keystore. For additional information on using this parameter, consult the configuration file.

Private Keystore Passphrase

Setting	`cleartrust.net.ssl.private.keystore_passphrase`
Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the password required to unlock the keystore that holds the private key. For additional information on using this parameter, consult the configuration file.
RSA Recommendations	Encrypt this parameter. For more information, see "Encrypting Configuration File Parameters" on page 23.

Private Key Alias

Setting	`cleartrust.net.ssl.private.key_alias`
Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the common name of the private key in the keystore. For additional information on using this parameter, consult the configuration file.

Private Key Passphrase

Setting	`cleartrust.net.ssl.private.key_passphrase`
Configuration File Location	*AXM_HOME*/conf/eserver.conf *AXM_HOME*/conf/dispatcher.conf *AXM_HOME*/conf/keyserver.conf *AXM_HOME*/conf/aserver.conf *AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the password required to unlock the private key specified by `cleartrust.net.ssl.private.key_alias`. Use a canonical path, or a relative path from the /conf folder.
RSA Recommendations	Use a strong ACL policy, and allow file access only to the RSA Access Manager service account. Encrypt this parameter. For more information, see “Encrypting Configuration File Parameters” on page 23. For additional information on using this parameter, consult the configuration file.

Peer Verification

All incoming connections to RSA Access Manager Servers should be from trusted sources. To help ensure this, you can configure the Authorization Server to verify the identity of the clients, typically RSA Access Manager Agents or Runtime API clients, that are connecting to it.

Verify Peer CN

Setting	`cleartrust.net.ssl.verify_peer_cn`
Configuration File Location	*AXM_HOME*/conf/aserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Available when mutually authenticated SSL is enabled. It determines whether the Server verifies the common name (cn) in client certificates. Used only when `cleartrust.net.ssl.use=Auth`. Use `true` or `false` (default). For more information, go to the *Servers Installation and Configuration Guide*, locate the "Implementing Security Features" chapter, locate the "Configuring Mutually Authenticated SSL" section, and see the "Peer Verification" section.

DN Checks

The following two parameters, when enabled and set, allow the Authorization Server to validate the DN in the certificate of the client (an RSA Access Manager Agent or the Runtime API) connecting to it. DN validation helps prevent token impersonation using the CTSESSION token, which is used to create the session cookie.

Authorization Server DN Checks

Setting	`cleartrust.aserver.token_api.enable`
Configuration File Location	*AXM_HOME*/conf/aserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Allows the Authorization Server to validate the DN in the certificate of the clients connecting to it. Valid values are `True` and `False` (default).
RSA Recommendations	Enable DN checks: `cleartrust.aserver.token_api.enable=True`

Authorization Server Trusted DN List

Setting	`cleartrust.aserver.token_api.trusted_dn_list`
Configuration File Location	*AXM_HOME*/conf/aserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Ensures that only clients whose DN has been specified can invoke APIs of the Authorization Server and get a token returned from the Authorization Server. For additional information on using this parameter, consult the configuration file.
RSA Recommendations	Enable this parameter to validate the Runtime API client connection.

SSL between RSA Access Manager Servers and Web Applications

In the *Servers Installation and Configuration Guide*, the "Implementing Security Features" chapter provides detailed information on configuring SSL between RSA Access Manager Servers and the following components:

- RSA Access Manager Administrative Console
- RSA Access Manager User Self Service
- RSA Access Manager Runtime Web Service
- RSA Access Manager Administrative Web Service

SSL between the Entitlements Server and the Administrative Console

Setting	`cleartrust.eserver.api_port.use_ssl`
Configuration File Location	*AXM_HOME*/conf/eserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path

Description	Specifies the type of encryption for communications between the Administrative Console (or Administrative API clients) and the Entitlements Server. Allows you to disable SSL for the Administrative API port on the Entitlements Server when the rest of the system is using SSL. Applies to both C and Java Administrative API clients. Another parameter, `cleartrust.net.ssl.use`, controls the SSL settings between the Entitlements Server and the other RSA Access Manager Servers. Allowed values are: • `Clear` - Clear text (no encryption) • `Anon` (default) - Anonymous SSL (SSL encryption with no certificate authentication) • `Auth` - Mutually authenticated SSL (SSL encryption with PKI certificate authentication)
RSA Recommendations	For stronger security, use `Auth`.

SSL between the Administrative Console and the Web Browser

Use the following parameter to enable encryption between the Administrative Console and the web browser.

Administrative Console SSL Encryption

Setting	`cleartrust.admingui.browser.use.ssl`
Configuration File Location	*AXM_HOME*/webapps/admingui.cfg where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the communication between the Administrative Console and the web browser. For more information on configuring the RSA Access Manager Administrative Console, go to the *Servers Installation and Configuration Guide*, locate the “Installing the RSA Access Manager Administrative Console” chapter, and see the “Configuring the RSA Administrative Console” section.
RSA Recommendations	Enable this parameter for the Administrative GUI to process only HTTPS requests: `cleartrust.admingui.browser.use.ssl=on`

Data Security Settings for Data at Rest

Data security settings are intended to prevent permanently stored product data from being disclosed in an unauthorized manner.

Encrypt Configuration Files

For all methods that can be used to encrypt configuration files or individual parameters, see “Encrypting Configuration File Parameters” on page 23.

Server Authenticated SSL

Use server authenticated SSL that helps to ensure secure communications between RSA Access Manager Servers and the LDAP data store.

Observe the following requirements:

- The LDAP directory host must be configured to accept SSL traffic.
- The SSL and keystore parameters must be set in **ldap.conf**.

For more information on server authenticated SSL, go to the *Servers Installation and Configuration Guide*, and see the “Implementing Security Features” chapter.

Server Authenticated SQL

Use server authenticated SQL that can help secure communications between RSA Access Manager Servers and the SQL datastore.

Observe the following requirements:

- The SQL server host must be configured to accept JDBC.
  For example:

  ```
  SQL Server: add encrypt=true to jdbc <URL>;
  ```

  where *URL* can be:

  jdbc:sqlserver://win2k.currey.com:1433;databaseName=CT;encrypt=true
- The TCPS protocol must be specified in the JDBC URL.
  For example:

  ```
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST
  =win2k.currey.com)(PORT=1521))(CONNECT_DATA=(SID=orcl)))
  ```

Password Hash Algorithm

Setting	**SQL:** `cleartrust.data.sql.user.password_hash_algorithm` **LDAP**: `cleartrust.data.ldap.user.password_hash_algorithm`
Configuration File Location	*AXM_HOME*/conf/sql-mssql.conf *AXM_HOME*/conf/sql-oracle.conf *AXM_HOME*/conf/sql-sybase.conf *AXM_HOME*/conf/ldap-activedirectory.conf *AXM_HOME*/conf/ldap-activedirectory-adam.conf *AXM_HOME*/conf/ldap-edirectory.conf *AXM_HOME*/conf/ldap-iplanet.conf where *AXM_HOME* is the RSA Access Manager Server installation path

Description	Specifies the algorithm that RSA Access Manager uses to encrypt user passwords in the LDAP user directory or SQL database. Used only when setting a new password value. All RFC-compliant hash algorithms are supported for password validation, regardless of what is entered here. Allowed values are: • `SSHA` (default) - Salted SHA1, which is more secure than SHA1. • `SSHA256` - Salted SHA-256, which is more secure than SSHA. • `SHA` - SHA1, which is generally considered more secure than MD5. • `MD5` - MD5 message digest algorithm. • `CRYPT` - UNIX-style CRYPT that uses a two letter salt and a variant of DES. Passwords encrypted in this format are compatible with standard UNIX /etc/password (or /etc/shadow) files. • `CLEAR` - Clear text, which is highly discouraged. • `PASSTHRU` - No password encryption or algorithm specifier. Equivalent to `CLEAR` without the {CLEAR} algorithm specifier when the password is stored. For use with directories that perform transparent password encryption on the server side.
RSA Recommendations	Use a strong password hashing algorithm for storing passwords. `SSHA256` is recommended.

Other Security Considerations

Session Replay Protection

RSA Access Manager Servers are designed to protect against cookie replay for logged out users. This feature is disabled by default, but RSA recommends enabling it.

For more information, go to the *Servers Installation and Configuration Guide*, locate the “Implementing Security Features” chapter, and see the “Session Replay Protection” section.

SNMP Configuration

RSA Access Manager supports SNMPv1, SNMPv2c, and SNMPv3. RSA recommends using SNMPv3 to help connect to an NMS server.

Instrumentation Server SNMP Version

Setting	`cleartrust.iserver.snmp.version`
Configuration File Location	*AXM_HOME*/conf/iserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path

Description	Specifies the SNMP version. Allowed values are: • `1` - SNMPv1 • `2` (default)- SNMPv2c • `3` - SNMPv3 For more information on configuring SNMP, go to the *Servers Installation and Configuration Guide*, locate the "Simple Network Management Protocol Support" chapter, and see the "Installing and Configuring the Instrumentation Server" section.
RSA Recommendations	Use SNMPv3 to connect to an NMS server.

Encrypting Configuration File Parameters

RSA Access Manager provides the following methods for encrypting configuration file parameters:

- The encryption utility **cryptedit** enables you to encrypt individual configuration file parameters that contain sensitive information, such as IP addresses, port numbers, and credentials. Using cryptedit, you may encrypt configuration parameters in the following files: **aserver.conf**, **eserver.conf**, **dispatcher.conf**, **keyserver.conf**, **ldap.conf**, **sql.conf**, and **adaptive_auth-onpremise-6021.conf**.

  For more information, go to the *Servers Installation and Configuration Guide*, locate the "Implementing Security Features" chapter, and see the "Encrypting Parameters in the Configuration Files" section.

- The encryption utility **manage-config** enables you to encrypt or decrypt all configuration files in *AXM_HOME*/conf.

  For more information, go to the *Servers Installation and Configuration Guide*, locate the "Implementing Security Features" chapter, and see the "Securing the Configuration Files" section.

- The encryption utility **encryptutil** enables you to encrypt the following configuration parameters:
  - `com.rsa.axm.selfservice.adapi.user_id` (*AxM_HOME*/webapps/ axm-selfservice-gui-*.war/selfservice.conf)
  - `com.rsa.axm.selfservice.adapi.user_password` (*AxM_HOME*/webapps/ axm-selfservice-gui-*.war/selfservice.conf)
  - `com.rsa.axm.selfservice.ssl.ca.keystore_passphrase` (*AxM_HOME*/webapps/ axm-selfservice-gui-*.war/selfservice.conf)
  - `com.rsa.axm.selfservice.ssl.private.keystore_passphrase` (*AxM_HOME*/webapps/ axm-selfservice-gui-*.war/selfservice.conf)
  - `com.rsa.axm.selfservice.ssl.private.key_passphrase` (*AxM_HOME*/webapps/ axm-selfservice-gui-*.war/selfservice.conf)

- `axm:securityPassphrase`
  (*AxM_HOME*/conf/snmp-access-policy.xml)
- `axm:privacyPassphrase`
  (*AxM_HOME*/conf/snmp-access-policy.xml)

For more information, go to the *Servers Installation and Configuration Guide*, locate the "Implementing Security Features" chapter, and see the "Running the EncryptUtil tool" section.

Securing the Web Services Description Language

You must use security constraints that are designed to secure Web Services Description Language (WSDL) generated by Administrative and Runtime web services.

To secure the WSDL generated by Administrative Web Services:

1. Go to **WEB-INF** in the directory where you unzipped the **ws-admin-api.war** file, and open **web.xml**.
2. Include the following text in **web.xml**.file of Administrative web service:

```
<context-param>
<param-name>
  cleartrust.ws.admin.api.secure_wsdl
</param-name>
<param-value>false</param-value>
</context-param>
<filter>
<filter-name>SecureWSDLFilter</filter-name>
<filter-class>sirrus.ws.admin.filters.SecureWSDLFilter
</filter-class>
<init-param>
  <param-name>ADMIN_ROLE</param-name>
  <param-value>Default Administrative Role</param-value>
</init-param>
<init-param>
  <param-name>ADMIN_GROUP</param-name>
  <param-value>Default Administrative Group</param-value>
</init-param>
<init-param>
<param-name>FORM_PAGE</param-name>
<param-value>displaywsdl.jsp</param-value>
</init-param>
</filter>
<filter-mapping>
<filter-name>SecureWSDLFilter</filter-name>
<url-pattern>/services/AdminAPI</url-pattern>
</filter-mapping>
```

3. Save **web.xml** and restart the application server.

To secure the WSDL generated by Runtime Web Services:

1. Go to **WEB-INF** in the directory where you unzipped the **ws-runtime-api.war** file, and open **web.xml**.
2. Include the following text in **web.xml**.file of Runtime web service:

```
<context-param>
<param-name>
    cleartrust.ws.rtapi.secure_wsdl
</param-name>
<param-value>false</param-value>

</context-param>
<context-param>
<param-name>
  cleartrust.ws.rtapi.admin_api.hostname
</param-name>
<param-value>localhost</param-value>
<description>
  This parameter is used to specify the hostname of the
  entitlement Server.
</description>
</context-param>
<context-param>
<param-name>cleartrust.ws.rtapi.admin_api.port</param-name>
<param-value>5601</param-value>
<description>
  This parameter is used to specify the port number of the
  entitlement Server.
</description>
</context-param>
<context-param>
<param-name>
  cleartrust.ws.rtapi.admin_api.timeout
</param-name>
<param-value>60000</param-value>
<description>
  This parameter is used to specify the timeout period in
  milliseconds for the entitlement server.
</description>
</context-param>

<filter>
<filter-name>SecureWSDLFilter</filter-name>
<filter-class>sirrus.ws.runtime.SecureWSDLFilter</filter-cla
ss>
<init-param>
  <param-name>ADMIN_ROLE</param-name>
  <param-value>Default Administrative Role</param-value>
</init-param>
<init-param>
  <param-name>ADMIN_GROUP</param-name>
  <param-value>Default Administrative Group</param-value>
</init-param>
<init-param>
<param-name>FORM_PAGE</param-name>
```

```
<param-value>displaywsdl.jsp</param-value>
</init-param>
</filter>

<filter-mapping>
<filter-name>SecureWSDLFilter</filter-name>
<url-pattern>/services/CTAuthService</url-pattern>
</filter-mapping>
```

3. Save **web.xml** and restart the application server.

SSL for Tomcat and WebLogic Application Servers

RSA Access Manager is designed to support secure connections with anonymous and mutually authenticated SSL between the Runtime and Administrative Web Services and your application server.

For information on setting up SSL for these instances, go to the *Servers Installation and Configuration Guide*, locate the "Deploying Runtime and Administrative Web Services" chapter, and see the "Application Server SSL Configuration" section.

Using Windows Authentication with Microsoft SQL Server

You can configure the SQL data adapter to use Windows authentication with Microsoft SQL Server. For more information, go to the *Servers Installation and Configuration Guide*, locate the "Installing and Configuring the SQL Data Adapter" chapter, and see the "Configuring SQL Adapter with Microsoft SQL Server for Integrated Authentication" section.

Server Platform Updates with Security Fixes

Apply all available security patches or fixes to the RSA Access Manager Server operating system.

Apache HTTP Server Default Cache Configuration and Cookie Security

For information on the Apache module "mod_cache", consult the Apache documentation at **http://www.apache.org/**.

With respect to RSA Access Manager, RSA notes the following:

By default, the Apache module "mod_cache" caches HTTP content including cookies. In this default configuration, when a user is accessing a protected resource, RSA's CTSESSIONS cookie is cached, and until it expires, it is sent to other users who request the same page. The result is that a user can access a resource using a previous user's logon credentials.

To prevent this scenario, modify your Apache configuration (**httpd.conf**) using the following methods:

a. Add the `CacheIgnoreHeaders` directive to specify that Set-Cookie and Set-Cookie2 headers should not be cached:
`CacheIgnoreHeaders Set-Cookie Set-Cookie2`

Note: This directive became available in Apache HTTP Server 2.0.54 and later, and is also available in versions 2.2 and 2.4.

b. Add the `Header` directive and the `Cache-Control` header to specify that Set-Cookie and Set-Cookie2 headers should not be cached at any level:
```
Header set Cache-Control "no-cache=set-cookie,
set-cookie2"
```

For additional Apache security considerations, refer to the *Apache Caching Guide* at **http://httpd.apache.org/docs/2.2/caching.html**.

Deploying Access Manager Server

You must plan the physical deployment of your organization like servers, data stores, and so on before you install the software to help ensure a smooth implementation that suits the specific needs of your organization.

You must also plan the logical deployment of your organization like protecting the resource, providing access to the resource, applying security policies and so on to take inventory for the security needs of your organization.

To deploy the components of your organization securely, see *Planning Guide*.

To deploy RSA Access Manager Applications like User Self-Service Console, Runtime and Administrative Web Services, and Adminitrative Console, see *Servers Installation and Configuration Guide*.

Secure Deployment and Usage Settings for Servers

Use the following configuration settings that helps secure the RSA Access Manager Server deployment.

HTTPS Settings

To help secure communications between web browsers and web applications RSA recommends HTTPS protocol. RSA also recommends to use non-self-signed SSL Certificate and the certificate that support high cipher suites.

The following components can be deployed in HTTPS mode:

- RSA Access Manager Administrative Console
- RSA Access Manager Self-Service Console
- RSA Access Manager Administrative web services
- RSA Access Manager Runtime web services

For more information on deploying the web application in HTTPS mode, consult the documentation for your application server.

Refer to your organization’s security policy to remove or harden security for the folders that are exposed by the application server. Also, on the application server, configure the **HTTPOnly** and **Secure** flags for cookies accordingly. For more information, consult the documentation on the application server.

Reverse Proxy in DMZ

If you are using the self-service console outside the enterprise network, in such cases, rather than deploying the self-service console in DMZ, it is recommended that you deploy a reverse proxy in DMZ, so that the reverse proxy then forwards the request to the self-service console deployed inside the network.

Configuring Shared Secret Encryption

The shared secret helps with authentication and secure communication with the Key Server. The secret is stored in a text file in the RSA Access Manager installation directory. It should be changed periodically in accordance with your organization's security policies.

For more information, go to the *Servers Installation and Configuration Guide*, locate the "Deploying RSA Access Manager in Production Environments" chapter, and see the "Generating a Shared Secret Using Keygen" section.

Deploying Components Across a Firewall

Each RSA Access Manager component is configured separately, and may be placed inside or outside the firewall, regardless of how the other components are configured.

For any two RSA Access Manager components to communicate across a firewall, you must configure the firewall to allow connections between these two systems on a specific port.

For more information, go to the *Servers Installation and Configuration Guide*, locate the "Deploying RSA Access Manager in Production Environments" chapter, and see the "Deploying Components Across a Firewall" section.

Configuring Two-Factor Authentication

RSA Authentication Manager

RSA Access Manager supports RSA SecurID two-factor authentication to validate the user's passcode against the credential stored in RSA Authentication Manager. A user account with the same user name must also exist in RSA Access Manager.

For more information, go to the *Servers Installation and Configuration Guide*, locate the "Supported Authentication Types" chapter, and see the "RSA SecurID Authentication" section.

RSA Adaptive Authentication

RSA Access Manager supports two-factor authentication with RSA Adaptive Authentication. First-level authentication is performed by the RSA Adaptive Authentication Server, and second-level authentication is performed by RSA Access Manager.

For more information, go to the *Servers Installation and Configuration Guide*, locate the "Supported Authentication Types" chapter, and see the "RSA Adaptive Authentication" section.

Physical Security Controls for Servers

Physical security controls help protect resources against unauthorized physical access and physical tampering.

RSA recommends the following:

- The physical servers in the RSA Access Manager deployment should be located in a secure data center that leverages the organization's best practices for physically securing a data center, server rack, and/or server.
- File-level permissions for configuration files, startup scripts, and log files should be hardened according to your organization's ACL policy.

FIPS Mode for RSA Access Manager Components

RSA Access Manager provides an option to run RSA Access Manager components in FIPS 140 mode. By enabling FIPS mode, RSA Access Manager uses only FIPS-approved algorithms for encryption processes. RSA recommends running RSA Access Manager Servers in FIPS mode.

Note: FIPS mode is disabled by default. For information on enabling FIPS mode, go to the *Servers Installation and Configuration Guide*, locate the "Implementing Security Features" chapter, and see the "Enabling FIPS Mode" section.

Use the following parameter to specify the algorithm for the token that sets the CTSESSION cookie.

Authorization Server Token Version

Setting	`cleartrust.aserver.token_version`
Configuration File Location	*AXM_HOME*/conf/aserver.conf where *AXM_HOME* is the RSA Access Manager Server installation path
Description	Specifies the algorithm for the token that is used to set the CTSESSION cookie. Allowed values are: • `2` for the algorithm MD5 • `3` for the FIPS-compliant algorithm SHA1 • `4` for the FIPS-compliant algorithm SHA256 • `5` for the FIPS-compliant algorithm SHA512
RSA Recommendations	Use `4` or `5`.

For more information on FIPS 140, go to **http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf**.

Additional Documentation on Server Security Features

The *Servers Installation and Configuration Guide* provides detailed information on product security configuration, which includes some features that are mentioned in this guide. It also includes information on the following:

- **Configuring server authenticated SSL** - This configuration helps to encrypt communications between the Entitlements and Authorizations Servers and your LDAP directory host. This section includes instructions on generating CA certificates using RSA Certificate Manager, and adding certificates to the keystore of each RSA Access Manager Server using the Access Manager Certificate Tool or Sun Java Keytool.

  For more information, go to the *Servers Installation and Configuration Guide*, locate the “Implementing Security Features” chapter, and see the “Configuring Server Authenticated SSL” section.

- **Configuring mutually authenticated SSL** - This configuration helps to ensure that only authorized clients, or “peers”, are using RSA Access Manager Servers. This section includes instructions on generating CA certificates using RSA Certificate Manager.

  For more information, go to the *Servers Installation and Configuration Guide*, locate the “Implementing Security Features” chapter, and see the “Configuring Mutually Authenticated SSL” section.

- **Using HTTPS with RSA Adaptive Authentication Servers** - For environments in which RSA Access Manager integrates with RSA Adaptive Authentication, this feature helps to secure the communication between RSA application servers.

  For more information, go to the *Servers Installation and Configuration Guide*, and see the “Integrating RSA Access Manager with RSA Adaptive Authentication” chapter.

- **Configuring SSL for the RSA Administrative Console** - This configuration helps secure browser-to-manager connections using anonymous SSL.

  For more information, go to the *Servers Installation and Configuration Guide*, locate the “Installing the RSA Access Manager Administrative Console” chapter, and see the “Configuring the RSA Administrative Console” section.

- **Applying custom password policy requirements** - During different phases of authentication and authorization, you can call custom code using listener classes, for example, if you want to run your own compliance tests for additional password policy requirements. For passwords that fail compliance tests, you can create custom error messages.

  For more information, go to the *RSA Access Manager Developer’s Guide*, locate “Code Examples” in the navigation pane, and see “PasswordHookEventExample.java”.

- **Configuring password restrictions** - In addition to the RSA Access Manager password policy feature, you can set password restrictions that are validated when a user is created or modified.

  For more information, go to the *Servers Installation and Configuration Guide*, locate the “Enhanced Functionality” appendix, and see the “Configuring Password Restrictions” section.

2 Security Configuration Settings on RSA Agents

Locations of Agent Configuration Files and Utilities

RSA Access Manager Agent utilities are located in ***AGENT_HOME*/bin** where *AGENT_HOME* is the Agent installation path.

RSA Access Manager Agent configuration parameters are located in ***CT_AGENT_ROOT*/conf/webagent.conf** where *CT_AGENT_ROOT* is one of the following:

Platform	Location
Windows	Agent installation path
UNIX (Domino only)	Agent installation path
UNIX (all servers except Domino)	*AGENT_HOME*/webservers/<instance -name> where *AGENT_HOME* is the Agent installation path

Security Configuration Settings for Agents

This section provides an overview of the settings available in the RSA Access Manager Agent that helps with its secure operation. Security settings are divided into the following categories:

Access Control Settings for User Authentication and Authorization. Describes settings that limit access by end users or external Agent components.

Log Settings. Describes settings related to event logging.

Intercomponent Security Settings. Describes security settings related to Agent network communications.

Data Security Settings. Describes settings that ensure protection of the data that is handled by the Agent.

Proxy Configurations. Describes security settings that are used to secure proxy configurations.

Access Control Settings for User Authentication and Authorization

Access control settings help in protecting the resources against unauthorized access.

User authentication settings control the process of verifying a user's identity, allowing access to the RSA Access Manager deployment, and authorizing access to requested resources.

The following configuration parameters help control access to protected resources, and work in conjunction with RSA Access Manager Servers to determine whether an URL is protected.

Agent Authentication Methods and Resources List

Setting	`cleartrust.agent.auth_resource_list`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies a list of comma-separated URLs and the authentication methods that are required to access to them.
RSA Recommendations	Run the Authorization Server in passive mode to ensure that all resources are protected by default. For more information, go to "Authorization Server Mode" on page 7.

Agent Default Auth Mode

Setting	`cleartrust.agent.default_auth_mode`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the default authentication type for protected resources that are not defined by the `cleartrust.agent.auth_resource_list` parameter. This configuration does not apply to resources that are not protected in the RSA Entitlements Server.
RSA Recommendations	Run the Authorization Server in passive mode to ensure that all resources are protected by default. For more information, go to "Authorization Server Mode" on page 7.

Agent for handling Intersite Single Sign-on Slave Authentication at Authorization Server

Setting	`cleartrust.agent.isso.handle_slave_auth_at_aserver`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	This parameter is provided to handle the creation and verification of signatures, using Authorization server for slave authentication. Agents prior to 5.0 used to handle it by retreiving the encryption and decryption keys from Key server. When this parameter is set to True, Agent uses Authorization runtime APIs for slave authentication. When this parameter is set to False, Agent retreives session keys from the Key server and handles signature verification by itself.
RSA Recommendations	Configure this parameter to a value of True so that Agent will use a runtime API to communicate with the Authorization server to create or verify a signature. This results in handling sensitive information within secure network.

Agent URL Exclusion List

Setting	`cleartrust.agent.url_exclusion_list`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies a list of URLs that are excluded from access control checks. URLs in this list are unprotected, and are not subject to Agent authentication.
RSA Recommendations	Configure this parameter using specific URLs instead of wildcards, which can unintentionally allow access to URLs that should be protected.

Agent Extension Exclusion List

Setting	`cleartrust.agent.extension_exclusion_list`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.

Description	Specifies a list of file extensions that are excluded from access control checks.
	RSA deprecates the use of this legacy parameter, and recommends using the Agent rules engine (**rules.xml**) to specify more specific URL patterns that are excluded from access control checks.
	Note: Any URL with a specified extension is excluded from access control checks. This can potentially exclude a lot of namespace URLs from access control checks. Also, this can expose the web server to URL exploits.
	For example, to exclude all .jpg and .gif files in /cleartrust/images/ from access control checks, configure a rule that is similar to the following:
	`<Rule>`
	`<argument type="URI"`
	`expression="^/cleartrust/images/[0-9a-z]*\.jpg"/>`
	`<action type="HTTP" argument="200"/>`
	`</Rule>`
	`<Rule>`
	`<argument type="URI"`
	`expression="^/cleartrust/images/[0-9A-Za-z_]*\.gif$"/>`
	`<action type="HTTP" argument="200"/>`
	`</Rule>`
	For rules.xml usage, refer to the `cleartrust.agent.rules_file` parameter in the configuration file.
	For more information on rules.xml, go to the *RSA Access Manager Agent 5.0 SP1 for Web Servers Installation and Configuration Guide*, locate the "General Configuration" chapter, and see the "Agent Rules Engine" section.

RSA Recommendations	Write exclusion rules as specific as possible, and apply them to a minimum set of resources. This reduces the risk of unintentionally excluding a resource that should be protected.

Important: To help protect all server resources, RSA recommends running the Authorization Server in passive mode, and providing granular access levels using the RSA Entitlements Server and a combination of the following:

- `cleartrust.agent.auth_resource_list` with chained authentication using OR(:) and AND(+) operators
- `cleartrust.agent.url_inclusion_list`
- `cleartrust.agent.url_exclusion_list`, leaving unspecified URLs to be protected under `cleartrust.agent.default_auth_mode`

URL definitions in the Entitlements Server should include all or most web server resources. The resources that do not need to be protected should be specifically listed using `cleartrust.agent.url_exclusion_list`, so that a web server with an unsecure configuration, such as directory listing enabled, remains protected. (Alternately, run the Authorization Server in passive mode, which protects all web server resources by default. For more information, go to "Authorization Server Mode" on page 7.)

For more information on using these methods, go to the *RSA Access Manager Agent 5.0 SP1 for Web Servers Installation and Configuration Guide*, and see the "Configuring and Specifying Authentication Types" chapter.

Log Settings

Error and Debug Logs

The RSA Access Manager Agent log location is configured in each instance's **webagent.conf** file. By default, the location is under the following instance directory: ***AGENT-ROOT*/logs/**

You can configure the log location at the installation level, which sets the default value for each instance. For each instance, you can use the default value or choose a different location. RSA recommends that you configure the default log location at the installation level, and use the default location for every instance.

Set the maximum log file size to 50 MB using `cleartrust.agent.log_file_rotation_maxsize`. (When the log file reaches the maximum size, the logs rotate.)

Do not set the log level above "Critical" for production web servers. This ensures that only important messages and errors are logged, while potentially sensitive information, such as user names and authentication results, are not logged.

Depending on the logging level that has been set for the instance, the following items may be logged:

- Server start/stop events
- Errors pertaining to configuration, communication, and security
- Information related to processing individual requests

Directory Permissions

To help secure logs directory, RSA recommends that you restrict permissions on the logs directory to the minimum required permissions: read and write.

Windows: Permissions must be assigned to "NETWORK_SERVICE", the service account for web server processes.

UNIX-based systems: Permissions must be assigned to the user account under which the web server runs.

To review the permissions on the logs directory:

1. Log on to the RSA Access Manager Server.
2. Do one of the following:
   - **Windows**: Locate the log file directory. Right-click on the folder, and select **Properties**. Go to the **Permissions** tab.
   - **UNIX**: Navigate to the log file directory in a terminal, and run the following command:
     ```
     ls -ld
     ```
3. Confirm that NETWORK_SERVICE (Windows) or the user account under which the web server runs (UNIX-based systems) has the required permissions.

Intercomponent Security Settings

Intercomponent security settings help with securing the communication channels between RSA Access Manager Servers and Agents, as well as between the RSA Access Manager web application and external systems or components.

SSL between Agent and Servers

Use the following parameter to enable SSL encryption and secure the communication between RSA Access Manager Servers and Agents.

Agent SSL Encryption

Setting	`cleartrust.agent.ssl.use`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.

Description	Specifies the communications mode used between RSA Access Manager Servers and Agents. Allowed values are: • `Clear` - Clear text (no encryption) • `Anon` - Anonymous SSL (SSL encryption with no certificate authentication) • `Auth` (default) - Mutually authenticated SSL (SSL encryption with PKI certificate authentication) For more information, go to the *RSA Access Manager Agent 5.0 SP1 for Web Servers Installation and Configuration Guide*, and see the "Configuring and Specifying Authentication Types" chapter.
RSA Recommendations	For stronger security, use `Auth`.

The following configuration parameters need to be set appropriately when this configuration is set to 'Auth'.

Agent Private Key Keystore

Setting	`cleartrust.agent.ssl.keystore`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the keystore name of the PKCS #12 keystore that contains the Agent's private key.
RSA Recommendations	Set this parameter to the PKCS #12 keystore file name, either an absolute file path or a file name relative to the Agent's conf folder. Ensure that only authorized users have access to the private key file.

Agent Keystore Passphrase

Setting	`cleartrust.agent.ssl.keystore_passphrase`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the passphrase that is used to verify the integrity of the PKCS #12 keystore containing the private key.

RSA Recommendations	Set this parameter as `cleartrust.agent.ssl.keystore_passphrase.cleartext=false` to ensure that the parameter is defined in an encrypted store instead of being stored as clear text in webagent.conf. For more information, refer to the `cleartrust.agent.encrypted_store` parameter in the configuration file. Set a strong passphrase in compliance with your organization's security policies. The passphrase is case-sensitive.

Agent Private Key Passphrase

Setting	`cleartrust.agent.ssl.private_key_passphrase`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the passphrase that is used to decrypt the private key in the PKCS #12 private-key keystore.
RSA Recommendations	Set this parameter as `cleartrust.agent.ssl.private_key_passphrase.cleartext=false` to ensure that the parameter is defined in an encrypted store instead of being stored as clear text in webagent.conf. For more information, refer to the `cleartrust.agent.encrypted_store` parameter in the configuration file. Set a strong passphrase in compliance with your organization's security policies. The passphrase is case-sensitive.

Agent Private Key Alias

Setting	`cleartrust.agent.ssl.private_key_alias`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the alias of the private key in the PKCS #12 private-key keystore.
RSA Recommendations	Specify an alphanumeric string (without spaces or special characters) for the private key alias.

Agent Certificate Keystore

Setting	`cleartrust.agent.ssl.ca_keystore`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the keystore name of the PKCS #12 keystore that contains the Agent's certificate.
RSA Recommendations	Set this parameter to the PKCS #12 keystore file name, either an absolute file path or a file name relative to the Agent's conf folder. Ensure that only authorized users have access to the file.

Agent CA Keystore Passphrase

Setting	`cleartrust.agent.ssl.ca_keystore_passphrase`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the passphrase that is used to verify the integrity of the PKCS #12 CA keystore.
RSA Recommendations	Set this parameter as `cleartrust.agent.ssl.ca_keystore_passphrase.cleartext=false` to ensure that the parameter is defined in an encrypted store instead of being stored as clear text in webagent.conf. For more information, refer to the `cleartrust.agent.encrypted_store` parameter in the configuration file. Set a strong passphrase in compliance with your organization's security policies. The passphrase is case-sensitive.

Web Server SSL

Use SSL encryption that helps secure the communications between the client browser and the web server. To do this, configure SSL-only connections between the client and the webservers. For more information on enabling SSL, consult the documentation on your web server.

Cookies over SSL

Restrict cookies to SSL connections. To do this, set the following parameter.

Agent Secure Cookie

Setting	`cleartrust.agent.secure`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies that the browser should accept and send cookies using only secure methods.
RSA Recommendations	Enable this parameter to restrict cookies to SSL connections: `cleartrust.agent.secure=True`

Data Security Settings

Data security settings are intended to prevent permanently stored product data from being disclosed in an unauthorized manner.

Encryption of Data at Rest: Cookie Security

Set the following configuration parameters that help ensure that cookies are stored securely in the client's browser, and that cookies are transferred securely between the Agent and client browser.

Agent Cookie Path

Setting	`cleartrust.agent.path`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the path on the web server where the SSO (single sign-on) cookie applies. **Note:** An empty value means that the current URL path is used, that is, '/cleartrust' is the path set for the cookie after successful authentication. This is not recommended.
RSA Recommendations	Set this parameter to be specific to the path to which the SSO cookie needs to be applied. Use '/' only if the SSO cookie should be applied to all resources on the web server.

Agent Cookie Expiration

Setting	`cleartrust.agent.cookie_expiration`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Sets the amount of time that a cookie persists in a browser.
RSA Recommendations	Set this parameter to `0 Mins` to ensure that the cookie is valid only until the browser exits.

Agent Cookie HttpOnly

Setting	`cleartrust.agent.httponly`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies whether the HttpOnly attribute is included in the SSO (single sign-on) cookie. Use `True` or `False` (default).
RSA Recommendations	Set this parameter to `True` so that cookies which are presented as part of http requests are not available to client-side scripts. This mitigates cross-site-scripting (XSS) attacks designed to steal session cookies.

Encryption of Data at Rest: Encryption Utilities

RSA Access Manager Agent is installed with utilities that helps you to encrypt sensitive configuration parameters in the **webagent.conf** file.

Unique User Sessions

Encrypted Store

Setting	`cleartrust.agent.encrypted_store`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.

Description	Specifies the filename for the encrypted store where sensitive configuration parameters can be stored. **Note:** This parameter needs to be enabled to use the **cryptedit** tool. When the cryptedit tool is run, it searches **webagent.conf** for `cleartext=false` entries and displays those parameters at the command prompt so the user can set their values. For more information on cryptedit, go to the *RSA Access Manager Agent 5.0 SP1 for Web Servers Installation and Configuration Guide*, and see the "Agent Utilities" chapter.
RSA Recommendations	Specify an absolute file path or a filename relative to the Agent's conf directory. Ensure that only authorized users have permissions to access to the file.

Agent Crypt Edit Utility

Setting	`ctagent_cryptedit[.exe]`
Configuration File Location	*AGENT_HOME*/conf/ctagent_cryptedit.exe where *AGENT_HOME* is the Agent installation path
Description	Encrypts sensitive configuration parameter settings for **webagent.conf**, such as the keystore passphrase.
RSA Recommendations	Encrypt all sensitive configuration parameters using **cryptedit**.

Agent Watchdog Utility

Setting	`ctagent_watchdog[.exe]`
Configuration File Location	*AGENT_HOME*/conf/ctagent_watchdog.exe where *AGENT_HOME* is the Agent installation path
Description	Stores the password that you assign to the file used for the cryptedit utility. Also, supplies the Agent with the password so that it can read the encrypted parameters, which allows the Agent to restart unattended.
RSA Recommendations	Use the **watchdog** utility to secure all encrypted configuration parameters using a master password. Record your master password in a secure location, where only authorized individuals are able to access it. For more information, go to the *RSA Access Manager Agent 5.0 SP1 for Web Servers Installation and Configuration Guide*, and see the "Agent Utilities" chapter.

Data Integrity: Cookie Integrity

Agent Cookie IP Check

Setting	`cleartrust.agent.cookie_ip_check`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Enables/disables session IP checking. When this setting is enabled, the Agent accepts cookies only from the same IP address to which they were originally issued. If the IP addresses do not match, the token is rejected as invalid, and the user is required to log on again. This feature safeguards against cookies that have been moved from one computer to another.
RSA Recommendations	Enable this parameter to mitigate cookie replay attacks: `cleartrust.agent.cookie_ip_check=True`

Agent Domain Checking

Setting	`cleartrust.agent.cookie_domain`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the domain name in the HTTP 'Set-Cookie' header for SSO (single sign-on) tokens.
RSA Recommendations	Restrict CTSESSION cookie distribution to the most restricted domain possible.

Agent Strict Cookie Set

Setting	`cleartrust.agent.strict_cookie_set`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies whether to set the CTSESSION SSO (single sign-on) cookie.

RSA Recommendations	Enable this parameter to ensure that the CTSESSION cookie is set only if the user has successfully authenticated with at least one of the supported authentication types: `cleartrust.agent.strict_cookie_set=True`

Data Integrity: URL Integrity

Agent Trusted Domains List

Setting	`cleartrust.agent.trusted_domains_list`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies a list of domain names to which the Agent is allowed to redirect users immediately after authentication. **Note:** You must add the domain name of the Agent's host if this parameter is enabled. For Agents in an ISSO environment, include master and slave domain names.
RSA Recommendations	Specify a list of URLs that the Agent can trust to prevent redirects to arbitrary URLs.

Data Erasure: Timeouts

Set the following configuration parameters to invalidate cookies after a period of inactivity.

Agent Idle Timeout

Setting	`cleartrust.agent.idle_timeout`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Sets the maximum amount of time between requests, after which sessions are considered idle and are invalidated, and the user is required to log on again. The default value is 15 minutes.

RSA Recommendations	Set this parameter to a value that is appropriate for your environment. A value that is too high or low may result in cookies not being invalidated or users being required to log on again frequently.

Agent POST URL Idle Timeout

Setting	`cleartrust.agent.post_url_idle_timeout`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Sets an additional amount of time for a session to remain valid when making HTTP POST requests to a specific set of URLs that are identified by the parameter `cleartrust.agent.post_url_idle_timeout_list`. Used primarily to work around the problem of a logged-on user's session timing out before he can submit a page due to the `cleartrust.agent.idle_timeout` setting.
RSA Recommendations	Do not set this parameter to a high value due to security implications.

Agent Session Lifetime

Setting	`cleartrust.agent.session_lifetime`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Sets the maximum lifetime of an SSO session. The default value is 8 hours.
RSA Recommendations	Set this parameter to a value that is appropriate for your environment. A value that is too low may result in users being required to log on again frequently.

Agent Cookie Touch Window

Setting	`cleartrust.agent.cookie_touch_window`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.

Description	Sets the amount of time that the Agent waits before updating the cookie for an authenticated user.
RSA Recommendations	Set this parameter to `<1 Minutes`. Do not set this parameter to a high value, such as greater than 5 minutes, because the "idle_timeout" is shortened by the period of time that is specified in this parameter.

Data Erasure: Cache Control

To help manage the caching of resources and cookies, RSA recommends that you use the following configuration settings:

Agent Protected Resources Cache TTL

Setting	`cleartrust.agent.protected_resource_cache_ttl`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Sets the protected resource status cache time to live (TTL). The default value is 10 minutes.
RSA Recommendations	Set this parameter to `10 Mins`, the default, so that cached entries are cleared after 10 minutes. Do not set this parameter to `0`, as the Agent would never prune the cache based on TTL.

Agent Unprotected Resources Cache TTL

Setting	`cleartrust.agent.unprotected_resource_cache_ttl`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Sets the unprotected resource status cache time to live (TTL). The default value is 5 minutes.
RSA Recommendations	Set this parameter to `5 Mins`, the default, so that cached entries are cleared after 5 minutes. Do not set this parameter to `0`, as the Agent would never prune the cache based on TTL.

Agent Token Cache TTL

Setting	`cleartrust.agent.token_cache_ttl`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Sets the cookie cache time to live (TTL). Cookies issued to the client browser can be cached in unencrypted form by the Agent for better performance. The default value is 5 Minutes.
RSA Recommendations	Set this parameter to `5 Mins`, the default, so that cached cookies are cleared after 5 minutes. Do not set this parameter to `0` or `> 5 Mins` to minimize cookie replay attacks. **Note:** Setting this parameter to `0` results in cached cookies never being cleared based on TTL.

Agent Token Cache Size

Setting	`cleartrust.agent.token_cache_size`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Sets the cookie cache size. When the maximum size is reached, cache entries are removed, oldest first. The default value is 10000.
RSA Recommendations	Set this parameter to `10000`, the default, so that the cache is pruned when it reaches 10000 entries. Do not set this parameter to `0`, as the Agent would never prune the cache based on cache size.

Note: The TTL and size-based cache control parameters work in conjunction with each other. For example, the Agent prunes a cache based on TTL or size, depending on which limit is exceeded first.

Proxy Configurations

Use the following configuration settings for securing proxy configurations.

Agent Trusted Proxy List

Setting	`cleartrust.agent.trusted_proxy_list`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies a comma-separated list of IP addresses which represent the hosts that are identified as trusted proxies. If `cookie_ip_check` is enabled and requests are from one of these hosts, and they contain a header as specified in `trusted_proxy_header_name`, this header IP is set in the cookie when the client authenticates. The proxies are "trusted" in the sense that if there was no list to check against, any client could spoof the header with any IP and it would be accepted as the client IP by the Agent.
RSA Recommendations	Set specific IP addresses, or a range of IP addresses instead of a broader subnet, to prevent spoofing a client address that is within the specified subnet but does not exist.

Agent Cookie IP Check

Setting	`cleartrust.agent.cookie_ip_check`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Enables/disables session IP checking. When session IP checking is enabled, the Agent accepts cookies only from the same IP address to which they were originally issued.
RSA Recommendations	Disable this configuration by setting it to `False` in load-balancing environments where the client IP address frequently changes, which results in cookies being rejected and users being required to log on again frequently. For proxies with static IP addresses, enable this parameter by setting it to `True` and exclude them from IP checks using `cleartrust.agent.ip_check_exclusion_list`.

Agent Cookie Exclusion List

Setting	`cleartrust.agent.cookie_exclusion_list`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies a comma-separated list of IP addresses which represent hosts that are not issued cookies.
RSA Recommendations	Set this parameter in proxy environments where both the proxy and content servers are protected by RSA Access Manager. This allows the content server to suppress generating a duplicate cookie, as the proxy has already performed this task.

Agent Cookie IP Check Exclusion List

Setting	`cleartrust.agent.ip_check_exclusion_list`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies a comma-separated list of host IP addresses that are allowed to act as proxies and forward cookies to this server, and that are not subjected to IP address checks.
RSA Recommendations	Use a specific list of IP addresses when possible. Specify proxy IP addresses to ensure that requests from hosts in this list are not subject to IP address checks. Use a restrictive subnet specification (in conjunction with `allow_subnet_masking`) to prevent unintended IP addresses from being treated like proxies and excluded from cookie checks.

Agent Trusted Proxy Strict Mode

Setting	`cleartrust.agent.trusted_proxy_strict_mode`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies the resulting behavior when a check against the `trusted_proxy_list` fails.
RSA Recommendations	For Internet sites that are accessible to the public, set this parameter to `False`, as users behind proxies not registered in the `trusted_proxy_list` would not be able to connect.

In environments without proxy servers, RSA recommends configuring the content servers to require IP checks.

In environments with proxy servers, RSA recommends that Agents are installed on both the proxy servers and the content servers. The content servers should be configured to IP check all cookies that come from machines other than the proxy servers (using `ip_check_exclusion_list`). Proxy server Agents are responsible for IP checking cookies in requests that are addressed to the proxy server(s). This effectively secures a reverse proxy configuration.

Note: The parameters `trusted_proxy_strict_mode`, `trusted_proxy_header_name`, and `trusted_proxy_list` apply only to configurations where:

- The Agent is installed only on the content web servers, and not on the proxy servers.
- The proxy servers can forward the client IP address in the headers.

Secure Deployment and Usage Settings for Agents

To help secure the deployment of the Agent, RSA recommends the following configuration settings.

Web Server Security

The web server where the Agent is deployed should be patched to the latest version, and hardened against misconfigurations, such as allowing malicious scripting, directory listing, etc. Refer to the respective web server's hardening guidelines for more information.

HTTP Settings

Agent Export Headers for Protected Resources Only

Setting	`cleartrust.agent.export_headers_for_protected_resources_only`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies whether HTTP Request headers should be published for protected resources only or for all resources.
RSA Recommendations	Enable this parameter to prevent HTTP Request Headers from being published for unprotected resources: `cleartrust.agent.export_headers_for_protected_resources_only=True`

Agent Strict Headers Export

Setting	`cleartrust.agent.strict_headers_export`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Specifies whether to publish CT_REMOTE_USER from the user header list even if the user has not successfully authenticated.
RSA Recommendations	Enable this parameter to ensure that CT_REMOTE_USER is not published as a HTTP header if user authentication failed due to account lockout or password expiration: `cleartrust.agent.strict_headers_export=True` Publishing this header for all valid users, regardless of their authentication status, might potentially enable an attacker to distinguish between valid and invalid users.

Agent Retain URL in Cookie Vs. Query String

Setting	`cleartrust.agent.retain_url.use_query_string`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Indicates how the Agent stores the original URL during URL retention. If the parameter is set to `True`, the original URL is appended as a query string to each logon form URL during authentication. If the parameter is set to `False` (default), a temporary cookie is used instead.
RSA Recommendations	Disable this parameter to have the Agent store the original URL in a cookie during URL retention: `cleartrust.agent.retain_url.use_query_string=False`

Agent Ignore HTTP Auth

Setting	`cleartrust.agent.ignore_http_auth`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.

Description	Instructs the Agent to ignore the user credential in HTTP-Authorization headers.
RSA Recommendations	Enable this parameter to prevent users from bypassing form logons: `cleartrust.agent.ignore_http_auth=True`

RSA Adaptive Authentication Settings

Agent AA Allow on Failure

Setting	`cleartrust.agent.aa.allow_on_failure`
Configuration File Location	*CT_AGENT_ROOT*/conf/webagent.conf For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.
Description	Determines the action to take when the Agent receives an Adaptive Authentication connection failure from an Authorization Server. The default value is `True`.
RSA Recommendations	Disable this parameter to avoid bypassing AA authentication when the AA servers are down: `cleartrust.agent.aa.allow_on_failure=False`

Generic Error Pages

RSA allows you to create custom error pages if you require additional usability in your environment.

Consider that custom error messages can increase an attacker's ability to confirm valid logon IDs. To help obtain optimum security, RSA recommends that logon failure pages be the same for all failures.

The Agent provides the following configurations for custom error pages. These configuration parameters are located in *CT_AGENT_ROOT*/conf/webagent.conf. For more information about *CT_AGENT_ROOT*, see "Locations of Agent Configuration Files and Utilities" on page 33.

Configuration	Description
`cleartrust.agent.login_error_user_location_basic`	Specifies the path and file location of the page RSA Access Manager issues when a user submits an invalid user ID for Basic authentication.

`cleartrust.agent.login_error_pw_location_basic`	Specifies the location of the page RSA Access Manager issues when a user submits an invalid password for Basic authentication.
`cleartrust.agent.login_error_location_securid`	Specifies the location of the page RSA Access Manager issues when an error occurs during RSA SecurID authentication.
`cleartrust.agent.login_error_user_location_nt`	Specifies the location of the page RSA Access Manager issues for Windows NT authentication.
`cleartrust.agent.login_error_pw_location_nt`	Specifies the location of the page RSA Access Manager issues when an invalid password error has occurred during Windows NT authentication.
`cleartrust.agent.login_error_password_expired`	Specifies the location of the page RSA Access Manager issues when the Basic user password is expired.
`cleartrust.agent.login_error_password_expired_forced`	Specifies the location of the page RSA Access Manager issues when the Basic user password is forced to expire by the administrator.
`cleartrust.agent.login_error_password_expired_new_user`	Specifies the location of the page RSA Access Manager issues when the user account is new and the Basic user password has not yet been set.
`cleartrust.agent.login_error_user_location_custom`	Specifies the location of the page RSA Access Manager issues when an invalid User ID error has occurred during Custom authentication.
`cleartrust.agent.login_error_pw_location_custom`	Specifies the location of the page RSA Access Manager issues when an invalid password error has occurred during Custom authentication.
`cleartrust.agent.login_cert_invalid_user`	Specifies the location of the page RSA Access Manager issues when the DN presented by the user certificate does not exist in the backend data store.
`cleartrust.agent.login_auth_inactive_account`	Specifies the location of the page RSA Access Manager issues when the user account is in an inactive state.

`cleartrust.agent.login_error_pw_location_basic`	Specifies the location of the page RSA Access Manager issues when a user submits an invalid password for Basic authentication.
`cleartrust.agent.login_error_location_securid`	Specifies the location of the page RSA Access Manager issues when an error occurs during RSA SecurID authentication.
`cleartrust.agent.login_error_user_location_nt`	Specifies the location of the page RSA Access Manager issues for Windows NT authentication.
`cleartrust.agent.login_error_pw_location_nt`	Specifies the location of the page RSA Access Manager issues when an invalid password error has occurred during Windows NT authentication.
`cleartrust.agent.login_error_password_expired`	Specifies the location of the page RSA Access Manager issues when the Basic user password is expired.
`cleartrust.agent.login_error_password_expired_forced`	Specifies the location of the page RSA Access Manager issues when the Basic user password is forced to expire by the administrator.
`cleartrust.agent.login_error_password_expired_new_user`	Specifies the location of the page RSA Access Manager issues when the user account is new and the Basic user password has not yet been set.
`cleartrust.agent.login_error_user_location_custom`	Specifies the location of the page RSA Access Manager issues when an invalid User ID error has occurred during Custom authentication.
`cleartrust.agent.login_error_pw_location_custom`	Specifies the location of the page RSA Access Manager issues when an invalid password error has occurred during Custom authentication.
`cleartrust.agent.login_cert_invalid_user`	Specifies the location of the page RSA Access Manager issues when the DN presented by the user certificate does not exist in the backend data store.
`cleartrust.agent.login_auth_inactive_account`	Specifies the location of the page RSA Access Manager issues when the user account is in an inactive state.

`cleartrust.agent.login_auth_expired_account`	Specifies the location of the page RSA Access Manager issues when the user account has expired.
`cleartrust.agent.login_auth_user_locked_out`	Specifies the location of the page RSA Access Manager issues when the user account is locked.
`cleartrust.agent.login_auth_url_access_denied`	Specifies the location of the page RSA Access Manager issues when the user does not have access to the requested resource.
`cleartrust.agent.login_server_error`	Specifies the location of the page RSA Access Manager issues when there is an internal error processing a request.
`cleartrust.agent.post_data_loss_url`	Specifies the path and configuration file of the logon page RSA Access Manager issues when post form data is lost because of idle timeout/session expiration/logout/token error.

For more information, see the *RSA Access Manager Agent 5.0 SP1 for Web Servers Installation and Configuration Guide*, or **webagent.conf**.

Agent Rules Engine

Use the xml-based rules engine, **rules.xml**, to filter or respond to certain requests without making calls to RSA Access Manager Servers.

RSA recommends using the rules engine to filter URLs/query strings with XSS/XST payloads, and to create a URL whitelist or blacklist for enhanced security.

For example, to filter a sample XSS payload that uses "<script>" or "<meta>" tags in a query string, the rule might look similar to the following example:

```
<Rule>
     <argument type="QueryString" filter="XSS" />
     <action type="HTTP" argument="500"/>
</Rule>
<SecurityFilter id="XSS">
     <regex pattern="&lt;[[:space:]]*script(.*)&gt;"/>
     <regex pattern="&lt;[[:space:]]*meta(.*)&gt;"/>
</SecurityFilter>
```

Note: This is an example that does not filter all XSS payloads. For a comprehensive list of XSS payloads and methods to filter them, consult the Open Web Application Security Project (OWASP) security guidelines.

For more information on the Agent rules engine, go to the *RSA Access Manager Agent 5.0 SP1 for Web Servers Installation and Configuration Guide*, locate the “General Configuration” chapter, and see the “Agent Rules Engine” section.

Physical Security Controls for Agents

Physical security controls enable the protection of resources against unauthorized physical access and physical tampering.

To help protect the resources, RSA recommends that the physical servers in the RSA Access Manager deployment be located in a secure data center that leverages the organization's best practices for physically securing a data center, server rack, and/or server.

Additional Documentation on Agent Security Features

The *RSA Access Manager Agent 5.0 SP1 for Web Servers Installation and Configuration Guide* provides detailed information on product security configuration, which includes some features that are mentioned in this guide. It also includes information on the following:

Configuring SSL for Agent hosts - Implements anonymous SSL or mutually authenticated SSL between RSA Access Manager Agent and RSA Access Manager components.

For more information, refer to the *RSA Access Manager Agent 5.0 SP1 for Web Servers Installation and Configuration Guide*, locate the "General Configuration" chapter, and see the "Connection Types" section.