RSA Access Manager 6.2 SP4

Security Configuration Guide

June 2016

01 06 16

Contents

Preface

This guide provides an overview of the settings available in RSA Access Manager Server (Access Manager Server), RSA Access Manager Agent (Access Manager Agent), and their components to help ensure secure operation of the product.

This guide is intended for administrators and other trusted personnel. Do not make this guide available to the general user population.

Topics:

Document Organization

This guide is organized into the following chapters:

- Chapter 1 Security Configuration Settings for Access Manager Servers. Provides guidance on how to collect logging and troubleshooting information, and provides the log file results codes for all server types.
- Chapter 2 Security Configuration Settings for Access Manager Agents. Details how to set up the supported debugging facilities for the Access Manager Agents.

Related Documentation

For more information about Access Manager Server, see the following documents available with this release, and on RSA SecurCare Online at https://knowledge.rsasecurity.com:

- *RSA Access Manager Server Release Notes*, in PDF, with the latest information about what is new and changed in this release, as well as workarounds for known issues.
- *RSA Access Manager Server Installation and Configuration Guide*. Provides instructions for installing and configuring Access Manager Server and additional components. This guide also contains descriptions for different configuration options, features, and production environment considerations.
- *RSA Access Manager Security Configuration Guide.* Provides an overview of the settings available in the Access Manager Server and compatible Access Manager Agents to ensure secure operation of the product.

For more information about Access Manager Server, see the complete Access Manager Server documentation set available from RSA SecurCare Online.

For more information about products related to Access Manager, see the following documentation available from RSA SecurCare Online:

- Access Manager Agents documentation set. The documentation related to agents.
- RSA Adaptive Authentication (Adaptive Authentication) documentation set.
- RSA enVision (enVision) documentation set.

RSA Customer Support

For support, go to Contact RSA Customer Support.

Access these locations for help with your RSA product.

- RSA SecurCare Online

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

- RSA Customer Support

  The RSA Customer Support site contains information on RSA support programs plus an extensive Content Library of product-related documents such as datasheets, guides and whitepapers.

- RSA Ready

  RSA Ready is a platform for customers, partners, and RSA enthusiasts to learn about products certified to interoperate with RSA products including access to integration guides.

Before You Call Customer Support

Make sure you have direct access to the computer running your RSA product software. Please have the following information available when you call:

- ❑ Your RSA Customer Serial Number.
- ❑ The software version number of your RSA product.
- ❑ The make and model of the machine on which the problem occurs.
- ❑ The name and version of the operating system under which the problem occurs.

1 Security Configuration Settings for Access Manager Servers

This chapter provides an details of the security configuration settings available for Access Manager Servers to help ensure secure operation.

The *RSA Access Manager Server Installation and Configuration Guide* provides detailed information about product security configuration, including some features mentioned in this guide.

Throughout this chapter, the Access Manager Server installation path is referred to as `<AXM_HOME>`.

Topics:

Security Configuration Settings for Servers

This section provides an overview of the settings available for 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 limiting access by end users, Access Manager Servers, and external components.

- Log Settings for Error and Debug Logs.

  Describes settings related to event logging.

- Inter-component Security Settings.

  Describes security settings related to Access Manager network communications.

- Data Security Settings for Data at Rest.

  Describes settings to help ensure protection of the data handled by Access Manager Servers.

- Session Replay Protection.

  Describes the security setting to protect against cookie replay for logged out users.

- SNMP Configuration.

  Describes the security setting to specify the Instrumentation Server SNMP version

Access Control Settings for User Authentication and Authorization

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

The following access control configuration parameters are located in the *<AXM_HOME>/conf/aserver.conf* file:

Parameter	Description
Authorization Server Mode	
`cleartrust.aserver.authorization_mode`	Controls access to unprotected resources, and works in conjunction with the Access Manager Agent to determine whether a URL is protected. Allowed values are `active` and `passive`. In passive mode, all resources on an Access Manager-protected web server are protected by default. For active mode, review the exclusion lists in the Access Manager Agents configuration. See Security Configuration Settings for Access Manager Agents. **RSA Recommendation:** 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	
`cleartrust.aserver.handle_invalid_user`	Controls the work flow of redirecting the user to the password screen for an invalid user ID instead of displaying the login failed error message.
Lockout Mode	
`cleartrust.aserver.lockout_mode`	Controls the behavior of the authentication result when the user is locked out. Allowed values are `1` or `2`. The default is `1`. If the value is 1, the user in context is locked out and the correct password is entered during authentication, `locked out` will be returned, otherwise, `invalid password` will be returned. If the value is 2 and the user in context is locked out, `locked out` will be returned as the authentication result until the user is unlocked.

The following access control configuration parameters are located in the *<AXM_HOME>/conf/keyserver.conf* file:

Parameter	Description
Key Server DNS Check	
`cleartrust.keyserver.session_key_dns_check`	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. Allowed values are `True` or `False`. **RSA Recommendation:** Set this parameter to `True` to help ensure the DNS in the environment is secure.
Key Server Token Lifetime	
`cleartrust.keyserver.token_lifetime`	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 Access Manager Agents' *webagent.conf* file. • Must be at least twice the value of `session_key_life` to prevent token decryption failure.
Key Server Session Key Life	
`cleartrust.keyserver.session_key_life`	Specifies how long a session key is valid for encrypting new SSO tokens. The default value is `30 mins`. Use an integer, a space, and one of the following time identifiers: `hour | mins | secs` **RSA Recommendation:** Use the lowest possible value based on the user's idle time with the system.

Unique User Sessions

Access Manager Server 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 provide increased security, RSA recommends disabling concurrent user sessions per IP address. For more information, see "Configure Unique User Session" in **Enhanced Functionality**, in the *RSA Access Manager Server Installation and Configuration Guide*.

Log Settings for Error and Debug Logs

The default location for the Access Manager Server logs is: `<AXM_HOME>/logs/`.

Logging Levels

The following items are logged by Access Manager Server, 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, see **Logging** in the *RSA Access Manager Troubleshooting Guide.*

For more information, see "Manage Log Files" in the *RSA Access Manager Installation and Configuration Guide.*

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 you grant log file access only to the most trusted administrators.

For more information, see "Secure Your Operating Environment" in Implement Security Features, in the *RSA Access Manager Server Installation and Configuration Guide*.

Inter-component Security Settings

Inter-component security settings, and system and security properties are designed to secure communication channels between Access Manager Servers and Access Manager Agents, as well as between the Access Manager Server web application and external systems or components.

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

Details of the following settings are provided:

- SSL between Access Manager Servers and Access Manager Agents
- SSL between Access Manager Servers and Web Applications
- SSLv3 Vulnerabilities & POODLE Protection
- Peer Verification
- Triple Handshake Vulnerability Protection
- Freak and LogJam Vulnerability Protection.

These security settings and properties are located in the following files. Select the configuration file appropriate for the relevant component:

- *<AXM_HOME>/conf/aserver.conf*
- *<AXM_HOME>/conf/eserver.conf*
- *<AXM_HOME>/conf/dispatcher.conf*
- *<AXM_HOME>/conf/iserver.conf*
- *<AXM_HOME>/conf/keyserver.conf.*

For more information about these parameters, see the configuration file.

These system properties are located in the following files. Select the batch file appropriate for the relevant component:

- *<AXM_HOME>/bin/aserver.bat*
- *<AXM_HOME>/bin/eserver.bat*
- *<AXM_HOME>/bin/dispatcher.bat*
- *<AXM_HOME>/bin/iserver.bat*
- *<AXM_HOME>/bin/keyserver.bat.*

SSL between Access Manager Servers and Access Manager Agents

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

Parameter	Description
Mutually authenticated SSL mode	
`cleartrust.net.ssl.use`	Specifies the communications mode used between Access Manager Servers and Access Manager 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 about setting mutually authenticated SSL between Servers and Access Manager Agents, see "Configuring Mutually Authenticated SSL" in **Implement Security Features**, in the *RSA Access Manager Server Installation and Configuration Guide*. **RSA Recommendation:** For stronger security, use `Auth`.
CA Keystore File	
`cleartrust.net.ssl.ca.keystore_file`	Specifies the name of the CA keystore file. This file is used to validate the certificate chain of clients and servers.
CA Keystore Type	
`cleartrust.net.ssl.ca.keystore_type`	Specifies the type of CA keystore.
CA Keystore Provider	
`cleartrust.net.ssl.ca.keystore_provider`	Specifies the provider of the keystore algorithm used for unlocking and using the CA keystore.
CA Keystore Passphrase	
`cleartrust.net.ssl.ca.keystore_passphrase`	Specifies the password required to unlock the CA keystore. **RSA Recommendation**: Encrypt this parameter. For more information, see Encrypt Configuration File Parameters

Private Keystore File

`cleartrust.net.ssl.private.keystore_file`

Specifies the keystore file where the private key of the server is stored.

Private Keystore Type

`cleartrust.net.ssl.private.keystore_type`

Specifies the type of keystore where the private key is stored.

Private Keystore Provider

`cleartrust.net.ssl.private.keystore_provider`

Specifies the keystore algorithm used for unlocking and using the private keystore.

Private Keystore Passphrase

`cleartrust.net.ssl.private.keystore_passphrase`

Specifies the password required to unlock the keystore holding the private key.

RSA Recommendation: Encrypt this parameter. For more information, see Encrypt Configuration File Parameters.

Private Key Alias

`cleartrust.net.ssl.private.key_alias`

Specifies the common name of the private key in the keystore.

Private Key Passphrase

`cleartrust.net.ssl.private.key_passphrase`

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.

Use a strong ACL policy, and allow file access only to the Access Manager Server service account.

RSA Recommendation: Encrypt this parameter. For more information, see Encrypt Configuration File Parameters.

SSL between Access Manager Servers and Web Applications

SSL communications between Access Manager Servers and the following Access Manager components is configurable:

- Administrative Console
- User Self-Service Console
- Runtime Web Service
- Administrative Web Service

For detailed information about configuring SSL communications between Access Manager Servers and these components, see "Implement Security Features" in the *RSA Access Manager Server Installation and Configuration Guide*

The following parameter specifies the type of encryption for communications between the Administrative Console or Administrative API clients, and the Entitlements Server. It allows you to disable SSL for the Administrative API port on the Entitlements Server when the rest of the system is using SSL. The setting applies to both C and Java Administrative API clients.

- `cleartrust.eserver.api_port.use_ssl`

The parameter is located in the *<AXM_HOME>/conf/eserver.conf* file

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 Recommendation: For stronger security, use `Auth`.

Another parameter, `cleartrust.net.ssl.use`, controls the SSL settings between the Entitlements Server and the other Access Manager Servers.

For more information about configuring the Administrative Console, see "Configure the Administrative Console" in the *RSA Access Manager Server Installation and Configuration Guide*.

SSLv3 Vulnerabilities & POODLE Protection

TLS clients are designed to communicate with TLS servers that support different versions of the TLS protocol, including SSLv3, TLS 1.0, TLS 1.1, and TLS 1.2.

Because SSLv3 uses no other ciphers than RC4 or CBC mode block ciphers, SSLv3 is not secure. Access Manager Server support for SSLv3 is disabled by default.

The following optional parameters can be used for the configuration of the SSL/TLS protocol and cipher suites.

Parameter	Description
SSL/TLS protocol	
`cleartrust.net.ssl.enabled_protocols`	Optional.Configures the SSL/TLS protocol over which the `SSL_ANON / SSL_AUTH` communication takes place. Access Manager Server and Access Manager Agent communication uses TLSv1 as the default protocol. Allowed Values: A comma-separated list of standard SSL/TLS protocol names. Default Value: A selected subset of SSL/TLS protocols supported by the configured Java Runtime, except the SSLv3 protocol.
Supported SSL Cipher Suites	
`cleartrust.net.ssl.cipher_suites`	Optional. Configures the cipher suites to be used in the SSL communication. A cipher suite is a set of cryptographic algorithms. Allowed Values: A comma-separated list of supported cipher suites. Default Value: A selected subset of cipher suites supported by the underlying JDK.
Excluded SSL Cipher Suites	
`cleartrust.net.ssl.excluded_cipher_suites`	Optional. Configures the exclusion of cipher suites used in ssl communication A cipher suite is a set of cryptographic algorithms. Allowed Values: A comma-separated list of supported cipher suites. Default Value:`_EXPORT_,_DES_,_3DES_,_DES40_,_NULL_,_RC4_`

Peer Verification

All incoming connections to 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 Access Manager Agents or Runtime API clients, that are connecting to it. The following peer verification parameters are located in the *<AXM_HOME>/conf/aserver.conf* file.

Verify Peer CN

The following parameter is available when mutually authenticated SSL is enabled. It determines whether the Server verifies the common name (`cn`) in client certificates:

- `cleartrust.net.ssl.verify_peer_cn`

Note: Used this setting only when `cleartrust.net.ssl.use=Auth`.

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 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.

Parameter	Description
Authorization Server DN Checks	
`cleartrust.aserver.token_api.enable`	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 Recommendation**: Enable DN checks
Authorization Server Trusted DN List	
`cleartrust.aserver.token_api.trusted_dn_list`	Ensures only clients whose DN has been specified can invoke APIs of the Authorization Server and get a token returned from the Authorization Server. **RSA Recommendation**: For more information about this parameter, see the configuration file. Enable this parameter to validate the Runtime API client connection.

For more information, see *Peer Verification* under "Configure Mutually Authenticated SSL" in Implement Security Features, in the *RSA Access Manager Server Installation and Configuration Guide*.

Triple Handshake Vulnerability Protection

Access Manager Servers have been designed to disallow a change of server certificate when renegotiating. The following security properties are statically registered at the JCE provider by default.

To disable all peer-initiated renegotiations

```
-Dcom.rsa.ssl.renegotiation=disabled
```

This parameter mitigates protocol downgrade attacks. It will prevent renegotiation with any client that does not include the extension `SCSV` or `renegotiation_info`.

To enable the peer update check:

```
-Dcom.rsa.ssl.server.compatibility.securerenegotiation=enabled'
-Dcom.rsa.ssl.server.compatibility.securerenegotiation.
   requireupdatedpeer=enabled
```

These security properties are applicable to all server components, and are set in the relevant batch files located in the `<AXM_HOME>/bin` directory.

Freak and LogJam Vulnerability Protection

Access Manager Servers have been designed to disallow all “export-grade” encryption ciphers and to use a 2048-bit Diffie-Hellman prime.

By default the following ciphers are not supported

- `*_EXPORT_*`
- `*_DES_*`
- `*_3DES_*`
- `*_DES40_*`
- `*_NULL_*`
- `*_RC4_*`.

The following system property is used to configure the default key size used in the Diffie-Hellman key exchange during a SSL handshake:

```
-Dcleartrust.dhe.keysize.limit=2048
```

This system property is applicable to all server components, and is set in the relevant server batch files located in the `<AXM_HOME>/bin` directory.

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 Encrypt Configuration File Parameters.

Server Authenticated SSL

Use server authenticated SSL to help to ensure secure communications between 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 about server authenticated SSL, see Implement Security Features, in the *RSA Access Manager Server Installation and Configuration Guide*.

Server Authenticated SQL

Use server authenticated SQL to help secure communications between 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)))
  ```

Use the following Password Hash Algorithm parameter to specify the algorithm the Access Manager Server uses to encrypt user passwords in the LDAP user directory or SQL database:

- For LDAP:

  ```
  cleartrust.data.ldap.user.password_hash_algorithm
  ```

- For SQL:

  `cleartrust.data.sql.user.password_hash_algorithm`

These parameters are located in the following files. Select the appropriate file for your system:

- *<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*

Note: Used the Password Hash Algorithm parameter 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 Recommendation: Use a strong password hashing algorithm for storing passwords. SSHA256 is recommended.

Session Replay Protection

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, see “Session Replay Protection” in Implement Security Features, in the *RSA Access Manager Server Installation and Configuration Guide*.

SNMP Configuration

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

Use the following parameter to specify the Instrumentation Server SNMP version:

`cleartrust.iserver.snmp.version`

The parameter is located in the *<AXM_HOME>/conf/iserver.conf* file

Allowed values are:

- `1` - SNMPv1
- `2`- SNMPv2c, the default
- `3` - SNMPv3

RSA Recommendation: Use SNMPv3 to connect to an NMS server.

For more information about configuring SNMP, see "Install and Configure the Instrumentation Server" in Simple Network Management Protocol Support, in the *RSA Access Manager Server Installation and Configuration Guide*.

Encrypt Configuration File Parameters

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

- The encryption utility `cryptedit` enables you to encrypt individual configuration file parameters containing 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*
  - *adaptive_auth-onpremise.conf*

  For more information, see "Encrypt Parameters in the Configuration Files" in Implement Security Features, in the *RSA Access Manager Server Installation and Configuration Guide*.

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

  For more information, see "Secure the Configuration Files" in Implement Security Features, in the *RSA Access Manager Server Installation and Configuration Guide*.

- The encryption utility `encryptutil` enables you to encrypt the following configuration parameters:

  In the *<AXM_HOME>/webapps/ axm-selfservice-gui-*.war/selfservice.conf* file:

  - `com.rsa.axm.selfservice.adapi.user_id`
  - `com.rsa.axm.selfservice.adapi.user_password`
  - `com.rsa.axm.selfservice.ssl.ca.keystore_passphrase`
  - `com.rsa.axm.selfservice.ssl.private.keystore_passphrase`
  - `com.rsa.axm.selfservice.ssl.private.key_passphrase`

  In the *<AXM_HOME>/conf/snmp-access-policy.xml* file:

  - `axm:securityPassphrase`
  - `axm:privacyPassphrase`

  For more information, see “Encrypt Configuration Parameters” in Implement Security Features, in the *RSA Access Manager Server Installation and Configuration Guide*.

Secure the Web Services

The following instructions are provided:

- Secure the Web Services Description Language
- SSL for Apache Tomcat and WebLogic Application Servers
- Apache HTTP Server Default Cache Configuration and Cookie Security
- Use Windows Authentication with Microsoft SQL Server
- Server Platform Updates with Security Fixes.

Secure the Web Services Description Language

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

The following instructions are provided:

- To secure the WSDL generated by Administrative Web Services:
- To secure the WSDL generated by Runtime Web Services:

To secure the WSDL generated by Administrative Web Services:

1. Navigate to `\WEB-INF` in the directory where you unzipped the *ws-admin-api.war* file.
2. Open the *web.xml* file of the Administrative web service.
3. Include the following text:

```
<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>
```

4. Save the *web.xml* file.
5. Restart the application server.

To secure the WSDL generated by Runtime Web Services:

1. Navigate to `\WEB-INF` in the directory where you unzipped the *ws-runtime-api.war* file.
2. Open the *web.xml* file of the Runtime web service in a text editor.
3. Include the following text:

```
<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-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/CTAuthService</url-pattern>
</filter-mapping>
```

4. Save the *web.xml* file.
5. Restart the application server.

SSL for Apache Tomcat and WebLogic Application Servers

Access Manager Server 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 about setting up SSL for these instances, see **Deploy Runtime and Administrative Web Services**, in the *RSA Access Manager Server Installation and Configuration Guide*.

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/.

Specifically for Access Manager Server, note that by default, the Apache module `mod_cache` caches HTTP content including cookies. In this default configuration, when a user accesses a protected resource, the RSA `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 in the *httpd.conf* file:

- Add the `CacheIgnoreHeaders` directive to specify 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.

- Add the `Header` directive and the `Cache-Control` header to specify Set-Cookie and Set-Cookie2 headers should not be cached at any level:

  ```
  Header set Cache-Control "no-cache=set-cookie, set-cookie2"
  ```

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

Use Windows Authentication with Microsoft SQL Server

You can configure the SQL data adapter to use Windows authentication with Microsoft SQL Server. For more information, see "Configure SQL Adapter with Microsoft SQL Server for Integrated Authentication" in Install and Configure the SQL Data Adapter, in the *RSA Access Manager Server Installation and Configuration Guide*.

Server Platform Updates with Security Fixes

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

Plan the Access Manager Server Deployment

You must plan the physical deployment of the 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 within your organization to protect the resources, providing access to the resources, 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 the *RSA Access Manager Server Planning Guide*.

To deploy Access Manager Applications, such as the Administrative Console, User Self-Service Console, Runtime and Administrative Web Services, see the *RSA Access Manager Server Installation and Configuration Guide*.

Secure Deployment and Usage Settings for Servers

Use the following configuration settings and system and security properties to help secure the your Access Manager Server deployment:

HTTPS Settings

To help secure communications between web browsers and web applications RSA recommends the HTTPS protocol. RSA also recommends using non-self-signed SSL certificates and certificates supporting strong cipher suites.

The following Access Manager components can be deployed in HTTPS mode:

- Administrative Console
- User Self-Service Console
- Administrative web services
- Runtime web services.

For more information about deploying the web applications in HTTPS mode, see the documentation for your application server.

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

Configure 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 Access Manager Server installation directory. It should be changed periodically in accordance with your organization's security policies.

For more information, see “Generate a Shared Secret Using Keygen” in **Deploy in Production Environments**, in the *RSA Access Manager Server Installation and Configuration Guide.*

Reverse Proxy in the DMZ

If you are using the User Self-Service Console outside the enterprise network, instead of deploying in the DMZ, it is recommended you deploy a reverse proxy in the DMZ, so the reverse proxy then forwards requests to the User Self-Service Console deployed inside the network.

Deploy Components Across a Firewall

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

For any two Access Manager Server 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, see "Deploy Components Across a Firewall" in **Deploy in Production Environments**, in the *RSA Access Manager Server Installation and Configuration Guide.*

Configure Two-Factor Authentication

RSA Authentication Manager

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

For more information, see "SecurID Authentication" in **Supported Authentication Types**, in the *RSA Access Manager Server Installation and Configuration Guide.*

RSA Adaptive Authentication

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

For more information, see **Integrate With Adaptive Authentication** in the *RSA Access Manager Server Installation and Configuration Guide.*

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 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 Access Manager Server Components

Access Manager Server provides an option to run Access Manager components in FIPS 140 mode (FIPS mode). By enabling FIPS mode, Access Manager Server uses only FIPS-approved algorithms for encryption processes. The following parameter should be configured in all the Access Manager Servers

```
cleartrust.fips_mode_enabled=true
```

Note: If `fips_mode_enabled` is enabled, the `aserver.token_version` should be either 3, 4 or 5. See Authorization Server Token Version below. For information about enabling FIPS mode, see "Enable FIPS Mode" in **Implement Security Features**, in the *RSA Access Manager Server Installation and Configuration Guide*.

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

Authorization Server Token Version

Use the following parameter located in the *<AXM_HOME>/conf/aserver.conf* file to specify the algorithm for the token that sets the `CTSESSION` cookie:

```
cleartrust.aserver.token_version
```

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 Recommendation: Use `4` or `5`.

Additional Documentation on Server Security Features

The *RSA Access Manager Server Installation and Configuration Guide* provides detailed information about product security configuration, including some features mentioned in this guide. It also includes information about how to:

- **Configure 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 Access Manager Server using the Access Manager Certificate Tool or Sun Java Keytool.

  For more information, see "Configure Server Authenticated SSL" in **Implement Security Features**.

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

  For more information, see "Configure Mutually Authenticated SSL" in **Implement Security Features**.

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

  For more information, see **Integrate with Adaptive Authentication**.

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

  For more information, see "Configure the Administrative Console" in **Deploy the Administrative Console**.

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

  For more information, see "Configure Password Restrictions" in **Enhanced Functionality**.

The *RSA Access Manager Server Developer*s *Guide* provides detailed information about API usage, such as how to:

- **Apply 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, see the *PasswordHookEventExample.java* example in "Code Examples".

2 Security Configuration Settings for Access Manager Agents

This chapter provides an details of the security configuration settings available for Access Manager Agents to help ensure secure operation.

The *RSA Access Manager Agent Installation and Configuration Guide* provides detailed information about product security configuration, including some features mentioned in this guide.

Topics:

Security Configuration Settings for Access Manager Agents

This section provides an overview of the settings available for Access Manager Agents to help ensure secure operation.

Details of the following security settings are provided:

- Access Control Settings for User Authentication and Authorization. Describes settings to limit access by end users or external Agent components.
- Log Settings. Describes settings related to event logging.
- Inter-component Security Settings. Describes security settings related to Agent network communications.
- Data Security Settings. Describes settings to ensure protection of the data handled by the Agent.
- Proxy Configurations. Describes security settings used to secure proxy configurations.

Access Manager Agent Configuration Files and Utilities

Access Manager Agent configuration files and utilities are located in `<AGENT_HOME>/bin` where `AGENT_HOME` is the Access Manager Agent installation path.

Access Manager Agent configuration parameters are located in *<AGENT_ROOT>/conf/webagent.conf* where `<AGENT_ROOT>` is one of the following:

Platform	Location
Windows	Access Manager Agent installation path
UNIX (Domino only)	Access Manager Agent installation path
UNIX (all servers except Domino)	`<AGENT_HOME>/webservers/<instance-name>`

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 Access Manager deployment, and authorizing access to requested resources.

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

Table 1 Access control parameters

Parameter	Description
Access Manager Agent Authentication Methods and Resources List	
`cleartrust.agent.auth_resource_list`	Specifies a list of comma-separated URLs and the authentication methods required to access to them. **RSA Recommendation:** Run the Authorization Server in passive mode to ensure all resources are protected by default. For more information, go to Access Control Settings for User Authentication and Authorization in **Security Configuration Settings for Access Manager Servers**.
Access Manager Agent Default Authentication Mode	
`cleartrust.agent.default_auth_mode`	Specifies the default authentication type for protected resources not defined by the `cleartrust.agent.auth_resource_list` parameter. This configuration does not apply to resources not protected in the Entitlements Server. **RSA Recommendation:** Run the Authorization Server in passive mode to ensure all resources are protected by default. For more information, go to Access Control Settings for User Authentication and Authorization in **Security Configuration Settings for Access Manager Servers**.
Agent for Handling ISSO Slave Authentication at Authorization Server	
`cleartrust.agent.isso.handle_slave_auth_at_aserver`	Handles the creation and verification of signatures, using Authorization Server for slave authentication. Allowed values are `True` or `False`. When set to `True`, Access Manager Agent uses Authorization runtime APIs for slave authentication. When set to `False`, the Agent retrieves session keys from the Key Server and handles signature verification by itself. **RSA Recommendation:** Set this parameter to `True` so Access Manager Agent uses a runtime API to communicate with the Authorization Server to create or verify a signature. This results in sensitive information being handled within a secure network.

Table 1 Access control parameters (Continued)

Agent URL Exclusion List
`cleartrust.agent.url_exclusion_list`
Specifies a list of URLs excluded from access control checks. URLs in this list are unprotected, and are not subject to Access Manager Agent authentication. **RSA Recommendation:** Configure this parameter using specific URLs instead of wildcards, which can unintentionally allow access to URLs that should be protected.
Agent Extension Exclusion List
`cleartrust.agent.extension_exclusion_list`
Specifies a list of file extensions excluded from access control checks. **This parameter is deprecated.** RSA recommends using the Access Manager Agent rules engine *rules.xml*, to specify more specific URL patterns to exclude 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. This can also expose the web server to URL exploits. For *rules.xml* usage, refer to the `cleartrust.agent.rules_file` parameter in the configuration file. For more information, see "Access Manager Agent Rules Engine" in **Configuration Options** in the *RSA Access Manager Agent Installation and Configuration Guide*. **RSA Recommendation:** Make 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.

Note: To help protect all server resources, RSA recommends running the Authorization Server in passive mode, and providing granular access levels using the 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 not needing protection 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, see Access Control Settings for User Authentication and Authorization in **Security Configuration Settings for Access Manager Servers**.

For more information about using these methods, see **Configure and Specify Authentication Types**, in the *RSA Access Manager Agent Installation and Configuration Guide*.

Log Settings

Error and Debug Logs

The 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/.`

The log location can be configured 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 configuring 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 the following parameter:

- `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 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 set for the instance, the following items might be logged:

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

Directory Permissions

To help secure the logs directory, RSA recommends restricting permissions on the logs directory to the minimum required permissions, read and write. for:

- 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 for Windows systems:

1. Log on to the Access Manager Server.
2. Locate the log file directory.

   Right-click on the folder, and select **Properties**. Go to the **Permissions** tab.
3. Confirm `NETWORK_SERVICE` has the required permissions.

To review the permissions on the logs directory for UNIX-based systems:

1. Log on to the Access Manager Server.
2. Navigate to the log file directory in a terminal
3. Run the following command:

   ```
   ls -ld
   ```
4. Confirm that the user account under which the web server runs has the required permissions.

Inter-component Security Settings

Inter-component security settings help to secure the communication channels between Access Manager Servers and Access Manager Agents, as well as between the Access Manager web application and external systems or components.

Details about the following settings are provided:

- SSL between Access Manager Agent and Access Manager Servers
- Web Server SSL
- Cookies over SSL.

SSL between Access Manager Agent and Access Manager Servers

To specify the communications mode used between Access Manager Servers and Access Manager Agents, use the following parameter:

- `cleartrust.agent.ssl.use`

Allowed values are:

- `Clear` - Clear text (no encryption)
- `Anon` - Anonymous SSL (SSL encryption with no certificate authentication)
- `Auth` (default) - Mutually authenticated SSL, that is, SSL encryption with PKI certificate authentication.

For more information, see **Configure and Specify Authentication Types** in the *RSA Access Manager Agent Installation and Configuration Guide*.

RSA Recommendation: For stronger security, use `Auth`.

The following configuration parameters must be set appropriately when this configuration is set to `Auth`:

Table 2 Mutually authenticated SSL parameters

Parameter	Description
Access Manager Agent Private key Keystore	
`cleartrust.agent.ssl.keystore`	Specifies the keystore name of the PKCS #12 keystore containing the Access Manager 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 only authorized users have access to the private key file.

Table 2 Mutually authenticated SSL parameters (Continued)

Access Manager Agent Keystore Passphrase
`cleartrust.agent.ssl.keystore_passphrase`
Specifies the passphrase used to verify the integrity of the PKCS #12 keystore containing the private key. **RSA Recommendations:** Set this parameter to `false` to ensure the parameter is defined in an encrypted store instead of being stored as clear text in *webagent.conf*. For more information, see 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.
Access Manager Agent Private Key Alias
`cleartrust.agent.ssl.private_key_alias`
Specifies the alias of the private key in the PKCS #12 private-key keystore. **RSA Recommendation:** Specify an alphanumeric string, without spaces or special characters, for the private key alias.
Access Manager Agent Certificate Keystore
`cleartrust.agent.ssl.ca_keystore`
Specifies the keystore name of the PKCS #12 keystore containing 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 only authorized users have access to the file.
Access Manager Agent CA Keystore Passphrase
`cleartrust.agent.ssl.ca_keystore_passphrase`
Specifies the passphrase used to verify the integrity of the PKCS #12 CA keystore. **RSA Recommendations:** Set this parameter as `.cleartext=false` to ensure the parameter is defined in an encrypted store instead of being stored as clear text in *webagent.conf*. For more information, see 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 to help 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 about enabling SSL, see your web server documentation.

Cookies over SSL

To specify that the browser should accept and send cookies using only secure methods, to restrict cookies to SSL connections, set the following parameter:

- `cleartrust.agent.secure`

Allowed values are `True` or `False`.

RSA Recommendation: Set this parameter to `True` to enable this parameter to restrict cookies to SSL connections.

Data Security Settings

Data security settings are intended to prevent permanently stored product data from being disclosed in an unauthorized manner. Details about the following settings are provided:

- Encryption of Data at Rest: Cookie Security
- Encryption of Data at Rest: Encryption Utilities
- Data Integrity: Cookie Integrity
- Data Integrity: URL Integrity
- Data Erasure: Timeouts
- Data Erasure: Cache Control.

Encryption of Data at Rest: Cookie Security

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

Parameter	Description
Cookie Path	
`cleartrust.agent.path`	Specifies the path on the web server where the SSO cookie applies. An empty value means the current URL path is 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.
Cookie Path	
`cleartrust.agent.cookie_expiration`	Sets the amount of time a cookie persists in a browser. **RSA Recommendation:** Set this parameter to `0 Mins` to ensure the cookie is valid only until the browser exits.
Cookie HTTP Only	
`cleartrust.agent.httponly`	Specifies whether the `HttpOnly` attribute is included in the SSO cookie. Allowed values are `True` or `False` (default). **RSA Recommendation:** Set this parameter to `True` so cookies 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

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

Parameter	Description
Encrypted Store Parameter	
`cleartrust.agent.encrypted_store`	Specifies the filename for the encrypted store where sensitive configuration parameters can be stored. **Note:** This parameter must 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, see **Access Manager Agent Utilities** in the *RSA Access Manager Agent Installation and Configuration Guide*. Specify an absolute file path or a filename relative to the Agent's `conf` directory. Ensure only authorized users have permissions to access to the file
Crypt Edit Utility	
`<AGENT_HOME>/conf/ctagent_cryptedit[.exe]`	Encrypts sensitive configuration parameter settings for *webagent.conf*, such as the keystore passphrase. **RSA Recommendation:** Encrypt all sensitive configuration parameters using `CryptEdit`.
Watchdog Utility	
`<AGENT_HOME>/conf/ctagent_watchdog[.exe]`	Stores the password you assign to the file used for the cryptedit utility. Also, supplies the Access Manager Agent with the password so it can read the encrypted parameters, which allows the Access Manager 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, see **Access Manager Agent Utilities** in the *RSA Access Manager Agent Installation and Configuration Guide*.

Data Integrity: Cookie Integrity

To help maintain cookie integrity, RSA recommends the following settings:

Parameter	Description
Cookie IP Check	
`cleartrust.agent.cookie_ip_check`	Enables/disables session IP checking. When this setting is enabled, the Access Manager 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 are moved from one computer to another. Allowed values are `True` or `False` **RSA Recommendation:** Set this parameter to `True` to mitigate cookie replay attacks.
Domain Check	
`cleartrust.agent.cookie_domain`	Specifies the domain name in the HTTP `Set-Cookie` header for SSO tokens. **RSA Recommendation:** Restrict `CTSESSION` cookie distribution to the most restricted domain possible.
Strict Cookie Set	
`cleartrust.agent.strict_cookie_set`	Specifies whether to set the `CTSESSION SSO` cookie. Allowed values are `True` or `False` **RSA Recommendation:** Set this parameter to `True` to ensure the `CTSESSION` cookie is set only if the user has successfully authenticated with at least one of the supported authentication types.

Data Integrity: URL Integrity

To specify a list of domain names to which the Access Manager Agent is allowed to redirect users immediately after authentication, set the following parameter:

- `cleartrust.agent.trusted_domains_list`

For Access Manager Agents in an ISSO environment, include master and slave domain names.

Note: You must add the domain name of the Agent's host if this parameter is enabled.

RSA Recommendation: Specify a list of URLs the Access Manager Agent can trust to prevent redirects to arbitrary URLs.

Data Erasure: Timeouts

To help manage timeouts, RSA recommends the following configuration settings:

Parameter	Description
Idle Timeout	
`cleartrust.agent.idle_timeout`	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 Recommendation:** Set this parameter to a value appropriate for your environment. A value too high or low might result in cookies not being invalidated or users being required to log on again frequently.
POST URL Idle Timeout	
`cleartrust.agent.post_url_idle_timeout`	Sets an additional amount of time for a session to remain valid when making HTTP POST requests to a specific set of URLs 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 they can submit a page, due to the `cleartrust.agent.idle_timeout` setting. **RSA Recommendation:** Do not set this parameter to a high value due to security implications.
Session Lifetime	
`cleartrust.agent.session_lifetime`	Sets the maximum lifetime of an SSO session. The default value is 8 hours. **RSA Recommendation:** Set this parameter to a value appropriate for the environment. A value too low might result in users being required to log on again frequently.
Cookie Touch Window	
`cleartrust.agent.cookie_touch_window`	Sets the amount of time the Access Manager Agent waits before updating the cookie for an authenticated user. Set this parameter to `<1 Minutes`. **RSA Recommendation:** Do not set this parameter to a high value, such as greater than 5 minutes, as the `idle_timeout` is shortened by the period of time specified in this parameter.

Data Erasure: Cache Control

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

Parameter	Description
Protected Resources Cache TTL	
`cleartrust.agent.protected_resource_cache_ttl`	Sets the protected resource status cache time to live (TTL). The default value is 10 minutes. **RSA Recommendations:** Maintain the default,`10 Mins`, so cached entries are cleared after 10 minutes. Do not set this parameter to `0`, as the Access Manager Agent would never prune the cache based on TTL.
Unprotected Resources Cache TTL	
`cleartrust.agent.unprotected_resource_cache_ttl`	Sets the unprotected resource status cache time to live (TTL). The default value is 5 minutes. **RSA Recommendations:** Maintain the default, `5 Mins`, so cached entries are cleared after 5 minutes. Do not set this parameter to `0`, as the Access Manager Agent will never prune the cache based on TTL.
Token Cache TTL	
`cleartrust.agent.token_cache_ttl`	Sets the cookie cache time to live (TTL). Cookies issued to the client browser can be cached in unencrypted form by the Access Manager Agent for better performance. The default value is 5 Minutes. **RSA Recommendations:** Maintain the default, `5 Mins`, so 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.
Token Cache Size	
`cleartrust.agent.token_cache_size`	Sets the cookie cache size. When the maximum size is reached, cache entries are removed, oldest first. The default value is 10000. **RSA Recommendations:** Maintain the default, `10000`, so the cache is pruned when it reaches 10000 entries. Do not set this parameter to `0`, as the Access Manager 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 Access Manager 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:

Table 3 Proxy configurations parameters

Parameter	Description
Trusted Proxy List	
`cleartrust.agent.trusted_proxy_list`	Specifies a comma-separated list of IP addresses which represent the hosts 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 Access Manager Agent. **RSA Recommendation:** Set specific IP addresses, or a range of IP addresses instead of a broader subnet, to prevent spoofing a client address within the specified subnet but does not exist.
Cookie IP Check	
`cleartrust.agent.cookie_ip_check`	Enables/disables session IP checking. When session IP checking is enabled, the Access Manager Agent accepts cookies only from the same IP address to which they were originally issued. **RSA Recommendations:** Disable this configuration 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 and exclude them from IP checks using `cleartrust.agent.ip_check_exclusion_list`.
Cookie Exclusion List	
`cleartrust.agent.cookie_exclusion_list`	Specifies a comma-separated list of IP addresses representing hosts not issued cookies. **RSA Recommendation:** Set this parameter in proxy environments where both the proxy and content servers are protected by Access Manager. This allows the content server to suppress generating a duplicate cookie, as the proxy has already performed this task.

Table 3 Proxy configurations parameters (Continued)

Cookie IP Check Exclusion List	
`cleartrust.agent.ip_check_exclusion_list`	
	Specifies a comma-separated list of host IP addresses allowed to act as proxies and forward cookies to this server, and are not subjected to IP address checks. **RSA Recommendations:** Use a specific list of IP addresses when possible. Specify proxy IP addresses to ensure 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.
Trusted Proxy Strict Mode	
`cleartrust.agent.trusted_proxy_strict_mode`	
	Specifies the resulting behavior when a check against the `trusted_proxy_list` fails. **RSA Recommendation:** For Internet sites accessible to the public, set this parameter to `False`, as users behind proxies not registered in the `trusted_proxy_list` are not 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 Access Manager Agents are installed on both the proxy servers and the content servers. The content servers should be configured to IP check all cookies coming from machines other than the proxy servers (using `ip_check_exclusion_list`). Proxy server Agents are responsible for IP checking cookies in requests 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 Access Manager 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 Access Manager Agents

To help secure the deployment of the Access Manager Agent, details about the following configuration settings are provided:

- Web Server Security
- Adaptive Authentication Settings
- HTTP Settings
- Generic Error Pages
- Access Manager Agent Rules Engine.

Web Server Security

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

Adaptive Authentication Settings

To determine the action to take when the Access Manager Agent receives an Adaptive Authentication connection failure from an Authorization Server, set the following parameter:

- `cleartrust.agent.aa.allow_on_failure`

Allowed values are `True` or `False`. The default value is `True`.

RSA Recommendation: Set this parameter to `False` to avoid bypassing authentication when the Adaptive Authentication servers are down.

HTTP Settings

Use the following configuration settings for securing HTTP configurations:

Parameter	Description
Export Headers for Protected Resources Only	
`cleartrust.agent.export_headers_for_protected_resources_only`	Specifies whether HTTP Request headers should be published for protected resources only or for all resources. Allowed values are `True` or `False`. **RSA Recommendation:** Set to `True`, to prevent publication of HTTP Request Headers for unprotected resources.
Strict Headers Export	
`cleartrust.agent.strict_headers_export`	Specifies whether to publish `CT_REMOTE_USER` from the user header list even if the user has not successfully authenticated. Allowed values are `True` or `False`. **RSA Recommendations:** Set to `True` to ensure `CT_REMOTE_USER` is not published as an HTTP header if user authentication failed due to account lockout or password expiration. Publishing this header for all valid users, regardless of their authentication status, might potentially enable an attacker to distinguish between valid and invalid users.
Retain URL in Cookie Vs. Query String	
`cleartrust.agent.retain_url.use_query_string`	Indicates how the Access Manager Agent stores the original URL during URL retention. Allowed values are `True` or `False`. 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 Recommendation:** Disable to have the Access Manager Agent store the original URL in a cookie during URL retention
Ignore HTTP Auth	
`cleartrust.agent.ignore_http_auth`	Instructs the Access Manager Agent to ignore the user credential in HTTP-Authorization headers. **RSA Recommendation:** Enable to prevent users from bypassing form log ons.

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 log on failure pages are the same for all failures.

The Access Manager Agent provides the following configurations for custom error pages:

Table 4 Custom Error pages

Parameter	Description
`cleartrust.agent.login_error_user_location_basic`	Specifies the path and file location of the page 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 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 Access Manager issues when an error occurs during SecurID authentication.
`cleartrust.agent.login_error_user_location_nt`	Specifies the location of the page Access Manager issues for Windows NT authentication.
`cleartrust.agent.login_error_pw_location_nt`	Specifies the location of the page 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 Access Manager issues when the Basic user password is expired.
`cleartrust.agent.login_error_password_expired_forced`	Specifies the location of the page 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 Access Manager issues when the user account is new and the Basic user password has not yet been set.

Table 4 Custom Error pages (Continued)

Parameter	Description
`cleartrust.agent.login_error_user_location_custom`	Specifies the location of the page 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 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 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 Access Manager issues when the user account is in an inactive state.
`cleartrust.agent.login_auth_expired_account`	Specifies the location of the page Access Manager issues when the user account has expired.
`cleartrust.agent.login_auth_user_locked_out`	Specifies the location of the page Access Manager issues when the user account is locked.
`cleartrust.agent.login_auth_url_access_denied`	Specifies the location of the page 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 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 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 Installation and Configuration Guide*, or *webagent.conf*.

Access Manager Agent Rules Engine

Use the xml-based rules engine, *rules.xml*, to filter or respond to certain requests without making calls to 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 using `<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 about the Access Manager Agent rules engine, see "Access Manager Agent Rules Engine" in **Configuration Options** in the *RSA Access Manager Agent Installation and Configuration Guide.*

Physical Security Controls for Access Manager Agents

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

To help protect the resources, RSA recommends the physical servers in the 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.