Abridged

RSA® Authentication Manager 8.1 Administrator's Guide

for Security Domain Administrators

IT Services
Iowa State University

Jan 2015

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 RSA 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.

Third-Party Licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed on the product documentation page on RSA SecurCare Online. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

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.

December 2013

1 RSA Authentication Manager Overview

Introduction to RSA Authentication Manager

RSA Authentication Manager is a multi-factor authentication solution that verifies authentication requests and centrally administers authentication policies for enterprise networks. Use Authentication Manager to manage security tokens, users, multiple applications, agents, and resources across physical sites, and to help secure access to network and web-accessible applications, such as SSL-VPNs and web portals.

Multifactor Authentication

Passwords are a weak form of authentication because access is protected only by a single factor — a secret word or phrase selected by the user. If this password is discovered by the wrong person, the security of the entire system is compromised. Multifactor authentication provides stronger protection by requiring two or more unique factors to verify a user's identity. Authentication factors in a multifactor system may include:

- Something the user knows (a password, passphrase, or PIN)
- Something the user has (a hardware token, laptop computer, or mobile phone)
- Something the user does (specific actions or a pattern of behavior)

Authentication Manager provides the following choices for strong authentication:

- RSA SecurID, which protects access using two-factor authentication with hardware and software-based tokens.
- On-demand authentication (ODA), which protects access using two-factor authentication by sending authentication credentials to users upon request through SMS text messaging or e-mail.
- Risk-based authentication (RBA), which protects access by assessing user behavior and matching the device being used to authenticate to assess the risk-level of an authentication attempt.

  By leveraging devices that the user already owns, for example, a mobile phone, PC, or laptop, RBA and ODA enable multifactor authentication with no tokens to manage.

Key Components for RSA Authentication Manager

An RSA Authentication Manager deployment may have the following components:

Primary Instance

Replica Instance

Identity Sources

RSA Authentication Agents

Risk-Based Authentication for a Web-Based Resource

RSA RADIUS Overview

Web Tier

Self-Service

SMS plug-ins. For more information see Deploying On-Demand Authentication on page 217.

Load Balancer

Note: RSA supports and certifies many third-party products for integration with RSA SecurID, risk-based authentication (RBA), on-demand authentication (ODA), or Short Message Service (SMS). For a list of supported products, go to **https://gallery.emc.com/community/marketplace/rsa** and search for Authentication Manager. Each certification has a step-by-step implementation guide for setting up the solution.

Primary Instance

The primary instance is the initial Authentication Manager system that you deploy. Once you deploy a primary instance, you can add replica instances. It is possible to promote a replica instance to replace the primary instance in maintenance or disaster recovery situations.

The primary instance is the only system in the deployment that allows you to perform all Authentication Manager administrative tasks. Some administrative tasks can be performed on a replica instance, for example, replica promotion and log file collection.

The main functions of the primary instance include the following:

- Authenticating users.
- Enabling administration of Authentication Manager data stored in the internal database. You can perform tasks such as importing and assigning SecurID tokens, enabling risk-based authentication (RBA), adding LDAP identity sources, configuring self-service, generating replica packages, and generating agent configuration files and node secrets.
- Replicating changes due to administration and authentication activities.
- Hosting the primary RSA RADIUS server.

- Handling self-service requests.
- Maintaining the most up-to-date Authentication Manager database.

Replica Instance

A replica instance provides deployment-level redundancy of the primary instance. You can view, but not update, administrative data on a replica instance.

A replica instance provides the following benefits:

- Real-time mirror of all user and system data
- Failover authentication if the primary instance becomes unresponsive
- Improved performance by load balancing authentication requests to multiple instances
- Ability to deploy a replica instance at a remote location
- Remote deployment must be done carefully to make sure that the replica instance is secure.For information about securely deploying a remote replica instance, see "Deployment Scenarios" in the *RSA Authentication Manager 8.1 Planning Guide*.Ability to recover administrative capabilities through replica promotion if the primary instance becomes unresponsive

Although a replica instance is optional, RSA recommends that you deploy both a primary and a replica instance.

Identity Sources

All users and user groups in your deployment are stored in identity sources. RSA Authentication Manager supports the following as identity sources:

- LDAP directory servers, either Active Directory or Oracle Directory Server.
- Active Directory Global Catalogs, when some or all of the Active Directory servers in its Active Directory forest are used as identity sources. In such a case, the Global Catalog is used for runtime activities, for example, looking up and authenticating users, and resolving group membership within the Active Directory forest. The Global Catalog cannot be used to perform administrative functions.
- The Authentication Manager internal database, used for administrative operations, such as enabling users for on-demand authentication and risk-based authentication.

You can integrate LDAP directory servers as identity sources in Authentication Manager without modifying the schema of the directories.

RSA Authentication Agents

An authentication agent is a software application installed on a machine, such as a domain server, web server, or personal computer, that enables authentication.

The authentication agent is the component on the protected resource that communicates with RSA Authentication Manager to process authentication requests. Any resource that is used with SecurID authentication, on-demand authentication (ODA) or risk-based authentication (RBA) requires an authentication agent.

Different types of authentication agents protect different types of resources. For example, to protect an Apache Web server, you need RSA Authentication Agent 5.3 for Web for Apache. You may also purchase products that contain embedded RSA Authentication Agent software. The software is embedded in a number of products, such as remote access servers, firewalls, and web servers. Information about implementing such devices is available from the Secured By RSA Partner Program. For more information, see **http://www.emc.com/partnerships/rsa-partners/secured-rsa-partner-program-overview.htm**.

Note: RBA only works with web-based authentication agents.

Risk-Based Authentication for a Web-Based Resource

Risk-based authentication (RBA) protects access to web-based resources and applications. Deploying RBA requires integrating the resource with Authentication Manager. Authentication Manager provides a template to facilitate the integration process. Once integrated, the web-based resource automatically redirects users to Authentication Manager, which does either of the following:

- Authenticates the user and returns proof of authentication to the resource
- When the risk level is high, prompts the user to provide further credentials, such as the correct answers to pre-configured security questions, before returning proof of authentication.

The web-based resource presents the proof of authentication to Authentication Manager for verification and allows the user access to the resource.

RSA RADIUS Overview

You can use RSA RADIUS with RSA Authentication Manager to directly authenticate users attempting to access network resources through RADIUS-enabled devices. A RADIUS server receives remote user access requests from RADIUS clients, for example, a VPN. The RADIUS server forwards the access requests to RSA Authentication Manager for validation. Authentication Manager sends accept or reject messages to the RADIUS server, which forwards the messages to the requesting RADIUS clients.

RADIUS is automatically installed and configured during the Authentication Manager installation. After installation, RADIUS is configured to run on the same instance with Authentication Manager.

You use the Operations Console to configure RSA RADIUS and manage settings that must be made on individual instances running RSA RADIUS.

You can use the Security Console to complete most tasks associated with managing RADIUS day-to-day operations.

Product or Application	Description
Wireless networking	Authentication Manager includes an 802.1-compliant RADIUS server.
Secure access to Microsoft Windows	Authentication Manager can be used to control access to Microsoft Windows environments both online and offline.
Network hardware devices	Authentication Manager can be used to control desktop access to devices enabled for SecurID, such as routers, firewalls, and switches.

RSA SecurID Authentication Process

The RSA SecurID authentication process involves the interaction of three distinct products:

- RSA SecurID authenticators, also known as tokens, which generate one-time authentication credentials for a user.
- RSA Authentication Agents, which are installed on client devices and send authentication requests to the Authentication Manager.
- RSA Authentication Manager, which processes the authentication requests and allows or denies access based on the validity of the authentication credentials sent from the authentication agent.

To authenticate a user with SecurID, Authentication Manager needs, at a minimum, the following information.

Element	Information
User record	Contains a User ID and other personal information about the user (for example, first name, last name, group associations, if any). The user record can come from either an LDAP directory server or the Authentication Manager internal database.
Agent record	Lists the name of the machine where the agent is installed. This record in the internal database identifies the agent to Authentication Manager and enables Authentication Manager to respond to authentication requests from the agent.
Token record	Enables Authentication Manager to generate the same tokencode that appears on a user's RSA SecurID token.
SecurID PIN	Used with the tokencode to form the passcode.

RSA SecurID Tokens

An RSA SecurID token is a hardware device or software-based security token that generates and displays a random number is called a tokencode.

In addition to the tokencode, RSA SecurID typically requires a PIN, either created by the user or generated by Authentication Manager. Requiring these two factors, the tokencode and the PIN, is known as two-factor authentication:

- Something you have (the token)
- Something you know (the PIN)

In Authentication Manager, the combination of the tokencode and the PIN is called a passcode. When users try to access a protected resource, they enter the passcode at the logon prompt. (To protect against the use of stolen passcodes, Authentication Manager checks that a passcode has not been used in any previous authentication attempt.)

RSA SecurID also supports tokens that do not require a PIN. The user can authenticate with the current tokencode only. In such a case, an alternative second factor, for example, a user's network password, is used.

Each shipment of tokens includes token seed records that you must import into the Authentication Manager. Each token seed record corresponds to an individual RSA SecurID token, and is used by Authentication Manager to generate the correct tokencode when a SecurID authentication request is received from an authentication agent.

Authentication Manager logs the serial numbers of SecurID tokens used to authenticate. By default, Authentication Manager logs the serial number in the clear, but you can mask the serial numbers of tokens when logging to syslog or using SNMP if you want to avoid transmitting and recording the serial number in the clear. RSA recommends masking token serial numbers for added security. For more information, see Mask Token Serial Numbers in Logs on page 344.

You can assign up to three RSA SecurID tokens to each authorized user on a protected system.

RSA SecurID Hardware Tokens

RSA SecurID hardware authenticators are available in a variety of convenient form factors.

- RSA SecurID 200 Authenticator

  This hardware token, the size of a credit card, is easily portable and extremely durable. It generates and displays a new tokencode at a predefined time interval, typically every 60 seconds.

- RSA SecurID 520 Authenticator

  With this device, the size of a credit card, the user enters the PIN on a 10-digit numeric keypad. The code displayed is a hash-encrypted combination of the PIN and the current tokencode. The RSA SecurID 520 Authenticator can offer protection against the key-logger type of malware, which may attempt to steal a user's PIN for other purposes.

- RSA SecurID 700 Authenticator

  This hardware device easily connects to any key ring. The user simply reads the changing display (typically every 60 seconds) and uses it as part of a dynamic and always-changing password.

- RSA SecurID 800 Hybrid Authenticator

  The RSA SecurID Authenticator SecurID 800 is both an RSA SecurID authenticator and a USB smart card (USB token) with a built-in reader. The two sets of electronics operate independently of each other.

  When disconnected, the SecurID 800 generates and displays tokencodes used in RSA SecurID authentication. When connected to a computer, the token serves two functions:

- For RSA SecurID authentication, users obtain their tokencodes through the supporting middleware installed on their desktop instead of reading the number off the token.
- With the token's smart card capabilities, users can store credentials, including multiple X.509 digital certificates, which enable authentication, digital signature, and file-encryption applications, and Windows logon accounts.

RSA SecurID Software Tokens

RSA SecurID tokens are also available in a software form-factor that you can install into an RSA SecurID software token application on a client workstation, or a mobile device. The RSA Authentication Manager provides a centralized administration interface for issuing RSA SecurID software tokens to the supported device types. You can add information to software tokens such as device type, device serial number, or token nickname using token extension fields. For more information about the software token, see Chapter 9, Deploying and Administering RSA SecurID Tokens, and the documentation that accompanies individual RSA SecurID software token products.

The Role of RSA Authentication Manager In SecurID Authentication

RSA Authentication Manager software, authentication agents, and RSA SecurID tokens work together to authenticate user identity. RSA SecurID patented time synchronization ensures that the tokencode displayed by a user's token is the same code that the RSA Authentication Manager software has generated for that moment. Both the token and the Authentication Manager generate the tokencode based on the following:

- The token's unique identifier (also called a "seed").
- The current time according to the token's internal clock, and the time set for the Authentication Manager system.

To determine whether an authentication attempt is valid, the RSA Authentication Manager compares the tokencode it generates with the tokencode the user enters. If the tokencodes do not match or if the wrong PIN is entered, the user is denied access.

~~On-Demand Authentication~~

~~On-demand authentication (ODA) delivers a one-time tokencode to a user's mobile phone or e-mail account. A tokencode is a randomly generated six-digit number. ODA protects company resources that users access through agent-protected devices and applications, such as SSL-VPNs and web portals.~~

~~ODA strengthens network security by requiring users to present two factors:~~

- ~~Something only the user knows (a PIN)~~
- ~~Something the user has (a tokencode)~~

~~ODA is easy to deploy because it does not require extra hardware, such as physical tokens. Employees already have and use mobile phones and e-mail accounts.~~

~~On-demand tokencodes can be used only once and expire after a specified time period, enhancing their security.~~

3 Deploying Authentication Agents

RSA Authentication Agents

Authentication agents are software applications that securely pass user authentication requests to and from RSA Authentication Manager. Authentication agents are installed on each machine, such as a domain server, web server, or a personal computer, that you protect with Authentication Manager.

For example, agent software residing on a web server intercepts all user requests for access to protected web pages. When a user attempts to access a protected URL, the agent requests the User ID and passcode and passes the User ID and passcode to the Authentication Manager for authentication. If the authentication is successful, the user is granted access to protected web pages.

Authentication Agent Types

When you deploy an agent, you specify whether the agent is unrestricted or restricted.

Unrestricted agents. Unrestricted agents process all authentication requests from all users in the same deployment as the agent.

However, to allow a user to authenticate with a logon alias, the user must belong to a user group that is associated with the logon alias and that is enabled on the unrestricted agent.

Restricted agents. Restricted agents process authentication requests only from users who are members of user groups that have been granted access to the agent. Users who are not members of a permitted user group cannot use the restricted agent to authenticate. Resources protected by restricted agents are considered to be more secure because they process requests only from a subset of users.

Obtaining RSA Authentication Agents

The agent that you need depends on the type of resource you want to protect. For example, to protect an Apache web server, you need to download the RSA Authentication Agent for Apache.

The download package includes an *Installation and Administration Guide* and a *Readme*. Read these documents before installing the agent. For information about installing agent software, see your agent documentation.

You may purchase products that contain embedded RSA Authentication Agent software. For example, these products include remote access servers and firewalls.

Procedure

Do one of the following.

- For an RSA agent and for a third-party agent that requires an RSA agent, go to **http://www.emc.com/security/rsa-securid/rsa-securid-authentication-agents.htm#!offerings**.

  Locate the agent software for your platform and download the agent software and the RBA Integration Script Template.

- For a third-party agent that has an embedded RSA agent, go to the RSA Secured web site at **https://gallery.emc.com/community/marketplace/rsa?view=overview**, and locate the listing for *RSA Implementation Guide for Authentication Manager* for your agent.

  Download the *RSA Implementation Guide for Authentication Manager* for your agent. Save it to your desktop or a local drive that you can access during the integration process.

  Note: Only certified partner solutions have an implementation guide. For other agents that are certified as RSA SecurID Ready, you can create a custom implementation.

Next Steps

See Deploying an Authentication Agent on page 64.

Deploying an Authentication Agent

Authentication agents are software applications that securely pass user authentication requests to and from RSA Authentication Manager. Before an authentication agent can communicate with RSA Authentication Manager, you must deploy the agent.

Procedure

1. Use the Security Console to generate a configuration file for the agent. This allows the agent to locate Authentication Manager servers. For more information, see Generate the Authentication Manager Configuration File on page 65.
2. Install an authentication agent on each machine that you want to protect. See your agent documentation for installation instructions.
3. Use the Security Console to add a record for the new agent to the internal database. In this step, you can specify whether you are creating a restricted agent. The agent record identifies the agent to RSA Authentication Manager. This process is called registering the agent. For more information, see Add an Authentication Agent on page 66.

Add an Authentication Agent

Before an authentication agent can communicate with Authentication Manager, you must add the agent to the internal database. This process is called registering the agent. The agent record identifies the agent to Authentication Manager.

Deployments that use risk-based authentication (RBA) require additional configuration. If you use RBA to protect a web-based application, such as an SSL-VPN, web portal, or a thin client, you must integrate the web-based application with Authentication Manager. For more information, see Implementing Risk-Based Authentication on page 273.

Procedure

1. In the Security Console, click **Access** > **Authentication Agents** > **Add New**.
2. From the **Security Domain** drop-down menu, select the security domain to which you want to add the new agent.
3. Under **Authentication Agent Basics**, do the following:
   a. For **Hostname**, enter a new hostname for the agent host, and then click **Resolve IP**.
      The IP address is automatically entered. If you enter a new name, the name must be unique.

      Note: For IPv4/IPv6 agents, the hostname can be any agent descriptor and does not necessarily need to be a fully qualified host name. IP address resolution is not supported for IPv4/IPv6 agents.

   b. (Optional) In the **IP Address** field, enter the IP address of the agent.
      If you use an existing server name, this field is automatically populated and read-only. If no address is specified, UDP agents will use auto-registration to provide the address to the server.

      Note: Do not enter IP addresses in the IPv6 format. IPv4/IPv6 agents will use the hostname to provide the address to the server.

   c. (Optional) In the **Alternate IP Addresses** field, enter alternate IP addresses for the agent.
      You enter alternate IP addresses if the agent has more than one network interface card, or is located behind a static network address translation (NAT) firewall.
      If you use an existing server name, this field is automatically populated and read-only.
4. (Optional) Under **Authentication Agent Attributes,** you can select the following options:
   - To specify the type of agent, select the type from the **Agent Type** list.
     If the agent is a web agent, select **Web Agent**, otherwise keep the default selection **Standard Agent**. The populated agent types are labels, there is no functional difference by choosing Web Agent or Standard Agent.

- To disable the agent, select **Agent is disabled**.

  You might select this option to stop access to a resource temporarily.
- To add a restricted agent, select **Allow access only to members of user groups who are granted access to this agent**.

  Only users who are members of user groups that have permission to access a restricted agent can use this agent to authenticate. Any user can use an unrestricted agent to authenticate.
- To assign a manual or automatic contact list to the new agent, use the Authentication Manager Contact List buttons.

5. (Optional) To configure how users from a trusted realm authenticate to this agent, select **Enable Trusted Realm Authentication**, and then select whether you want to allow all trusted users to authenticate through the new agent or only those trusted users who belong to a trusted user group that has been granted explicit permission to use the agent.
6. (Optional) To allow users to authenticate to this agent using RBA, do the following.
   - Select **Enable this agent for risk-based authentication**.
   - If you want to restrict RBA access on this server agent, select **Allow access only to users who are enabled for risk-based authentication**.
   - Select an authentication method for RBA users.
7. Choose one of the following options to save the settings for this agent.
   - If you enabled this agent for RBA, click **Save Agent and Go to Download Page**.

     The system saves the settings and displays the Integration Script page, where you select and download the integration script for this agent.
   - If you did not enable this agent for RBA, click **Save**.

Note: If the hostname is not a fully qualified host name or the IP address is not specified, a Confirmation Required dialog, summarizing the hostname and the IP address is displayed. Here, you can either edit the agent details or save the agent information.

Next Steps

- Review the configuration settings. See the Security Console Help topic "Configure Agent Settings."
- (Optional) Generate an integration script. See the Security Console Help topic "Generate an integration Script for a Web-based Application."

Node Secret for Encryption

The node secret is a shared secret known only to the authentication agent and Authentication Manager. Authentication agents use the node secret to encrypt authentication requests that they send to Authentication Manager.

Authentication Manager automatically creates and sends the node secret to the agent in response to the first successful authentication on the agent.

The agent and the Authentication manager server must agree on the state of the node secret. For example, if the server expects the agent to have a node secret but the agent does not have one, or if the agent thinks it has a node secret and the server does not think the agent has one.

Manual Delivery of the Node Secret

In most deployments, automatically delivering the node secret is sufficient. However, you can choose to manually deliver the node secret for increased security. When you manually deliver the node secret, you must:

- Use the Security Console to create the node secret. For instructions, see Manage the Node Secret on page 69.
- Deliver the node secret to the agent, for example, on a disk, and use the Node Secret Load utility to load the node secret on to the agent.

The Node Secret Load utility does the following:

- Decrypts the node secret file.
- Renames the file after the authentication service name, usually **securid**.
- Stores the renamed file on your machine. For more information on where the renamed node secret file is stored, see your agent documentation.

When you manually deliver the node secret, take the following security precautions:

- Use the longest possible, alphanumeric password. The maximum length is 16 characters. The minimum length, required special characters, and excluded characters are determined by these default password policy for the deployment.
- If possible, deliver the node secret on external electronic media to the agent administrator, and verbally deliver the password. Do not write down the password. If you deliver the node secret through e-mail, deliver the password separately.
- Make sure that all personnel involved in the node secret delivery are trusted personnel.

For additional information about creating and sending the node secret file, see Manage the Node Secret on page 69.

Manage the Node Secret

To ensure a secure transaction the first time a user attempts to authenticate with a SecurID passcode, the authentication agent and Authentication Manager automatically communicate using a hashed value of the unique node secret and store it on the agent computer. From then on, each authentication interaction uses the node secret to encrypt the communication between the two systems.

Procedure

1. In the Security Console, click **Access** > **Authentication Agents** > **Manage Existing**.
2. Click the **Restricted** or **Unrestricted** tab, depending on whether the agent that you want to search for is restricted or unrestricted.
3. Use the search fields to find the agent with the node secret that you want to manage.
4. Click the agent with the node secret that you want to manage, and click **Manage Node Secret**.
5. If you want to clear the node secret from the Authentication Manager server, select the **Clear Node Secret** checkbox.

   To allow the agent to authenticate to the server, you must also clear the node secret on the agent.
6. (Optional) If you want to create a new node secret, select the **Create Node Secret** checkbox.
7. (Optional) If you chose to create a new node secret, enter and confirm a password to encrypt the node secret file.

   When you create a password, the maximum length is 16 characters. The minimum length, required characters, and excluded characters are determined by the default password policy for the deployment.
8. Click **Save**.
9. Click **Download Now.**

Refresh the Node Secret Using the Node Secret Load Utility

The node secret rarely needs to be refreshed, however there are times when it is necessary. Problems with the node secret can result in authentication or node verification errors. Refresh the node secret when:

- The node secret on the agent is lost, for example, when you perform a factory reset or reinstall the agent.
- The authentication agent record is either deleted or re-added.
- The node secret is deleted from one end of the connection but not the other, for example, the node secret is deleted from the Authentication Manager appliance but not from an associated agent.

You do not need to refresh the node secret when you:

- Change the agent name.
- Change the IP address.

The Node Secret Load utility, agent_nsload, is located in the RSA Authentication Manager 8.1 download kit.

Procedure

1. Create a node secret using the Security Console. For more information, see Manage the Node Secret on page 69.
2. From the RSA Authentication Manager 8.1 download kit, copy **agent_nsload** from the **rsa-ace_nsload** directory for the agent's platform to the agent host.

   RSA provides the following platform-specific versions of the utility:

   - Windows
   - LINUX
   - HP-UX
   - IBM AIX
3. From a command line on the agent host, run the Node Secret Load utility. Type:

   ```
   agent_nsload -f path -p password
   ```

   where:

   - *path* is the directory location and name of the node secret file.
   - *password* is the password used to protect the node secret file.

Automatic Agent Registration

The Automated Agent Registration and Update utility (**sdadmreg.exe)**, included with the RSA Authentication Agent software, enables new authentication agents to automatically add an agent record to the Authentication Manager internal database. This process is called registering the agent. Allowing authentication agents to self-register saves time and money by eliminating the need for an administrator to perform these tasks.

By default, when the agent host starts, the Automated Agent Registration and Update utility automatically runs to allow any IP address changes to be registered in the internal database. You can also run this utility whenever IP address of the agent host changes. This is useful for systems that use the Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. If you use DHCP and do not enable this utility, you must manually update the IP addresses each time the agent host changes its IP address.

You can also run the Automated Agent Registration and Update utility manually whenever the IP address of an agent host changes, to update the IP address in the internal database.

Note: The RSA Authentication Agent 6.1.2 for Microsoft Windows automatically updates the internal database with any IP address changes. If you are using this agent, you do not need to manually run the utility.

It is important that you protect your critical IT infrastructure from potential Denial of Service (DOS) attacks. To reduce the vulnerability of your system:

- Disable agent auto-registration on critical machines, such as e-mail and VPN servers.
- In your IT infrastructure, give critical agents static IP addresses.
- Protect IP addresses within Authentication Manager. To do this, select Protect IP Address on the Authentication Agent page in the Security Console.

Allow an Agent to Auto-Register

Authentication agents can automatically add an agent record to the internal database. The process of adding an agent record is called registering the agent.

If your network uses Dynamic Host Configuration Protocol (DHCP) to assign IP addresses, consider enabling agent auto-registration. When enabled, agent IP addresses are automatically updated whenever the IP address of the agent host changes.

The Auto-Registration utility automatically registers users' computers in the Authentication Manager database the first time users start their computers with Authentication Agent installed. This utility and eliminates the need for an administrator to manually create the agent host record.

Before You Begin

- Install an authentication agent.
- Configure the agent host. See your agent documentation for information.

Procedure

1. In the Security Console, click **Setup > System Settings> Agents**.
2. Select **Allow authentication agent auto-registration**.
3. Click **Save**.

Download an RSA Authentication Manager Server Certificate

The RSA Authentication Manager server certificate is required to set up agent auto-registration.

Procedure

1. In the Security Console, click **Access** > **Authentication Agents** > **Download Server Certificate File**.
2. Click **Download Now**.
3. Select **Save it to disk**.

Next Step

To set up agent auto-registration, copy the file to the auto-registration installation directory on the agent host machine and run the **.exe** file.

Contact Lists for Authentication Requests

Contact lists are ordered lists of instances available to accept authentication requests, and are created either automatically by Authentication Manager, or manually by an administrator.

Authentication Manager uses contact lists to determine to which instance authentication requests are sent. Authentication Manager sends contact lists to each agent after the initial contact between the agent and Authentication Manager.

Depending on your license type, your Authentication Manager deployment can have a primary instance and up to 15 replica instances. To increase efficiency, use contact lists to route authentication requests from agents to the instances that can respond the quickest.

Agents request new contact lists as a part of subsequent authentications. Periodically, the agent reviews all the instances listed in the contact list to determine where to send authentication requests. The agent uses metrics, such as the amount of time it takes the instance to respond to authentication requests, to determine where to send requests.

If none of the servers on the contact list respond to authentication requests, the agent reverts to the Authentication Manager configuration file and uses an IP address in the configuration file to reconnect with Authentication Manager.

RSA RADIUS supports contact lists for the RADIUS server agent. RADIUS client agents do not support contact lists because there is no authentication agent software installed on RADIUS clients. The associated agent record in the internal database enables Authentication Manager to track RADIUS authentication attempts made through the RADIUS server. For more information, see RADIUS Clients on page 297.

IPv4/IPv6 agents do not support contact lists.

Automatic Contact Lists

An automatic contact list is assigned to each instance in your deployment. The list contains the IP addresses of each instance the contact list is assigned to, up to a limit of 11. Agents receive automatic contact lists by default.

Authentication Manager automatically updates these lists each time a new instance is added to the deployment. When the list is updated, a time stamp associated with the list is also updated. Agents use this time stamp to determine when to request an updated list.

The Super Admin can edit an automatic contact list in the Security Console on the Edit Authentication Manager Contact List page. Any edits that you make to an automatic contact list may be overwritten when a new instance is added to the deployment.

Manual Contact Lists

The Super Admin updates manual contact lists to reflect the most recent list of instances. Manual lists can contain the IP address of any instance in the deployment, up to a limit of 11.

For many organizations, automatic contact lists are sufficient. However, you may choose to create a manual contact list if you have a specific way that you want to route authentication requests.

For example, suppose that you are an administrator at a company that has Boston, New York, and San Jose locations. The New York and San Jose locations are small and all authentications are routed to Authentication Manager replica instances at each site. The Boston location, however, is largest, and the primary instance at that location handles all Boston location users, as well as all VPN requests from external users. You can create a manual contact list that routes authentication requests to the replica instances. This leaves the primary instance free to replicate data to the replica instances in New York and San Jose.

For instructions, see the Security Console Help topics, "Add a Manual Contact List," "Assign a Manual Contact List to an Authentication Agent," and "Edit a Manual Contact List."

6 Administering Users

Common User Administration Tasks

You use the Security Console to provide and maintain authentication service for users. The Security Console enables you to perform the following common tasks:

- Manage user data
- Restrict access to specific resources
- Specify user access privileges
- Resolve user access problems
- Respond to user security issues
- Maintain user data

The administrative tasks that you can perform are defined by the scope of your administrative role.

~~Add a User to the Internal Database~~

~~You can use the Security Console to add users to the internal database even if an LDAP directory is the primary identity source. Adding users directly to the internal database allows you to create a group of users different from those in identity source. For example, you might store a group of temporary contractors or a specific group of administrators in the internal database. You might also use the internal database to store a small number of users for a pilot project.~~

~~User data in an LDAP directory is read-only. You must add users to the LDAP directory using the directory tools. However, you can use the Security Console to perform certain administrative functions, such as assigning tokens or enabling a user for risk-based authentication.~~

~~Procedure~~

1. ~~In the Security Console, click **Identity** > **Users** > **Add New**.~~
2. ~~In the Administrative Control section, from the **Security Domain** drop-down list, select the security domain where you want the user to be managed. The user is managed by administrators whose administrative scope includes the security domain you select.~~

c. (Optional) Select **Disabled** if you want to disable the new account.

d. If a Locked Status option is selected, you can unlock the user by clearing all selected options.

6. (Optional) Under **Attributes**, enter the user's mobile phone number in the **Mobile Number (String)** field.

7. Click **Save**.

User Status

To increase security, you can disable users who take an extended absence, and enable these users when they return to work.

Before enabling or disabling users, know the following:

- Disabling a user does not delete that user from the identity source.
- When a user account is disabled, any tokens belonging to the user remain assigned. Disabling a user account does not unassign the user's assigned tokens.
- Authentication Manager verifies whether a user is enabled or disabled each time a user authenticates. If a user in a linked identity source is disabled, that user cannot authenticate.

Disable a User Account

When you disable a user account, you suspend the user's permission to authenticate, which prohibits access to protected resources. Disabling a user does not delete the user from the identity source. To delete a user, see the Security Console Help topic "Delete a User."

If you want to disable a user in an LDAP directory that is linked to RSA Authentication Manager, you must use the native LDAP directory interface.

Procedure

1. In the Security Console, click **Identity** > **Users** > **Manage Existing**.
2. Use the search fields to find the user that you want to disable. Some fields are case sensitive.
3. Click the user that you want to disable, and select **Edit**.
4. Under **Account Information**, select **Account is disabled**.
5. Click **Save**.

Enable a User Account

When you enable a user account, the user can authenticate and access protected resources.

To authenticate users to a directory server, you must enable the user in both the directory server and in the Security Console. Only users who are enabled in the directory server can authenticate to the directory server.

Procedure

1. In the Security Console, click **Identity** > **Users** > **Manage Existing**.
2. Use the search fields to find the user that you want to enable. Some fields are case sensitive.
3. Click the user that you want to enable, and select **Edit**.
4. Under **Account Information**, clear **Account is disabled**.
5. Click **Save**.

Security Domains to Organize Users

After you create the security domain hierarchy and link the identity source to the system, all users are added to the top-level security domain. To help you organize users, manage the deployment, and limit administrative scope, you may want to move users to another security domain in the hierarchy.

Just as you have likely created security domains to match either your organization's structure or geographic locations, you can use the Security Console to transfer users from each department or location to their respective security domains.

For example, if the top-level security domain is named SystemDomain, and you have lower-level security domains named Boston, New York, and San Jose, you would likely move users from SystemDomain to their respective security domains.

For more information about security domains, see Security Domain Overview on page 39.

Move Users Between Security Domains

You can manually move users whose accounts are stored in the internal database to other security domains. You can also move user groups.

When you move users to another security domain, the policies for the new security domain take effect immediately. Also, after you move users, only administrators with permissions to manage users in that security domain can manage the users you moved.

When you move users, consider that users who are enabled for risk-based authentication (RBA) before the move retain their RBA user settings after the move. If users are disabled for RBA before the move, the users remain disabled for RBA after the move.

You can automatically move LDAP directory users to other security domains by mapping directory objects, such as organizational units, to the security domain of your choice. Authentication Manager uses security domain mappings to add users to the appropriate security domain when new user records are added to the database.

Procedure

1. In the Security Console, click **Identity** > **Users** > **Manage Existing**.
2. Use the search fields to find the users that you want to move. Some fields are case sensitive.

3. Select the users that you want to move.
4. From the Action menu, select **Move to Security Domain**, and click **Go**.
5. From the **Move to Security Domain** drop-down list, select the security domain where you want to move the user.
6. Click **Move**.

Duplicate User IDs

If two users with the same user name attempt to access the same protected resource, authentication will fail. This may occur if you link multiple identity sources to the same deployment and users with the same User ID exist in each identity source. In these cases, you have the following options:

- Map the User ID to another field where there are no duplicate values. For example, for an Active Directory identity source, you might be able to map to the UPN field or to a user's email address.
- Change one of the User IDs in the identity source so that both User IDs become unique. This option may not be practical if the User ID is used for other applications.
- Assign authenticators to only one of the users with the duplicate User ID. This option is not practical if authenticators must be assigned to more than one user with the duplicate User ID.
- You can allow one user to authenticate with a logon alias, and you can prevent this user from authenticating with the default User ID.

User Authentication

You use the Security Console to manage user authentication. You can:

- Modify user authentication settings
- Resolve user access problems

The user authentication tasks that you can perform are defined by the scope of the administrative role.

Manage User Authentication Settings

User authentication settings allow you to create exceptions to authentication policies for individual users. These settings also allow you to troubleshoot user authentication issues.

Before You Begin

You must have a restricted or unrestricted agent. If you plan to configure a logon alias, the user must belong to a user group that has access to a restricted agent or has been enabled on an unrestricted agent.

Procedure

1. In the Security Console, click **Identity** > **Users > Manage Existing**.
2. Use the search fields to find the user that you want to manage.
3. From the search results, click the user that you want to manage.
4. From the context menu, click **Authentication Settings**.
5. If you want to assign a fixed passcode to the user, select the **Fixed Passcode** checkbox.

   RSA recommends that you do not use fixed passcodes because they eliminate all the advantages of two-factor authentication.
6. Select the **Clear Incorrect Passcodes** checkbox to clear any incorrect passcodes. The count of incorrect passcodes is reset, and the user is not prompted for the next tokencode. However, if the user continues to enter incorrect passcodes and exceeds the number of failed logon attempts allowed by the lockout policy, the user is locked out of the system.
7. Select **Clear cached copy of selected user's Windows credential** to clear a cached version of a user's password.

   If your deployment uses RSA SecurID for Windows, Authentication Manager saves a cached version of the user's Windows logon password. This information may need to be cleared, if the Windows password has been changed in Active Directory.
8. If you want to assign a default shell to the user, enter it in the **Default Shell** field.
9. To configure a logon alias for the user:
   a. Select whether you want to allow users to use their own User IDs and the alias.

      You can use this option to prevent a conflict between users who share the same default User IDs.
   b. Select the user group to which you want to assign the alias.
   c. In the **User ID** field, enter the User ID that you want to assign to the alias. In the **Shell** field, enter the shell that you want assigned to the alias. If you are using RADIUS, from the **RADIUS Profile** drop-down menu, select the RADIUS profile to assign to the alias. Click **Add**.
10. If you use RADIUS, select the RADIUS profile and RADIUS user attributes to assign to the user:
    a. From the **User RADIUS Profile** drop-down menu, select a RADIUS profile to assign to the user.

       If you set up logon aliases for the user and you do not specify a RADIUS profile for each alias in step 9, Authentication Manager assigns the user RADIUS profile to each alias.
    b. In **RADIUS User Attributes**, select the attribute that you want to assign to the user, enter the value for the attribute in the **Value** field, and click **Add**. RADIUS user attributes take precedence over attributes in a RADIUS profile.

A RADIUS user attribute can be mapped to an identity source attribute. For more information, see Map a RADIUS User Attribute Definition to an Identity Source Attribute on page 313.

11. Click **Save**.

Logon Alias

A logon alias allows users to log on with a user group ID. The user group ID is associated with a user group that has access to a restricted agent or that has been enabled on an unrestricted agent.

For example, users can have a User ID based on their first initial and last name, such as, "kmiller," as well as an administrative User ID with a specific name, for example "root." If a logon alias is established, Authentication Manager verifies the authentication using the user's passcode, regardless of the User ID that the user entered to log on to the operating system. For backward compatibility, a shell value is also maintained by the system.

A logon alias can also be used in deployments where there are users with the same User ID. An alias that further identifies the user may prevent conflicts when these users attempt to authenticate. In the authentication settings for a user, you can also prevent a user from authenticating with the default User ID and instead require that the user authenticate with an alias.

To allow a user to authenticate with a logon alias on a restricted agent, you must grant the user group that is associated with the alias access to the agent. Although all users within a deployment can access an unrestricted agent, a user cannot authenticate with a logon alias until you enable the user group that is associated with the alias on the unrestricted agent.

You can assign logon aliases on the Authentication Settings page in the Security Console. This page is accessed through the user Context menu.

For instructions, see Manage User Authentication Settings on page 123.

Unlock a User

RSA Authentication Manager locks out users who violate the lockout policy. Locked out users cannot authenticate until they are unlocked.

The lockout policy specifies the number of failed authentication attempts allowed before the system locks the account. A lockout policy can unlock users after a specific time period, or you can require an administrator to manually unlock the user.

Procedure

1. In the Security Console, click **Identity > Users > Manage Existing**.
2. Use the search fields to find the user that you want to unlock. Some fields are case sensitive.
3. Click the user that you want to unlock, and select **Edit**.
4. Under **Account Information**, go to **Locked Status**, and clear all options that are selected.
5. Click **Save**.

Incorrect Passcode Count

The system counts each time the assigned user enters an incorrect passcode, clearing this count automatically with each correct passcode. If a user enters more incorrect passcodes than are allowed by the SecurID token policy and then enters a correct passcode, the user is prompted for the next tokencode. If you do not want a user to be prompted for the next tokencode, you can use the Security Console to clear the number of incorrect passcodes entered.

This operation only clears the existing count. To clear future counts, you must perform the procedure again.

For instructions, see Manage User Authentication Settings on page 123.

Managing Security Questions

Security questions is an authentication method that requires users to answer questions in order to authenticate. During enrollment or when users access the Self-Service Console for the first time, users are presented with several questions, which they must answer. Later when users authenticate, the users must answer a subset of these questions with the same answers that they provided during enrollment.

Security questions are used under the following conditions:

- when the primary authentication method results in a failed authentication
- to confirm identity for risk-based authentication (RBA)

If you want to allow users to change their answers, you must clear their existing answers. For example, you might need to do this when users forget their answers, or when users believe that their answers are compromised. After you clear a user's answers, the user is prompted to provide new answers at the next logon. For instructions, see the Security Console Help topic "Clear Security Question Answers."

A file of questions is provided for English-speaking users, which you can modify to create a new question file. You can also create a file of non-English questions in any supported language. When you create a new set of questions or modify just one question, the new file replaces the existing file.

You specify the number of questions that users must answer during enrollment or when accessing the Self-Service Console for the first time. You also specify the number of questions that users must answer during authentication. The number of questions that you specify for enrollment should be greater than the number of questions that you specify for authentication. If you specify fewer questions for authentication than you specify for enrollment, users can choose which questions to answer for authentication.

For self-service troubleshooting, the number of available questions must exceed the number of questions required for authentication.

Next Step

Import each new security questions file. For instructions, see the Security Console Help topic "Import Security Questions."

Emergency Online Authentication

Online authentication provides emergency access for users with missing or damaged tokens. Even with a missing token, users can continue to have two-factor authentication using an online emergency access tokencode, an 8-character alphanumeric code generated by Authentication Manager.

The format of the online emergency access tokencode is determined by the token policy of the associated security domain. For example, if the security domain's token policy allows special characters, the online emergency access tokencode can include special characters.

If a user has an expired token, assign a new token, and then provide temporary access. An online emergency access tokencode cannot be assigned to a user with an expired token. For instructions, see Provide an Offline Emergency Passcode on page 133.

The following table lists the types of online emergency access tokencodes. Both tokencode types replace the tokencode generated by the user's token.

Tokencode Type	Description
Temporary fixed tokencode	• Can be used more than once. • Must be combined with the user's RSA SecurID PIN to create a passcode. • Is displayed on the Self-Service Console.
One-time tokencode	• Issued in sets. You can determine the number of tokencodes in a set. • Must be combined with the user's RSA SecurID PIN to create a passcode. • Is displayed on the Self-Service Console. • Users can download the set of one-time tokencodes in a file. • Each tokencode in the set can only be used once.

Users can also use the Self-Service Console to request temporary access to Authentication Manager without the assistance of an administrator. For more information, see RSA Self-Service on page 233.

Assign a Set of One-Time Tokencodes

You can provide temporary access for a user whose token has been permanently lost or destroyed by assigning a set of one-time tokencodes. A one-time tokencode replaces the tokencode generated by the user's missing token. Users must enter their PIN and a one-time tokencode to perform two-factor authentication.

Each one-time tokencode in a set can be used once. A set of tokencodes allows a user to authenticate multiple times without contacting an administrator each time.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Use the search fields to find the appropriate token.
3. From the search results, click the token with which you want to work.
4. From the context menu, click **Emergency Access Tokencodes**.
5. On the Manage Emergency Access Tokencodes page, select the **Online Emergency Access** checkbox to enable authentication with an online emergency access tokencode.
6. Select **Set of One-Time Tokencodes**.
7. Enter the number of tokencodes that you want to generate.
8. Click **Generate Codes**. The set of tokencodes displays below the Generate Codes button.
9. Record the set of one-time tokencodes so you can communicate them to the user.
10. Select one of the following options for the **Emergency Access Tokencode Lifetime**:
    - No expiration.
    - Set an expiration date for the tokencode.
11. In the **If Token Becomes Available** field, configure how Authentication Manager handles lost or unavailable tokens that become available.
    - Deny authentication with the recovered token.

      If a token is permanently lost or stolen, deny authentication with the recovered token so that it cannot be used for authentication if recovered by an unauthorized individual. This is essential if the lost token does not require a PIN.
    - Allow authentication with the recovered token while simultaneously disabling the emergency access tokencode.
    - Allow authentication with the recovered token only after the emergency access tokencode has expired.
12. Click **Save**.

Assign a Temporary Fixed Tokencode

Occasionally, it is necessary to give a user temporary access to resources protected by RSA SecurID. For example, you can give temporary access to a user whose existing token has been lost or destroyed. Temporary access allows a user to access protected resources while waiting for a replacement token.

This procedure provides a user with temporary emergency access using a temporary fixed tokencode.

A temporary fixed tokencode replaces the tokencode generated by the user's token. Similar to the regular tokencode, the temporary fixed tokencode is entered with the user's PIN to create a passcode. By using a PIN with the temporary fixed tokencode, the user can still achieve two-factor authentication.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. On the **Assigned** tab, use the search fields to find the lost or destroyed token.
3. From the search results, click the lost or destroyed token, and from the context menu, select **Emergency Access Tokencodes**.
4. On the Manage Emergency Access Tokencodes page, select **Online Emergency Access**.
5. For **Type of Emergency Access Tokencode(s)**, select **Temporary Fixed Tokencode**.
6. Click **Generate New Code**. The tokencode displays next to the **Generate New Code** button.
7. Record the emergency access tokencode so that you can communicate it to the user.
8. For **Emergency Access Tokencode Lifetime**, select either **No expiration** or select **Expire on** and specify an expiration date.

   You may want to limit the length of time the one time tokencode can be used. Because the one-time tokencode is a fixed code, it is not as secure as the pseudorandom number generated by a token.
9. For **If Token Becomes Available**, select one of the following options:
   - **Deny authentication with token**.

     Select this option if the token is permanently lost or stolen. This option prevents the token from being used for authentication if recovered. This safeguards the protected resources in the event the token is found by an unauthorized individual who attempts to authenticate.
   - **Allow authentication with token at any time and disable online emergency tokencode**.

     Select this option if the token is temporarily unavailable (for example, the user left the token at home). When the user recovers the token, he or she can immediately resume using the token for authentication. The online emergency access tokencode is disabled as soon as the recovered token is used.
   - **Allow authentication with token only after the emergency code lifetime has expired and disable online emergency tokencode**.

     You can choose this option for misplaced tokens. When the missing token is recovered, it cannot be used for authentication until the online emergency access tokencode expires.
10. Click **Save**.

Emergency Offline Authentication

Offline authentication provides emergency access for RSA SecurID for Windows users who require emergency access while authenticating offline. These are users with lost or stolen tokens, or users who have forgotten their PIN. Temporary emergency access can be provided in two ways:

- **Offline emergency access tokencode.** Use this option if the user's token is unavailable. The Offline Emergency Access Tokencode is used with the user's PIN.
- **Offline emergency passcode.** Use this option if a user has forgotten his or her PIN. The offline emergency passcode is used in place of the user's PIN and tokencode.

RSA SecurID for Windows users may need temporary emergency access so that they can authenticate while working offline. Temporary emergency access is necessary for users with misplaced, lost, or stolen tokens, or users who have forgotten their PIN.

For offline authentication, the system generates and downloads an offline passcode or tokencode before the user needs it. Providing emergency offline authentication codes must be done in advance. Authentication codes cannot be sent to a user who is offline.

Provide an Offline Emergency Access Tokencode

An offline emergency access tokencode replaces the tokencode generated by the user's token. Similar to the tokencode used in non-emergency circumstances, the user enters the offline emergency access tokencode with a PIN to create a passcode, thus achieving two-factor authentication.

You can configure the following:

- Specify that a new offline emergency access tokencode is downloaded the next time the user authenticates online.
- Allow the offline emergency access tokencode to be used for online and offline authentication.

Before You Begin

- The user's security domain must allow offline authentication and permit the user to download offline emergency access tokencodes.
- The user must have authenticated to an agent that supports offline authentication and the agent has downloaded days of offline authentication data.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Use the search fields to find the token for the user who needs an offline emergency access tokencode.
3. From the search results, click the token.
4. From the context menu, click **Emergency Access Tokencodes**.

5. On the Manage Emergency Access Tokencodes page, note the Offline Emergency Access Tokencode and its expiration date.
6. Select **Reset Offline Emergency Access Tokencode**, if you want the user to download a new offline emergency access tokencode the next time he or she authenticates online. If selected, the new tokencode downloads automatically.
7. Click **Use offline code for online access**, if you want the offline emergency access tokencode used for online authentication.
8. Click **Save**.

Provide an Offline Emergency Passcode

An offline emergency passcode takes the place of the passcode (PIN + tokencode) that the user normally enters. The user does not need to possess the token or know the PIN to authenticate offline.

Before You Begin

Confirm the following:

- The user's security domain allows offline authentication and permits the user to download offline emergency access tokencodes.
- The user has authenticated to an agent that supports offline authentication and the agent has downloaded days of offline authentication data.

Procedure

1. In the Security Console, click **Identity > Users > Manage Existing**.
2. Use the search fields to find the user who needs an offline emergency passcode.
3. From the search results, click the user.
4. From the context menu, click **Manage Emergency Offline Access**.
5. On the Manage Emergency Access Passcodes page, note the Offline Emergency Passcode and its expiration date.
6. Select **Reset Offline Emergency Access Passcode**, if you want the user to download a new offline emergency passcode the next time he or she authenticates online. If selected, the new passcode downloads automatically.
7. Click **Update**.

RSA SecurID PINs

A personal identification number (PIN) is a numeric password used to authenticate a user.

To increase security, you can set the token policy to require users to create PINs containing both letters and numbers and to change their PINs at regular intervals. See Token Policy on page 76.

Misplaced or stolen PINs puts protected resources at risk. For this reason, you should instruct users to report compromised PINs as soon as possible.

When a user reports a compromised PIN, you can require the user to change his or her PIN after the next successful authentication.

When a user is required to change a PIN, the user must know his or her current PIN. To change a PIN, the user authenticates using the existing PIN and tokencode. After successfully authenticating, the user is prompted to create and confirm a new PIN, and the PIN is associated with the user's token.

For example, suppose a user reports that she used her computer at a local coffee shop, and now she is worried that someone may have seen her type her PIN. After you receive the report, you use the Security Console to require the user to change her PIN. For instructions, see Require Users to Change Their RSA SecurID PINs on page 135.

The token policy may require the user to use a system-generated PIN instead of creating one. After the next authentication, the system provides the user with a new, system-generated PIN. The user then authenticates again using the new, system-generated PIN.

If users forget their PINs, you cannot require them to change their PINS in order to obtain a new one because users need to know their PINs in order to change them. When a user forgets his or her PIN, you must clear the PIN before the user can create a new one. For instructions, see, Clear an RSA SecurID PIN on page 135.

Users can also use Self-Service to reset their PINs.

Set an Initial On-Demand Authentication PIN for a User

On-demand authentication (ODA) always requires a PIN to request a tokencode. You can set the initial PIN for the user. The following procedure is for a user who is not yet enabled for ODA.

Procedure

1. In the Security Console, click **Identity** > **Users** > **Manage Existing**.
2. Use the search fields to find the user for whom you want to enable ODA and set an initial PIN.
3. Click the user.
4. Select **SecurID Tokens**.
5. Under **On-Demand Authentication**, select **Enable User**.
6. For **Expiration Date**, specify **No expiration** or the date when on-demand authentication expires.
7. For **Send On-Demand Tokencodes to**, specify the delivery method.
8. For **Associated PIN**, do one of the following:
   - Select **Require user to setup the PIN through RSA Self-Service Console**.
   - Select **Set initial PIN to**, and enter the initial PIN.
9. Click **Save**.
10. If you have set the initial PIN, securely communicate the initial PIN to the user.

Clear a User's On-Demand Authentication PIN

You might clear a user's on-demand authentication (ODA) PIN when the PIN is compromised, forgotten, or when your company policy requires the PIN change. You must always set a temporary PIN when you clear a user's PIN because ODA requires a PIN.

The user must change a temporary PIN the first time it is used.

Procedure

1. In the Security Console, click **Identity** > **Users** > **Manage Existing**.
2. Use the search fields to find the user for whom you want to set a temporary PIN.
3. Click the user.
4. From the context menu, select **SecurID Tokens**.
5. Under **On-Demand Authentication**, for **Associated Pin**, select **Clear existing PIN and set a temporary PIN for the user**.
6. Enter a temporary PIN.
7. Click **Save**.
8. Securely communicate the temporary PIN to the user.

Require Users to Change Their RSA SecurID PINs

When you require a user to change a SecurID PIN, the user is prompted to create a new PIN after successfully authenticating with the token.

You can require a PIN change only when a user knows the existing PIN. For example, you might require a user to change a SecurID PIN if the current PIN has been compromised. If a user has forgotten the PIN, clear the PIN.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Click the **Assigned** tab.
3. Use the search fields to find the token that you want to edit.
4. From the search results, click the token with the PIN that you want the user to change.
5. Select **Require SecurID PIN Change**.

Clear an RSA SecurID PIN

When a user forgets a SecurID PIN, you can clear the PIN so that the user can create a new one. When you clear a user's PIN, the user can create a new PIN the next time the user authenticates.

For example, suppose a user has forgotten a PIN and calls for help. You verify the user's identity and clear the PIN. You tell the user to enter just the tokencode when prompted for the passcode the next time user authenticates. After entering the tokencode, the user is prompted to create a new PIN for the user's token.

9 Deploying and Administering RSA SecurID Tokens

RSA SecurID Tokens

RSA SecurID tokens offer RSA SecurID two-factor authentication. An RSA SecurID token generates a 6-digit or 8-digit pseudorandom number, or tokencode, at regular intervals. When the tokencode is combined with a personal identification number (PIN), the result is called a passcode. Users enter passcode values, along with other security information, to verify their identity to resources protected by Authentication Manager. If Authentication Manager validates the passcode, the user is granted access. Otherwise, the user is denied access.

There are two kinds of SecurID tokens, hardware tokens and software tokens. Hardware tokens generate tokencodes using a built-in clock and the token's factory-encoded random key, known as the "seed." Hardware tokens come in several models, such as key fobs and PINPads. Software tokens consist of two components that are installed separately, an application specific to the intended device platform and a token seed record. Software token applications generate tokencodes on the device and offer the same passcode functionality as hardware tokens. Devices include smart phones, computers, and tablets.

Hardware and software tokens require similar administrative tasks. Following deployment, you can perform many token-related administrative tasks with the User Dashboard in the Security Console. For more information, see the Security Console Help topic "User Dashboard."

By default, RSA provides tokens that require a PIN and strongly recommends that you use PINs for all tokens. PINs provide the second factor in RSA SecurID two-factor authentication. RSA Authentication Manager also supports authentication with tokens that do not require an RSA SecurID PIN.

Deploying RSA SecurID Tokens

The following steps outline the tasks required to deploy RSA SecurID tokens.

Procedure

1. Import a Token Record File on page 188.
2. (Optional) Move a Token Record to a New Security Domain on page 189.
3. Assign Tokens to Users on page 189.
4. For software tokens, Add a Software Token Profile on page 192. Otherwise continue to step 5.
5. Distribute the token. Choose one of the following:

- Distribute a Hardware Token on page 194
- Distribute Multiple Software Tokens Using File-Based Provisioning on page 194
- Distribute One Software Token Using File-Based Provisioning on page 196
- Distribute Multiple Software Tokens Using Dynamic Seed Provisioning (CT-KIP) on page 197
- Distribute One Software Token Using Dynamic Seed Provisioning on page 198
- Distribute Multiple Software Tokens Using Compressed Token Format (CTF) on page 200
- Distribute One Software Token Using Compressed Token Format (CTF) on page 201

Import a Token Record File

RSA manufacturing provides an XML file that contains the token records that your organization has purchased. Before you can work with individual token records, you must import the token record XML file into Authentication Manager.

For hardware tokens, each token record in the file corresponds to a hardware token that your organization has purchased.

For software tokens, token record data will eventually be transferred into a software token application. Each token record contains the token seed and metadata such as the token serial number, expiration date, and the tokencode length and interval.

Before You Begin

- Decide which security domain will own the imported tokens. The security domain must be in the administrative scope of the administrator who will deploy and manage the tokens.
- Your administrative role must permit you to manage tokens.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Import Tokens Job > Add New**.
2. Enter a name for the import job. The job is saved with this name so that you can review the details of the job later. The name must be from 1 to 128 characters. The characters & % > < are not allowed.
3. From the **Security Domain** drop-down menu, select the security domain into which you want to import the tokens. The tokens are managed by administrators whose scope includes this security domain. By default, tokens are imported into the top-level security domain.
4. Browse to select the token files that you want to import.
5. In the **File Password** field, enter a password if the file is password protected.

6. Use the **Import Options** radio buttons to specify handling for duplicate tokens.
7. Click **Submit Job.**

Next Steps

Assign tokens to users. For more information, see Assign Tokens to Users on page 189.

Move a Token Record to a New Security Domain

You can select a new security domain to move token records. If you do not select a security domain, the token records are imported into the top-level security domain by default. You can move token records between security domains within a deployment using the Security Console. You can also change an administrator's scope to manage token records, or you can move the records to a security domain that is associated with the location where the tokens will be used.

Procedure

1. In the Security Console, click **Authentication** > **SecurID Tokens** > **Manage Existing**.
2. Use the search fields to find the tokens that you want to move.
3. Select the checkboxes next to the tokens that you want to move.
4. From the **Action** menu, click **Move to Security Domain**.
5. Click **Go**.
6. From the **Move to Security Domain** drop-down list, select the security domain to which you want to move the tokens.
7. Click **Move**.

Assign Tokens to Users

Assigning a token associates the token with a specific user. You must assign a token to a user before the user can authenticate. You can assign a maximum of three tokens to each user. RSA recommends that you do not assign more than one hardware token to a user as this may increase the likelihood that users will report a lost or stolen token. Some software token applications may not allow multiple tokens. For platform-specific information on token limitations, see the RSA SecurID software token documentation.

You can also assign tokens with the User Dashboard. For instructions, see the Security Console Help topic "User Dashboard."

Before You Begin

- Import the token records from the token record file to the internal database. For more information, see Import a Token Record File on page 188.
- Make sure a user record exists in Authentication Manager for each user to whom you want to assign a token.

Procedure

1. In the Security Console, click **Identity > Users > Manage Existing**.
2. Use the search fields to find the users to whom you want to assign tokens.
3. From the search results, click the user(s) to whom you want to assign tokens.
4. From the context menu, under SecurID Tokens, click **Assign More**.
5. From the list of available RSA SecurID tokens on the Assign to Users page, select the checkboxes for the tokens that you want to assign. Record which tokens you assign so you can deliver them later.
6. Click **Assign**.

Next Steps

Add a Software Token Profile on page 192

Software Token Profiles

Software token profiles specify software token configurations and distribution processes. A software token profile is required for each platform for which you plan to distribute software tokens. Only a Super Admin can add software token profiles to the deployment. Software token profiles are available to the entire deployment and are not specific to a security domain.

Device Definition File

A device definition file is an XML file that defines the capabilities and attributes of software tokens used on a specific platform, for example, the Android platform or the iOS platform. The file identifies the supported tokencode characteristics, the token type, whether the token is CT-KIP capable, CT-KIP link format, whether the token is CTF capable, and the supported binding attributes.

When RSA releases applications for new software token types, the applications often require new device definition files. You must import the new device definition file when you create a software token profile using the new token type. To determine whether you need to import a new device definition file, see the *Administrator's Guide* for your software token application.

Software Token Configuration

RSA SecurID software tokens are factory-set as PINPad PIN type (PIN integrated with tokencode), 8-digit tokencode length, and 60-second tokencode interval. However, you can configure the tokencode interval, PIN type, and tokencode length of software tokens for each software token profile that you create. Depending on configuration options set in the device definition file, you can set the tokencode length to 6 or 8 digits and the tokencode interval to 30 or 60 seconds. You can change the PIN type so that the token behaves like a hardware fob. You can also reconfigure the token to be tokencode only.

Software Token Delivery Methods

The following methods are available for providing token data to a software token application:

Dynamic Seed Provisioning (CT-KIP). The dynamic seed provisioning method uses the four-pass Cryptographic Token Key Initialization Protocol (CT-KIP) to exchange information between an RSA SecurID client application running on a mobile device, desktop, or desktop, and the CT-KIP server, which is a component of the Authentication Manager server. The information exchanged between the client and server is used to generate a unique shared secret (token seed). Information critical to the seed generation is encrypted during transmission using a public-private key pair. The generated token seed value is never transmitted across the network. Dynamic seed provisioning is preferred over file-based provisioning because the four-pass protocol prevents the potential interception of the token's seed during the provisioning process.

If you configured activation codes to expire, a user must provide the activation code to the client application before the code expires. If the activation code is not used before the expiration time, you must redistribute the token, and provide the CT-KIP URL and the new activation code to the user.

The four-pass CT-KIP protocol is initiated by a request from the client application to the CT-KIP server when the user selects an import token option on the client device. Dynamic seed provisioning uses a unique one-time provisioning activation code to ensure that the request is legitimate The client application must be provided with the activation code, either through manual user entry or as part of a URL string sent to the user's device e-mail. The CT-KIP server evaluates the activation code, and if the server determines that the request is valid, the four-pass process continues, ultimately resulting in a successful import operation.

File-Based Provisioning. With file-based provisioning, Authentication Manager generates token data contained within a file, which is added to a ZIP file for download. Software token files provisioned using this method have the extension .sdtid. The data in the token file includes the seed used by the SecurID algorithm and other metadata, including the token serial number, expiration date, number of digits in the tokencode, and so on. To protect the seed against attack, the seed is encrypted using the AES encryption algorithm and an optional password that you can assign during the configuration process. RSA recommends protecting file-based tokens with a strong password that conforms to guidelines provided in the *RSA Authentication Manager 8.1 Security Configuration Guide.*

Compressed Token Format (CTF) Provisioning. E-mail programs on some mobile device platforms cannot interpret .sdtid file attachments. In such cases, you can deliver file-based tokens using Compressed Token Format (CTF). Authentication Manager generates token data in the form of a CTF URL string, which you deliver to the user's device by e-mail as a URL link. CTF URL strings contain the encoded token data needed by the software token application. This encoded data includes the seed used by the SecurID algorithm and other metadata, including the token serial number, expiration date, number of digits in the tokencode, and so on. The URL format signals the device that the URL link contains data relevant to the software token application. RSA recommends protecting CTF format tokens with a strong password that conforms to guidelines provided in the *RSA Authentication Manager 8.1 Security Configuration Guide*.

Device Attributes

Device attributes are used to add information to software tokens. You can use the **DeviceSerialNumber** field to restrict the installation of a token to a device platform or to a specific device. The default **DeviceSerialNumber** value associated with the device type binds the token to a specific platform. Binding to a device platform allows the user to install the token on any device that runs on that platform, for example, any supported Android device. The user cannot install the token on a different platform, such as Apple iOS.

For additional security, you can bind a token to a single, device-specific identification number, for example, a separate, unique device ID assigned by the RSA SecurID software token application. In this case, the token can only be installed on the device that has the device-specific ID. If a user attempts to import the token to any other device, the import fails. You must bind tokens before you distribute them. In some cases, you must obtain the device binding information from the user. The user must install the software application before providing the binding information. For more information, see your RSA SecurID software token documentation.

The **Nickname** field allows you to assign a user-friendly name to the token. When the token is installed into the application on the device, the application displays the token nickname. If you do not assign a nickname, the application displays a default name, for example, the token serial number. Not all software token applications support nicknames.

Add a Software Token Profile

Software token profiles specify software token configuration and distribution options. You must configure a software token profile for each platform to which you plan to distribute software tokens.

Before You Begin

You must be a Super Admin.

Procedure

1. In the Security Console, click **Authentication** > **Software Token Profiles** > **Add New**.
2. Enter the **Profile Name**. Try to include the device type or distribution method in the profile name. For example, "Android_CT_KIP."
3. Do one of the following:
   - To choose an existing device type, select one from the **Device Type** drop-down list.
   - To load a new device type, click **Import New Device Definition File**, browse to the device definition file, and click **Submit**.
4. Complete the specific fields for the device type. You may have to configure these settings:
   a. In the **Tokencode Duration** field, select the duration for the tokencode display.
   b. In the **Tokencode Length** field, select the number of digits in the tokencode.

- Distribute Multiple Software Tokens Using Dynamic Seed Provisioning (CT-KIP) on page 197
- Distribute One Software Token Using Dynamic Seed Provisioning on page 198
- Distribute Multiple Software Tokens Using Compressed Token Format (CTF) on page 200
- Distribute One Software Token Using Compressed Token Format (CTF) on page 201

Distribute a Hardware Token

Before You Begin

Assign Tokens to Users on page 189

Procedure

Do one of the following:

- If users are located within close proximity, instruct the users to physically collect the tokens.
- If your organization is large and geographically dispersed, distribute tokens by mail.

Distribute Multiple Software Tokens Using File-Based Provisioning

When you distribute software tokens using file-based provisioning, token data is stored in a token distribution file (SDTID file). The SDTID file is added to a ZIP file for download.

Before You Begin

- Instruct users to install the software token application on their devices. For installation instructions, see the *Administrator's Guide* for your software token application.
- Add a Software Token Profile on page 192.
- Assign Tokens to Users on page 189.

Important: When you redistribute tokens using this method, any existing users of these tokens may no longer be able to authenticate. Users must import the new token data before they can authenticate.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Distribute Software Tokens in Bulk > Generate Software Token Files**.
2. In the **Job Name** field, enter a name for the job, or accept the default name. The job is saved with this name so that you can review the details of the job later. Enter a unique name from 1 to 128 characters. The characters & % > < are not allowed.

3. From the **Software Token Profile** drop-down list, select a software token profile with file-based provisioning as the delivery method.
4. In the **DeviceSerialNumber** field, do one of the following:
   - To bind the token to the device class, leave the default setting.
   - To bind the token to a specific device, clear the field and enter the device ID you obtained from the user.
5. Enter a nickname or leave the **Nickname** field blank.
6. You can choose to **Password Protect** the token file. The user must enter the password when adding the token to the SecurID application on the device. Select an option:
   - **Password.** Enter a password of your choice. This password applies to all software tokens in the token distribution file. A password can be up to 24 characters long for 128-bit tokens and 8 characters long for 64-bit tokens.
   - **No password**. The user does not enter a password.
   - **User ID**. The user enters his or her user ID.
   - **Combination User ID followed by Password**. The user enters his or her user ID and the password that you set. The user ID and password combination can be up to 24 characters long for 128-bit tokens and 8 characters long for 64-bit tokens.
7. If you selected **Password**, enter the password in the **Password** and **Confirm Password** fields.
8. Click **Next**.
9. Enter the token selection criteria to find the tokens that you want to distribute. For example, enter the range of serial numbers for the tokens that you want to distribute.
10. Click **Next**.
11. Review the distribution summary and click **Submit Job**.
12. Click the **Completed** tab to view completed jobs.
13. From the context menu, click **Download Output File**.
14. Save the output file to your machine.
15. Safely deliver the token files to users.

Distribute One Software Token Using File-Based Provisioning

When you distribute software tokens using file-based provisioning, token data is stored in a token distribution file (SDTID file). The SDTID file is added to a ZIP file for download.

Before You Begin

- Instruct the user to install the software token application on a device. For installation instructions, see the *Administrator's Guide* for your software token application.
- Add a Software Token Profile on page 192. Only a Super Admin can add software token profiles.
- Assign Tokens to Users on page 189.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Use the search fields to find the software token that you want to distribute.
3. From the search results, click the software token that you want to distribute.
4. From the context menu, click **Distribute**.
5. From the **Select Token Profile** drop-down list, select a software token profile with file-based provisioning as the delivery method.
6. In the **DeviceSerialNumber** field, do one of the following:
   - To bind the token to the device class, leave the default setting.
   - To bind the token to a specific device, clear the field and enter the device ID you obtained from the user.
7. Enter a nickname or leave the **Nickname** field blank.
8. You can choose to **Password Protect** the token file. The following options are available:
   - **Password.** Enter a password of your choice. This password applies to all software tokens in the token distribution file. A password can be up to 24 characters long for 128-bit tokens and 8 characters long for 64-bit tokens.
   - **No password**. The user does not enter a password.
   - **User ID**. The user enters his or her user ID.
   - **Combination User ID followed by Password**. The user enters his or her user ID and the password that you set. The user ID and password combination can be up to 24 characters long for 128-bit tokens and 8 characters long for 64-bit tokens.
9. If you select **Password or Combination**, create a password, and enter it in the **Password** and **Confirm Password** fields.
10. Click **Save and Distribute**.
11. Click **Download Now.**

12. Securely deliver the token file to the user.
13. Click **Done**.

Distribute Multiple Software Tokens Using Dynamic Seed Provisioning (CT-KIP)

Dynamic Seed Provisioning uses the CT-KIP protocol to generate token data without the need for a token file. After you complete the provisioning steps, RSA Authentication Manager displays the URL link of the CT-KIP server and a unique, one-time token activation code. You need these two pieces of information to deliver the token to a device as a URL. Authentication Manager 8.1 now generates custom CT-KIP URLs for certain mobile platform device types. For example, Android, iPhone, Nokia, Browser Toolbar, and Windows Phone.

Before You Begin

- Decide how to safely deliver the URL link of the CT-KIP server and the CT-KIP activation code to users. RSA Authentication Manager does not encrypt e-mail. For secure delivery, you can do the following:
  - Provide the information offline, such as by calling the users on the telephone.
  - Copy the information into e-mail that you encrypt.
  - Use a Simple Mail Transfer Protocol (SMTP) e-mail encryption gateway if the end-user device supports encrypted e-mail.
- Instruct users to install the software token application on their devices. For installation instructions, see the documentation for the software token application.
- Add a Software Token Profile on page 192.
- Assign Tokens to Users on page 189
- RSA recommends that you replace the default certificates in Authentication Manager with trusted certificates. If you do not replace the default certificates, end users are prompted to accept untrusted certificates before proceeding. If you want to use dynamic seed provisioning with CT-KIP, you must have a trusted certificate on the Authentication Manager server or web-tiers.

Important: When you redistribute tokens using this method, any existing users of these tokens may no longer be able to authenticate. Users must import the new token data before they can authenticate.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Distribute Software Tokens in Bulk > Generate Dynamic Seed Provisioning Credentials**.
2. In the **Job Name** field, enter a name for the job, or accept the default name. The job is saved with this name so that you can review the details of the job later. The name must be a unique name from 1 to 128 characters. The characters & % > < are not allowed.
3. From the **Software Token Profile** drop-down list, select a software token profile with dynamic seed provisioning as the delivery method.

4. In the **DeviceSerialNumber** field, do one of the following:
   - To bind the token to the device class, leave the default setting.
   - To bind the token to a specific device, clear the field and enter the device ID you obtained from the user.
5. Enter a nickname or leave the **Nickname** field blank.
6. Click **Next.**
7. Enter the token selection criteria to find the tokens that you want to distribute. For example, enter the range of serial numbers for the tokens that you want to distribute.
8. Click **Next.**
9. Review the distribution summary and click **Submit Job**.
10. Click the **Completed** tab to view completed jobs.
11. Click the job with which you want to work.
12. From the context menu, click **Download Output File**.
13. Save the output file to your machine.
14. Open the output file, copy the activation codes and CT-KIP URL and safely deliver them to the users.

    Note: When you download the output file, some spreadsheet applications will remove the leading zeroes from the activation codes. To import activation codes successfully, open the file in an application that does not remove any characters, such as a text editor, to copy the activation code accurately.

15. Instruct users on how to import tokens.

    If you configured activation codes to expire, advise users to import tokens before the expiration time. If the activation codes are not used before the expiration time, you must redistribute the tokens, and provide the CT-KIP URL and the new activation codes to users.

    For more information, see the software token *Administrator's Guide* for your platform.

Distribute One Software Token Using Dynamic Seed Provisioning

Dynamic seed provisioning uses the CT-KIP protocol to generate token data without the need for a token file. After you complete the provisioning steps, RSA Authentication Manager displays the URL link of the CT-KIP server and the unique, one-time token activation code. You need these two pieces of information to deliver the token to a device as a URL. Authentication Manager 8.1 now generates custom CT-KIP URLs for certain mobile platform device types. For example, Android, iPhone, Nokia, Browser Toolbar, and Windows Phone.

Before You Begin

- Decide how to safely deliver the URL link of the CT-KIP server and the CT-KIP activation code to the user. RSA Authentication Manager does not encrypt e-mail. For a more secure delivery option, you can do the following:
  - Provide the information offline, such as by calling the user on the telephone.
  - Copy the information into an e-mail that you encrypt.
  - Use a Simple Mail Transfer Protocol (SMTP) e-mail encryption gateway.
- Instruct users to install the software token application on their devices. For installation instructions, see the documentation for the software token application.
- Add a Software Token Profile on page 192. Only a Super Admin can add software token profiles.
- Assign Tokens to Users on page 189.
- RSA recommends that you replace the default certificates in Authentication Manager with trusted certificates. If you do not do this, end users are prompted to accept untrusted certificates before proceeding. Certain mobile device platforms only support an SSL certificate with a server that has a trusted certificate installed. If you want to use dynamic seed provisioning with CT-KIP, you must have a trusted certificate on your Authentication Manager server.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Use the search fields to find the software token that you want to distribute.
3. From the search results, click the software token that you want to distribute.
4. From the context menu, click **Distribute**.
5. From the **Select Token Profile** drop-down list, select a software token profile with dynamic seed provisioning as the delivery method.
6. In the **DeviceSerialNumber** field, do one of the following:
   - To bind the token to the device class, leave the default setting.
   - To bind the token to a specific device, clear the field and enter the device ID you obtained from the user.
7. Enter a nickname or leave the **Nickname** field blank.
8. From the **CT-KIP Activation Code** drop-down list, select an activation code for the software token.
9. Click **Save and Distribute**.

10. Copy the activation code and CT-KIP URL and safely deliver them to the user.
11. Instruct the user on how to import the token.

    If you configured the activation code to expire, advise the user to import the token before the expiration time. If the activation code is not used before the expiration time, you must redistribute the token, and provide the CT-KIP URL and the new activation code to the user.

    For more information, see the software token *Administrator's Guide* for your platform

Distribute Multiple Software Tokens Using Compressed Token Format (CTF)

When you distribute software tokens using Compressed Token Format (CTF), you generate a URL, which you deliver to the user. This URL contains the token data needed by the software token application.

Before You Begin

- Instruct users to install the software token application on their devices. For installation instructions, see the documentation for the software token application.
- Add a Software Token Profile on page 192. Only a Super Admin can add software token profiles.
- Assign Tokens to Users on page 189.

Important: If you use this method to redistribute existing tokens, the users of these tokens cannot authenticate until they import the new token data.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Distribute Software Tokens in Bulk > Generate Compressed Token Format Credentials**.
2. In the **Job Name** field, enter a name for the job, or accept the default name. The job is saved with this name so that you can go back and review the details of the job later. The name must be a unique name containing 1 to 128 characters. The characters & % > < are not allowed.
3. From the **Software Token Profile** drop-down list, select a software token profile with CTF as the delivery method.
4. In the **DeviceSerialNumber** field, do one of the following:
   - To bind the token to the device class, leave the default setting.
   - To bind the token to a specific device, clear the field and enter the device ID you obtained from the user.
5. Enter a nickname or leave the **Nickname** field blank.

6. You can choose to **Password Protect** the token file. The following options are available:
   - **Password.** Enter a password of your choice. This password applies to all software tokens in the token distribution file. A password can be up to 24 characters long for 128-bit tokens and 8 characters long for 64-bit tokens.
   - **No password**. The user does not enter a password.
   - **User ID**. The user enters his or her user ID.
   - **Combination User ID followed by Password**. The user enters his or her user ID and the password that you set. The user ID and password combination can be up to 24 characters long for 128-bit tokens and 8 characters long for 64-bit tokens.
7. If you select **Password or Combination**, create a password, and enter it in the **Password** and **Confirm Password** fields.
8. Click **Next.**
9. Enter the token selection criteria to find the tokens that you want to distribute. For example, enter the range of serial numbers for the tokens that you want to distribute.
10. Click **Next.**
11. Review the distribution summary and click **Submit Job**.
12. Click the **Completed** tab to view completed jobs.
13. Click the job with which you want to work.
14. From the context menu, click **Download Output File**.
15. Save the output file to your machine.
16. Open the output file, copy the CTF URLs and safely deliver them to the users.
17. Instruct users on how to import the token. For more information, see the software token *Administrator's Guide* for your platform.

Distribute One Software Token Using Compressed Token Format (CTF)

When you distribute a software token using Compressed Token Format (CTF), you generate a URL, which you deliver to the user. This URL contains the token data needed by the software token application.

Before You Begin

- Instruct users to install the software token application on their devices. For installation instructions, see the documentation for the software token application.
- Add a Software Token Profile on page 192. Only a Super Admin can add software token profiles.
- Assign Tokens to Users on page 189.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Click the **Assigned** tab.
3. Use the search fields to find the software token that you want to distribute.
4. From the search results, click the software token that you want to distribute.
5. From the context menu, click **Distribute**.
6. From the **Select Token Profile** drop-down list, select a software token profile with Compressed Token Format (CTF) as the delivery method.
7. In the **DeviceSerialNumber** field, do one of the following:
   - To bind the token to the device class, leave the default setting.
   - To bind the token to a specific device, clear the field and enter the device ID you obtained from the user.
8. Enter a nickname or leave the **Nickname** field blank.
9. You can choose to **Password Protect** the token file. The following options are available:
   - **Password.** Enter a password of your choice. This password applies to all software tokens in the token distribution file. A password can be up to 24 characters long for 128-bit tokens and 8 characters long for 64-bit tokens.
   - **No password**. The user does not enter a password.
   - **User ID**. The user enters his or her user ID.
   - **Combination User ID followed by Password**. The user enters his or her user ID and the password that you set. The user ID and password combination can be up to 24 characters long for 128-bit tokens and 8 characters long for 64-bit tokens.
10. If you select **Password or Combination**, create a password, and enter it in the **Password** and **Confirm Password** fields.
11. Click **Save and Distribute**.
12. Copy the CTF URL and safely deliver it to the user.

    Note: If you navigate away from this page before you copy the CTF URL, you must perform the distribution process from the beginning to generate a new URL.

13. Instruct the user on how to import the token. For more information, see the software token *Administrator's Guide* for your platform.

Administering RSA SecurID Tokens

Administering a token includes token management and providing temporary emergency access for users. You can perform the following tasks with tokens:

- Enable a Token on page 203
- Disable a Token on page 204
- Delete a Token on page 204
- Edit a Token on page 204
- Assign a Replacement Token on page 205

Enabled and Disabled Tokens

An enabled token can be used for authentication. A disabled token cannot be used.

After Authentication Manager is installed, tokens must be imported into the deployment. All imported tokens are automatically disabled. This security feature protects the deployment if the tokens are lost or stolen.

Tokens are automatically enabled when they are assigned by an administrator.

Tokencode-only software tokens that are provisioned through Self-Service are disabled by default. Users can enable these tokens through the Self-Service Console.

A disabled token does not lock a user's account. Lockout applies to a user's account, not to a user's token. Disabling a token does not remove the user's account from the deployment. You can view disabled tokens using the Security Console.

You can enable and disable tokens only in security domains that are included in your administrative scope.

You should disable an assigned token in the following situations:

- Before a hardware token is mailed to a user. Re-enable the token after you know that it has been successfully delivered to the user to whom it has been assigned and the user is ready to use it.
- If you know that a user does not need to authenticate for an extended period of time. For example, you may want to disable a token before a user takes a short-term leave or an extended vacation. After you disable the token, that user cannot authenticate with that token until it is re-enabled.

Enable a Token

Before a user can use an assigned token to authenticate, you must enable the token.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Click the **Assigned or Unassigned** tab to view the list of tokens that you want to enable.
3. Select the checkbox next to the tokens that you want to enable.

4. From the Action menu, click **Enable**.
5. Click **Go**.

Disable a Token

When you disable a token, the assigned user can no longer use the token to authenticate.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Click the **Assigned or Unassigned** tab to view the list of tokens that you want to disable.
3. Select the checkbox next to the tokens that you want to disable.
4. From the Action menu, click **Disable**.
5. Click **Go**.

Delete a Token

When you delete a token, the token is removed from the internal database and can no longer be assigned. If it is already assigned to a user, the user cannot use the token to authenticate.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage**
2. **Existing**.
3. Click the **Assigned or Unassigned** tab, depending on whether the tokens you want to delete are assigned to a user or are unassigned.
4. Use the search fields to find the token that you want to delete.
5. From the Search results, select the checkbox next to the token or tokens that you want to delete.
6. From the Action menu, click **Delete**.
7. Click **Go**.
8. Click **OK** in the delete dialog box to confirm deletion.

Edit a Token

Edit a token to update information about the token, such as the security domain to which the token is assigned. You can also:

- Enable or disable the token.
- Clear the SecurID PIN.

- Require the user to change the SecurID PIN the next time he or she authenticates with the token. The user can view, clear, and change the PIN options only when the token is assigned.
- Specify whether the user must enter a SecurID PIN.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Click the **Assigned** or **Unassigned** tabs, depending on whether the token you want to edit is assigned to a user.
3. Use the search fields to find the token that you want to edit.
4. From the search results, click the token that you want to edit.
5. From the context menu, click **Edit**.
6. Make any necessary changes to the token record.
7. Click **Save**.

   If you did not save your edits, you can click **Reset** to reset the token record to be as it was before you began editing.

User Assistance for Lost, Stolen, Damaged, or Expired Tokens

Users can request replacement tokens and temporary emergency access tokencodes using the Self-Service Console.

You can provide temporary emergency access tokencodes for users whose tokens are damaged, lost, temporarily misplaced, stolen, or expired. Authentication Manager generates emergency access tokencodes and provides two-factor authentication so that users can authenticate until their token issue is resolved. For more information, see Exporting and Importing Users and Tokens Between Deployments.

Note: You cannot assign emergency access tokencodes to an expired token. Replace the token before providing temporary access.

You must replace permanently lost, stolen, damaged, or expired tokens. For more information, see Assign a Replacement Token.

Assign a Replacement Token

You can assign a replacement token to a user if a user's token has been permanently lost or destroyed, or if the current token has expired.

Procedure

1. In the Security Console, click **Authentication > SecurID Tokens > Manage Existing**.
2. Use the search fields to find the token that you want to replace.
3. From the search results, click the token that you want to replace.
4. Click **Replace with Next Available Token**.

Resynchronize a Token

A token must be resynchronized when the tokencode displayed on the token does not match the tokencode generated by Authentication Manager. When the tokencodes do not match, authentication attempts fail. Depending on your configuration, users can resynchronize tokens with the Self-Service Console. You can also use the Security Console to resynchronize tokens.

Before You Begin

You need access to the tokencodes. The user can read tokencodes to you over the phone.

Procedure

1. In the Security Console, click **Identity > Users > Manage Existing**.
2. Use the search fields to find the user whose token needs to be resynchronized.
3. Click the **Assigned** or **Unassigned** tab.
4. From the search results, click the user whose token needs to be resynchronized.
5. From the context menu, click **SecurID Tokens.**
6. Click the token that you want to resynchronize.
7. Click **Resynchronize Token**.
8. Enter the tokencode that is displayed on the user's token.
9. Wait for the tokencode to change, and then enter the new tokencode.
10. Click **Resynchronize**.

~~Exporting and Importing Users and Tokens Between Deployments~~

~~You can export and import token records and user records between two Authentication Manager 8.1 deployments. You might do this when merging two deployments or to move tokens from one deployment to another. You can export and import tokens, or you can export and import users with tokens. You export from the source deployment and import to the target deployment.You can move tokens that are assigned or unassigned, and reassign the tokens to different users on the target deployment. Authentication Manager encrypts the export file to ensure that the process is secure.~~

~~Impact of Export and Import on Authentication~~

~~Moving users to a new deployment does not interrupt authentication because the exported data from the source deployment is not deleted. Users can still authenticate to the source deployment even after you have imported their records to the target deployment. You must manually delete the users from the source deployment.~~

Glossary

Active Directory

The directory service that is included with Microsoft Windows Server 2003 SP2, Microsoft Windows Server 2008, and Microsoft Windows Server 2008 R2.

Active Directory forest

A federation of identity servers for Windows Server environments. All identity servers share a common schema, configuration, and Global Catalog.

administrative role

A collection of permissions and the scope within which those permissions apply.

administrator

Any user with one or more administrative roles that grant administrative permission to manage the system.

agent host

The machine on which an agent is installed.

appliance

The hardware or guest virtual machine running RSA Authentication Manager. The appliance can be set up as a primary instance or a replica instance.

approver

A Request Approver or an administrator with approver permissions.

assurance level

For risk-based authentication, the system categorizes each authentication attempt into an assurance level that is based on the user's profile, device, and authentication history. If the authentication attempt meets the minimum assurance level that is required by the RBA policy, the user gains access to the RBA-protected resource. Otherwise, the user must provide identity confirmation to access the RBA-protected resource.

attribute

A characteristic that defines the state, appearance, value, or setting of something. In Authentication Manager, attributes are values associated with users and user groups. For example, each user group has three standard attributes called Name, Identity Source, and Security Domain.

attribute mapping

The process of relating a user or user group attribute, such as User ID or Last Name, to one or more identity sources linked to the system. No attribute mapping is required in a deployment where the internal database is the primary identity source.

audit information

Data found in the audit log representing a history of system events or activity including changes to policy or configuration, authentications, authorizations, and so on.

audit log

A system-generated file that is a record of system events or activity. The system includes four such files, called the Trace, Administrative, Runtime Audit, and System logs.

authentication

The process of reliably determining the identity of a user or process.

authentication agent

A software application installed on a device, such as a domain server, web server, or desktop computer, that enables authentication communication with Authentication Manager on the network server. See agent host.

authentication method

The type of procedure required for obtaining authentication, such as a one-step procedure, a multiple-option procedure (user name and password), or a chained procedure.

authentication protocol

The convention used to transfer the credentials of a user during authentication, for example, HTTP-BASIC/DIGEST, NTLM, Kerberos, and SPNEGO.

authentication server

A component made up of services that handle authentication requests, database operations, and connections to the Security Console.

authenticator

A device used to verify a user's identity to Authentication Manager. This can be a hardware token (for example, a key fob) or a software token.

authorization

The process of determining if a user is allowed to perform an operation on a resource.

backup

A file that contains a copy of your primary instance data. You can use the backup file to restore the primary instance in a disaster recovery situation. An RSA Authentication Manager backup file includes: the internal database, appliance-only data and configuration, keys and passwords used to access internal services, and internal database log files. It does not include all the appliance and operating system log files.

certificate

An asymmetric public key that corresponds with a private key. It is either self-signed or signed with the private key of another certificate.

certificate DN

The distinguished name of the certificate issued to the user for authentication.

command line utility (CLU)

A utility that provides a command line user interface.

core attributes

The fixed set of attributes commonly used by all RSA products to create a user. These attributes are always part of the primary user record, whether the deployment is in an LDAP or RDBMS environment. You cannot exclude core attributes from a view, but they are available for delegation.

Cryptographic Token-Key Initialization Protocol (CT-KIP)

A client-server protocol for the secure initialization and configuration of software tokens. The protocol requires neither private-key capabilities in the tokens, nor an established public-key infrastructure. Successful execution of the protocol results in the generation of the same shared secret on both the server as well as the token.

custom attributes

An attribute you create in Authentication Manager and map to a field in an LDAP directory. For example, you could create a custom attribute for a user's department.

data store

A data source, such as a relational database (Oracle or DB2) or directory server (Microsoft Active Directory or Oracle Directory Server). Each type of data source manages and accesses data differently.

delegated administration

A scheme for defining the scope and responsibilities of a set of administrators. It permits administrators to delegate a portion of their responsibilities to another administrator.

delivery address

The e-mail address or the mobile phone number where the on-demand tokencodes will be delivered.

deployment

An installation of Authentication Manager that consists of a primary instance and, optionally, one or more replica instances.

demilitarized zone

The area of a network configured between two network firewalls.

device history

For risk-based authentication, the system maintains a device history for each user. It includes the devices that were used to gain access to protected resources.

device registration

For risk-based authentication, the process of saving an authentication device to the user's device history.

distribution file password

A password used to protect the distribution file when the distribution file is sent by e-mail to the user.

distributor

A Token Distributor or an administrator with distributor permissions.

DMZ

See demilitarized zone.

dynamic seed provisioning

The automation of all the steps required to provide a token file to a device that hosts a software token, such as a web browser, using the Cryptographic Token-Key Initialization Protocol (CT-KIP).

e-mail notifications

Contain status information about requests for user enrollment, tokens, and user group membership that is sent to users who initiated the request. For token requests, e-mail notifications also contain information about how to download and activate tokens. Request Approvers and Token Distributors receive e-mail notifications about requests that require their action. See e-mail templates.

e-mail templates

Templates that administrators can use to customize e-mail notifications about user requests for user enrollment, tokens, user group membership, or the on-demand tokencode service. See e-mail notifications.

excluded words dictionary

A dictionary containing a record of words that users cannot use as passwords. It prevents users from using common, easily guessed words as passwords.

fixed passcode

Similar to a password that users can enter to gain access in place of a PIN and tokencode. The format for fixed passcodes is defined in the token policy assigned to a security domain. An administrator creates a fixed passcode in a users authentication settings page. Fixed passcodes can be alphanumeric and contain special characters, depending on the token policy.

Global Catalog

A read-only, replicated repository of a subset of the attributes of all entries in an Active Directory forest.

Global Catalog identity source

An identity source that is associated with an Active Directory Global Catalog. This identity source is used for finding and authenticating users, and resolving group membership within the forest.

identity attribute

Customer-defined attributes that are mapped to an existing customer-defined schema element. They are always stored in the same physical repository as the user's or user group's core attribute data. You can search, query, and report on these attributes. Each identity attribute definition must map to an existing attribute in an LDAP directory or RDBMS.

identity confirmation method

For risk-based authentication, an authentication method that can be used to confirm a user's identity.

identity source

A data store containing user and user group data. The data store can be the internal database or an external directory server, such as Microsoft Active Directory.

instance

An installation of RSA Authentication Manager that can be set up as a primary instance or a replica instance. An instance also includes a RADIUS server.

internal database

The Authentication Manager proprietary data source.

keystore

The facility for storing keys and certificates.

load balancer

A deployment component used to distribute authentication requests across multiple computers to achieve optimal resource utilization. The load balancer is usually dedicated hardware or software that can provide redundancy, increase reliability, and minimize response time. See Round Robin DNS.

lower-level security domain

In a security domain hierarchy, a security domain that is nested within another security domain.

minimum assurance level

See assurance level.

node secret

A long-lived symmetric key that the agent uses to encrypt the data in the authentication request. The node secret is known only to Authentication Manager and the agent.

on-demand tokencode

Tokencodes delivered by SMS or SMTP. These tokencodes require the user to enter a PIN to achieve two-factor authentication. On-demand tokencodes are user-initiated, as Authentication Manager only sends a tokencode to the user when it receives a user request. An on-demand tokencode can be used only once. The administrator configures the lifetime of an on-demand tokencode. See on-demand tokencode service.

on-demand tokencode service

A service that allows enabled users to receive tokencodes by text message or e-mail, instead of by tokens. You configure the on-demand tokencode service and enable users on the Security Console.

Operations Console

An administrative user interface through which the user configures and sets up Authentication Manager, for example, adding and managing identity sources, adding and managing instances, and disaster recovery.

permissions

Specifies which tasks an administrator is allowed to perform.

preferred instance

The Authentication Manager instance that the risk-based authentication service in the web tier communicates with first. Also, the instance that provides updates to the web tier. Any instance can be the preferred instance. For example, you can configure a replica instance as the preferred instance.

primary instance

The installed deployment where authentication and all administrative actions are performed.

promotion, for disaster recovery

The process of configuring a replica instance to become the new primary instance. During promotion, the original primary instance is detached from the deployment. All configuration data referring to the original primary instance is removed from the new primary instance.

promotion, for maintenance

The process of configuring a replica instance to become the new primary instance when all instances are healthy. During promotion, a replica instance is configured as a primary instance. The original primary instance is demoted and configured as a replica instance.

provisioning

See token provisioning.

provisioning data

The provisioning server-defined data. This is a container of information necessary to complete the provisioning of a token device.

RADIUS

See Remote Authentication Dial-In User Service.

RBA

See risk-based authentication.

RBA integration script

A script that redirects the user from the default logon page of a web-based application to a customized logon page. This allows Authentication Manager to authenticate the user with risk-based authentication. To generate an integration script, you must have an integration script template.

realm

A realm is an organizational unit that includes all of the objects managed within a single deployment, such as users and user groups, tokens, password policies, and agents. Each deployment has only one realm.

Remote Authentication Dial-In User Service (RADIUS)

A protocol for administering and securing remote access to a network. A RADIUS server receives remote user access requests from RADIUS clients, for example, a VPN.

replica instance

The installed deployment where authentication occurs and at which an administrator can view the administrative data. No administrative actions are performed on the replica instance.

replica package

A file that contains configuration data that enables the replica appliance to connect to the primary appliance. You must generate a replica package before you set up a replica appliance.

requests

Allows users to enroll, as well as request tokens, the on-demand tokencode service, and user group membership.

Request Approver

A predefined administrative role that grants permission to approve requests from users for user enrollment, tokens, or user group membership.

risk-based authentication (RBA)

An authentication method that analyzes the user's profile, authentication history, and authentication device before granting access to a protected resource.

risk engine

In Authentication Manager, the risk engine intelligently assesses the authentication risk for each user. It accumulates knowledge about each user's device and behavior over time. When the user attempts to authenticate, the risk engine refers to its collected data to evaluate the risk. The risk engine then assigns an assurance level, such as high, medium, or low, to the user's authentication attempt.

round robin DNS

An alternate method of load balancing that does not require dedicated software or hardware. When the Domain Name System (DNS) server is configured and enabled for round robin, the DNS server sends risk-based authentication (RBA) requests to the web-tier servers. See Load Balancer.

scope

In a deployment, the security domain or domains within which a role's permissions apply.

Secure Sockets Layer (SSL)

A protocol that uses cryptography to enable secure communication over the Internet. SSL is widely supported by leading web browsers and web servers.

Security Console

An administrative user interface through which the user performs most of the day-to-day administrative activities.

security domain

A container that defines an area of administrative management responsibility, typically in terms of business units, departments, partners, and so on. Security domains establish ownership and namespaces for objects (users, roles, permissions, and so on) within the system. They are hierarchical.

security questions

A way of allowing users to authenticate without using their standard method. To use this service, a user must answer a number of security questions. To authenticate using this service, the user must correctly answer all or a subset of the original questions.

self-service

A component of Authentication Manager that allows the user to update user profiles, change passwords for the Self-Service Console, configure life questions, clear devices enabled for risk-based authentication, change e-mail addresses or phone numbers for on-demand authentication, and manage on-demand authentication PINs. The user can also request, maintain, and troubleshoot tokens.

Self-Service Console

A user interface through which the user can update user profiles, change passwords for the Self-Service Console, configure life questions, clear devices enabled for risk-based authentication, change e-mail addresses or phone numbers for on-demand authentication, and manage on-demand authentication PINs. Users can also request, maintain, and troubleshoot tokens on the Self-Service Console.

session

An encounter between a user and a software application that contains data pertaining to the user's interaction with the application. A session begins when the user logs on to the software application and ends when the user logs off of the software application.

shipping address

An address used by distributors to distribute hardware tokens.

silent collection

For risk-based authentication, a period during which the system silently collects data about each user's profile, authentication history, and authentication devices without requiring identity confirmation during logon.

SSL

See Secure Sockets Layer.

Super Admin

An administrator with permissions to perform all administrative tasks in the Security Console. A Super Admin:

- Can link identity sources to system
- Has full permissions within a deployment
- Can assign administrative roles within a deployment

system event

System-generated information related to nonfunctional system events, such as server startup and shutdown, failover events, and replication events.

System log

A persistable store for recording system events.

time-out

The amount of time (in seconds) that the user's desktop can be inactive before reauthentication is required.

token distributor

A predefined administrative role that grants permission to act upon requests from users for tokens. Distributors record how they plan to deliver tokens to users and close requests.

token provisioning

The automation of all the steps required to provide enrollment, user group membership, RSA SecurID tokens, and the on-demand tokencode service to users. See also self-service.

top-level security domain

The top-level security domain is the first security domain in the security domain hierarchy. The top-level security domain is unique in that it links to the identity source or sources and manages the password, locking, and authentication policy for the entire deployment.

Trace log

A persistable store for trace information.

trusted realm

A trusted realm is a realm that has a trust relationship with another realm. Users on a trusted realm have permission to authenticate to another realm and access the resources on that realm. Two or more realms can have a trust relationship. A trust relationship can be either one-way or two-way.

trust package

An XML file that contains configuration information about the deployment.

UDP

See User Datagram Protocol.

User Datagram Protocol (UDP)

A protocol that allows programs on networked computers to communicate with one another by sending short messages called datagrams.

User ID

A character string that the system uses to identify a user attempting to authenticate. Typically a User ID is the user's first initial followed by the last name. For example, Jane Doe's User ID might be *jdoe*.

virtual host

Physical computer on which a virtual machine is installed. A virtual host helps manage traffic between web-based applications, web-tier deployments, and the associated primary instance and replica instances.

virtual hostname

The publicly-accessible hostname. End users use this virtual hostname to authenticate through the web tier. The system also generates SSL information based on the virtual hostname. The virtual hostname must be same as the load balancer hostname.

web tier

A web tier is a platform for installing and deploying the Self-Service Console, Dynamic Seed Provisioning, and the risk-based authentication (RBA) service in the DMZ. The web tier prevents end users from accessing your private network by receiving and managing inbound internet traffic before it enters your private network.

workflow

The movement of information or tasks through a work or business process. A workflow can consist of one or two approval steps and a distribution step for different requests from users.

workflow participant

Either approvers or distributors. Approvers review, approve, or defer user requests. Distributors determine the distribution method for token requests and record the method for each request. See also workflow.