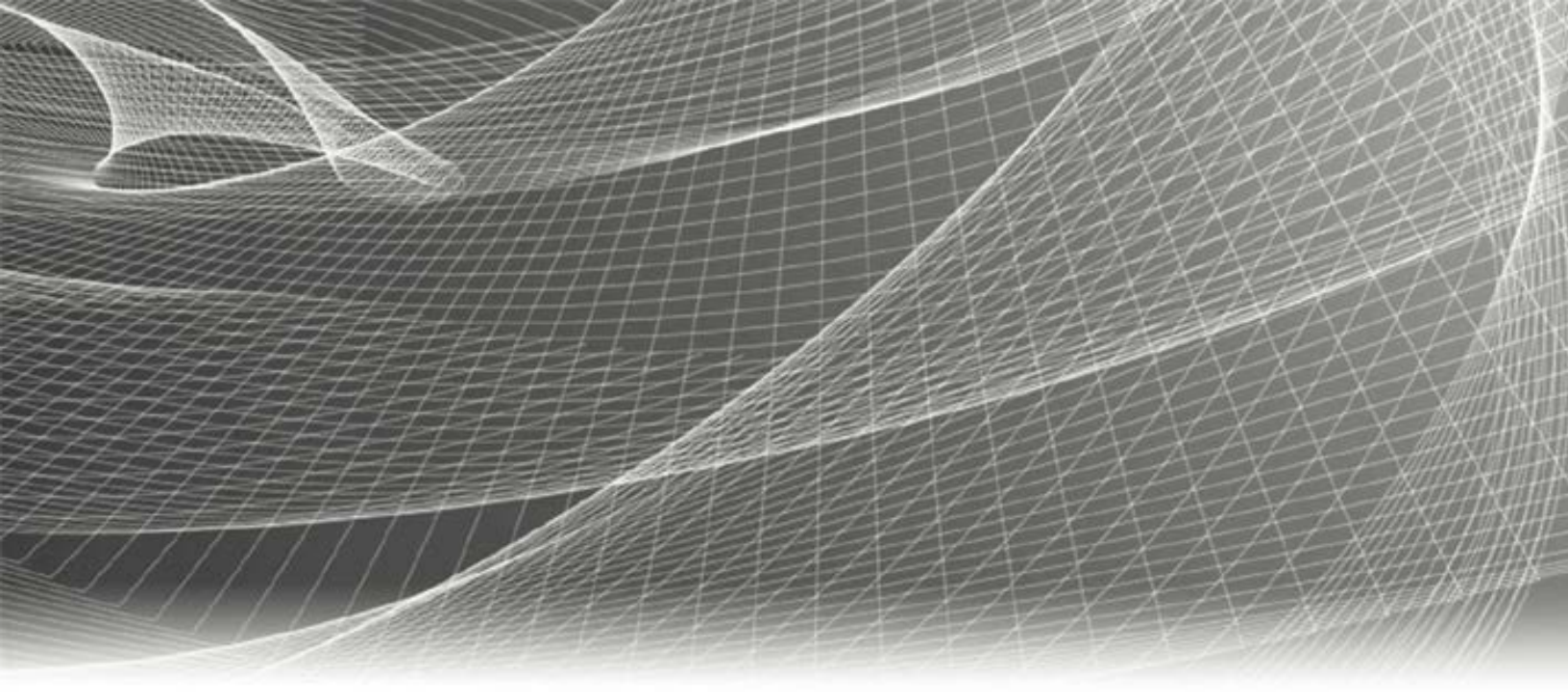

RSA | Identity Governance and Lifecycle

Lieberman Software™ Rapid Enterprise Defense™ Identity Management Application Guide

Contact Information

RSA Link at https://community.rsa.com contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

License Agreement

Third-Party Licenses

Note on Encryption Technologies

Distribution

Contents

February 2017

Revision History

Revision Number	Description
Version 1	Enterprise Random Password Manager Collection & Provisioning
Version 2	Updated product name

Preface

This guide describes how to set up the Lieberman Software™ Rapid Enterprise Defense™ (RED) Identity Management collector and connector for data collection, provisioning and de-provisioning of RED entities. The collector and connector use REST Web Services to communicate with Lieberman RED. The guide outlines the required configurations, parameters and mappings of different attributes between the collector, connector and Lieberman RED. The guide also includes use cases and troubleshooting tips.

Audience

This guide is intended for the users of RSA Identity Governance and Lifecycle, including security administrators. Lieberman RED can be integrated with RSA Identity Governance and Lifecycle using the Lieberman RED Collector and Connector. Any end-point administrator having access to Lieberman RED end-point can refer to this guide.

RSA recommends that users of this guide have basic REST Web Services knowledge.

Supported RSA Identity Governance and Lifecycle Version(s)

- RSA Identity Governance and Lifecycle 7.0.2 and later

Supported Lieberman Software™ Rapid Enterprise Defense Identity Management Version

- Lieberman RED 5.5.2

About Data Collection and Provisioning with Lieberman RED

Lieberman Software™ Rapid Enterprise Defense™ (RED) Identity Management is a Proactive Cyber Defense Platform that protects organizations against malicious insiders, advanced persistent threats (APTs) and other sophisticated cyber-attacks on premise, in the cloud and in hybrid environments. Lieberman RED simplifies the management of your privileged credentials, delivering automated protection at scale, with a rapidly deployed and affordable solution.

Integrating Lieberman Software Rapid Enterprise Defense Identity Management with RSA Identity Governance and Lifecycle helps you improve access decision, reduce the risk of inappropriate access, and better analyze security incidents by providing access to identity context and application entitlement data.

RSA Identity Governance and Lifecycle's collector for Lieberman Software Rapid Enterprise Defense Identity Management provides a rich data context about delegation identity (such as their permissions of different resources such as management set, account, system etc.) from Lieberman RED.

The Lieberman Software Rapid Enterprise Defense Identity Management Connector helps you govern and provision delegation identity access to Lieberman RED. You can use the business governance processes within RSA Identity Governance and Lifecycle to request, provision, and de-provision user access to workspaces within Lieberman Software Rapid Enterprise Defense Identity Management.

Account Data Collection is a process of gathering accounts from Lieberman RED. The gathered data is further processed to perform User-Account resolution (mapping RSA Identity Governance and Lifecycle users to accounts collected). Account Data Collection is done to associate the users of RSA Identity Governance and Lifecycle to the collected Accounts.

Account data collection for Lieberman RED is domain specific, requiring integration of Active Directory/LDAP with RSA Identity Governance and Lifecycle.

Entitlement Data Collection is a process of collecting all the resources and their actions from Lieberman RED. Each Resource-Action pair forms an entitlement in Lieberman RED. Lieberman RED users having assigned permissions are also collected using the Entitlement Data Collector. The collected data is further processed to perform the User-Entitlement resolution in RSA Identity Governance and Lifecycle.

Lieberman RED has several types of accounts. The Account Data Collector (ADC) only collects the following account types: Explicit, Windows Domain User, Windows Domain Group and LDAP User. These accounts are the only accounts that are collected and managed by the Entitlement Data Collector (EDC).

Collectors and Connector for Lieberman RED

RSA Identity Governance and Lifecycle support the following collectors and connector for Lieberman RED.

Name	Data Source Type/Connector Template	Description
Lieberman RED Account Data Collector	Lieberman Enterprise Random Password Manager	Lieberman RED Collector collects the Identities as accounts from Lieberman RED. This collector uses REST APIs internally.
Lieberman RED Entitlement Data Collector	Lieberman Enterprise Random Password Manager	Lieberman RED Entitlement Collector collects permissions as entitlements from Lieberman RED. This collector uses REST APIs internally.
Lieberman RED Connector	Lieberman Enterprise Random Password Manager	Lieberman RED Connector can provision data on Lieberman RED. This Connector uses REST APIs internally.

Prerequisites

1. **Install the Lieberman RED Web Service**
   Lieberman RED web services need to be installed.

2. **Determine the URL for API access**
   Non-SSL Web Service -
   The Lieberman RED web service installation wizard installs a REST-based web service. The REST service base URI has following formats for Non-SSL and SSL web services respectively:
   http://<REDServerName>/ERPMWebService/json/V2/AuthService.svc

   SSL Web Service -
   Follow the documentation on enabling SSL on Lieberman RED. Once SSL is enabled, access the RED web service with SSL using the following URL format.
   https://<REDServerName>:<SSL_Port>/ERPMWebService/json/V2/AuthService.svc

   A. Setting Hostname Verification property:

   -DverifyHostnameForSSL property needs to be set in ACM and AFX server in order to enable/disable Hostname verification.

   Steps for setting hostname verification property in ACM - WildFly Server:
   Edit the standalone.conf from /home/oracle/wildfly-8.2.0.Final/bin to add a property as follows:
   JAVA_OPTS="$JAVA_OPTS -DverifyHostnameForSSL=false"
   Save the file and restart the Server.

   Run: afx stop
   Run: acm stop
   Run: acm start
   Run: afx start

   Steps for setting hostname verification property in ACM – WebLogic Server:
   1. Log in to WebLogic Administrative console. (http://<HOST_NAME>.aveksa.local:7001/console/login/LoginForm.jsp)
   2. Under Domain Configurations, in the Environment section, click Servers link.
   3. Click aveksaServer link.
   4. Click the SSL tab.
   5. Click advanced link.
   6. Select HostName Verification = None.

Save the settings and restart the Server:
Login to the VM using Putty as: oracle and password: secret and follow the steps given below:

1. cd /home/oracle/Oracle/Middleware/user_projects/domains/aveksaDomain/bin
2. ./stopWebLogic.sh
3. nohup ./startWeblogic.sh &

Steps for setting hostname verification property in ACM - WebSphere Server:

1. In the WebSphere Administration Console select Servers.
2. Expand Server Type and select WebSphere application servers.
3. Click on the name of your server.
4. Expand Java and Process Management and select Process Definition.
5. Under the Additional Properties section, click Java Virtual Machine.
6. Scroll down and locate the textbox for Generic JVM arguments.
7. Add property "–DverifyHostnameForSSL=false" to JVM and save the configuration.

Save the settings and restart the Server.
Login to the VM using Putty as: root and password: Av3k5a and follow the steps given below:

1. cd /opt/IBM/WebSphere/AppServer/bin
2. stopServer.sh server1
3. startServer.sh server1

Steps for setting hostname verification property in AFX:
Edit the wrapper.conf file from /home/oracle/AFX/esb/conf/ to add a property as follows:
wrapper.java.additional.<n>=-DverifyHostnameForSSL=false

If –DverifyHostnameForSSL=false, hostname verification will be disabled
If –DverifyHostnameForSSL=true, hostname verification will be enabled

B. Installing required certificates:

Lieberman Software Rapid Enterprise Defense Identity Management certificate should be added to the appropriate trust-stores. Follow the steps mentioned below for adding certificates to the trust-stores of WebSphere, WebLogic and WildFly application servers.

a) WildFly

1. Download/retrieve the Lieberman Enterprise Random Password Manager SSL certificate in PEM format e.g. liebermanCert.pem and save at some location.
2. cd <$JAVA_HOME>/jre/lib/security
3. Add certificates in cacerts by using keytool:
   keytool -import -file <Path to certificate> -alias <name of alias> -keystore cacerts

4. Password for keystore (unless you have made any changes): changeit
5. Restart the server:
   Run: afx stop
   Run: acm stop
   Run: acm start
   Run: afx start

b) WebLogic

[For ACM]

1. Login into WebLogic machine using SSH (e.g. putty).
2. cd /home/oracle/
   keytool -import -file <Path to certificate> -alias <alias name> -keystore /home/oracle/server.keystore -storepass Av3k5a15num83r0n3

[For AFX]

1. Import the certificate to default java_homeCommand. Example: keytool -import -keystore $JAVA_HOME/jre/lib/security/cacerts -storepass changeit -file <Path to certificate> -alias <alias name>.
2. Restart Server:
   Login to the VM using Putty as: oracle and password: secret and follow the steps given below:
   1. cd /home/oracle/Oracle/Middleware/user_projects/domains/aveksaDomain/bin
   2. ./stopWebLogic.sh
   3. nohup ./startWeblogic.sh &

c) WebSphere

Changes at WebSphere Administrative console:
[For ACM]

1. Log in to WebSphere Administrative console (http://<HOST_NAME>:9060/ibm/console/login.do)
2. In left panel, expand the Security menu.
3. Click on SSL certificate and then click the key management link.
4. Under Configuration Settings, click the Manage endpoint security configurations link.
5. Select outbound properties for the appropriate node.
6. Click on appropriate node link to get the properties.
7. Under Related Items, click Key stores and certificates and then click the 'NodeDefaultTrustStore' key store.

8. Under Additional Properties, click Signer certificates.
9. Click Retrieve from Port.
10. In the Host field, enter <host_name> enter 8443 in the Port field, and erpm_cert in the Alias field.
11. Click Retrieve Signer Information.
12. Verify that the certificate information is for a certificate that you trust.
13. Click Apply and then click Save.
14. Add certificate to WebSphere java home.
    Path: /opt/IBM/WebSphere/AppServer/java_1.7.1_64/jre/lib/security
    Keytool command Example: keytool -import -keystore /opt/IBM/WebSphere/AppServer/java_1.7.1_64/jre/lib/security/cacerts -storepass changeit -file <Path to certificate> -alias <alias name>

[For AFX]:

1. Import the certificate to default java_home
2. Command: keytool -import -keystore $JAVA_HOME/jre/lib/security/cacerts -storepass changeit -file <Path to certificate> -alias <alias name>

Save the settings and restart the Server:
Login to the VM using Putty as: root and password: Av3k5a and follow the steps given below:

1. cd /opt/IBM/WebSphere/AppServer/bin
2. stopServer.sh server1
3. startServer.sh server1

3. Create a Lieberman RED User (Delegation Identity) for API access

Complete the following steps to create a Lieberman RED explicit account (also called native static account) to authenticate to the web service.

a) In the Lieberman RED management console, choose **Delegation > Delegation Identities.**
   The Enroll Identities dialog opens.
b) Click **Add.**
   The Add Delegation Role dialog opens.
c) Choose **Explicit Identity**, enter a **Username** and **Password**, and click **OK**.
   The identity is added to the list of identities in the Enroll Identities dialog.

4. Set Permissions in Lieberman RED to enable the account to use the Web Service

The Delegation Identity created in the step above needs to access the web service and perform operations in Lieberman RED.

To assign permissions to the Delegation Identity:

a) In Lieberman RED management console, choose **Delegation > Web Application Global Delegation Permissions.**

The Web Application Global Delegation Permissions dialog opens.

b) In the **Enrolled Identities** section, select the explicit account just created in the above step.

c) In the **Global Identity Rules** section, select **Logon** and **Grant All Access**.

Logon is the minimum permission required to obtain an authentication token, which is necessary for subsequent API calls. For the purpose of using the API, the **Grant All Access** setting needs to be enabled.

In production, the account type that best meets the security and operations requirements must be used and only the minimum required permissions must be configured.

Note: Any time you change permissions for the identity that logs in to the web service, follow up by getting a new authentication token using the DoLogin2 call. The new authentication token will use the latest permissions.

Using the Lieberman RED Application Wizard to Configure Connector and Collectors

RSA Identity Governance and Lifecycle provides an Application Wizard that simplifies the process of setting up Lieberman Enterprise Random Password Manager Connector and Collectors. RSA recommends that you use the Application Wizard to initially set up Lieberman Enterprise Random Password Manager Connectors and Collectors. To modify the collectors/connector in future, please refer to the collectors/connector sections respectively.

1) Log in to RSA Identity Governance and Lifecycle.
2) Go to Resources > Applications and click Create Application.
3) From the list of applications, select Lieberman Software Rapid Enterprise Defense Identity Management.
4) Click Next. The Setup page provides an overview of the Lieberman RED endpoint, as well as collector and connector information.
5) Click Next.
6) Fill out the Connect page with connection information relevant to Lieberman RED.

Parameter Name	Description
Application Name	Any name to identify this application
Scheme	HTTP or HTTPS
Host	Host name of the Lieberman RED endpoint server
Port	Port number of the Lieberman RED endpoint server
Base URL	Base URL to connect to Lieberman RED server E.g. ERPMWebService/JSON/V2/AuthService.svc
Collect Explicit Delegation Identities	To be checked if explicit delegation identities are to be collected from Lieberman RED.
Domain	Provide the Domain Name of Active Directory/LDAP for collection. Accounts collected are filtered based on the domain name provided.
Username	Username used to login to Lieberman RED server
Password	Password for user of Lieberman RED
Authenticator	The authentication server configured in Lieberman RED, for example, an Explicit/Domain Account. **Note:** Use type 'Explicit' when using a delegation identity not associated to any domain

	User type 'Domain Account' when using a delegation identity associated to some domain.
Domain Name *appears when the Authenticator field value selected is Domain account.	If Authenticator is Domain account, then provide the domain name for the username used to log in to the Lieberman RED server.
LoginType	The type of account being used for logging in to Lieberman RED. One of the following values: Unknown NativeStaticAccount: for a local account (explicit) without a domain/authenticator. FullyQualifiedAccount: for a domain/LDAP account. IntegratedAuthentication: for Windows integrated authentication. CertificateAuthentication: for certificate-based authentication. **Note:** Use NativeStaticAccount for a local account without a domain /authenticator. Use FullyQualifiedAccount for a domain/LDAP account. Use IntegratedAuthentication for Windows integrated authentication. Use CertificateAuthentication for certificate-based authentication.
AFX Server	Select Available AFX server from the drop down list.
Proxy Host	Hostname of the proxy server
Proxy Port	Port of the proxy server
Proxy User Name	Username for the proxy server
Proxy Password	Password for the proxy server

Click Test Connection to check the connectivity to the endpoint from RSA Identity Governance and Lifecycle instance.

7) Click Next.
8) On the Confirm Changes page, confirm all the provided details. If there are any corrections required, click Back to return to previous page.
9) Click Next.
10) The Change Summary page lists all the components created by this Application Wizard.

Following Components will be created using the Lieberman RED Application Wizard:

- Application – Lieberman RED Applications with all components
- Custom Attributes for Account – AccountName, DisplayName, EmailAddress, IsDomainAccount, EntityName
- Custom Attributes for Entitlement – ResourceType, Namespace, SystemName, EntityName

- Lieberman RED Account Data Collector (ADC)
- Lieberman RED Entitlement Data Collector (EDC)
- Lieberman RED Connector
- Lieberman RED Request Form
- Lieberman RED Account Template

RSA Identity Governance and Lifecycle Lieberman RED Collectors

The following sections describe how to use the Lieberman RED Collector to get data from Lieberman RED into RSA Identity Governance and Lifecycle.

Lieberman RED Collectors

These collectors communicate with Lieberman RED and collect the Account and Entitlement data.

Lieberman RED Account Data Collector

Lieberman RED Account Data Collector (ADC) collects RED users and associates them with the users of RSA Identity Governance and Lifecycle. If RED delegation identities of type Domain user/group having different domains must be collected as accounts, multiple Account Data Collectors (ADCs) with corresponding domain must be configured.

These are the RED users collected as accounts in RSA Identity Governance and Lifecycle. They use delegation rules (permissions) to control access to the web client and the web service APIs. Delegation Identities have several types and they are described as follows.

1. Windows Domain Group - RED can provide access to groups of users from Windows domains.
2. Windows Domain User - RED can provide access to users from Windows domains.
3. Explicit Identity - RED can create explicit accounts (program local accounts) that exist only in the confines of the program. They have no association with any directory or system.
4. LDAP user – RED can provide access to users from LDAP servers.

Data/Attributes to be collected using Account Data Collector (ADC)

- AccountName
- DisplayName
- EmailAddress
- IsDomainAccount
- EntityName (required for account-entitlement resolution)

Adding Additional Attributes for Account Data Collector (ADC)

Additional attributes are required to hold the data collected in RSA Identity Governance and Lifecycle system. If they do not exist then you must add them.

1. Go to Admin > Attributes.
2. In the “Account” tab add the following attributes.

Attribute Name	Data Type	Database ID	Data Source
AccountName	String	<one of available>	Collected

Attribute Name	Data Type	Database ID	Data Source
DisplayName	String	<one of available>	Collected
EmailAddress	String	<one of available>	Collected
IsDomainAccount	String	<one of available>	Collected
EntityName	String	<one of available>	Collected

Creating a New Account Data Collector (ADC)

To set up a new Lieberman RED Account Data Collector without using the Application Wizard, follow steps below:

1. Log in to RSA Identity Governance and Lifecycle instance.
2. Select Collectors > Account Data Collector/Entitlement Data Collector.
3. Click Create Account Collector. Configure the Collector Description screen with these values:

Field Name	Value
Collector Name	Unique Collector name
Description	Collector description
Business Source	Select any available application
Data Source Type	Lieberman Enterprise Random Password Manager
Agent	AveksaAgent
Status	Active
Copy From	Select Existing Lieberman Enterprise Random Password Manager Account Collector template If you want to use its configuration
Scheduled	Default : No

4. Click Next.
5. Configure the Configuration Information screen with these values:

Field Name	Value
Scheme	HTTP or HTTPS
Host	Host name of the Lieberman RED instance.
Port	Port of Lieberman RED server E.g. for http, use port 80.
Base URL	Base URL to connect to Lieberman RED server. E.g. /ERPMWebService/JSON/V2/AuthService.svc
Collect Explicit Delegation Identities	To be checked if explicit delegation identities are to be collected from Lieberman RED.
Domain	Provide the Domain Name of Active Directory/LDAP for collection. Accounts collected are filtered based on the domain name provided.
Username	Username used to login to Lieberman RED server.
Password	Password for user of Lieberman RED
Authenticator Type	The authentication server configured in RED, for example, an Explicit or domain Account.
Domain Name *appears when the Authenticator field value selected is Domain account.	If Authenticator is Domain account, then provide the domain name for the username used to login to Lieberman RED server.
LoginType	The type of account being used for logging in to Lieberman RED. One of the following values: Unknown NativeStaticAccount: for a local account (explicit) without a domain/authenticator. FullyQualifiedAccount: for a domain/LDAP

	account. IntegratedAuthentication: for Windows integrated authentication. CertificateAuthentication: for certificate-based authentication. **Note:** Use NativeStaticAccount for a local account without a domain/authenticator. Use FullyQualifiedAccount for a domain/LDAP account. Use IntegratedAuthentication for Windows integrated authentication. Use CertificateAuthentication for certificate-based authentication.
Proxy Host	Hostname of the proxy server
Proxy Port	Port of the proxy server Default Port is 0, Note: Keep Proxy Port 0 if you are not using any proxy server to connector Lieberman Server.
Proxy User Name	Username for the proxy server
Proxy Password	Password for the proxy server

6. Click Next.
7. Configure the Map Collector Attributes to Account Attributes screen with these values:

Field Name	**Value**
Last Login Date	Last login date is not collected from RED
*Account Name	AccountName
*Display Name	DisplayName

*Email Address	EmailAddress
*Entity Name	EntityName
*Is Domain Account	IsDomainAccount

***AccountName, DisplayName, EmailAddress, EntityName and IsDomainAccount are custom attributes created in RSA Identity Governance and Lifecycle.**

8. Click Next.
9. Configure the Map Collector Attributes to Account Mapping Attributes screen with these values:

Field Name	Value
User Reference	Account Name

10. Click Next.
11. Configure the Edit User Resolution Rules screen with these values:

Field Name	Value
Target Collector	Select already created IDC. Default: Users
User Attribute	Email Address Default: User Id

12. Click Finish to save this Collector.

Lieberman RED Explicit Delegation Identities are collected as orphan accounts in RSA Identity Governance and Lifecycle.

Lieberman RED Entitlement Data Collector

Lieberman RED Entitlement Data Collector (EDC) collects Lieberman RED resources and their corresponding actions to form RSA entitlements and associates them with the Accounts collected from Lieberman RED.

Data/Attributes to be collected using Entitlement Data Collector (EDC)

- ResourceType

- Namespace
- SystemName
- EntityName

Adding Additional Attributes for Entitlement Data Collector (EDC)

Additional attributes are required to hold the data collected in RSA Identity Governance and Lifecycle system. If they do not exist then you must add them.

1. Go to Admin > Attributes.
2. In the “Entitlement” tab add the following attributes.

Attribute Name	Data Type	Database ID	Data Source
ResourceType	String	<one of available>	Collected
Namespace	String	<one of available>	Collected
SystemName	String	<one of available>	Collected
EntityName	String	<one of available>	Collected

Permission Types to be collected with Entitlement Data Collector (EDC)

Global Permission Set:

Permissions applied using the "Web Application Global Delegation Permissions" dialog in the management console are visible in the Lieberman RED web client. Listed below are the permissions names of permissions in the console are on the left side and names of permissions in the website are listed on the right side.

1. PermissionLogon
2. PermissionRequire2Factor
3. PermissionRequireOATH
4. PermissionChangePasswords
5. PermissionAccessJobs
6. PermissionViewWebLogs
7. PermissionElevateAnyAccount
8. PermissionViewDelegation
9. PermissionViewFileStore
10. PermissionChangeDelegation
11. PermissionViewDashboards
12. PermissionIgnorePasswordCheckout
13. PermissionAllAccess

14. PermissionChangeSharedCredentialLists
15. PermissionRequestPasswords
16. PermissionRequestRemoteAccess
17. PermissionGrantPasswordRequests
18. PermissionViewPasswords
19. PermissionSelfElevation
20. PermissionViewAccounts
21. PermissionViewSystems
22. PermissionChangePasswordsOnManagedSystems
23. PermissionViewPasswordHistory
24. PermissionViewPasswordActivity
25. PermissionModifyRefreshJob
26. PermissionModifyPasswordChangeJob
27. PermissionModifyElevationJob

Management Set Permission:

Configuring delegation on a per system management set basis allows you to grant delegates the ability to recover passwords for one set of systems while requiring them to request passwords (workflow) for another, and granting them View system permission for yet another.

1. PermissionGrantPasswordRequests
2. PermissionRequestPasswords
3. PermissionViewAccounts
4. PermissionViewPasswords
5. PermissionViewSystems
6. PermissionRequestRemoteAccess
7. PermissionSelfElevation

System Permission Set:

Configuring delegation on a per system management set basis allows you to grant delegates the ability to recover passwords for one set of systems while requiring them to request passwords (workflow) for another, and granting them View system permission for yet another.

1. PermissionGrantPasswordRequests
2. PermissionRequestPasswords
3. PermissionViewAccounts
4. PermissionViewPasswords
5. PermissionViewSystems
6. PermissionRequestRemoteAccess
7. PermissionSelfElevation

Account Permission Set:

Configuring delegation on a per-account basis makes it possible to grant a delegate different access privileges for separate accounts.

1. PermissionAccessRemoteSesisons
2. PermissionGrantPasswordRequests
3. PermissionRequestPasswords
4. PermissionViewAccounts
5. PermissionViewPasswords
6. PermissionRequestRemoteAccess

Shared Credential List Permission Set:

Once a shared credential list is created, it can be edited by users with the "Web Application Global Delegation" permission for Manage External Lists, or who have access to the management console. Permissions on these lists can be set using the web application or the admin console.

1. PermissionRequestPassword
2. PermissionChangeDelegation
3. PermissionAddPassword
4. PermissionDeletePassword
5. PermissionEditPassword
6. PermissionGrantRequest
7. PermissionRecoverPassword
8. PermissionViewList

Creating a New Entitlement Data Collector (EDC)

To set up a new Lieberman RED Entitlement Data Collector without using the Application Wizard, follow steps below:

1. Log in to RSA Identity Governance and Lifecycle.
2. Select Collectors > Entitlement Data Collector.
3. Click Create Account Collector. Configure the Collector Description screen with these values:

Field Name	Value
Collector Name	Unique Collector name
Description	Collector description
Business Source	Select any available Application

Data Source Type	Lieberman Enterprise Random Password Manager
Agent	AveksaAgent
Status	Active
Copy from	Select Existing Lieberman Enterprise Random Password Manager Account Collector template If you want to use its configuration
Scheduled	Default : No

4. Click Next.
5. Configure the Configuration Information screen with these values:

Field Name	**Value**
Scheme	HTTP or HTTPS
Host	Host name of the Lieberman RED instance.
Port	Port of Lieberman RED server
Base URL	Base URL to connect to Lieberman RED server.
Username	Username used to log in to Lieberman RED server.
Password	Password for user of Lieberman RED
Authenticator Type	The authentication server configured in RED, for example, an Explicit or domain Account.
Domain Name *appears when the Authenticator field value selected is Domain account.	If Authenticator is Domain account, then provide the domain name for the username used to log in to Lieberman RED server.

LoginType	The type of account being used for logging in to Lieberman RED. One of the following values: Unknown NativeStaticAccount: for a local account (explicit) without a domain/authenticator. FullyQualifiedAccount: for a domain/LDAP account. IntegratedAuthentication: for Windows integrated authentication. CertificateAuthentication: for certificate-based authentication. **Note:** Use NativeStaticAccount for a local account without a domain/authenticator. Use FullyQualifiedAccount for a domain/LDAP account. Use IntegratedAuthentication for Windows integrated authentication. Use CertificateAuthentication for certificate-based authentication.
Proxy Host	Hostname of the proxy server
Proxy Port	Port of the proxy server Default Port is 0 **Note**: Keep Proxy Port 0 if you are not using a proxy server to connect to the Lieberman Server.
Proxy User Name	Username for the proxy server
Proxy Password	Password for the proxy server

6. Click Next.
7. Configure Account Evaluation screen with these values

Field Name	**Value**

Associated Account Collector	Lieberman RED Account Data Collector **Note:** In case of multiple Lieberman RED ADCs for different domains, map EntityName to each ADC.
Account Value Evaluates To	Entity Name

8. Click Finish to save the Collector.

Note: Permissions of Delegation Identities not being collected in RSA Identity Governance and Lifecycle will result in rejection.

RSA Identity Governance and Lifecycle RED Connector

The following sections describe how to use the Lieberman RED Connector to update Lieberman RED with changes from RSA Identity Governance and Lifecycle.

Lieberman RED Connector

This REST API-based connector can be used to communicate with and provision data to Lieberman RED.

Configuration

The Connector is configured by completing the three sections: General, Settings, and Capabilities.

Perform the following steps to get to the Connector wizard:

1. Log in to RSA Identity Governance and Lifecycle.
2. Click Connectors under AFX menu.
3. Click Create Connector.
4. Enter the appropriate information for each of the sections.

General – General details about the Connector like the name, type.

The settings for the tab are listed in the following table:

Field Name	Value
Name	<Connector instance name>
Description	<Connector instance description>
Server	Select available AFX server
Connector Template	Lieberman Enterprise Random Password Manager
State	Test/Active Note that on satisfactory Connector testing, change the state to *Active*. No automated provisioning occurs while in the *Test* state. RSA recommends that you test all enabled commands using the *Test* Connector Settings and check Connector Capabilities prior to changing to the *Active* state.

Export As Template	Use this field to define name of this Connector export zip file. It is used while exporting Connector instance.

Settings – The connection settings required to connect RSA Identity Governance and Lifecycle to Lieberman RED.

The settings for the tab are listed in the following table:

Field Name	**Value**
Connection Details	
Scheme	Protocol (http/https) to be used for connection
Host	Fully qualified host name of the Lieberman RED instance
Port	Port of Lieberman RED server 80 for HTTP and 8443 for HTTPS.
Base URL	Base REST URL to connect to Lieberman RED server
Username	Username used to log in to Lieberman RED server
Password	Password for user of Lieberman RED
Authenticator	The authentication type – Lieberman RED Explicit User or a Domain Account
Domain Name	The name of the Domain. Used when the 'Authenticator' is of type Domain Account
Login Type	The type of account being used for logging in to Lieberman RED. One of the following values: Unknown NativeStaticAccount: for a local account (explicit) without a domain/authenticator. FullyQualifiedAccount: for a domain/LDAP account. IntegratedAuthentication: for Windows integrated authentication. CertificateAuthentication: for certificate-based authentication.

Proxy Settings	
Proxy Host	Hostname of the proxy server **Note:** Proxy settings are available in 6.9.1Patch-02 and above.
Proxy Port	Port of the proxy server
Proxy User Name	Username for the proxy server
Proxy Password	Password for the proxy server

Capabilities – This tab has a list of capabilities supported by this Connector.

When creating the Connector, by default, all of the capabilities have parameters pre-populated. Each capability has same configurations.

- Create Delegation Identity
- Update Delegation Identity
- Delete Delegation Identity
- Assign Permission to Delegation Identity
- Remove Permission from Delegation Identity

Command Input Parameters

Create Delegation Identity

Field Name	**Value**
Parameter Name	AccountName
Type	STRING
Default Value	None
Is the parameter required?	Yes

Is the parameter encrypted?	No
Display Name	Delegation Identity Name
Mapping	${AccountTemplate.AccountName}
Description	Lieberman RED Delegation Identity Name - must be unique

Field Name	**Value**
Parameter Name	DomainName
Type	STRING
Default Value	None
Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Domain Name
Mapping	${AccountTemplate.DomainName}
Description	Provide the Domain Name of Active Directory/Ldap for Delegation Identity type Domain User/Domain Group.

Field Name	**Value**
Parameter Name	DisplayName

Type	STRING
Default Value	None
Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Display Name
Mapping	${AccountTemplate.DisplayName}
Description	Lieberman RED Delegation Identity Display Name

Field Name	**Value**
Parameter Name	EmailAddress
Type	STRING
Default Value	None
Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Email Address
Mapping	${AccountTemplate.EmailAddress}
Description	Lieberman RED Delegation Identity Account email address

Field Name	**Value**
Parameter Name	Password
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	Yes
Display Name	Password
Mapping	${AccountTemplate.Password}
Description	Lieberman RED Delegation Identity Password

Field Name	**Value**
Parameter Name	IsDomainAccount
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No

Display Name	Account Role Type
Mapping	${AccountTemplate.IsDomainAccount}
Description	One of the following values: Explicit User Windows Domain User Windows Domain Group LDAP User

Field Name	**Value**
Parameter Name	Privilege
Type	STRING
Default Value	Logon
Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Privileges
Mapping	${AccountTemplate.Privileges}
Description	Lieberman account privileges - permissions, a comma separated List (Logon, AllAccess) Following is the list of values that can be added: AddPasswordsForManagedSystems AllAccess AllowRemoteSessions CreateRefreshSystemJob EditDelegation EditPasswordLists EditStoredPasswords ElevateAccountPermissions

	ElevateAnyAccountPermissions GrantPasswordRequests IgnorePasswordCheckout Logon PersonalStore RequestPasswords RequestRemoteAccess RequireOATH RequireRSASecurID SelfRecovery ViewAccounts ViewDashboards ViewDelegation ViewFileStore ViewJobs ViewPasswordActivity ViewPasswordHistory ViewPasswords ViewScheduler ViewSystems ViewWebLogs

Update Delegation Identity

Field Name	**Value**
Parameter Name	AccountName
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Delegation Identity Name

Mapping	${Account.AccountName}
Description	Lieberman RED Delegation Identity Name - must be unique

Field Name	**Value**
Parameter Name	DisplayName
Type	STRING
Default Value	None
Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Display Name
Mapping	${Account.DisplayName}
Description	Lieberman RED Delegation Identity Display Name

Field Name	**Value**
Parameter Name	EmailAddress
Type	STRING
Default Value	None

Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Email Address
Mapping	${Account.EmailAddress}
Description	Lieberman RED Delegation Identity email address

Field Name	**Value**
Parameter Name	IsDomainAccount
Type	STRING
Default Value	None
Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Account Role Type
Mapping	${Account.IsDomainAccount}
Description	One of the following values: Explicit User Windows Domain User Windows Domain Group LDAP User

Field Name	Value
Parameter Name	Password
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	Yes
Display Name	Password
Mapping	${AccountPassword}
Description	Lieberman RED Delegation Identity Password

Delete Delegation Identity

Field Name	Value
Parameter Name	AccountName
Type	STRING
Default Value	None
Is the parameter required?	Yes

Is the parameter encrypted?	No
Display Name	Delegation Identity Name
Mapping	${Account.AccountName}
Description	Lieberman RED Delegation Identity to be deleted

Assign Permission to Delegation Identity

Field Name	**Value**
Parameter Name	AccountName
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Delegation Identity Name
Mapping	${Account.AccountName}
Description	Lieberman RED Delegation Identity to be assigned permission

Field Name	**Value**

Parameter Name	ResourceType
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Lieberman RED Resource Type
Mapping	${Entitlement.ResourceType}
Description	Lieberman RED Resource Type - can be Account /ManagementSet/ Global/ System

Field Name	**Value**
Parameter Name	ResourceName
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Lieberman RED Resource Name

Mapping	${Entitlement.EntityName}
Description	Lieberman RED Resource Name

Field Name	**Value**
Parameter Name	ActionName
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Lieberman RED Action Name
Mapping	${Entitlement.Action_Name}
Description	Lieberman RED Action Name - like PermissionLogon/ PermissionViewAccounts

Field Name	**Value**
Parameter Name	SystemName
Type	STRING
Default Value	None

Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Lieberman RED System Name
Mapping	${Entitlement.SystemName}
Description	Lieberman RED System Name where the resource type Account resides - required only in case of resource type 'Account'

Field Name	**Value**
Parameter Name	Namespace
Type	STRING
Default Value	None
Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Lieberman RED Namespace
Mapping	${Entitlement.Namespace}
Description	Lieberman RED Namespace associated with the resource type Account - required only in case of resource type 'Account'

Remove Permission from Delegation Identity

Field Name	Value
Parameter Name	AccountName
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Delegation Identity Name
Mapping	${Account.AccountName}
Description	Lieberman RED Delegation Identity to be assigned permission

Field Name	Value
Parameter Name	ResourceType
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter	No

encrypted?	
Display Name	Lieberman RED Resource Type
Mapping	${Entitlement.ResourceType}
Description	Lieberman RED Resource Type - can be Account /ManagementSet/ Global/ System

Field Name	**Value**
Parameter Name	ResourceName
Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Lieberman RED Resource Name
Mapping	${Entitlement.EntityName}
Description	Lieberman RED Resource Name

Field Name	**Value**
Parameter Name	ActionName

Type	STRING
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Lieberman RED Action Name
Mapping	${Entitlement.Action_Name}
Description	Lieberman RED Action Name - like PermissionLogon/ PermissionViewAccounts

Field Name	**Value**
Parameter Name	SystemName
Type	STRING
Default Value	None
Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Lieberman RED System Name
Mapping	${Entitlement.SystemName}

Description	Lieberman RED System Name where the resource type Account resides - required only in case of resource type 'Account'

Field Name	**Value**
Parameter Name	Namespace
Type	STRING
Default Value	None
Is the parameter required?	No
Is the parameter encrypted?	No
Display Name	Lieberman RED Namespace
Mapping	${Entitlement.Namespace}
Description	Lieberman RED Namespace associated with the resource type Account - required only in case of resource type 'Account'

Lieberman RED Integration with RSA Identity Governance and Lifecycle - Use Case

The customer has Delegation Identities configured on Lieberman RED, which include some Explicit and some Windows domain users (with multiple Active Directory domains). Now the customer wants to manage these accounts and all corresponding entitlements using RSA Identity Governance and Lifecycle.

1. Collect all Identities using Active Directory Identity Data Collector.

   Configure Active Directory Identity Data Collector for two different domains so you can collect all identities from Active Directory.

2. Collect accounts (Delegation Identities) for Lieberman RED.

   Use the application wizard to configure Lieberman RED in RSA Identity Governance and Lifecycle. An ADC will be configured for one domain name provided. Collection of Explicit identities from Lieberman RED is optional.

   Configure a different Lieberman Account Data Collector for another domain.
   Mapping for IDC to ADC will be as follows:
   Domain 1 -> IDC 1-> ADC 1
   Domain 2 -> IDC 2-> ADC 2

3. Collect Entitlements for accounts that have been collected.

If you configured Lieberman RED using the Application Wizard, an ADC and EDC is automatically configured. Multiple ADCs, if required can be configured after the application has been created. To add mappings for multiple ADCs you must edit the EDC using the application wizard.

Tips and Troubleshooting

For any Connector, errors are written to the Server log files – AFX/esb/logs/esb.AFX-CONN-<ConnectorName>.log.

Status codes help you troubleshoot issues related to endpoint connectivity and responses received. Lieberman RED provides two status codes in its response:

Expected Condition	HTTP Response Code	Meaning	Example
The business process succeeded or failed in an expected way	200	Success	Request for non-existent entity (user, group, role etc.)
A system process failed (at a deeper level than the business process)	400	Bad Request	A malformed request body

- ***Why can't I obtain the authentication token using my valid credentials?***

  You can't obtain an authentication token because the account being used to authenticate to the web service doesn't have '**Logon**' permission. You must assign the required permission to the account and try obtaining the authentication token again.

- ***Why, even after obtaining the authentication token, am I unable to execute any capability?***

  The permissions required for obtaining an authentication token and executing various operations in Lieberman RED are different. You must assign '**Grant All Access**' to the account being used to perform the operations on Lieberman RED.

- ***Why does the request time out even though it is provided with valid host name and credentials?***

  The given port number is incorrect and thus is unable to make a connection to Lieberman RED.

- ***Why does the test collection fail with DataReadException for Lieberman RED ADC/EDC with SSL even though the credentials are valid?***

The Hostname Verification property is not set. To set the Hostname Verification property, see this section.

Known Issues

Known API Limitations

1. GetIdentities API call - PermissionRequire2Factor does not get reflected in the APIs.
2. SetPermissionOnSystem, SetPermissionOnAccount API calls - 'PermissionRequestRemoteAccess' is missing.
3. If a Delegation Identity has no permissions associated with any resource type, the corresponding resource type will not be collected in RSA Identity Governance and Lifecycle.

The following table contains mapping between permission names used in Lieberman RED API to permission display names on UI of Lieberman RED Admin Console.

Global Permission Set:

Permission name used in Lieberman API	Display name on Lieberman User Interface
PermissionLogon	Logon
-	Require Ext 2-Factor Auth
PermissionRequireOATH	Require OATH/Yubico
PermissionEditStoredPasswords	Add/Edit/Delete Password
PermissionViewJobs	Managed Scheduled Jobs
PermissionViewWebLogs	View Web Activity Logs
PermissionElevateAnyAccountPermissions	Elevate Any Account
PermissionViewDelegation	View Delegation
PermissionViewFileStore	Access File Repository
PermissionEditDelegation	Manage Delegation
PermissionViewDashboards	View Dashboards
PermissionIgnorePasswordCheckout	Ignore Password checkout
PermissionAllAccess	Grant All access
PermissionEditPasswordLists	Manage External Lists
PermissionRequestPasswords	Request Password Access
PermissionRequestRemoteAccess	Request Remote Access
PermissionGrantPasswordRequests	Grant Requests
PermissionViewPasswords	Recover Password

PermissionElevateAccountPermissions	Elevate Account Access
PermissionViewAccounts	View Account
PermissionViewSystems	View System
PermissionAddPasswordsForManagedSystems	Add/Edit/Delete Password for only Managed System
PermissionViewPasswordHistory	View Password History
PermissionViewPasswordActivity	View Password Activity
-	Edit Refresh Jobs
-	Edit Password Change Jobs
-	Edit Elevation Jobs

Management Set Permissions:

Permission name used in Lieberman API	Display name on Lieberman User Interface
PermissionGrantPasswordRequests	Grant Password Requests
PermissionRequestPasswords	Request Password Access
PermissionViewAccounts	View Accounts
PermissionViewPasswords	Recover Passwords
PermissionViewSystems	View Systems
PermissionRequestRemoteAccess	Request Remote Access
PermissionElevateAccountPermissions	Elevate Account
PermissionChangeGroupMembership	Change Group Membership
PermissionAllowRemoteSessions	Allow Remote Session

System Permission Set:

Permission name used in Lieberman API	Display name on Lieberman User Interface
PermissionGrantPasswordRequests	Grant Password Requests
PermissionRequestPasswords	Request Password Access
PermissionViewAccounts	View Accounts
PermissionViewPasswords	Recover Passwords
PermissionViewSystems	View Systems

PermissionRequestRemoteAccess	Request Remote Access
PermissionElevateAccountPermissions	Elevate Account
PermissionAllowRemoteSessions	Allow Remote Session

Account Permission Set:

Permission name used in Lieberman API	Display name on Lieberman User Interface
PermissionAccessRemoteSesisons	Allow Remote Session
PermissionGrantPasswordRequests	Grant Password Requests
PermissionRequestPasswords	Request Password Access
PermissionViewAccounts	View Account
PermissionViewPasswords	Recover Password

Shared Credential List Permission Set:

Permission name used in Lieberman API	Display name on Lieberman User Interface
PermissionChangeDelegation	Delegate Control
PermissionGrantRequest	Grant Password Requests
PermissionRequestPassword	Request Password Access
PermissionViewList	View Accounts
PermissionRecoverPassword	Recover Passwords
PermissionAddPassword	Add Password
PermissionEditPassword	Edit Password
PermissionDeletePassword	Delete Password

Copyrights

Trademarks