IBM Security

IBM

IBM Resilient SOAR Platform
Software Installation Guide
V34

Date: August 2019

Resilient Security Orchestration, Automation and Response (SOAR) Platform User Guide

Platform Version	Publication	Notes
34.0	August 2019	Initial publication.

Contents

Chapter 1. Introduction

Based on a knowledgebase of incident response best practices, industry standard frameworks, and regulatory requirements, the Resilient SOAR Platform makes incident response efficient and compliant.

There are three variations of the Resilient platform:

- Standalone installed on a Red Hat Enterprise Linux (RHEL) server **(this package)**
- Standalone installed on a FIPS compliant RHEL server
- VMware package installed on a RHEL host

You cannot upgrade from one variation to another, or install different variations on the same system.

MSSP add-on

The Resilient for Managed Security Service Providers (MSSP) add-on, licensed separately, is an optional feature that allows you to manage multiple Resilient child organizations from a single global dashboard. Each child organization can be assigned to a different group, division, or company to meet their incident response requirements.

Many of the administrative procedures remain the same; however, you manage the administrative settings in the configuration organization. If you have the MSSP add-on, you need to use the MSSP Add-on Configuration Guide to configure and manage the MSSP add-on components.

Important: If you are configuring Resilient for an MSSP deployment, you not need to create a regular Resilient organization. In addition, do not configure LDAP, as it is not currently supported for Resilient for MSSP.

Chapter 2. Prerequisites

Ensure that the operating system is Red Hat Enterprise Linux and the prerequisite software is installed on your system.

The requirements for the Resilient platform are:

- Red Hat Enterprise Linux 7.4 or 7.6.
- User account on the system with sudo privileges.
- The Resilient platform installation creates two directories, /usr/share/co3, and /crypt. The minimum required free disk space in the partition that hosts the /usr/share/co3 directory is 5 GB, and the minimum required free disk space in the partition that hosts the /crypt directory is 10 GB. IBM Resilient recommends using the Logical Volume Manager (LVM) to manage your partitions.
- Minimum 8 GB of RAM.
- The following packages must be installed (see the Chapter 3, “Deployment,” on page 5 section for information about how to install them):
  - wget package
  - Extra Packages for Enterprise Linux (EPEL)
  - pip Python installer program
  - setuptools Python module
  - PostgreSQL 9.6
  - javapackages-tools RPM
  - policycoreutils-python RPM
  - semver Python module
- The firewall must allow access to port numbers 443, 65000, and 65001 (see the Deployment section).

For additional guidelines about minimum configurations based on your expected workloads, refer to the Resilient Incident Response Platform Sizing Guidelines .

Chapter 3. Deployment

Check that your system has all of the required packages and then download the installer and install the Resilient platform.

Before you begin

Contact IBM Resilient Customer Support to acquire the software package.

If not already installed, install the `wget` package:

```
sudo yum install wget
```

If not already installed, install the Extra Packages for Enterprise Linux (EPEL):

```
sudo rpm --import http://dl.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL-7
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
sudo rpm -i epel-release-latest-7.noarch.rpm
```

If not already installed, install the `policycoreutils-python` and `javapackages-tools` packages:

```
sudo yum install policycoreutils-python javapackages-tools
```

If not already installed, install the pip Python Installer and then upgrade it to the latest version.

```
sudo yum install python-pip
sudo pip install --upgrade pip
```

If not already installed, install and upgrade the latest `setuptools` Python module:

```
sudo pip install --upgrade setuptools
```

If not already installed, install the semver Python module:

```
sudo pip install --upgrade semver
```

If not already installed, install and initialize the PostgreSQL on your system:

```
sudo rpm -Uvh https://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm
sudo yum install postgresql96-server
sudo yum install postgresql96-contrib
sudo /usr/pgsql-9.6/bin/postgresql96-setup initdb
```

If postgresql has already been initialized, you see a message: `Data directory is not empty!`

If not already done, configure the firewall, as follows:

```
sudo firewall-cmd --zone=public --add-port=443/tcp --permanent
sudo firewall-cmd --zone=public --add-port=65000/tcp --permanent
sudo firewall-cmd --zone=public --add-port=65001/tcp --permanent
sudo firewall-cmd --reload
```

Note: Port 65000 and 65001 are required only if you are using the Resilient Action Module.

About this task

Complete the following to download and install the Resilient platform on your system.

Procedure

1. If downloaded to a different machine, transfer the software package that you acquired from IBM to the system that is to host the Resilient platform.

2. Enter the following command to install the Resilient platform, using the actual version number (in the format *x.x.x*) in the file name:

```
sudo bash resilient-<version>.run
```

3. Make sure your system is up-to-date with all required OS security updates.

What to do next
After you complete the installation, import the Resilient license and then complete the tasks in the following sections to configure the Resilient platform.

Importing the Resilient license

About this task

Before you can start the Resilient platform, you must import the license that you obtained from IBM Resilient. To import the license, you must log in to the Resilient system using an SSH client, such as PuTTY.

To import the license:

Procedure

1. Copy the license file that you received from IBM Resilient for the Resilient system.
2. Log in to the system using SSH as the user account you created in the previous section. You can use PuTTY or connect from a terminal client as follows:

```
ssh <username>@<Resilient Platform hostname or IP Address>
```

3. To import the license, enter the following command:

```
sudo license-import -file <Resilient License File>
```

A message, similar to the following message, appears on the screen, indicating successful import:

```
Successfully imported license
Customer name:  <customer>
Expiration:  No expiration
US regulators enabled:  true
CA regulators enabled:  true
EU regulators enabled:  true
APAC regulators enabled:  true
Security module enabled:  true
Actions framework enabled: true
Users:  Unlimited
```

To display information about the currently installed license, enter the following command:

```
sudo resutil license
```

The system displays the following information:

- Customer name, which is the name of your company.
- Expiration, which is the expiration date of the license, or no expiration if the license does not expire.
- US regulators enabled, which displays true or false.
- CA regulators enabled, which displays true or false.
- EU regulators enabled, which displays true or false.
- APAC regulators enabled, which displays true or false.
- Security module enabled, which displays true or false.

- Actions framework module enabled, which displays true or false.
- Users, which displays the number of users the license allows, or Unlimited if there are an unlimited number of users allowed.

Results

If you do not have an installed license when you run this command, the system informs you that no license is installed.

Chapter 4. Setting the time zone

About this task

The Resilient platform uses dates for different purposes and therefore needs to know the time zone of your location. By default, the Resilient system time zone is set to UTC. Follow these steps to change the time zone:

Procedure

1. Enter the following command in the SSH client to list the available timezones:

```
timedatectl list-timezones
```

2. Determine the time zone that you want to use. For example, America/New_York.
3. Enter the following command to change the time zone:

```
sudo timedatectl set-timezone <time_zone>
```

4. Restart the `resilient-messaging.service` by entering the following command:

```
sudo systemctl restart resilient-messaging.service
```

Chapter 5. SSL certificate

The Resilient appliance comes with a self-signed Secure Socket Layer (SSL) certificate. However, it is not recommended that you use it in a production environment. For optimal security, we recommend that you obtain a certificate from a trusted certificate authority (CA).

To obtain and use the SSL certificate, follow these steps, which are outlined in more detail in the following sections:

1. If you do not have a signed certificate, you can create a certificate request then submit it to a CA, such as Thawte or Verisign, for signing.
2. Import the signed certificate into the Resilient platform then restart the Resilient service so that it recognizes the new certificate.

Creating and submitting the certificate request

About this task

To create a certificate request:

Procedure

1. Enter the following command in your SSH client:

```
sudo cert-req
```

2. When prompted, enter the qualified domain name of the host, for example, `resilient.example.com`
3. When prompted, enter the subject alternate name of this certificate, for example `res.example.com` or `res2.example.com`. Some browsers, such as Chrome and Firefox, require the certificate alternate name while others browsers do not.
4. When prompted, enter the name of your company, for example, `My Company, Inc`.
5. When prompted, enter the name of your group in the company, for example, `Incident Response`.
6. When prompted, enter your city, for example, `Cambridge`.
7. When prompted, enter your state. Most CAs do not accept your request if you use a state abbreviation, for example, enter `Massachusetts`.
8. When prompted, enter the abbreviation for your country, for example, US.

   You can locate the certificate request in `/crypt/certs/certreq.pem` directory and it appears on the screen, as follows:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIDAjCCAeoCAQAwgYwxCzAJBgNVB...
-----END NEW CERTIFICATE REQUEST-----
```

9. Copy the content of the certificate request to the clipboard, starting with the "-----BEGIN NEW CERTIFICATE REQUEST-----" and ending with the "-----END NEW CERTIFICATE REQUEST-----".

   In PuTTY, you use the left mouse button to select text. The act of selection automatically copies the text to the clipboard. You do not need to press any other key. The only thing you need to do to copy text to the clipboard is to select it.

Results

Once you have the request, you need to have it signed. The procedure for getting your certificate signed depends on which certificate authority (CA) you use. If you choose a CA such as Verisign (http://www.verisign.com/) or Thawte (http://www.thawte.com/), go to their web site to obtain a signed certificate. You can submit the certificate request you generated in the previous section to your CA through their web site. If the CA asks for the server platform that the certificate applies to, you should choose Tomcat. They then contact you with information on how to obtain your signed certificate. After you obtain a signed certificate, you can import it into the Resilient platform.

Importing the signed certificate

About this task

To import the signed certificate, copy the certificate file to your Resilient system and enter the following command in your SSH client:

```
sudo cert-import <cert>
```

Where <*cert*> can be an end user certificate, such as `cert.cer`, or a certificate chain, such as `ca-chain.p7b`.

After you import the signed certificate, restart the `resilient-messaging.service` by entering the following command:

```
sudo systemctl restart resilient-messaging.service
```

Open a browser and connect to the Resilient platform by entering the following location:

```
https://<hostname for Resilient Platform>
```

Troubleshooting tip: If you see an error message, `java.security.cert.CertificateException: Fail to parse input stream` after the cert-import command, you may have extra characters (even whitespace) in the certificate file. To correct the error, open the certificate file to ensure the content starts with "-----BEGIN CERTIFICATE-----" and ends with "-----END CERTIFICATE-----". After fixing the certificate file, import it again.

Chapter 6. Accounts and additional configuration

Before you can use the Resilient platform, you must configure an initial user account in the platform and an organization to which this user belongs. This initial user is the platform's system administrator. After you create the system administrator account, you can use the account to create other users in the Resilient platform.

Optionally, you can configure the Resilient platform to use LDAP, SAML, or two-factor authentication. You can use either SAML authentication or LDAP authentication, but not both. Two-factor authentication is a second layer authentication and you can use it with LDAP or SAML.

Accounts and groups

Some user accounts and groups are created and available by default after a successful installation.

The following users are created by default:

- `postgres`
- `elasticsearch`
- `co3`
- `res-scripting`
- `res-email`
- `res-messaging`
- `irhub` (for the Email Connector only)

The following groups are created by default:

- `postgres`
- `elasticsearch`
- `co3`
- `res-attachments`
- `res-scripting`
- `res-email`
- `res-messaging`
- `res-keystore`
- `irhub` (for the Email Connector only)
- `irhubadmin` (for the Email Connector only)

Creating the initial Resilient user account

Create a system administrator account for the Resilient platform and an organization to which the administrator account belongs.

Enter the following command in your SSH client to determine how to create the platform's system administrator account and the organization to which the administrator belongs:

```
sudo resutil newuser -help
```

This command has the following options and defaults:

- `-createorg` creates the organization that contains the system administrator.

- `-createrole` creates a role, if it does not exist.
- `-email` provides an email address for the user. This is a required option.
- `-first` provides the first name of the system user.
- `-last` provides the last name of the system user.
- `-org` provides an organization for the user. This is a required option.
- `-orglocale` provides the default language of the organization. The default setting is English. Use the `-help` option to see the list of supported languages. The language values are case-sensitive. When the organization is created, you cannot change the locale. Note that some text, such as regulatory and legal-related information, is available only in English.
- `-role` assigns an existing role to the user. If unspecified, the default role, Master Administrator, is assigned.

This command prompts you to enter and confirm the password for this user (no keystrokes appear on the screen). The following is an example of the command:

```
sudo resutil newuser -createorg -email "jsmith@example.com" -first "John"
-last "Smith" -org "My Company, Inc."
Enter the password for the user:
Confirm the password for the user:
Creating a new user John Smith <jsmith@example.com
Creating a new organization My Company, Inc.
Adding the user John Smith <jsmith@example.com> to the organization My Company, Inc.
Assigning the following roles to user jsmith@example.com: Master Administrator
Upon successful completion of this command, you will be able to login to the
application and finish setup.
```

You can create multiple organizations in your system by running the above command multiple times. You only need to provide the `-first` and `-last` options the first time the user is created.

To subsequently edit an organization, use the `editorg` command. Enter the following to get information about the options to use with the `editorg` command:

```
sudo resutil editorg -help
```

The `editorg` command has the following options:

- `-address1` specifies the first line or the address for the organization.
- `-address2` specifies the second line address for the organization.
- `-city` specifies the city for the organization.
- `-name` specifies the organization's new name.
- `-orgname` specifies the name of the organization to be updated.
- `-state` specifies the organization's new state.
- `-zip` specifies the organization's new ZIP code.

LDAP authentication

To configure the Resilient platform to use LDAP authentication, you must have an Active Directory Server. The Resilient platform supports Active Directory only.

Note: LDAP is not currently supported for Resilient for MSSPs. If you are installing and configuring the Resilient platform for MSSPs, do not configure LDAP.

You should have at least one master administrator or equivalent account per organization whose email address is not managed by the Active Directory. When LDAP authentication is enabled, any email address managed by the Active Directory (based on your configuration) has its authentication and authorization to organizations determined by LDAP.

IMPORTANT: For users who had a previously configured Resilient account, logging into the platform using LDAP authentication clears the password for that account. If a user does not log in using LDAP, the account is still valid.

Basic configuration

The basic procedure to configure LDAP is to obtain the LDAP configuration values, use the ldapedit command to enter the values, enter the user password, and configure a Resilient organization to use LDAP authentication.

1. Make sure that you have the following values available before configuring the Resilient platform. You may need to consult with your LDAP administrator to get the appropriate values for your setup.

Value	Example
LDAP server name.	myldap.example.com
Distinguished Name (DN) of the user account that the Resilient platform can use to perform LDAP queries, such as determining if the user accessing the system is managed by LDAP. This account must have at least Read-Only permission to view all the necessary users. You also need the user password.	"cn=John Smith,cn=Users,dc=Example,dc=com"
LDAP search root for all directory searches. If not specified, the Resilient platform attempts to locate and use the Root DSE. If you wish to constrain all queries to a sub-tree in the LDAP, you can specify it here. For performance reasons, it is recommended that you select the lowest search root in the directory that contains all users and groups that you wish included in the queries.	cn=Users,dc=Example,dc=com
LDAP server port number. Determine if this is an SSL port, such as port 636.	389 (No SSL)
LDAP config name.	resilientLDAP
LDAP domain name. Only used when you have an LDAP configuration with multiple trees and you wish to extend LDAP searches to different trees. The LDAP domain name is the name the platform matches when it receives an LDAP search reference. You must provide the actual host name for each domain.	sales.division.company.com marketing.division.company.com executive.division.company.com

2. Use the ldapedit command to enter the LDAP values. For example, using the values in the previous table the command would be:

```
sudo resutil ldapedit -name myLDAP -bindname "cn=John Smith,
cn=Users,dc=Example,dc=com" -host myldap.example.com -port 389
```

   If enabling LDAP over SSL/TLS (using port 636), see the LDAP Over SSL/TLS section in this guide.
3. When prompted, enter the password of the user to complete the setup.
4. Test the LDAP configuration, For example:

```
sudo resutil ldaptest -name myLDAP
```

5. After completing the configuration, you must enable it in your Resilient organization as follows:

a. Log in to your organization as a Master Administrator or equivalent account.
b. Click on **Administrator settings** (from the menu by your username).
c. Click on the **Organization** tab.
d. Locate the **LDAP Authentication** section, under Settings.
e. Switch the indicator to **On**.
f. Select the LDAP group that you wish to have access to the Resilient organization.

Note: The information in the last step is also in the *System Administrator Guide*.

LDAP trees

The Resilient platform primary LDAP configuration points to a single LDAP tree. By default, all LDAP queries are sent to this tree. If you have an LDAP configuration with multiple trees, you can configure the platform to have a search in this LDAP tree point to a different tree. This is called an LDAP search reference, and it is usually in the form of an LDAP URL, ldap://<domainname>:<port>/<optional parameters>. To do this, you need to provide one or more domain names with the host and port. When there is an LDAP reference to ldap://<domainname>, the platform searches for a sub-configuration that has that domain name. If there is a match, the platform sends the LDAP query to that sub-configuration's host and port.

NOTE: The LDAP domain name is the name the platform matches when it receives an LDAP search reference. The LDAP domain name does not have to be the same as the host name of that LDAP server.

If you wish to search additional LDAP trees, create an additional LDAP configuration, called a sub-configuration, for each tree. The following example shows how to create three sub-configurations. The bindname can be different for each sub-configuration.

```
sudo resutil ldapedit -name salesSub -bindname "cn=John
Smith,cn=Users,dc=Example,dc=com" -domainname sales.division.company.com -
subdomainof myLDAP -host host1.sales.division.company.com -port 389

sudo resutil ldapedit -name marketingSub -bindname "cn=John
Smith,cn=Users,dc=Example,dc=com" -domainname sales.division.company.com -
subdomainof myLDAP -host host2.marketing.division.company.com –port 389

sudo resutil ldapedit -name execSub -bindname "cn=John
Smith,cn=Users,dc=Example,dc=com" -domainname sales.division.company.com
-subdomainof myLDAP -host host3.executive.division.company.com –port 389
```

If prompted, enter the password of the user to complete the setup.

NOTE: Using the examples in the basic procedure, you now have four LDAP configurations. The configuration defined in step 2 of the basic procedure is the primary configuration.

Troubleshooting tip: If you need to determine the number of external references a single LDAP search can perform, use the following command. For example, a search on ldap1 could reference ldap2, which in turn could reference ldap1. The default is 3.

```
sudo resutil configset -key ldap.ref_limit -ivalue <VALUE>
```

LDAP over SSL/TLS

If configuring LDAP using port 636, SSL/TLS, perform the following:

- Verify that the service is using a certificate signed by a trusted CA. If not, follow the instructions in the Importing Untrusted Certificates section of this guide.
- Use the following ldapedit command to enter the LDAP values. For example, using the values in the basic procedure the command would be:

```
sudo resutil ldapedit -name myLDAP -bindname "cn=John
Smith,cn=Users,dc=Example,dc=com" -host myldap.example.com -port 636
-usessl -wlhost myldap.example.com
```

The -wlhost option is not required if the common name on the certificate matches the name in –host in the above command.

Additional information

The following lists other helpful `ldapedit` commands:

- View the command help:

```
sudo resutil ldapedit -help
```

- To change the configuration value that syncs data between LDAP and the Resilient platform:

```
sudo resutil configset -key elastic_server.principal_indexer_sync_interval -ivalue <interval
 in minutes>
```

  The default value is 4 hours, which means that if you make updates in LDAP, such as adding, deleting or disabling users, there is a delay of 4 hours for the updates to be propagated to the Resilient platform.

- View your configuration:

```
sudo resutil ldapshow
```

- Delete your configuration:

```
sudo resutil ldapdel -name myLDAP
```

Troubleshooting tip: If you cannot log in after enabling LDAP, check the following:

- If you are unable to log in as an LDAP user, verify that you can use the same account to login using an LDAP browser, such as JXPLORER.
- The master administrator or equivalent account has an email address managed by Active Directory. Accounts tied to email addresses with Active Directory must be authorized by authorizing a desired LDAP group. Until a group is authorized, no email address tied to an Active Directory can access the organization. Add a new master administrator account to the desired organization using the following command line. Make sure that the email address is not tied to LDAP; in fact, you can use a fake address. (Consider keeping this user for one-off circumstances.)

```
sudo resutil newuser -email "jsmith@example.com" -first "John"
-last "Smith" -org "My orgname"
```

  After you confirm that this user can log in and access the Administrator settings, then enable LDAP again as before, to authorize your desired LDAP group.

SAML authentication

SAML authentication allows users to use their organization's login credentials to authenticate to the Resilient platform. The SAML specification identifies two different types of endpoints that are relevant to the Resilient platform:

- Identity Providers
- Service Providers

The Resilient platform serves as a SAML Service Provider. An authentication and identification system that you provide (such as Microsoft Active Directory Federation Services) serves as the Identity Provider.

To configure the Resilient platform to function as a SAML Service Provider, follow these steps, which are outlined in more detail in the following sections:

1. Create a SAML federation.
2. Import the SAML metadata into your Identity Provider.
3. Test the configuration.

Important: For users who had a previously configured Resilient account, logging into the platform using SAML authentication clears the password for that account. If a user does not log in using SAML, the account is still valid.

Create a SAML federation

SAML federations are created in the Resilient platform using the resutil tool. To create the SAML federation, you need the following information from your Identity Provider:

- Identity Provider Authentication URL
- Identity Provider public certificate

Additionally, you need the organization name on the Resilient platform to which this federation applies. You also need to assign an "alias" for this federation. The alias appears in the URL to users when initially connecting to the Resilient platform through SAML.

Important: Only alphanumeric characters and the underscore character are supported for the SAML federation name.

The instructions in this section assume that:

- The authentication URL for your identity provider is https://adfs.example.com/adfs/ls/
- You have copied the Identity Provider's certificate file to the system using a tool such as `scp` and the file name of the certificate file is `idp.cer` in the current working directory.
- The "alias" for the SAML federation is "resilient".
- The organization name on the Resilient platform is "My Test Org".

```
sudo resutil samledit -alias resilient -certfile idp.cer -org "My Test Org"
-createusers -url https://adfs.example.com/adfs/ls/
```

- If your identity provider has set up single logout functionality, you can have the Resilient system specify that as well in the `samledit` command:

```
sudo resutil samledit -alias resilient -org "Production" -org "Development"
-certfile idp.cer -loginurl https://adfs.example.com/adfs/ls/ -logouturl
https://adfs.example.com/adfs/ls/
```

This command prints out the SAML federation to the console. It also writes out the following files:

- *alias*-metadata.xml - SAML XML metadata that can be imported into the Identity Provider to complete the configuration.
- *alias*-sp-cert.pem - Service Provider certificate that was automatically generated.

Troubleshooting tip:By default, the Resilient system verifies the signature on all incoming identity provider messages. If an incoming message is not signed, the Resilient system rejects the message. To have the Resilient system not verify identity provider message signatures, use the command line option, `-requiresignedidprequests false`.

A federation can be associated with multiple organizations. Consider the situation where your Resilient platform is configured to have two organizations: Production and Development. You can create a single federation that allows access to both organizations. For example:

```
sudo resutil samledit -alias resilient -org "Production" -org "Development"
-certfile idp.cer -url https://adfs.example.com/adfs/ls/
```

By default, users are only granted access to the organization via the federation if they already exist in that organization. Consequently, in the above example, users would have to exist in both "Production" and "Development" organizations, in order to access them. If, however, you want users to be automatically be

added to the “Production” organization then you would specify –createusers for just that organization. For example:

```
sudo resutil samledit -alias resilient -org "Production" -createusers
```

If the **–createusers** argument is specified for both organizations then users who authenticate via the federation are automatically created in both organizations.

You can unlink the federation from an organization by using the -clearorgs flag.

Import the SAML metadata into your identity provider

The SAML metadata written out in the previous step can now be imported into your SAML Identity Provider. The specific instructions vary by product. Please consult the documentation for your Identity Provider for instructions on how to create a federation (also called a "relying party").

The Resilient platform requires that the Identity Provider provide the following attributes:

- E-mail address
- First name (given name)
- Last name (surname)

The Resilient platform does not function if the above attributes are missing.

The Resilient platform utilizes the following attributes if they are present (they are not required for proper operation):

- Phone number
- Mobile phone
- Title
- Groups

Consult your Identity Provider documentation for details on configuring which attributes are sent during authentication.

Authenticated users are added to the groups listed by the Identity Provider in the SAML response. For example, if the user is a member of the "IT" group and it is sent then the user is added to the "IT" group in the Resilient platform, if it exists. Groups that are in the SAML response that do not already exist in the Resilient platform are not automatically created.

Test the configuration

Once you have configured SAML in Identity Provider, you can test the authentication by using the Authentication URL for your organization. You can check this value by running the following command:

```
sudo resutil samlshow
```

For example:

```
https://resilient.example/saml2/resilient
```

Using this URL redirects you to the Identity Provider for authentication. After you have authenticated, you are redirected to the Resilient platform and logged in to the platform. Note that authentication may be done without prompting (single sign-on).

You can send the above URL to users who you wish to grant access to the Resilient platform. Note that all users who are authorized to use the Identity Provider are granted access to the Resilient platform. If you want to restrict access, you must do so through the Identity Provider configuration.

Two-factor authentication

Two-factor authentication provides unambiguous identification of users by means of the combination of two different components. These components may be something that the user knows, something that the user possesses or something that is inseparable from the user. Combining two components as a means of identification adds a second layer of security to your accounts. Verifying your identity using a second factor (like your phone or other mobile device) prevents anyone but you from logging in, even if they know your password.

The Resilient platform uses Duo Security, a third-party vendor, as its two-factor authentication provider. When you enable two-factor authentication, users can still log in with their email address and password but are also presented with a challenge - an additional second layer of security provided by Duo Security - to verify their identity. This challenge appears anytime a user, who has not been previously authenticated via two-factor, tries to access an organization.

How to set up two-factor authentication

1. Sign-up for a Duo Account at https://www.duosecurity.com/pricing.
2. Create and configure a Duo application to use with the Resilient platform.

   During this stage, you receive an Integration Key, Secret Key, and API hostname. You need these items to configure two-factor authentication on the Resilient platform.

   When prompted for the application type, select **Web SDK**.
3. Locate the Integration Key, Secret Key, and API hostname from your Duo application.
4. Determine the names of one or more Resilient organizations that can enable the two-factor domain. (When this procedure is completed, a master administrator can then choose to enable the two-factor domain for each of those organizations.)
5. Create a two-factor domain:

   ```
   $ sudo resutil twofactoredit -name <domain_name> -org <org_name>
   -integrationkey <duo_integration_key> -host <duo_api_hostname>
   -integrationsecret <duo_integration secret>
   ```

   Where:

   <domain_name> is the name of the two-factor authentication domain. This name is presented to your Organization Administrator in a drop down.

   <duo_integration_key>, <duo_api_hostname>, and <duo_integration secret> are values obtained from Duo Security after completing their configuration.

   <org_name> is the name of an organization in the Resilient platform that may utilize the two-factor domain. Multiple -org <org_name> arguments can be provided.

You can also use the resutil twofactoredit command to change the name of an existing domain and clear any orgs associated with it. Use this command to check all the options:

```
sudo resutil twofactoredit -help
```

To display the details of an existing two-factor domain:

```
sudo resutil twofactorshow -name <domain_name>
```

Note: If the -name is not specified, all the organizations with two-factor authentication are displayed.

To delete a two-factor domain:

```
sudo resutil twofactordel -name <domain_name>
```

In some cases, you may want to exclude a specific user belonging to an organization configured for two-factor authentication; for example, you want to programmatically access the Resilient REST API with a system account. This can be done by using the twofactorexcluser command:

```
sudo resutil twofactorexcluser -email user@example.com
```

This enables user@example.com to access the two-factor org without providing the Duo authentication.

You can re-enable two-factor authentication by using the -clearemail flag. For example:

```
sudo resutil twofactorexcluser -clearemail user@example.com
```

Enabling your authentication domain

While logged in as a master administrator, you can enable a two-factor authentication domain under organizational settings on the administrator's settings page. If you have set up multiple two-factor authentication domains, you can select which domain you would like to authenticate your users against here. On this page, you can also set the cookie lifetime, which sets an expiration in days for when a user needs to re-authenticate via two-factor authentication.

Note: Authentication domains are set at the organizational level. You can use the same authentication domain for multiple organizations or set a different domain for each organization. This means that a user who authenticates against an organization under one domain, who then tries to access an organization under another domain, needs to separately authenticate for the other organization.

Registering users

Once two-factor authentication has been configured, users need an account on the Resilient platform and a corresponding account in your Duo Security account.

Management of user registration with Duo security is handled in the Policy settings of your Duo Security application. The "New user policy" allows you to select:

- Require Enrollment - users who are not already registered with Duo security are provided a self-enrollment process that makes it easy for users to register their devices and install the Duo mobile app (if necessary). When a user logs into the Resilient platform for the first time after two-factor authentication is enabled, Duo Security begins this self-enrollment process.
- Allow Access - users who are not already registered with Duo security are not challenged. We recommend AGAINST using this option.
- Deny Access - only users who are already configured with the Duo account are allowed access. This means that you need to configure your users using your Duo account in the "Users" tab.

The email address of the Resilient platform must match the Duo account username. In the Duo application settings, "Username normalization" allows you to specify whether or not "DOMAIN \username", "username@example.com" and "username" are all treated as the same user.

Two-factor authentication and user experience

When two-factor authentication is enabled, if a user has not been "previously authenticated" via two-factor authentication, they are presented with a two-factor challenge whenever they try to access an organization.

A user is considered as being "previously authenticated" under the following circumstances:

- They have successfully passed the challenge presented via two-factor authentication in their current session.
- They have successfully passed the challenge presented via two-factor authentication in a session then started a new session within the number of days set by the cookie lifetime value. In this situation, the user authenticates as normal (email and password) when starting the new session, but is not presented

with the challenge. The master administrator sets the cookie lifetime value in the administrator settings organization page.

Add additional user accounts

Now that you have successfully installed and set up the Resilient platform, you can open it and begin adding additional user accounts for the users you want to have access to the platform. See the *System Administrator Guide* for more information on adding additional Resilient user accounts.

Importing untrusted certificates

The Resilient platform might interact with services such as proxies, SMTP servers, and custom threat services, that do not use trusted SSL certificates. Instead, they might use self-signed certificates or certificates issued by an internal certificate authority. You must trust these certificates explicitly to use these services.

About this task

You perform the following steps to trust the certificates:

- Obtain the certificate from the service you need to trust (for example, Active Directory Server, SMTP Server, or Custom Threat Service).
- Add this certificate to a custom Java KeyStore called custcerts in `/crypt/certs/` folder on the Resilient system. It contains customer specific certificates for communication with external systems, for example, SMTP servers, custom threat feeds, and so on. The Resilient platform explicitly trusts all certificates within this keystore. This file is not changed during upgrades.

These steps are described in more detail as follows.

Procedure

1. Obtain the certificate from the service. You can request it from the administrator of that service or use the `openssl` utility installed on the Resilient system as follows:

   ```
   openssl s_client -connect active-directory.example.com:636 -tls1
   -showcerts </dev/null 2>/dev/null|openssl x509 -outform PEM > active-directory.pem
   ```

   In this example, the Active Directory's certificate is stored in a file called `active-directory.pem`. The host and port depend on the service you are trying to use. Additional parameters, such as `-starttls` might be required for SMTP servers.
2. Check that the `/crypt/certs/custcerts` directory exists.

   ```
   sudo ls -al /crypt/certs/custcerts
   ```
3. If it exists, verify that you can list the contents:

   ```
   sudo keytool -list -keystore /crypt/certs/custcerts
   ```

   You need to enter the keystore password to view the contents.
4. If the directory does not exist, add the certificate to the `/crypt/certs/custcerts` keystore:

   ```
   sudo keytool -importcert -trustcacerts -file <certificate name> -alias
   <name identifying the service e.g. MyCompanyActiveDirectory> -keystore
   /crypt/certs/custcerts
   ```

   You must enter the keystore password. A new keystore is created if it does not exist.

   Enter "yes" to trust the certificate.

   Repeat the command in the previous step to verify that you can list the keystore entries.

5. Add the custcerts password into the Resilient KeyVault:

```
sudo resutil keyvaultset -name custcerts -stdin
<Password for custcerts>
Ctrl+D (to send an EOF signal to standard input)
```

Chapter 7. Network configuration

For some Resilient functions to operate properly, the system requires access to services on the Internet. Work with your network administrators to ensure the system has access to the following URLs to support the corresponding services.

URL	Purpose
https://websvc.resilientsystems.com	IBM Resilient cyber threat service
https://repo.co3sys.com	Resilient software updates
ntp:ntp.org (udp port 123)	Network time synchronization

Chapter 8. Log file configuration

The Resilient platform logs various client and server activity in log files, located in the following directory:

```
/usr/share/co3/logs/
```

Log files in this directory include:

- **catalina.out.** Tomcat Catalina output file.
- **client.log.** Main Resilient platform log file.
- **client_access_log.log.** Tomcat-based log that keeps track of all HTTP request made to the Resilient platform server.
- **monitoring.log.** Resilient platform log file containing timing-related information.

PostgreSQL logs database access in log files located in the following directory:

```
/var/lib/pgsql/9.6/data/pg_log
```

Most logs roll daily and the rolled file is named with the date that it was rolled. The **daily** folder contains the rolled **client.log** and **monitoring.log** files.

By default, the Resilient client log files (client.log and monitoring.log) use a timestamp that includes only the current time of day. This is because the logs roll over daily and the date of the log is included in the filename. However, you can change the date format in order to keep it consistent across all of your logs. To do this, create a file named **logback-custom-pre.xml** in the /crypt folder of the Resilient system and add the following:

```
<included>
   <property name="customTimeStamp" value=MyFormat/>
</included>
```

where MyFormat is a valid logback time/date stamp format. For example, "%d{yyyy-MM-dd HH:mm:ss.SSS}" generates log messages with the following date format:

```
2016-01-14 16:34:39.218 [main] INFO ...
```

This file must be readable by the co3 group, so you may need to change the group associated with the file using:

```
sudo chgrp co3 /crypt/logback-custom-pre.xml
```

To implement your changes immediately, restart the Resilient service using the following command:

```
sudo systemctl restart resilient.service
```

Chapter 9. Email configuration

The Resilient platform sends email messages to users for notifications, such as when a new user becomes a member or when the platform assigns a user a task. Therefore, the Resilient platform must use an SMTP server to send these messages. After you install the platform, stay in the SSH client you use and enter the following command with the options you want to use to edit the SMTP configuration:

```
sudo resutil smtpedit
```

This command has the following options and defaults:

- `-help` prints the SMTP edit configuration help. The default is false.
- `-email` provides the email address in the From field of the email message.
- `-host` provides the host name of the mail server.
- `-name` provides the name in the From field of the email message.
- `-nostarttls` provides the option to not issue a StartTLS when connecting to the mail server.
- `-port` provides the port of the mail server.
- `-user` provides the user of the mail server. The system prompts you for the password if you use this option.
- `-wlhost` allows you to specify the host name when it is different from the certificate common name to avoid certificate name mismatch errors.

The following is an example that shows how to configure the system so that email messages sent from the Resilient platform appear to be from Resilient Incident Management *<user@example.com>*. In this example, the SMTP server is *<smtp.example.com>* and the port is 2525. The SMTP server requires authentication in this example and the account used is the Resilient account. If your SMTP server does not require authentication, you can omit the –user option.

```
sudo resutil smtpedit -email user@example.com -name "Resilient Incident
Management" -host smtp.example.com -port 2525 -user resilient

Enter the password for the user: <SMTPpassword>

Confirm the password for the user: <SMTPpassword>

Successfully edited the SMTP configuration
SMTP Host: smtp.example.com
SMTP Port: 2525
SMTP User: resilient
SMTP Password: hidden
SMTP From Email: user@example.com
SMTP From Name: Resilient Incident Management
```

If you want to use an encrypted connection, you must ensure that the SMTP server's certificate is trusted. If unsure, you can follow the instructions in the Importing Untrusted Certificates section of this guide.

After you configure email, you can test the configuration by entering the following command with the options you want to use:

```
sudo resutil smtptest
```

This command has the following options and defaults:

- -help prints the SMTP test configuration help. The default is false.
- -email provides the email address where you want to send the test email message.

The following is an example:

```
sudo resutil smtptest -email joe@example.com
Successfully sent the test email to joe@example.com
```

Email security – defanging URLs

When sending the contents of an artifact within an email notification, any web and IP addresses are automatically “defanged” to prevent the user from inadvertently clicking a malicious link.

The following occurs when URLs are defanged:

- “http” is replaced with “hxxp”
- “ftp” is replaced with “fxp”
- Brackets are added to domain names; for example, www.example.com is replaced with www[.]example[.]com
- Brackets are added to the IP address; for example, 8.8.8.8 is replaced with 8[.]8[.]8[.]8

You may have a number of legitimate domains that you do not wish to be defanged. In this case, you can create a whitelist that allows the specific domains to remain untouched. To see the current setting of the whitelist, enter the following command:

```
sudo resutil configget –key whitelist_defang_domains
```

Use the following command to create the whitelist. For multiple domains, use a comma (,) as a separator.

```
sudo resutil configset –key whitelist_defang_domains -svalue ${domain}
```

The following example adds the example.com and example.org domains to the whitelist:

```
sudo resutil configset –key whitelist_defang_domains -svalue example.com,example.org
```

Chapter 10. Changing ciphers and protocols

The Resilient platform is configured to use the most secure ciphers first, and then any deprecated ciphers as necessary. You can modify the list of ciphers by updating the `co3.properties` file. You can also specify which cryptographic protocols to use, for example, you can ensure that only TLS v1.2 connections are allowed.

About this task

A cipher suite is a collection of cryptographic algorithms that are used to create secure (TLS) internet connections, and to encrypt and verify data that is sent over these connections. The Resilient platform uses TLS cipher suites to establish TLS connections to external hosts such as email and threat information servers.

To view the list of ciphers currently in use, enter the following command. You need to know the IP address of the Resilient platform and you must have `nmap` installed:

```
nmap -p 443 --script ssl-enum-ciphers <ip_address>
```

To change the list of ciphers and to specify cryptographic protocols, complete the following procedure.

Procedure

1. Use an editor to open the `co3.properties` file, located in the `/usr/share/co3/conf/` directory.
2. Add a `resCiphers` variable if necessary.
3. In the variable, specify the ciphers that you want to use in the order in which you want to use them:
   - To use the most secure ciphers, while maintaining backwards compatibility, specify `resCiphers=MOST_SECURE,DEPRECATED_CIPHERS`. This is the default setting.
   - To use a custom cipher followed by the most secure ciphers, specify the custom cipher and then the MOST SECURE variable separated by a comma, for example: `resCiphers=SSL_RSA_WITH_RC4_128_MD5,MOST_SECURE`.
4. To limit the cryptographic protocols that are used and disable older versions of protocols:
   - Add a `resSslEnabledProtocols` variable to the `co3.properties` file, if necessary.
   - In the variable, specify the protocols to use. For example, to limit connections to the TLS V1.2 protocol, specify `resSslEnabledProtocols=TLS1.2`

   Note: Connections from older protocols will be rejected and this might prevent access from older browsers and API programs.
5. Restart the Resilient messaging service, as follows:

   ```
   sudo systemctl restart resilient-messaging.service
   ```
6. Use the `nmap` command again to verify that your changes were made. If not, you need to modify the properties file and try again. Changes might not take effect immediately because of caching.

Chapter 11. Manage Resilient services

The `resilient.service` and `resilient-email.service` are dependent on the `resilient-messaging.service` and cannot function unless it is up and running. Elastic search and Java and Python based integrations are also dependent on the `resilient-messaging.service`.

The `resilient-messaging.service` is a separate service that is used by Resilient email and other Resilient components. The `resilient-messaging.service` is supported only if you have SSL enabled.

To use the `resutil` tool, the `resilient-messaging.service` must be up and running.

If you make changes to the `co3.properties` file, you must restart the `resilient-messaging.service` for the changes to take effect. Restarting the `resilient-messaging.service` also restarts `resilient.service` and `resilient-email.service`.

You can start and stop some services independently. However, `resilient.service` and `resilient-email.service` are dependent on the `resilient-messaging.service` as follows:

- If you stop `resilient-messaging.service`, the `resilient.service` and `resilient-email.service` services are also stopped.
- If you restart the `resilient-messaging.service`, the `resilient.service` and `resilient-email.service` services are also restarted.
- If all three services are inactive and if you start or restart `resilient.service`, `resilient-messaging.service` and `resilient-email.service` are also started or restarted. If all three services are active and if you start or restart `resilient.service`, only the `resilient.service` is started or restarted.

Use the `systemctl` command to manage services.

To check the status of a service:

```
sudo systemctl status <servicename>
```

To stop a service:

```
sudo systemctl stop <servicename>
```

To start a service:

```
sudo systemctl start <servicename>
```

To restart a service:

```
sudo systemctl restart <servicename>
```

Chapter 12. KeyVaults

The KeyVault feature combines all relevant application “secrets” into a single Java keystore.

Combining the secrets into a single place provides the following benefits:

- Provides cryptographic protection for all application secrets.
- Simplifies access control to application secrets.
- Provides a single master key that unlocks all secrets.

By default, the KeyVault password is stored in cleartext; however, you have the option to use an encrypted KeyVault password file as described in Encrypting the KeyVault Password.

Storage format, location and key

The application secrets are stored in a Java JCEKS KeyStore. The following files are relevant, which are in the /crypt/keyvault directory by default.

File	Purpose
keyvault	The Java JCEKS keystore containing all application secrets. Each entry represents a single secret, and is encrypted with the KeyVault password.
.keyvaultpassword	Holds the randomly generated KeyVault password. The permissions are set to minimize who on the system has access to the file.
.keyvaultpassword.gpg	Optional encrypted KeyVault password. If this file exists, the system requires that the user decrypt it when the system starts, and in other cases where it is needed (such as resutil command, system upgrades, etc.). If present, the .keyvaultpassword file is not used and can be removed from the system. The Resilient platforms allows only the .keyvaultpassword.gpg or .keyvaultpassword file. The system uses the gpg command to decrypt this file. See Encrypting the KeyVault Password for additional information.
keys.properties	Configuration file for the KeyVault, if using a keystore other than the default Resilient KeyVault. This is empty by default. See Configuration Options for more information.

Configuration options

KeyVault configuration settings are stored in the `/crypt/keyvault/keys.properties` file. The file is a standard Java properties file, where each line contains a key and value separated by an `=`.

The following table contains the KeyVault configuration options.

Key	Description
passwordfile	Location of the master key file. The default is /crypt/keyvault/.keyvaultpassword
keystorefile	Location of the keystore file. The default is /crypt/keyvault/keyvault
should_backup_password	Whether the backup operation should include the KeyVault password (true) or not (false). Default: true If you do not include the KeyVault password in the backup, you must ensure that it is backed up independently. If the KeyVault password is lost, then all secrets are lost. This may not be a major problem for some secrets, such as the database password since that password can be reset. However, other secrets, such as the attachment encryption key, cannot be recovered.
lockretry_max	Specifies the maximum number of retries to get the KeyVault lock. The default is 25. Increase this value if you see an error like the following in the logs Timeout getting KeyVault file lock ...
lockretry_sleep_msec	Milliseconds to sleep between retries of request for KeyVault lock. The default is 200.

The following is an example that does not back up the password file but does change its location:

```
should_backup_password=false
passwordfile=/some/other/directory/mykeyvaultpassword
```

Note: Contact Resilient Support for assistance if you need to move the stored secrets to your keystore.

Encrypting the KeyVault password

The KeyVault password is stored in an unencrypted file by default (/crypt/keyvault/.keyvaultpassword). This file can be encrypted using GPG to protect it and is decrypted whenever the value is needed. The disadvantage to this approach is that decrypting the file causes a prompt and prevents the Resilient service from being automatically started.

Before you begin

Before you can use this procedure, you must first set up your GPG keypair on the system. Run the following commands and follow the prompts to set up your keypair:

```
sudo mkdir -m 700 /crypt/keyvault/.gnupg
sudo GNUPGHOME=/crypt/keyvault/.gnupg gpg --gen-key
```

You are prompted to select the kind of key you want, keysize, and key validity period. For example:

```
Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
```

```
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y
```

You need to create a user ID to identify your key. You are prompted to enter your Real Name, Email Address, and a Comment to generate this user ID. For example:

```
Real name: Robert Smith
Email address: rsmith@example.com
Comment: Resilient Administrator
You selected this USER-ID:
    "Robert Smith (Resilient Administrator) <rsmith@example.com>"
```

Use a secure passphrase to protect the secret key.

Note: This process requires a lot of random bytes to be generated. Depending on the activity on your system, it could take a long time. You may see a message similar to the following:

```
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 92 more bytes)
```

When completed, you see a message similar to the following:

```
gpg: key 03A371CE marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
pub   2048R/03A371CE 2017-11-13
      Key fingerprint = 9580 0B86 3FAF FBD2 527C  3DE9 AE29 6DD0 03A3 71CE
uid                  Robert Smith (Resilient Administrator) <rsmith@example.com>
sub   2048R/FC0BF124 2017-11-13
```

About this task

You can create multiple user IDs if needed.

After the IDs are created, complete the following steps to encrypt your KeyVault password file:

Procedure

1. If necessary, log on to the system and then run the following command to encrypt the KeyVault password file:

   ```
   sudo useEncryptedKeystore
   ```

2. You are prompted to enter the user ID that you created previously when you encrypted the keyword password file using the gpg command. You can enter multiple user IDs if you have created multiple users. For example:

   ```
   Enter the user ID.  End with an empty line: Robert Smith
   Current recipients:
   2048R/F19B9CFA 2017-11-14 "Robert Smith (Resilient Administrator)
   <rsmith@example.com>"
   ```

   A message displays indicating that you are using an encrypted file for keystore authentication. For example:

   ```
   Switched to using an encrypted file for keystore authentication.
   The Resilient services must now be started manually after a reboot
   ```

3. Reboot the system and then log in again.

4. Unlock the keystore using the following command:

```
sudo resUnlockKeystore
```

   You are prompted to enter the password of the user ID used in step 2.

5. Start the Resilient service:

```
sudo systemctl start resilient.service
```

   The Resilient services are started and users can log on to the application. **Troubleshooting tip**: In some cases, the login page may not be displayed after the Resilient service is restarted. If this happens, you need to manually restart the services individually using the following commands:

```
sudo systemctl start elasticsearch.service
sudo systemctl start resilient-scripting.service
sudo systemctl start resilient.service
```

   To check the status:

```
sudo systemctl status <service_name>.service
```

   Note: When the keyvault password file is encrypted, the Resilient platform does not start automatically if you reboot the system. You need to log in to the Resilient system using SSH, unlock the keystore, and restart the Resilient service as follows:

```
sudo resUnlockKeystore
sudo systemctl start resilient.service
```

   Note: The `resUnlockKeystore` command might not prompt you for the password if you recently provided the password to unlock the KeyVault using another command. For example, you may have provided this password on running a `resutil` command. For convenience, the password is cached for 10 minutes.

   The `resUnlockKeystore` command might not prompt you for the password if you have run any `resutil` command after the reboot. Instead, the prompt is displayed when you run the `resutil` command.

   To subsequently use an unencrypted keyvault password, complete the following steps:

   a. Enter the following command to stop services:

```
sudo systemctl stop resilient-messaging.service
```

   b. Enter the following command to specify the clear text keystore:

```
sudo useCleartextKeystore
```

   c. Enter the following command to start the services:

```
sudo systemctl start resilient.service
```

KeyVault backup

The KeyVault stores all of the passwords used in the Resilient platform. If the KeyVault is lost, it results in a considerable loss of data. For that reason, the Resilient platform writes a backup of the KeyVault files to the system database any time passwords are added or removed, and after each system upgrade. For example, a backup is written to the database when you add a new Threat Source, such as IBM X-Force Exchange.

The default installation includes the KeyVault password in this backup. If the KeyVault password has been encrypted, the encrypted password is backed up.

The net result of this approach is that if you are currently backing up your database, it includes your KeyVault backup. If you choose to NOT backup your KeyVault password (should_backup_password is set to `false` in `keys.properties`), then you must ensure that the KeyVault files are backed up separately.

To restore the most recent backup, use the following command:

```
sudo resutil keyvaultrestore -dir <directory>
```

The `-dir` argument specifies the location where you want to restore the backup. This command restores the backup from the database to the directory that you specified. If the existing KeyVault is lost or corrupted, you can use the backup by renaming the directory to `/crypt/keyvault`. Make sure that the permissions and ownership of the files are the same as the original.

To restore a different backup, you must provide the `-date` argument, which is specified in this format, `yyyy-MM-ddThh:mm:ss`. For example:

```
sudo resutil keyvaultrestore -dir somedir -date  2016-09-26T11:00:00
```

Secrets

Secrets in the KeyVault can be retrieved using the following command:

```
sudo resutil keyvaultget -name <secret name>
```

Similarly, secrets can be saved to the KeyVault using the following command:

```
sudo resutil keyvaultset -name <secret name> -stdin
<secret value>
Ctrl+D (to send an EOF signal to standard input)
```

The following table lists the secrets that you can retrieve from or save to the KeyVault.

Secret	Description
cacerts	Password protecting the CA certs keystore.
custcerts	Password protecting the custcerts keystore.
jms_file	Password protecting the embedded broker keystore.
jms_key	Embedded broker's password.
proxy	Proxy password.
keystore	Tomcat web server certificate's password.

Chapter 13. Configuring maximum image size

You can set the size limit for images that users paste or drag and drop into rich text editors in the Resilient interface. Any individual images that users add in the Resilient interface cannot exceed this file size.

About this task

The default image size limit is 5 MB. You can increase or decrease this size limit, as required.

Procedure

1. To view the current configuration value, if one is set, enter:

   ```
   sudo resutil configget -imagemb
   ```

2. Use the following command to configure the size limit for individual images:

   ```
   sudo resutil configset -imagemb <size in MB>
   ```

   where <*size in MB*> is a numerical value that specifies the maximum allowed size in MB. For example, to set the size limit to 4 MB:

   ```
   sudo resutil configset -imagemb 4
   ```

   To disable image size limit, enter the following:

   ```
   sudo resutil configset -imagemb -1
   ```

   To view the help on this configuration value, enter the following command:

   ```
   sudo resutil configset --help
   ```

Chapter 14. Resilient audit logs

You can configure the Resilient platform to log audit messages for user logins and logouts and for administrative actions taken from the Resilient user interface. Audit logging is disabled by default.

After you enable audit logging, audit log messages are created for all login and logout actions, both successful and unsuccessful. Messages are also logged for administrative create, update, and delete actions taken from the Resilient user interface on users, roles, groups, and workspaces. The messages provided are similar to the Syslog format. The messages show event key/value pairs and values are semi-colon separated. For new actions, the new state is logged. For deleted events, the prior state is logged. For updated actions, the prior state and the new states are logged. All logged events have a message ID. Messages are output to `client.log`. If you have Syslog set up and configured, audit messages are also sent to Syslog.

Actions are logged for Resilient users, external LDAP, SAML, and two factor authentication users. Audit messages are logged for the following actions through the Resilient user interface and REST API:

Login and logout

Audit messages are generated for user logins and logouts on the Resilient user interface for SAML, LDAP, two factor authentication and standard Resilient system users. All login and logout messages have an easily readable message, for example, `User login successful`. The following information is logged:

- All login attempts, successful or unsuccessful, showing the user IP address, user email and ID, and time and date of the login.

  Note: For two factor authentication users, an additional message is logged, showing that two factor authentication is successful.

- A message is also logged for session timeouts and user logouts.

User

- Create user actions.
- Update actions on users, including deactivation and reactivation of users, and updates to user details on the **My Settings** tab, including password changes, but not including changes to the **Notifications** section. Audit messages for password changes are logged for regular Resilient users only.
- Delete user actions.

Note: Actions for SAML and standard Resilient application users produce similar messages. Logging of changes to LDAP users on Active Directory is not managed by the Resilient platform.

API keys

- Create API keys.
- Edit API keys.
- Delete API keys.

Roles

- Create role actions.
- Update role actions.
- Delete role actions.

Groups

- Create group actions.
- Update group actions.
- Delete group actions.

Workspaces

- Create workspace actions.
- Update workspace actions, including changing the default workspace.
- Delete workspace actions.

Message destinations

- Create message destination actions.
- Update message destination actions.
- Delete message destination actions.

Workflows

- Create workflow actions.
- Edit workflow actions.
- Delete workflow actions.

Rules

- Create rule actions.
- Edit rule actions.
- Delete rule actions.

Configuring syslog

If you want the Resilient platform to send audit logging messages to Syslog, you must configure the Syslog service. You can set up Syslog to work for logging on the server on which the Resilient platform is installed (local server) or you can configure Syslog for a remote server.

About this task

To configure the Syslog service for audit logging on the local server, complete a procedure similar to the following example.

For information about how to set up Syslog for logging to a remote server, refer to this Red Hat documentation.

Procedure

1. Open the `rsyslog.conf` file, located in the `/etc` directory, as follows:

```
sudo vi /etc/rsyslog.conf
```

2. Uncomment the following lines in the `rsyslog.conf` file:

```
#$ModLoad imudp
#$UDPServerRun 514
#$ModLoad imtcp
#$InputTCPServerRun 514
```

3. Run the following command:

```
systemctl restart rsyslog.service
```

4. Verify that rsyslog is listening on port number 514 by running the following command:

```
sudo netstat -antup | grep 514
```

Configuring audit logging

You can configure the Resilient platform to send audit log messages to the Resilient `client.log` file and to Syslog, if you have set up and configured Syslog.

About this task

Complete the following steps to enable audit logging on the Resilient platform.

Procedure

1. Start an SSH session to the Resilient system.
2. Enter the following command to enable audit logging:

```
sudo resutil audit -on
```

A confirmation message is displayed, indicating that audit logging was successfully enabled:

```
Command successful. The Audit Logging configuration is as follows
Status: On    Type : Syslog    Host : localhost    Port : 514
```

If you have a remote Syslog server set up, enter an IP address and port number for the remote system:

```
sudo resutil audit -on [-host hostname] [-port port]
```

If you do not specify a host name or port number, the defaults are used. If you do not have Syslog set up, audit messages are logged to `client.log` only.

The changes occur during run time.

To see all options for the `resutil audit` command, enter the following:

```
sudo resutil audit -help
```

Results

Audit messages are logged to the `client.log` file, located in the `/usr/share/co3/logs` directory. If you have configured Syslog, the audit messages are logged to the default Syslog file in `var/log/messages`.

If you want to subsequently disable audit logging, enter the following command:

```
sudo resutil audit -off
```

Sending audit data to Splunk Cloud

You can configure the Resilient platform to send audit logging data to Splunk Cloud. The audit messages that are sent to Splunk are similar to the audit messages that are included in Syslog.

Before you begin

- You must have audit logging enabled on the Resilient platform as described in "Configuring audit logging" on page 45.
- You must have an administrator account on your Splunk Cloud instance.
- You must have the HEC token value for your Splunk deployment.

About this task

Sending audit data to Splunk Cloud depends on Splunk HEC and the network connection between the Resilient platform and your Splunk server.

Complete the following steps to configure the Resilient platform to send audit data to Splunk Cloud.

Procedure

1. Determine the Splunk host name to which you want to send data for your Splunk server deployment. To determine if your deployment is self-service or managed, examine the format of the URL that you use to connect to Splunk Cloud. For more information, see the Splunk Cloud User Manual.
2. If the domain name is not listed on a DNS server, add the IP address and domain name pair to the `/etc/hosts` file, as follows:
   a. Open the `/etc/host` file by entering the following on the command line:

   ```
   sudo vi /etc/hosts
   ```

   b. Add the following line to the `/etc/hosts` file:

   ```
   <IP Address> <hostname>
   ```

   For example:

   ```
   11.22.33.4444 splunk3-01.internal.examplecompany.com
   ```

3. Enable HEC on your Splunk Web application. The steps to enable HEC depend on whether your deployment is self-service or managed. For information about how to enable HEC, refer to the Splunk documentation.
4. Create a new HEC token for the Resilient platform on your Splunk instance. This procedure differs, depending on whether your Splunk deployment is self-service or managed. For more information, refer to the Splunk documentation.
5. Set the Splunk properties on the Resilient platform for the connection to Splunk Cloud using the HEC token value and the host name that you modified in the previous step. To do this, SSH to Resilient platform and enter the following commands in the order shown below. Note that the host name requires a prefix for self-service or managed Splunk Cloud, for example, for self-sevice it is `HOSTNAME = input-<host>` and for managed Splunk Cloud it is `HOSTNAME = http-inputs-<host>`. See this Splunk documentation for more information.

   ```
   sudo resutil configset -key "splunk_audit_connection.is_enabled" -bvalue true
   sudo resutil configset -key "splunk_audit_connection.hostname" -svalue <http-inputs-
   resilient.splunkcloud>
   sudo resutil configset -key "splunk_audit_connection.port" -ivalue <port_number>
   sudo resutil keyvaultset -name "splunk_audit_connection_token" -value <token>
   ```

   When prompted, enter the token value. Then enter the following commands:

   ```
   sudo resutil configset -key "splunk_audit_connection.source" -svalue "Resilient"
   sudo resutilconfigset -key "splunk_audit_connection.source_type" -svalue "Resilient"
   ```

   To verify that the properties are set correctly, enter the following commands:

   ```
   sudo resutil configget -key "splunk_audit_connection.is_enabled"
   sudo resutil configget -key "splunk_audit_connection.hostname"
   sudo resutil configget -key "splunk_audit_connection.port"
   sudo resutil keyvaultget -name "splunk_audit_connection_token"
   sudo resutil configget -key "splunk_audit_connection.source_type"
   sudo resutil configget -key "splunk_audit_connection.source"
   ```

6. Restart the Resilient service, as follows:

   ```
   service resilient restart
   ```

7. Verify that the configuration is successful by searching for `Resilient` audit messages in your Splunk instance. Enter the following search:

```
source="Resilient"
```

Results

The Resilient platform is configured to send audit logging data to Splunk Cloud.

To disable sending audit messages to the Splunk Cloud, enter:

```
resutil configset -key "splunk_audit_connection.is_enabled" -bvalue false
```

Then restart the Resilient service.

What to do next

Monitor audit messages on Splunk. If the Splunk HEC fails to receive or handle an audit log message, for example, if the connection is refused or dropped, or if there is a disabled HEC token, an error might be logged in the Resilient `client.log` file. To detect this, monitor the audit messages on Splunk and if you suspect that there is missing audit data, search the `client.log` for ERROR `com.co3.audit.SplunkOnErrorHandler - Error sending audit log message from Resilient to Splunk`. If required, turn auditing off and on again to force the Resilient platform to establish a new connection to the Splunk server.

Chapter 15. Backup and restore

You can back up the Resilient platform, which consists of the database, keyvault and attachments. You can restore the backup to another Resilient platform with the same version. You can restore a backup from a FIPS-compliant Resilient platform to a FIPS-compliant Resilient platform only.

To complete a backup, ssh to the Resilient platform and run the following command:

```
sudo resSystemBackup
```

This creates a backup in the `/crypt/backups/` folder in the form of a gz file; for example, `resilient-backup-20170426201138.tar.gz`. The timestamp is appended to the file name for uniqueness. You can rename this file for clarity, and move it to a secure location.

The backup file remembers the KeyVault password scheme (cleartext or gpg encrypted as described in KeyVaults). When running a restore on that file, it restores that scheme.

You can encrypt the backup by using the –encrypt option as follows:

```
sudo resSystemBackup --encrypt
```

It is recommended that you store the backup and its corresponding backup_passphrase file to a secure location for future use.

Use the `–help` option to view all the options on the `resSystemBackup` and `resSystemRestore` commands.

To restore a backup, use the `resSystemRestore` command and the name of the backup file. For example:

```
sudo resSystemRestore -f /crypt/backups/resilient-backup-20170426201138.tar.gz

**** POTENTIALLY DESTRUCTIVE BEHAVIOR ****

The existing database will be dropped
Are you sure? YES/NO: YES
[ ok ] Stopping Resilient Application: resilient.
[ ok ] Stopping Resilient Scripting Application: resilient-scripting.
```

If the backup was encrypted, you must supply the passphrase file to restore the system. Also, if the build number of the Resilient platform is different than the one used to create the backup, you have to specify the –c flag to ignore the version. For example:

```
sudo resSystemRestore -f /crypt/backups/resilient-backup-
20170426201138.tar.gz –p backup_passphrase -c
```

If the backup included a protected KeyVault, you need to supply the password of the user, as described in the Encrypting the KeyVault Password section in this guide.

Troubleshooting tip: You might need to reboot the system after using the resSystemRestore command.

Chapter 16. Upgrade Procedure

You can upgrade from a V33.*x* platform on RHEL to V34 on RHEL.

Before you begin

Contact IBM Resilient Customer Support to acquire the software package.

Verify that the following users and groups are available before you begin the upgrade and if necessary create them:

- `res-email` user in a `res-email` group.
- `res-messaging` user in a `res-messaging` group.
- `res-keystore` group.

About this task

Perform the following to upgrade your Resilient platform.

Procedure

Use one of the following commands to install the file the you acquired from IBM using the actual version number (typically in the format x.x.x) in the file name.

During the upgrade, the script automatically backs up the database. To allow the backup, use this command:

```
sudo bash resilient-<version>.run
```

If you do not want to have your database backed up, use these commands:

```
export RES_SKIP_DBBACKUP=1
sudo -E bash resilient-<version>.run
```

Depending on the amount of data, you may experience longer upgrade times. You can monitor the progress by examining the `update_database_`x`.log` as follows, where the x represents the timestamp of the database upgrade:.

```
sudo tail -f /usr/share/co3/logs/update_database_x.log
```

If you have any questions regarding this update, please contact our support team at support@resilientsystems.com.

Results

The upgrade of the Resilient platform proceeds. If the upgrade fails, output similar to the following is displayed:

```
Post-installation setup has failed
Failed to upgrade resilient
To roll back the installation, run the command:
sudo bash /crypt/resRollbackServerUpgrade /crypt/backups/resilient-xxxx.sql.gz
```

To revert to the previous state, enter the following command, which are based on the preceding output:

```
sudo bash /crypt/resRollbackServerUpgrade /crypt/backups/resilient-xxxx.sql.gz
```

Note: After the upgrade has completed, no changes are made to existing roles to add permissions for new features. Master administrators can add permissions for new features to existing roles, as required.