IBM Resilient

Incident Response Platform

ON-PREMISES INSTALLATION AND CONFIGURATION GUIDE v29

Licensed Materials – Property of IBM

© Copyright IBM Corp. 2010, 2018. All Rights Reserved.

US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Resilient Incident Response Platform On-Premises Installation and Configuration Guide

Platform Version	Publication	Notes
29.4	March 2018	Added step about stopping integrations during a backup.
29.3	February 2018	Added information about using a static IP address.
29.2	January 2018	Addressed changes in the V29.2 release.
29.1	December 2017	Corrected minor grammatical errors.
29.0	November 2017	Initial publication.

Table of Contents

1. Introduction

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

If your organization experiences a malware outbreak, system intrusion, or advanced blended attack, the Resilient platform instantly generates detailed dynamic playbooks and provides a platform for team member investigation, collaboration, and reporting. The Resilient platform turns a privacy breach response from a lengthy, tedious, expensive, and stressful process to one that is easily monitored and always up-to-date. Based on an industry-leading knowledgebase of privacy-related regulations with breach notification requirements, the Resilient platform instantly gives you a platform for breach preparation, assessment, and management.

1.1. Prerequisites

The Resilient appliance is a self-contained server that runs the Resilient platform. The server contains Tomcat and PostgreSQL. The following lists the default Resilient appliance configuration, which you can modify during deployment:

- Two CPUs
- Eight gigabytes (GBs) of memory
- 100GB thin provision disk

The Resilient appliance is provided as a VMware virtual application (vApp) in Open Virtualization format (.ova file). The VMware image is based on Red Hat Enterprise Linux (RHEL). The Resilient appliance runs on VMware vSphere Hypervisor (ESXi) 6.0 U2 or later. The Resilient appliance requires that the host for the appliance is on a network that is accessible by SSH (for administration access) and by web browser for user access.

IMPORTANT: Previous releases of the Resilient platform were based on the Turnkey Linux project, which is a Debian Linux derivative. If you wish to migrate to the RHEL based platform, refer to the Migrate to the RHEL Platform section in this guide.

1.2. Installation Overview

This document guides you through the following steps to install the Resilient platform:

1. Deploy the appliance and perform first boot configuration.
2. Connect through PuTTY/ssh.
3. Install the Resilient license.
4. Perform system updates.
5. Configure a Secure Socket Layer (SSL) certificate.
6. Create an initial user account.
7. Configure authentication.
8. Add additional Resilient user accounts.
9. Test the system configuration by connecting to the Resilient platform.
10. Configure the appliance to access specific URLs on the Internet.
11. Configure the log file.
12. Configure email notifications.
13. Configure the KeyVault.

The entire installation should take about an hour to complete.

1.3. Getting Started

Before you start the installation, make sure that you have the following:

- Resilient appliance ova file downloaded from the web site.
- Hypervisor credentials required to create virtual instances.
- VMware vSphere client.
- An IP address for the Resilient appliance, only if you want to use a static IP address.
- An SSH client, such as PuTTY, to connect to the Resilient appliance.
- Certificate authority (CA) that signs the Resilient appliance SSL certificate. This may be an internal CA used within your company or a third party CA such as Verisign or Thawte.
- SMTP server and credentials that the Resilient appliance can use to send email notifications.
- Resilient license.

2. Deployment

This section provides the procedure to deploy and install the Resilient appliance. After the deployment steps and your system reboots, the first boot script starts automatically and installs the appliance. This process may take a few minutes.

IMPORTANT: Do not cancel the first boot script or reboot the system until after the installation is complete; otherwise, you may need to re-import the ova file or, if you cancel the script before a root password is set, you may not be able to access the system.

During the first boot script, you will be asked to provide passwords for root and resadmin accounts, and to verify or adjust the VM's network configuration. Once completed, the first boot script is removed and cannot be run a second time.

Perform the following to deploy and install the resilient appliance.

1. Open the VMware vSphere client.
2. Select **File>Deploy OVF Template**.
3. Browse to the location of the ova file you downloaded from the IBM Resilient web site and select the ova file.
4. When prompted, enter a name for the deployed template.
5. When prompted by the Disk Format screen, select the Thick Provision Lazy Zeroed, Thick Provision Eager Zeroed, or Thin Provision option. If you choose one of the thick provisions, the appliance initially begins using 100 gigabytes of storage. If you choose Thin Provision, the appliance initially begins using a smaller amount of storage and grows over time.
6. In the Ready to Complete screen, select the **Power on after Deployment** option and click **Finish**. A status bar appears during deployment. VMware notifies you when the appliance successfully deploys.
7. Click **Close** in the Deployment Completed Successfully dialog box.

   After the reboot, the first boot script starts automatically and installs the appliance. This may take a few minutes.

8. When prompted, enter a root password then enter a password for the resadmin account. For security, you should use a strong password with at least four character classes (e.g., lowercase, uppercase, digits, symbols, etc.).

9. When prompted, review the network settings. You can choose to accept the default configuration or modify it. For example, you may wish to use a static IP address instead of using DHCP.

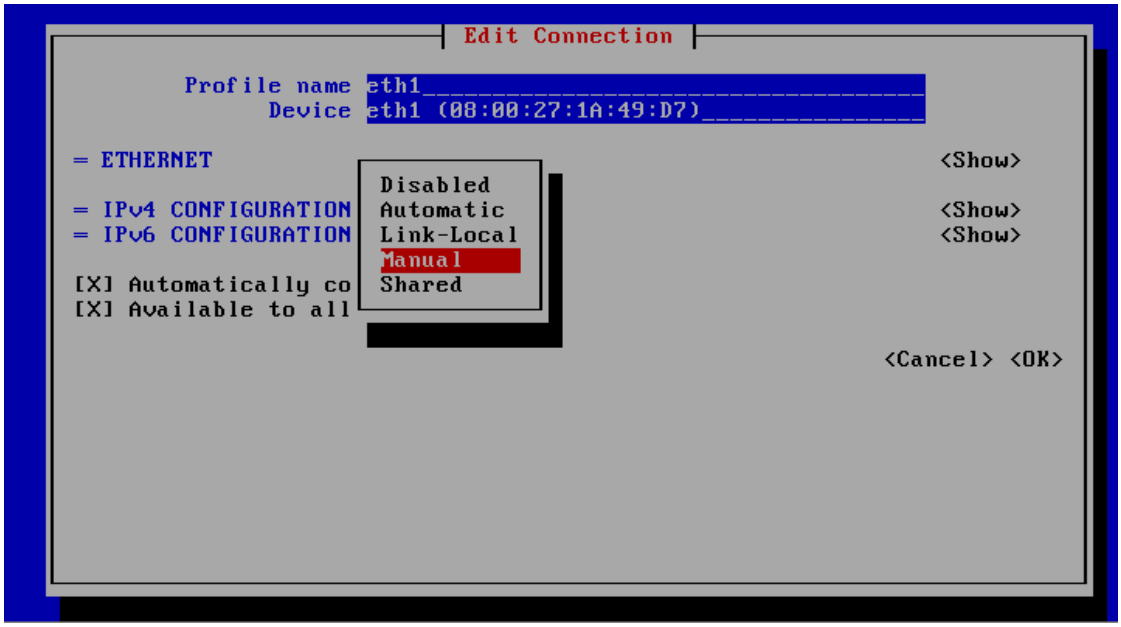
10. Perform the following if you wish to modify the settings:

    a. Start the NetworkManager text user interface (nmtui) tool by issuing the following command as root:

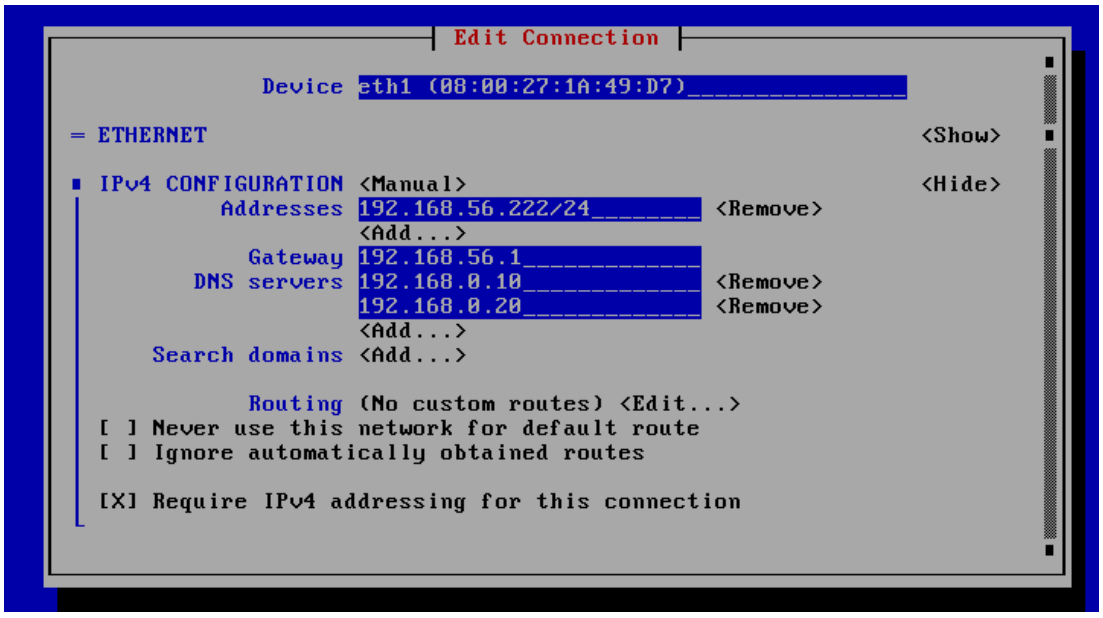
    ```
    #nmtui
    ```

    b. To navigate, use the arrow keys or press Tab to step forwards and press Shift+Tab to step back through the options. Press Enter to select an option. The Space bar toggles the status of a check box. Edit the connection you wish to modify:

c. To use a static IP address, select the configuration and select Manual.



d. Enter the IP address you want to use and your Domain Name Servers as shown below. Make sure that the host name resolves to that IP address. Note that the Resilient appliance does not automatically register the IP address in DNS. Also, enter the IP address in your Domain Name Server (DNS).



e. Save your settings and reboot the appliance.

NOTE: If using a static IP address and DNS is unavailable, you can append a mapping of the IP address to /etc/hosts. Otherwise, you may see performance issues, such as slow service startup or command execution, if the host name is not resolved to an IP address.

The appliance reboots automatically after you complete the network settings.

3. Connection and Updates

This section provides the procedures to import the Resilient license, connect to and update the Resilient appliance, and set the time zone.

3.1. Importing the Resilient License

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 appliance using an SSH client, such as PuTTY. To import the license:

1. Copy the license file (e.g., Reslient_License.lic) that you received from IBM Resilient to the Resilient appliance.

2. Log in to the appliance using SSH as the resadmin user account you created in the previous section. You can use PuTTY or connect from a terminal client as follows:

   ```
   ssh resadmin@<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.

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

3.2. Updating the Resilient Appliance Software

Perform the following procedure to make sure you have the latest Resilient software patch for this version. This step is needed if Resilient has released an update since the time you downloaded the ova file.

1. Enter the following command in your SSH client to connect to the Resilient appliance. The command downloads a resilient-<*version*>.run file.

```
wget --trust-server-names
https://repo.co3sys.com/public/6c69b7e4327a13b52033a54dbb210651/latest.php
```

2. Enter the following command to install the file. Use the actual version number (typically in the format *x.x.x*) in the file name.

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

3. Reenter the resadmin password when the system prompts you.

4. Restart the Resilient service by entering the following command:

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

The Resilient appliance is now installed and running.

3.3. Setting the Time Zone

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

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 service by entering the following command:

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

4. SSL Certificate

The Resilient appliance comes with a self-signed Secure Socket Layer (SSL) certificate. However, we do not recommend 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.

4.1. Creating and Submitting the Certificate Request

To create a certificate request:

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-----".

   TIP: 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.

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.

4.2. Importing the Signed Certificate

To import the signed certificate, copy the certificate file to your Resilient appliance 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 service by entering the following command:

```
sudo systemctl restart resilient.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.

5. Accounts and Authentication

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.

5.1. Creating the Initial Resilient User Account

Enter the following command in your SSH client 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.
- -role assigns an existing role to the user.

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.

5.2. 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.

You should have at least one master administrator or equivalent account per organization whose email address is not managed by the Active Directory. Once 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.

5.2.1. 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 *Resilient Incident Response Platform Master Administrator Guide*.

5.2.2. 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>
```

5.2.3. 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.

5.2.4. Additional Information

The following lists other helpful ldapedit commands:

- View the command help:

```
sudo resutil –help
```

- 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"
```

  Once 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.

5.3. 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.

5.3.1. Create a SAML Federation

SAML federations are created in the Resilient platform using the resutil tool. In order 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.

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 appliance 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 -loginurl https://adfs.example.com/adfs/ls/
```

- If your identity provider has set up single logout functionality, you can have the Resilient appliance 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 appliance verifies the signature on all incoming identity provider messages. If an incoming message is not signed, the Resilient appliance rejects the message. To have the Resilient appliance 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 -loginurl 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.

5.3.2. 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 Identity Provider. 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.

5.3.3. 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 wish to restrict access, you must do so through the Identity Provider configuration.

5.4. 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.

5.4.1. 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. If you are a SAAS customer, contact support@resilientsystems.com to complete your setup. Afterwards, proceed to Enabling your authentication domain.
4. For an on-premises deployment, follow the instructions below to configure two-factor authentication with the Resilient platform.
5. Locate the Integration Key, Secret Key, and API hostname from your Duo application.
6. 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.)

7. 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
```

5.4.2. 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.

5.4.3. 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.

5.4.4. 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.

5.5. 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 *Resilient Incident Response Platform Master Administrator Guide*, included in the Resilient platform, for more information on adding additional Resilient user accounts.

5.6. Importing Untrusted Certificates

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

Basically, you perform the following steps to trust the certificates:

- Obtain the certificate from the service you need to trust (e.g., 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 appliance. It contains customer specific certificates for communication with external systems, e.g., SMTP servers, custom threat feeds, etc. 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.

1. Obtain the certificate from the service. You may request it from the administrator of that service or use the openssl utility installed on the Resilient appliance 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 may be required for SMTP servers.

2. Check that the /crypt/certs/custcerts directory exists.

   ```
   sudo ls -al /crypt/certs/custcerts
   ```

3. If it does exist, verify that you can list the contents:

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

   You need to enter the keystore password in order 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 need to 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 –value <Password for custcerts>
   ```

6. Network Configuration

In order for some Resilient appliance functions to operate properly, the system requires access to services on the Internet. Work with your network administrators to ensure the appliance 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

7. 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 appliance 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
```

8. 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 hostname 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 hostname 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 wish to use an encrypted connection, you need to 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 edit 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
```

8.1. 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. Specifically, this means the following:

- “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 configget -key whitelist_defang_domains -svalue ${domain}
```

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

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

9. KeyVaults

Before version 28.0, the secrets were stored in various locations. 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.

9.1. 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.

9.2. 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. Default: /crypt/keyvault/.keyvaultpassword
keystorefile	Location of the keystore file. Default: /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.

The following is an example that does not backup 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.

9.3. 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 can use this procedure, you must first set up your GPG keypair on the appliance. 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
```

NOTE: If you are using the V29 version of the Debian-based Resilient platform, use these commands to set up your GPG keypair:

```
sudo mkdir -m 700 /crypt/keyvault/.gnupg
sudo gpg --homedir /crypt/keyvault/.gnupg --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)
```

Once 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
```

You can create multiple user IDs if needed. Once the IDs are created, perform the following steps to encrypt your KeyVault password file:

1. Stop the Resilient service:

   ```
   sudo systemctl stop resilient.service
   ```

2. Encrypt the KeyVault password file by running the following command:

   ```
   sudo useEncryptedKeystore
   ```

3. You are prompted to enter the user ID 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
   ```

4. Unlock the keystore using the following command:

   ```
   sudo resUnlockKeystore
   ```

   You are prompted to enter the password of the user ID selected in the previous step.

5. Restart the Resilient service. You are prompted to enter the password for the user which you encrypted the file with from the previous step.

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

NOTE: The Resilient platform does not start automatically if you reboot the virtual appliance. You need to log in to the appliance using SSH, unlock the keystore, and restart the Resilient service as follows:

```
sudo resUnlockKeystore

sudo systemctl restart resilient.service
```

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 need to 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
```

If you are using the Turnkey Debian appliance, the corresponding commands are:

```
sudo service <service_name> status

sudo service elasticsearch start

sudo service resilient-scripting start

sudo service resilient start
```

NOTE: The resUnlockKeystore command may 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 may 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.

You can switch the KeyVault back to using the unencrypted password by stopping the Resilient service and running the following command. This prompts you to enter the password of the user ID which was used to encrypt the GPG file. Make sure to restart the Resilient service afterwards.

```
sudo useCleartextKeystore
```

9.4. KeyVault Backup

The KeyVault stores very important data that, if lost, would cause a considerable loss of data. For that reason, it must be backed up. The default installation of the Resilient platform writes a backup of the KeyVault files to the system database any time the KeyVault changes and after each system upgrade.

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 specifies where to restore the backup. To restore a different backup, you need to supply 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
```

9.5. 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> -value <secret value>
```

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.

10. 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 virtual appliance running the same version.

Perform the following to back up the platform:

1. Before starting the backup, make sure to stop any integration packages, such as QRadar, and perform a backup of their databases.
2. Use ssh to connect to the appliance.
3. Run the backup 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 filename 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.

Perform the following to restore the platform:

1. Use ssh to connect to the appliance.
2. Enter 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.
```

3. If you have any integration packages, make sure you also restore its backup from the same time. The Resilient platform and any integration database backups should be in a consistent state. Refer to the integrated app's documentation for the backup procedure.

If the backup was encrypted, you need to 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. For the Debian based platform, you need to enter this password multiple times to restart the services used by the Resilient platform.

Troubleshooting tip: You may need to reboot the appliance after using the resSystemRestore command.

11. Upgrade Procedure

You can upgrade from a V28.*x* platform. If you are using an older version of the Resilient platform, you must first upgrade to version 28.

You cannot upgrade a Resilient platform on a Debian platform to the Resilient platform VMware image based on the RHEL platform. Instead, you can migrate to the RHEL version. See the Migrate to the RHEL Platform section in this guide.

If your existing Resilient platform is on the Debian platform, you can only upgrade your platform to version 29. You cannot perform a new installation.

Perform the following to upgrade your Resilient platform.

1. Run the following command from the Resilient appliance to download a resilient-<*version*>.run file.

   ```
   wget --trust-server-names
   https://repo.co3sys.com/public/CCA1A8F2BFB071A5D560601326CE4DE0/latest.php
   ```

   If your appliance does not have internet access, run the wget command above from another system then copy the downloaded file to your appliance.

2. Use one of the following commands to install the file 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 wish to have your database backed up, use these commands:

   ```
   export RES_SKIP_DBBACKUP=1

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

If you perform a backup, you should also back up the database of any integrated applications, such as QRadar.

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. The x represents the build number of the new version.

```
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.

12. Migrate to the RHEL Platform

You cannot upgrade to the RHEL based release of the Resilient platform, but you can migrate your platform's settings to a new installation of the platform.

Perform the following procedure to migrate your Resilient platform to the V29 RHEL based Resilient platform:

1. If not done already, upgrade your Resilient platform to V29, as described in the Upgrade Procedure section of this guide.
2. Perform a backup of the Resilient platform using the procedure in the Backup and Restore section in this guide.
3. At the new installation of the RHEL based platform, use the restore command to update the platform with your custom settings:

```
sudo resSystemRestore -f /crypt/backups/resilient-backup-<timestamp>.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.
```

4. Verify that the restore was successful by logging in to the Resilient platform and reviewing the various settings.

Troubleshooting tip: You may need to reboot the appliance after using the resSystemRestore command.

13. Debian Only Configuration

This section applies only to the Resilient platform based on Debian Linux. Only the Debian based Resilient platform performs security updates automatically. The RHEL based platform does not.

The Debian based Resilient appliance automatically performs security updates over the network every night at 4:00 a.m. local time. These updates come directly from the Turnkey Linux and Debian projects. IBM Resilient monitors security fixes to ensure that they do not interfere with the operation of the Resilient appliance. Automatic updates can introduce some risk because they can occasionally, compromise existing functionality. However, the Turnkey and Debian projects mitigate this risk by carefully separating out security fixes and only accepting the minimum possible fix required to resolve security issues. If you wish to disable automatic security updates, you can do so by entering the following command from the Resilient appliance console:

```
sudo rm /etc/cron.d/cron-apt
```

To perform the equivalent update manually, enter the following command:

```
sudo /usr/sbin/cron-apt
```

To support the updates, make sure that the Resilient appliance can access the following URLs.

http://archive.turnkeylinux.org	Turnkey Linux software updates
http://security.debian.org	Debian Linux security software updates
http://cdn.debian.net	Debian Linux standard software updates