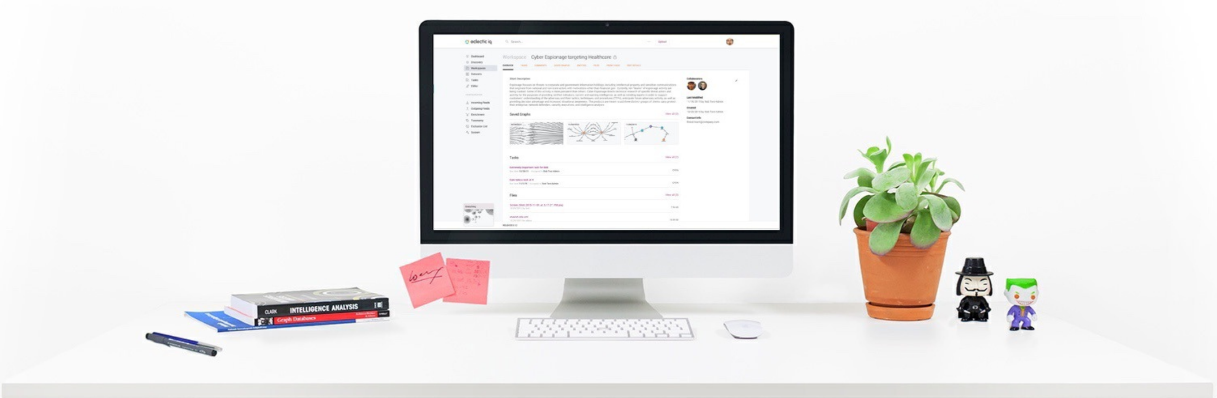
eclectic iq

EclecticIQ Platform troubleshooting

Solve common issues and hiccups

Last generated: March 06, 2018



Last generated on Mar 6, 2018

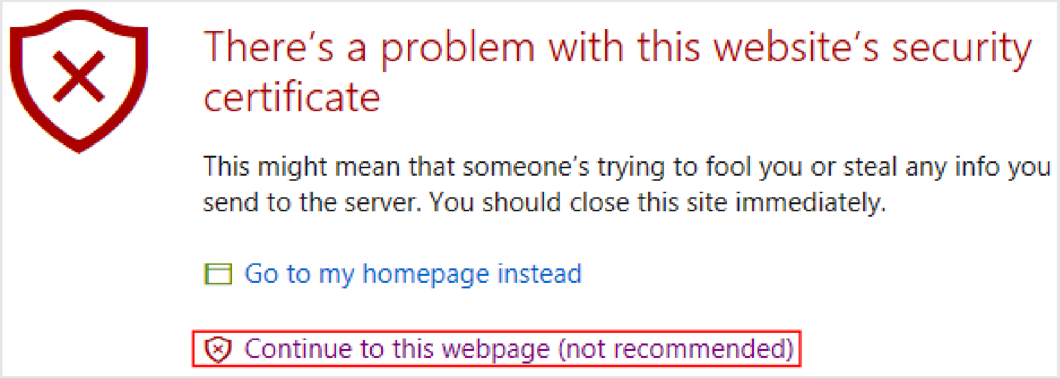
Table of contents

Web browser certificate error

Some web browsers, for example Microsoft Edge, may display a certificate error and prevent users from signing in to the platform.

Scenario

- The OS is Microsoft Windows 10 or later.
- The web browser in use is Microsoft Edge.
- When trying to reach the platform login page, MS Edge displays a certificate error page.



There's a problem with this website's security certificate

This might mean that someone's trying to fool you or steal any info you send to the server. You should close this site immediately.

Go to my homepage instead
Continue to this webpage (not recommended)

Issue

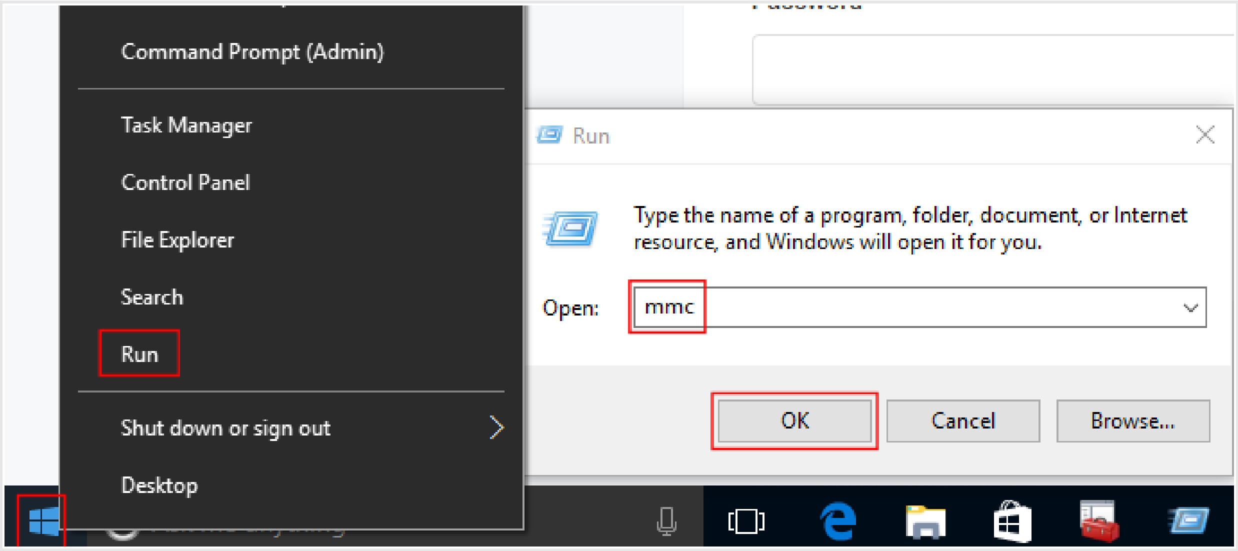
When you continue and land on the platform login page, it is not possible to successfully sign in and access the platform.

This problem can occur when a web browser does not accept a third-party or a self-signed certificate. The EclecticIQ Platform uses a self-signed certificate.

Mitigation

Manually install the platform self-signed certificate to the **Trusted Root Certification Authorities Certificate Store** in Windwos 10:

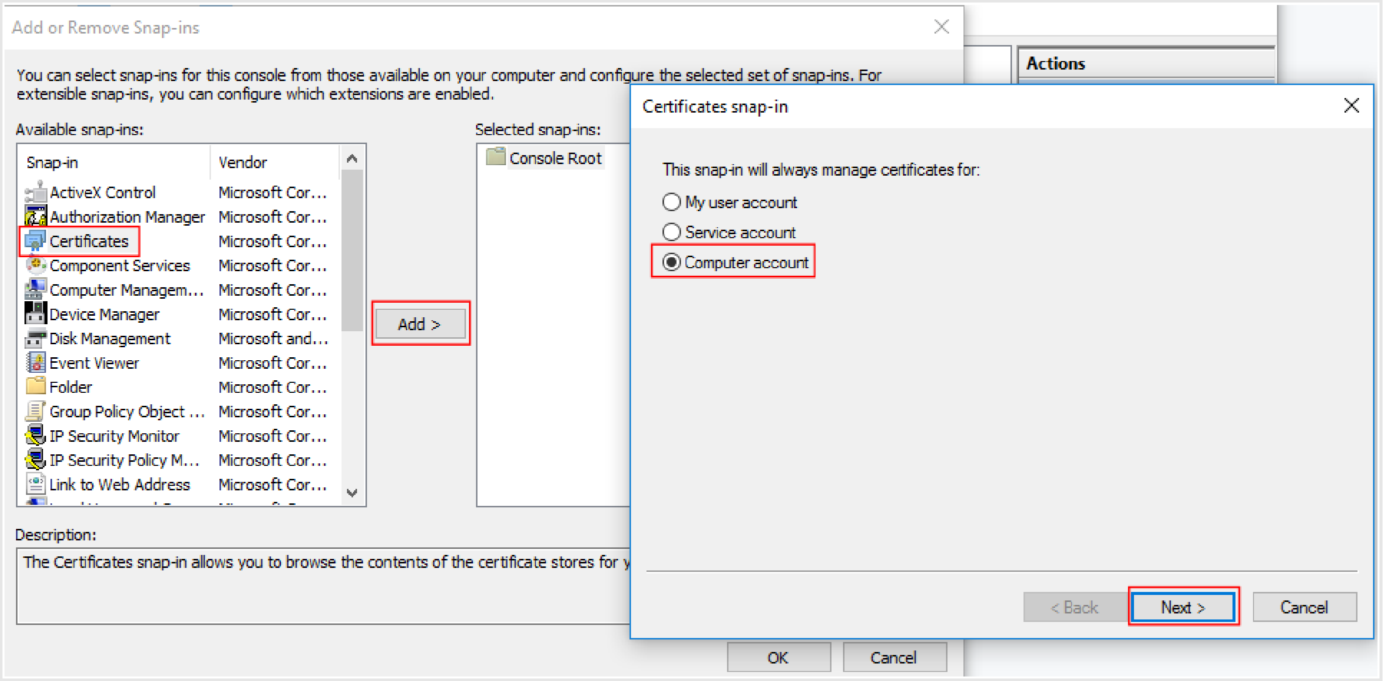
- Make sure you have a local copy of the platform self-signed *.crt* certificate. Contact our support people for any assistance or to request a new copy of the certificate.
- In Microsoft Windows 10, right-click the **Start** button.
- From the pop-up context menu, select **Run**.
- In the **Open** input field on the **Run** dialog window, enter *mmc*, and then click **OK**.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: mmc

- In the Microsoft Management Console main window, select **File > Add/Remove Snap-in…**.
- In the **Add or Remove Snap-ins** dialog window, under **Available snap-ins** select **Certificates**, and then click **Add**.
- In the **Certificates snap-ins** dialog window, select the **Computer account** radio button, and then click **Next**.



You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy M...	Microsoft Cor...
Link to Web Address	Microsoft Cor...

Selected snap-ins:

Console Root

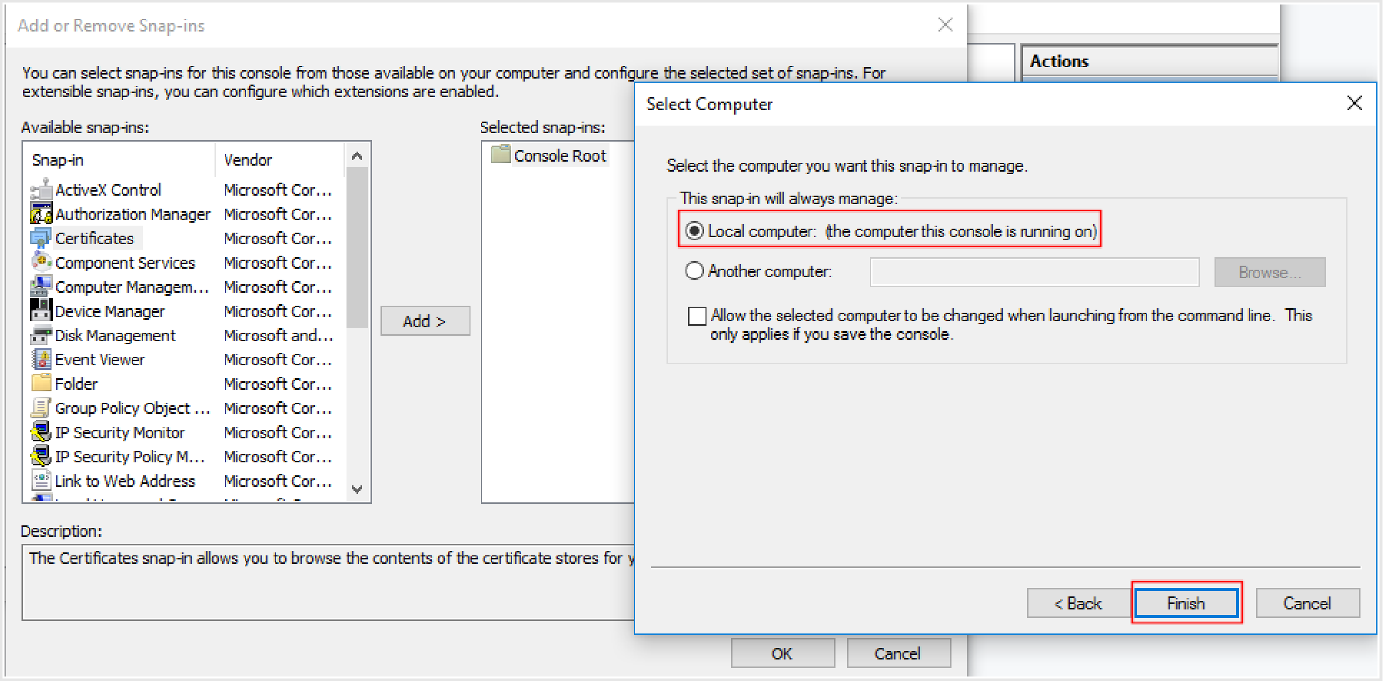
Description:

The Certificates snap-in allows you to browse the contents of the certificate stores for y

This snap-in will always manage certificates for:

○ My user account
○ Service account
● Computer account

- In the **Select Computer** dialog window, select the **Local computer** radio button to install the certificate on the current machine, and then click **Finish**.



You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy M...	Microsoft Cor...
Link to Web Address	Microsoft Cor...

Selected snap-ins:

Console Root

Description:

The Certificates snap-in allows you to browse the contents of the certificate stores for y

Select the computer you want this snap-in to manage.

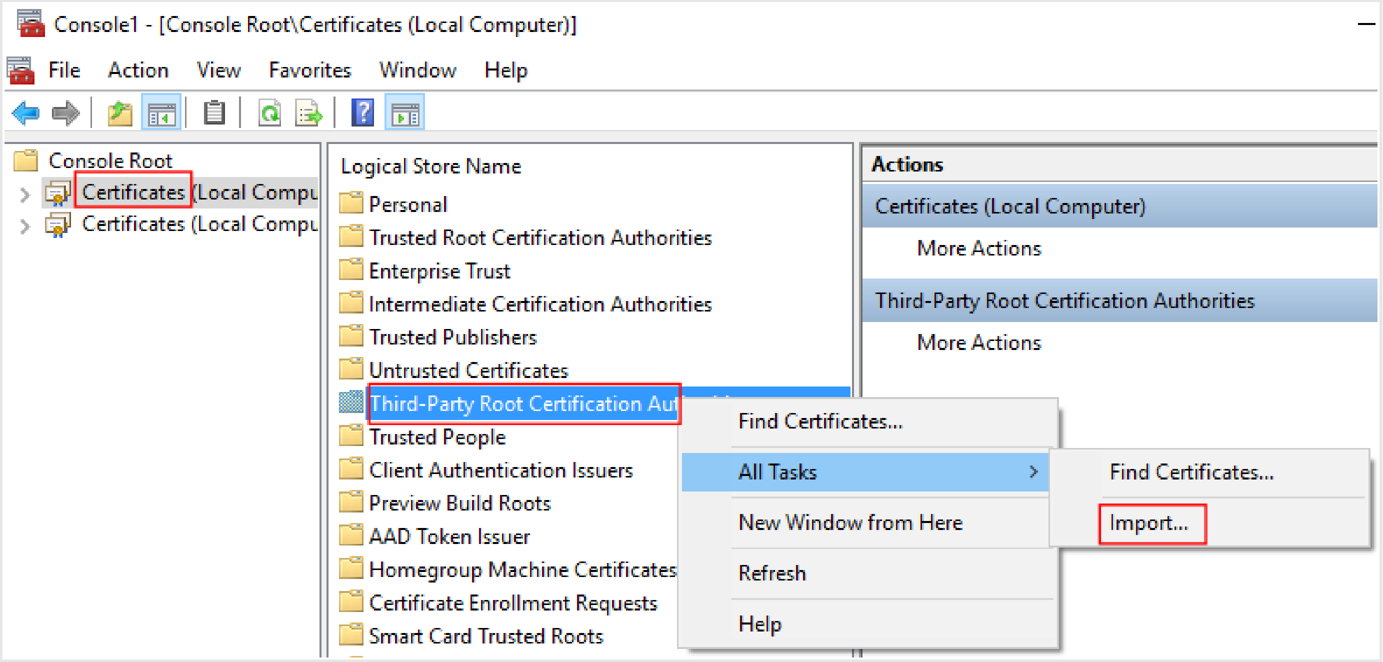
This snap-in will always manage:

● Local computer: (the computer this console is running on)

○ Another computer:

☐ Allow the selected computer to be changed when launching from the command line. This only applies if you save the console.

- In the Microsoft Management Console main window, click **Certificates** to expand the **Logical Store Name** list.
- Under **Logical Store Name**, right-click **Trusted Root Certification Authorities**, and then select **All Tasks > Import** from the context menu.


Console1 - [Console Root\Certificates (Local Computer)]
File
Action
View
Favorites
Window
Help
Console Root
Certificates (Local Compu
Certificates (Local Compu
Actions
Certificates (Local Computer)
More Actions
Third-Party Root Certification Authorities
More Actions
Find Certificates...
All Tasks
New Window from Here
Refresh
Help
Find Certificates...
Import...

Logical Store Name

- Personal
- Trusted Root Certification Authorities
- Enterprise Trust
- Intermediate Certification Authorities
- Trusted Publishers
- Untrusted Certificates
- Third-Party Root Certification Aut
- Trusted People
- Client Authentication Issuers
- Preview Build Roots
- AAD Token Issuer
- Homegroup Machine Certificates
- Certificate Enrollment Requests
- Smart Card Trusted Roots

The **Certificate Import Wizard** starts. It guides you through the import operation.
Make sure you choose the correct certificate to import and the appropriate certificate store:

- Select the platform certificate when the wizard prompts you for a certificate to import.
- Import the certificate to the **Trusted Root Certification Authorities** certificate store.

Errno 111 Connection refused

It is possible to sign in to the platform web-based GUI, tasks are running, but the platform log file returns a connection error when retrieving Supervisor-related system health information from the backend.

Scenario

- It is possible to sign in to the platform web-based GUI. Tasks are running, but the `platform-api.log` file reports a connection error when trying to retrieve system health information from Supervisor. On the system health monitor, Supervisor-managed tasks on the **Processes** tab are down and flagged with a red dot.

Issue

- The `/var/log/eclecticiq/platform-api.log` log file reports an `[Errno 111] Connection refused` error message.
- The platform and its tasks are up and running. However, it is not possible to retrieve system health information from Supervisor via HTTP.

Mitigation

Possible causes:

- Incomplete or incorrect platform server configuration (for example by assigning an erroneous server name or listening port)
- Firewall settings block communication between the platform and Supervisor
- The port specified in the platform server configuration is not open
- Missing properties in the Supervisor and in the platform configuration files.

Check the platform system configuration to identify and fix any incorrect or missing parameters, as well any active firewall rules to make sure they are allowing communication between the platform and Supervisor.

Check the log

Have a look at the platform log file with the error message. If the error is recent, it is probably at the end of the log file. To view it, run the following command(s):

```
$ tail -1 /var/log/eclecticiq/platform-api.log
```

The command returns the bottom end of the log file content. The error message can look like the following example:

```
{
  "cause": "[Errno 111] Connection refused",
  "event": "request.failed.internal",
  "exception": "Traceback (most recent call last):\n [...] File
\"/usr/lib64/python3.4/socket.py\", line 503, in create_connection\n sock.connect(sa)\n
ConnectionRefusedError: [Errno 111] Connection refused",
  "level": "error",
  "logger": "eiq.platform.error_handling",
  "request_method": "GET",
  "request_path": "/private/status",
  "timestamp": "2016-07-22T12:44:45.090774Z",
  "user": ""
}
```

Call Supervisor

Typically, you can reach Supervisor by making a call to `http://localhost:9001` or to `http://127.0.0.1:9001`.

cURL call example:

```
$ curl http://localhost:9001
```

Check the Supervisor configuration

Verify that the *supervisord.conf* file includes the following sections enabling the `supervisord` daemon to respond to requests from the platform:

- `[inet_http_server]` **section** `(http://supervisord.org/configuration.html#inet-http-server-section-settings)` to define the Supervisor host.
  To enable `inet_http_server`, *supervisord.conf* should contain the following properties under `[inet_http_server]`:

```
[inet_http_server]
port=127.0.0.1:9001
```

- `[rpcinterface:supervisor]` **section** `(http://supervisord.org/configuration.html#rpcinterface-x-section-settings)` to allow communication between the Supervisor XML-RPC interface and the platform API to retrieve status information.
  To enable `rpcinterface`, *supervisord.conf* should contain the following properties under `[rpcinterface:supervisor]`:

```
[rpcinterface:supervisor]
supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinterface
```

Check the platform configuration

Last generated on Mar 6, 2018

SELinux is not enabled or not installed

After installing the platform, a script fails to set the SELinux security labels for the platform files.

> If you are not using SELinux and are not planning to implement it in the environment where the platform is installed, you do not need to do anything and you can safely disregard this section.

Scenario

- The platform has been installed on the target system, but it is not yet configured or bootstrapped.
- SELinux is either installed, but is not enabled, or it is not installed.
- The post-install platform script does not set the SELinux security labels to platform files in the */opt/eclecticiq* directory.

Issue

Check SELinux status

If SELinux is installed, check if it is enabled or disabled. Run the following command(s):

```
$ sestatus -v
```

If SELinux is disabled, the response includes the following line:

```
SELinux status: disabled
```

Check SELinux mode

You can check also which SELinux mode is currently active. Run the following command(s):

```
$ getenforce
```

The allowed modes are `enforcing`, `permissive`, and `disabled`.
The active mode may not be the same as the `SELINUX` value defined in the SELinux global configuration file:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

This can happen after changing and saving **SELinux global configuration file** `(https://selinuxproject.org/page/configurationfiles)`, and before executing a system reboot for the changes to become effective.

Set SELinux to permissive mode

The recommended SELinux mode to offload complexity during installation and configuration is `permissive`.
To set SELinux to work permissively, run the following command(s):

```
$ setenforce permissive
```

Post-installation check

The platform post-install script included in the RPM package attempts to automatically set the appropriate SELinux security labels for the files that are deployed to */opt/eclecticiq*.

- If SELinux is not installed or if it is disabled, the post-install script included in the RPM install package does not attempt to configure any SELinux file security labels for the files that are deployed to */opt/eclecticiq*.
- If SELinux is installed and it is enabled, and if the platform post-install script does not set the SELinux security labels to the applicable platform files, run the following command(s):

```
$ semanage fcontext -a -t var_log_t -f d "/opt/eclecticiq"
```

- If SELinux policy-related errors occur, the command returns a response that can be similar to this example:

```
SELinux:  Could not downgrade policy file /etc/selinux/targeted/policy/policy.29, searching for
an older version.
SELinux:  Could not open policy file <= /etc/selinux/targeted/policy/policy.29:  No such file or
directory
/sbin/load_policy:  Can't load policy:  No such file or directory
libsemanage.semanage_reload_policy: load_policy returned error code 2.
```

The response provides more context about the affected files and the reasons why it was not possible to set the security labels.

Mitigation

SELinux is not installed

If SELinux is not installed on the target system, do the following:

- After completing the platform installation, install and enable SELinux.
- To set the correct security contexts, execute the following script:

```
BASE_PATH="/opt/eclecticiq"

if [ -x "$(command -v semanage)" ]; then

    SELINUX_MODE=$(getenforce)

    if ! [ $SELINUX_MODE == "Disabled" ]; then

        semanage fcontext -a -t etc_t "$BASE_PATH/etc(/.*)?"
        semanage fcontext -a -t etc_t "$BASE_PATH/etc-extras(/.*)?"

        semanage fcontext -a -t httpd_config_t "$BASE_PATH/etc/nginx(/.*)?"
        semanage fcontext -a -t httpd_config_t "$BASE_PATH/etc-extras/nginx(/.*)?"

        # By default, newly created files and directories inherit the SELinux type
        # of the corresponding parents, so that log files have the correct type.
        # However, we do not want to relabel existing logs.
        semanage fcontext -a -t var_log_t -f d "$BASE_PATH/logs"

        restorecon -RF $BASE_PATH

        echo "SELinux security labels configured."
    else
        echo "SELinux is not enabled. Security labels won't be configured."
    fi
else
    echo "SELinux is not installed. Security labels won't be configured."
fi
```

- You may need to reboot the system for the changes to become effective.

SELinux is installed but it is not enabled

If SELinux is installed on the target system but it is not enabled, do the following:

- Enable SELinux, either by editing its configuration file, and then by rebooting the system, or by running one of the following commands:

```
# Set SELinux to permissive mode
$ setenforce 0

# Set SELinux to enforcing mode
$ setenforce 1
```

- Create the following bash script:

```
BASE_PATH="/opt/eclecticiq"

if [ -x "$(command -v semanage)" ]; then

    SELINUX_MODE=$(getenforce)

    if ! [ $SELINUX_MODE == "Disabled" ]; then

        semanage fcontext -a -t etc_t "$BASE_PATH/etc(/.*)?"
        semanage fcontext -a -t etc_t "$BASE_PATH/etc-extras(/.*)?"

        semanage fcontext -a -t httpd_config_t "$BASE_PATH/etc/nginx(/.*)?"
        semanage fcontext -a -t httpd_config_t "$BASE_PATH/etc-extras/nginx(/.*)?"

        # By default, newly created files and directories inherit the SELinux type
        # of the corresponding parents, so that log files have the correct type.
        # However, we do not want to relabel existing logs.
        semanage fcontext -a -t var_log_t -f d "$BASE_PATH/logs"

        restorecon -RF $BASE_PATH

        echo "SELinux security labels configured."
    else
        echo "SELinux is not enabled. Security labels won't be configured."
    fi
else
    echo "SELinux is not installed. Security labels won't be configured."
fi
```

- Save it, make it executable, and then run it.
- You may need to reboot the system for the changes to become effective.

Broken dashboard gauges

The platform dashboard is loaded, but one or more graph gauges are broken or they are displayed incorrectly.

Scenario

- An EclecticIQ Platform installation with Kibana as the UI frontend to display graph data.
- Kibana gauges on the platform dashboard are broken.

Issue

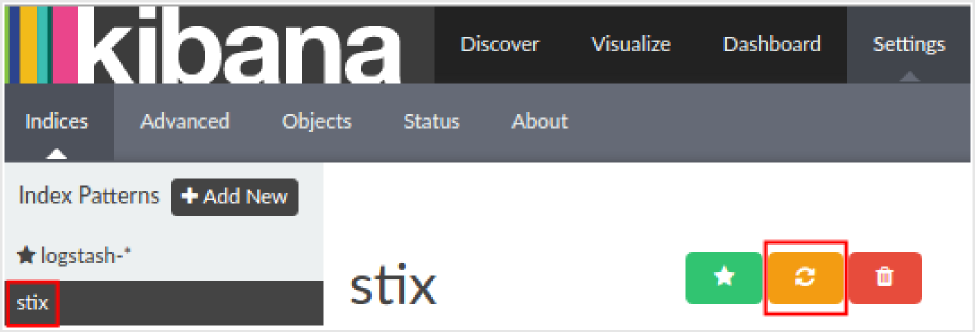
- After successfully signing in to the platform the dashboard is loaded, but one or more Kibana gauges are not displayed correctly, or they return the *“Could not locate that index-pattern-field”* error message.

Mitigation

The dashboard is configured to expect populated index fields whose data it can retrieve upon loading. If the fields contain no data or if the Kibana index is very recent, some gauges may fail to display. This is expected behavior.

If the index is populated with data, you can try to solve the dashboard gauge display issue by refreshing the Kibana index:

- To access Kibana, in the web browser address bar enter a URL with the following format:
  https://${platform_host}/private/kibana/app/kibana#
  Keep the trailing **#**
  Example: *https://${platform_host}.com/private/kibana/app/kibana#*
- On the top navigation bar, click **Settings**.
- Under **Settings**, click **Indices**.
- From the available index list, select **stix**.
- Click the yellow/orange **Refresh field list** button above the field list to refresh the Kibana index:



Last generated on Mar 6, 2018

PostgreSQL logs errors during UPSERT

PostgreSQL version 9.4 and earlier may log error messages during concurrent ingestion and/or UPSERT operations.

> PostgreSQL 9.5 and later versions natively support UPSERT. You can ignore this article if your platform installation uses PostgreSQL 9.5 or later.

Scenario

- The PostgreSQL version in use is 9.4 or earlier.
- PostgreSQL logs error messages during data ingestion.

Issue

During concurrent data ingestion of similar entity content sharing the same observables, PostgreSQL may log error messages that can look like the following example:

```
< 2016-05-13 17:36:05.767 CEST >ERROR:      duplicate key value violates unique constraint
"extract_kind_value_key"
< 2016-05-13 17:36:05.767 CEST >DETAIL:     Key (kind, value)=([...], [...]) already exists.
< 2016-05-13 17:36:05.767 CEST >STATEMENT:  INSERT INTO extract (kind, value, meta, created_at,
last_updated_at) VALUES ('[...]', '[...]', '{}', now(), now()) RETURNING extract.id
```

Mitigation

You can safely disregard these errors. The database integrity is not affected.

PostgreSQL may generate such error messages when concurrently ingesting a large data corpus where similar entities share the same observables. This is a side-effect of an **UPSERT** `(https://wiki.postgresql.org/wiki/upsert)` operation. PostgreSQL natively supports **UPSERT** `(http://git.postgresql.org/gitweb/?p=postgresql.git;a=commit;h=168d5805e4c08bed7b95d351bf097cff7c07dd65)` as per version **9.5** `(https://www.postgresql.org/docs/9.5/static/sql-insert.html)`. An UPSERT operation in earlier versions may generate errors, but PostgreSQL handles them gracefully. In case of a conflict, and provided there is no independent error, the application inserts or updates the affected records correctly.

In short, it does log the errors, but it handles them by doing the right thing with the ingested data. For PostgreSQL versions until 9.4, this standard behavior; apart from the error logging, the application works as expected.

Common proxy issues

If the system hosting the platform routes traffic through a proxy server, system administrators may need to tweak the proxy configuration files to address connectivity issues to and from the platform.

Scenario

- The EclecticIQ Platform is configured to route data communication through a proxy server.

Issue

- Data communication from and to the platform does not work as expected because the proxy settings are not correctly configured.

Mitigation

The recommendations in this section are the first steps you should take when troubleshooting common proxy issues that may hinder platform connectivity.

Check the proxy settings in the GUI

The first and easiest place to look for proxy settings is under **System > Proxies** in the web-based GUI.
This is the recommended course of action.

> **Warning:** It may be necessary to manually edit the *proxy_url* configuration file only in a few edge-case scenarios. The platform GUI allows to control all standard proxy configuration options within the platform context.
> Therefore, manually editing *proxy_url* is *not* recommended, since the manual edits applied directly to the file may conflict with the proxy settings defined through the platform GUI.

Check the proxy settings from the command line

Proxy config file	Description
`/opt/eclecticiq/etc/eclecticiq/proxy_url`	Contains the IP addresses and host names that should bypass the proxy. Multiple values are comma-separated. The no-proxy list needs to always include the following entries: `127.0.0.1,localhost`.

To retrieve `proxy_url`, run the following command(s):

```
$ find /opt/eclecticiq/etc/eclecticiq/ -name proxy_url
```

The proxy configuration file default location is:

```
/opt/eclecticiq/etc/eclecticiq/proxy_url
```

Open the file::

```
$ vi /opt/eclecticiq/etc/eclecticiq/proxy_url
```

An example of a proxy URL configuration file may look like this:

```
$ export HTTP_PROXY=http://example_proxy_server1.com:8080
$ export HTTPS_PROXY=https://example_proxy_server2.com:443

# Default values for NO_PROXY
# Always bypass proxy for localhost and 127.0.0.1
$ export NO_PROXY="localhost,127.0.0.1"
```

Check the logs

The Nginx web server log files can provide some insightful help when trying to identify proxy issue causes:

```
/var/log/nginx/access.log
/var/log/nginx/error.log
/var/log/nginx/eclecticiq-platform-ui-nginx-access.log
/var/log/nginx/eclecticiq-platform-ui-nginx-errors.log
```

Bypass the proxy

Always bypass proxying for the following addresses: `"localhost,127.0.0.1"`.
If the Nginx logs contain many errors, you may need to add to the no-proxy list also the IP address of the server hosting the platform.

Edit the proxy configuration

> **Warning:** It may be necessary to manually edit the *proxy_url* configuration file only in a few edge-case scenarios. The platform GUI allows to control all standard proxy configuration options within the platform context.
> Therefore, manually editing *proxy_url* is *not* recommended, since the manual edits applied directly to the file may conflict with the proxy settings defined through the platform GUI.

Open the file:

```
$ vi /opt/eclecticiq/etc/eclecticiq/proxy_url
```

For example, the current proxy configuration including user name and password may look like this:

```
$ export HTTP_PROXY=http://guest:guest@example_proxy_server1.com:8080
$ export NO_PROXY="localhost,127.0.0.1"
```

Remove user name and password from the proxy URL. In the example, this means removing `guest:guest@`:

```
$ export HTTP_PROXY=http://example_proxy_server1.com:8080
$ export NO_PROXY="localhost,127.0.0.1"
```

Save the file, and then restart all processes for the changes to become effective.

Restart all processes after proxy changes

When you modify or update proxy settings, the following notification message is displayed:

> Proxy configuration updated. The process needs to be restarted in order for these settings to be applied.

You need to restart all Supervisor-managed processes for the changes to become effective.
To do so, run the following command(s):

Test the proxy configuration

Before testing the changed proxy settings, make sure **Cabby is installed** `(https://cabby.readthedocs.io/en/latest/installation.html)`, and `venv` is enabled.

- Activate a Python virtual environment:

```
$ source /opt/eclecticiq/platform/api/bin/activate
```

An easier way to test the platform proxy settings uses a wrapper. This approach provides an environment that is as close as possible to the platform host environment.

Test the proxy to see if the new settings are picked up and are correct. For example, try to reach the **Hailataxii** `(http://hailataxii.com/)` discovery service.

Run the following command(s):

```
/opt/eclecticiq/bin/run-platform-component taxii-discovery --host hailataxii.com --path /taxii-discovery-service --username guest
```

A successful response from the Hailataxii discovery service may look like this:

```
2015-12-18 11:34:55,576 INFO: Sending Discovery_Request to http://hailataxii.com/taxii-discovery-service
2015-12-18 11:34:55,894 INFO: Response received for Discovery_Request from http://hailataxii.com/taxii-discovery-service
2015-12-18 11:34:55,895 INFO: 3 services discovered

=== Service Instance ===
  Service Type: DISCOVERY
  Service Version: urn:taxii.mitre.org:services:1.1
  Protocol Binding: urn:taxii.mitre.org:protocol:https:1.0
  Service Address: http://hailataxii.com:80/taxii-data
  Message Binding: urn:taxii.mitre.org:message:xml:1.1
  Available: True
  Message: None

=== Service Instance ===
  Service Type: COLLECTION_MANAGEMENT
  Service Version: urn:taxii.mitre.org:services:1.1
  Protocol Binding: urn:taxii.mitre.org:protocol:https:1.0
  Service Address: http://hailataxii.com:80/taxii-data
  Message Binding: urn:taxii.mitre.org:message:xml:1.1
  Available: True
  Message: None

=== Service Instance ===
  Service Type: POLL
  Service Version: urn:taxii.mitre.org:services:1.1
  Protocol Binding: urn:taxii.mitre.org:protocol:https:1.0
  Service Address: http://hailataxii.com:80/taxii-data
  Message Binding: urn:taxii.mitre.org:message:xml:1.1
  Available: True
  Message: None
```

Common ingestion issues

Workarounds and mitigation measures to address issues affecting intel and graph ingestion queues.

Scenario

- An EclecticIQ Platform installation with active data and intel ingestion processes feeding the PostgreSQL, Elasticsearch, and Neo4j databases.
  The platform relies on three databases to store, index, and manage ingested data:
  - *PostgreSQL* is the main database and the repository of all ingested BLOBs, as well as platform information such as user profiles and so on.
  - *Elasticsearch* is the search and indexing database. By taking care of these tasks, it contributes to making data indexing and searching faster and more efficient.
  - *Neo4j* is the graph database driving graph views.

Issue

- Common data ingestion issues like:
  - Adding entities to the graph through the GUI hangs or does not work
  - The graph database goes out of sync and it does not resync correctly
  - Recently added entities are not visible or findable in the GUI — this is a hint at a possible ingestion queue problem.

Mitigation

Inspect ingestion queues

Begin by taking a look at ingestion queues to look for anomalies like no entities being ingested. Redis acts as a message broker for task queues.

Within the platform Redis manages task queues. Possibly the only command you need is the one that allows you to check queue length: `llen`.

To inspect the platform data ingestion queue length, run the following command(s):

```
# Launch redis-cli
$ redis-cli

$ > llen "queue:ingestion:inbound"
```

To inspect the graph database queue length, run the following command(s):

```
$ > llen "queue:graph:inbound"
```

To inspect the Elasticsearch data update queue length, run the following command(s):

```
$ > llen "queue:search:inbound"
```

Example

```
# Launch redis-cli
$ redis-cli

> llen queue:ingestion:inbound
// response example
(integer) 1234567

> llen queue:search:inbound
// response example
(integer) 0

> llen queue:graph:inbound
// response example
(integer) 456789
```

Remove graph ingestion from the queue

This mitigation measure adresses the scenario where entity enqueuing to the graph database unexpectedly stops.
The command that starts enqueuing entities to the graph database is:

```
$ sudo -u eclecticiq -i EIQ_PLATFORM_SETTINGS=/opt/eclecticiq/etc/eclecticiq/platform_settings.py
/opt/eclecticiq/platform/api/bin/eiq-platform graph reindex
```

If the entity enqueuing process does not behave as expected, the graph database may go out of sync. To address the problem, do the following:

- Remove graph ingestion from the Redis (message broker) queues by running the following command(s):

```
# Launch redis-cli
$ redis-cli

$ > del queue:graph:inbound
```

- Restart enqueuing entities by running the following command(s):

```
$ sudo -u eclecticiq -i EIQ_PLATFORM_SETTINGS=/opt/eclecticiq/etc/eclecticiq/platform_settings.py
/opt/eclecticiq/platform/api/bin/eiq-platform graph reindex
```

After starting, the ingestion queue should begin to grow. Check the relevant log files to monitor ingestion progress.

Neo4j does not start or is very slow

Neo4j may not start after an update or a reinstallation, or the graph database performance may decline unexpectedly. Possible causes can be the mapped memory value defined for the Neo4j store or issues in the ingestion queues.

Scenario

- An EclecticIQ Platform installation with Neo4j 3.3.3 Community as the graph database.
- Neo4j does not behave as expected.

Issue

- Data funneling to the graph database may hang or it may be unusually slow.
- Neo4j does not start after a platform update or a platform reinstall operation.

Mitigation

Neo4j performance decay may be caused by issues in the Redis-managed data ingestion queues.
To investigate the issue, begin by checking ingestion queues and graph ingestion logs.

- Check the Redis graph ingestion queue to see if any anomalous values are returned.
  Example:

```
# Launch redis-cli
$ redis-cli

$ > llen queue:graph:inbound
// response example
(integer) 456789
```

The default locations of the graph ingestion and Neo4j log files are */var/log/eclecticiq/graph-ingestion.log* and */var/log/eclecticiq/neo4j.log*, respectively.

- Examine the graph ingestion log for errors, and review the sampling interval: if the corresponding value is too small, it can prevent Neo4j from starting correctly.
  For example the following warning extract from the graph ingestion log indicates that Neo4j has not been started, yet:

```
{
  "event": "graph.db.down",
  "level": "warning",
  "logger": "graph_ingestion.cli",
  "message": "Waiting for graph db",
  "timestamp": "2016-01-03T11:15:30.580337Z"
}
```

A possible cause of the problem preventing Neo4j from starting after an update or a reinstallation can be an invalid mapped memory value defined for the Neo4j store.
To investigate the issue, check the Ǹeo4j status and the Neo4j log.

- Check if Neo4j is running.
  For example the following response example indicates that Neo4j is not running:

```
// response example (extract)
neo4j.service - (null)
  Loaded: loaded (/etc/rc.d/init.d/neo4j)
   Active: failed (Result: exit-code) since Fri 2016-01-03 11:19:42 CET; 3s ago

...

  Process: 21816 ExecStop=/etc/rc.d/init.d/neo4j stop (code=exited, status=0/SUCCESS)
  Process: 21776 ExecStart=/etc/rc.d/init.d/neo4j start (code=exited, status=0/SUCCESS)
  Main PID: 12345 (code=exited, status=1/FAILURE)

...

Jan 03 11:20:33 <host_server> systemd[1]>: Starting (null)...

...

Jan 03 11:20:34 <host_server> neo4j[123456]: Starting neo4j: [  OK  ]
Jan 03 11:20:35 <host_server> systemd[1]: Started (null).
Jan 03 11:20:36 <host_server> systemd[1]: neo4j.service: main process exited, code=exited,
status=1/FAILURE
Jan 03 11:20:37 <host_server> neo4j[123567]: Stopping neo4j: /etc/rc.d/init.d/neo4j: line 66:
kill: (123466) - No such process
Jan 03 11:20:38 <host_server> systemd[1]: Unit neo4j.service entered failed state.
Jan 03 11:20:39 <host_server> systemd[1]: neo4j.service failed.
```

- Examine the Neo4j log and look for any failure or error messages to understand why it did not start.
  For example the following sample error message points at an incorrect Neo4j memory size value:

```
ERROR Failed to start Neo Server on port <unknown_port> 0.5G is not a valid size, must be e.g.
10, 5K, 1M, 11G

java.lang.IllegalArgumentException: 0.5G is not a valid size, must be e.g. 10, 5K, 1M, 11G
```

In this case, the affected key/value pair is `neostore.propertystore.db.mapped_memory=0.5G` in the */usr/share/neo4j/conf/neo4j.properties* configuration file.
This key takes **integer values** `(http://neo4j.com/docs/stable/configuration-settings.html#config_neostore.propertystore.db.mapped_memory).`
To assign Neo4j a 0.5 GB memory size, you need to set its value as
`neostore.propertystore.db.mapped_memory=500M`.

> **Warning:** As per Neo4j 2.3.1, `neostore.propertystore.db.mapped_memory` is deprecated, and **it has been replaced** `(http://neo4j.com/docs/stable/configuration-settings.html#config_dbms.pagecache.memory)` by `dbms.pagecache.memory`.

- After solving the problem, restart Neo4j to launch the graph database, and the related batch process to enable graph ingestion:

```
$ supervisorctl start neo4j-batching
```

- Lastly, check the statuses of the tasks:

```
$ supervisorctl status neo4j-batching
```

Pushing content to a TAXII inbox shows no new content

After successfully pushing content to the TAXII inbox service of an incoming feed, no new content is displayed or made available in the platform.

Scenario

- An incoming feed is configured to ingest STIX content using a TAXII inbox service.
- New content is pushed to the incoming feed TAXII inbox using **Cabby** `(https://cabby.readthedocs.org/)`, a TAXII client with a command line interface that allows executing data discovery, polling, and fetching tasks.

Issue

- You push new content to the TAXII inbox service of an incoming feed, so that the feed can retrieve it.
- The task is executed and it completes correctly.
- However, no new content shows or is available in the platform.

Mitigation

- Make sure the outgoing feed is correctly configured.
- Check the Cabby log file, where a successful push-to-inbox action should return the following message:

```
INFO: Content block successfully pushed
```

- Verify that the actual content type being pushed to the incoming feed TAXII inbox service matches the configured content type for that feed. To do so, you need to inspect the value of the `binding` parameter you pass when you **push content to the inbox service** `(https://cabby.readthedocs.org/en/latest/user.html#using-cabby-as-a-command-line-tool)` with `taxii-push`.
  The default **content binding** `(https://taxiiproject.github.io/releases/1.1/taxii_contentbinding_reference_v3.pdf)` type is `urn:stix.mitre.org:xml:1.1.1`.
  For example if you are fetching content in STIX 1.2 data format, change the content binding value to reflect the correct STIX version: `--binding "urn:stix.mitre.org:xml:1.2"`

```
(venv) $ taxii-push --host test.taxiistand.com \
                    --https \
                    --discovery /read-write/services/discovery \
                    --content-file /tmp/stuxnet.stix.xml \
                    --binding "urn:stix.mitre.org:xml:1.2" \
                    --subtype custom-subtype
```

- If the push action is successful, but it fails to return new content, the cause of the problem may be:
  - *Binding/Content type mismatch*: the content blocks being pushed have a content type that does not match the `binding` value you passed with `taxii-push`.
    The configured content type for the feed and the actual content are different. For example, the configured content type is STIX, but the pushed content is in PDF format. In this case, either reconfigure the content type as PDF to match the pushed PDF format, or push content in STIX format instead of PDF.
  - *Ingestion problem*: if the content type matches the content binding, the content block ingestion process failed. Check ingestion logs heck ingestion and OpenTAXII logs for to investigate any issues:
    - Go to `/var/log/eclecticiq/`.
    - Browse for `intel-ingestion.log` and `opentaxii.log`.