



Carbon Black Enterprise Response Unified View User Guide

Carbon Black Unified View Version 1.0

*****Important Note: As of this release, Carbon Black Federated Server is undergoing a name change. This product will be known as Cb Enterprise Response Unified View in the next release. In the current documentation where applicable, please treat Unified View and Federation as interchangeable.**

15 March 2016

Carbon Black, Inc.
1100 Winter Street, Waltham, MA 02451 USA
Tel: 617.393.7400 Fax: 617.393.7499
E-mail: support@carbonblack.com
Web: <http://www.carbonblack.com>

Copyright © 2004-2016 Carbon Black, Inc. All rights reserved. This product may be covered under one or more patents pending. Carbon Black is a trademark of Carbon Black, Inc. in the United States and other countries. Any other trademarks and product names used herein may be the trademarks of their respective owners.

Contents

[Introduction](#)

[Architecture](#)

[Terminology](#)

[Quick Start](#)

[Requirements](#)

[Installation](#)

[The Federation Service](#)

[Configuration](#)

[Other Configuration Options](#)

[Certificates](#)

[Facet Configuration](#)

[Federated Operating Modes](#)

[Changes to Carbon Black in Federated Context](#)

[Process Search Page](#)

[Process Analysis Page](#)

[Binary Search Page](#)

[Binary Details Page](#)

[Binary Preview Pop-ups](#)

[Per-Cluster Context Changes](#)

[Security Considerations in Per-cluster Context](#)

[Federated User Management](#)

[Carbon Black Cluster/Server Management](#)

[Cluster Health Performance](#)

[Cluster/Server Authentication](#)

[SSL Verification](#)

[Disabling Clusters from Queries](#)

[Logging](#)

[Troubleshooting](#)

[Command Line Tools](#)

[User Management](#)

[Cluster Management](#)

[Other Carbon Black Documentation](#)

[Contacting Carbon Black Support](#)

Introduction

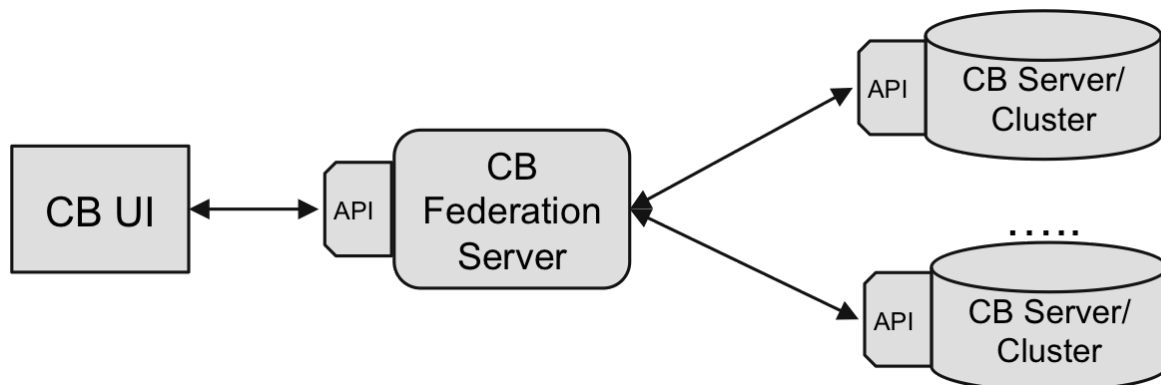
The Carbon Black Federation Server is a stand-alone server that ties multiple Carbon Black Enterprise servers together and provides a single user interface (UI). The federated UI provides the ability to perform queries across multiple Carbon Black servers with a unified result set. Additionally, the Carbon Black Federation UI can be used to manipulate the full functionality of a single Carbon Black Enterprise Server.

This document assumes familiarity with Carbon Black. It is recommended that users read the Carbon Black User Guide for the version they are installing to provide the necessary background on how Carbon Black functions.

Architecture

The Carbon Black Federation Server is a standalone server instance that runs on a separate system from the Carbon Black Enterprise Server. The Carbon Black Federation Server is an API proxy. It takes queries made to the Federation Server, performs those queries against each of the configured Carbon Black servers, merges the results into a unified result set, and presents them to the end user. The Federation Server has its own user store. It also has configuration store for the servers it queries. It does not store any queried data on the Federation Server.

The below figure illustrates the architecture of a Carbon Black Federation deployment. The single Carbon Black Federation Server uses its own modified version of a Carbon Black User Interface (UI).



Terminology

In order to avoid some confusion, this document and the Carbon Black Federation Server use a set of terminology to help the user understand different parts of the Carbon Black Federated Server architecture.

- Cluster - Commonly called cluster, or Carbon Black server. This refers to the Carbon Black Enterprise Servers that store and index sensor data. The term could refer to a single physical Carbon Black Enterprise Server or an entire cluster. For the sake of clarity, the term cluster was chosen over server.
- Federation Server - The Carbon Black Federation Server. This is the separate server used to run and host the federated capabilities.

Quick Start

The following section contains the necessary information to install the Carbon Black Federation Server.

Requirements

The Carbon Black Federation Server requires a hardware server with a base install of 64-bit Centos OS 6 server or Redhat Enterprise 6. The server can be either physical or virtualized. There are two recommended server configurations depending on the expected federation size:

- Smaller Federations – Federations with less than 10 Carbon Black servers and fewer than 100 concurrent federation users should use a configuration of 8 GB of RAM and 4 CPU Cores.
- Larger Federations – Federations with more than 10 Carbon Black servers should provide at least 16 GB of RAM and at least 8 CPU Cores.

A Carbon Black Federation Server does not store search data, so the systems do not require significant storage space and do not have substantial disk IO requirements. A typical enterprise level hard drive or equivalent should be sufficient. It is recommended that customers configure at least 40 GB of log space in `/var/log/cbfed`.

The Carbon Black Federation Server requires at least one Carbon Black Enterprise server for which it aggregates the results. The Carbon Black Federation Server uses the login credentials from any user account within that Carbon Black cluster. The Carbon Black Federation Server requires Carbon Black server version 5.1 or newer.

The Carbon Black Federation Server requires HTTPS query access to each underlying Carbon Black server. It must be able to make HTTP API queries to the restful API, using the configured Carbon Black port, usually 443. The Carbon Black Federation Server does not support HTTP proxies and must have direct HTTPS access to the servers.

Installation

The first step in the installation process is to add a new repo. This can be accomplished, as root, by creating a new file in `/etc/yum.repos.d` called `CarbonBlackFederation.repo`. The file should have the following contents:

```
[CarbonBlackFed]
name=CarbonBlackFed
baseurl=http://yum.carbonblack.com/cbfed/production/release/
gpgcheck=0
enabled=1
metadata_expire=60
```

Once this has been added, you can install Carbon Black Federation Server by running:

```
yum install cbfed
```

This installs all the necessary rpm packages for the Federation Server. The server should be initialized through the script `/usr/share/cbfed/cbfedinit`. This script creates the initial admin user account and initialize the federation internal storage.

The Federation Service

The Carbon Black Federation service runs under the service name `cbfed`. It can be started and stopped using the standard Linux service command:

```
service cbfed start
and
service cbfed stop
```

Configuration

The Carbon Black Federation Server configuration file is located in `/etc/cbfed/federation.config`. Generally speaking, the default values are sufficient for any federation, and the user should not need to change the configuration values. A detailed description of each configuration value can be found here.

The following configuration values control how the Federation Server handles logging.

Table 1. General Configuration Options

Column	Description
<code>max_clusters_per_monitor</code>	The maximum number of clusters to query health checks for at any given time. The default value is 50.
<code>ssl_cert_file</code>	The location of the SSL certificate used for the Federation Server. The default value is <code>/etc/cbfed/certs/server.crt</code>
<code>ssl_key_file</code>	The location of the SSL key used for the Federation Server. The default value is <code>/etc/cbfed/certs/server.key</code>

flask_secret	A random secret key used by flask.
Workers	The number of worker processes to start for the webserver.
max_request_per_worker	The number of HTTP requests a worker can handle before it's torn down and recreated. We do not recommend changing this value.
timeout_per_worker	The wait timeout, in seconds, for a worker process before it's considered hung and must be torn down. This value should always be larger than any timeout values.
graceful_timeout_per_worker	The time number of seconds the server waits for a worker process to be torn down. We don't recommend changing this value.
federated_request_timeout	The timeout value in seconds for the Federation Server to wait on a query response from a Carbon Black server. The default value is 2 minutes.
facet_mode	This configuration value controls how facets are rendered in the UI. See the detailed description below.
db_uri	The location of the cbfed Sqlite database. We don't recommend changing this value.

Table 2. Configuration Options for Health Monitoring

Column	Description
health_check_interval	The period of time in seconds over which the Federation Server performs a heartbeat to ensure that the back-end clusters are functional. This has the default value of 30 seconds.
max_clusters_per_monitor	The maximum number of clusters to query health checks for at any given time. The default value is 50.
stats_aggregation_interval	The time window, in seconds, over which average heartbeat and average query times are calculated. The default value is 5 minutes (300 seconds).
stats_aggregation_to_keep	The number of intervals of health stats to store. The default value is 1440, which results in a default storage of 5 days.
stats_aggregation_interval	The time window, in seconds, over which average heartbeat and average query are calculated.
max_number_of_db_error_logs	The maximum number of error logs to store internal to the health monitoring database. This is the number of errors that is displayed in the federation console. All errors are logged to the error log.
max_health_failures	The maximum number of failed API calls that can occur before a cluster is marked as Red (Poor) cluster health status.
unstable_avg_query_time	The threshold after which average query time results in a Yellow (Fair) cluster health status.
unavailable_avg_query_time	The threshold after which average query time results in a Red (Poor) cluster health status.
unstable_avg_heartbeat_time	The threshold after which average heartbeat time results in a Yellow (Fair) cluster health status.

unavailable_avg_heartbeat_time	The threshold after which average heartbeat time results in a Red (Poor) cluster health status.
--------------------------------	---

Table 3. Configuration Options for Logging

Column	Description
debug	Enable verbose debug logging in the application log.
app_log_output_path	The path of the Federation Server application log.
access_log_output_path	The path of the Federation Server access log. We don't recommend changing this value.
errors_log_output_path	The path of the Federation Server error log. We don't recommend changing this value.
rollover	The interval in days that the logs are stored. After this period, the oldest log file is deleted.
log_appender	Specifies the output for internal application logging. We don't recommend changing this value.

Other Configuration Options

There are several other configuration options stored within the federation configuration file. These configuration values control things such as rate limiting and caching. We don't recommend changing these configuration values unless prompted by a support professional.

Certificates

The SSL certificates used for the Federation Server are stored by default in `/etc/cbfed/certs`. The `cbinit` script generates an initial set of certificates. At any one point, these certificates can be changed to valid certificate authority certs. The federation configuration file contains two configuration values `ssl_cert_file` and `ssl_key_file` which can be used to change the location of the ssl keys if desired. Note, any changes to the values of `ssl_key_file` and `ssl_cert_file` must also be reflected in `/etc/init.d/cbfed-nginx`.

Facet Configuration

Facets, the search filters shown as graphics on Carbon Black console pages, can be configured to render with different modes when viewing federated process and binary search results. This value is controlled by the `facet_mode` configuration value. There are three possible values for this configuration setting:

- *asynchronous* – With Carbon Black 5.1.1, a user interface optimization was introduced to render facets asynchronously, after the table of search results has been displayed to the user. This allows the user to view search results without having to wait for the facets to be created. This is the preferred and default value, but may introduce performance issues for federations of 100s of Carbon Black servers because each server has to be queried multiple times.
- *synchronous* – This value causes the federation user interface to render facets at the same time as the federated search results. This was the default behavior on Carbon Black 5.1 and previous versions. This is recommended for larger federations, with hundreds of Carbon Black servers.

- *disabled* – This value causes the federation user interface to not render facets at all. This is only recommended in environments with poor search performance.

Federated Operating Modes

The Carbon Black Federation Server’s console operates in one of two contexts or modes. These contexts control how the federation searches are applied across the federation.

The first context is the “federated” or “global context”. When in the federated context, all searches done within the federated console are applied across all enabled servers within the federation, and the results returned contain matches from any server within the federation. The global context is the default context for the Carbon Black Federation Server and is indicated by the purple menu bar at the top of the console:

Figure X



When a user logs in, the initial context is the global context. There are four operations supported by the global context:

1. **Process Search.** The Process Search is performed across the entire federation. The matching results are returned for all clusters within the federation.
2. **Process Analysis Page.** When, within the global context, a process is selected from the Process Search page, a global context view of the Process Analysis page is presented. This page is similar to a regular Carbon Black Process Analysis page, but the links within the page that perform searches perform the searches across the entire federation, instead of performing them against a single cluster.
3. **Binary Search.** Similar to the process search, the binary search is done across the entire federation and the results are displayed. It’s important to note that in the global context, binaries are represented once for each occurrence within each individual Carbon Black cluster. Thus a binary, or md5, seen by four clusters within the federation appears four times.
4. **Binary Details.** Any binary that is selected from the binary search within the global context is taken to a global context version of the Binary Details page. This page contains aggregated information about the binary from across the federation. Searches selected on this page are also applied across the federation.

The global context operations are very similar to the regular Carbon Black Enterprise Server counterparts, but have had some functionality removed that is not available in the global context. Some information, such as that on the Sensor Details page, is not supported in the global context; it’s only supported in the per-cluster context. Whenever a link causes a change in global context to per-cluster context, a new browser tab is opened and the context changes to per-cluster context. The per-cluster context can be used to drive all the functions of a particular cluster/server.

Whenever operating in federated or “global context”, the Federation Server must query the results for each cluster, perform merge operations, and present the result set to the user. This means that

the page always waits for the slowest server to return before returning results to the end user. There is a configurable time-out value for how long the Federation Server can wait for a cluster to return results. Whenever an error or timeout occurs for a query performed against a cluster, the global context page presents the user with an error bar.



When this bar is clicked, the cluster name, along with any relevant error details, is displayed to the user.

Changes to Carbon Black in Federated Context

The federated or “global context” has some modified functionality from the normal Carbon Black functionality.

Process Search Page

The process search page allows for searches of process activity across the entire federation. The page has been modified, replacing the group facet with a per-cluster facet that allows for filtering results to a specific cluster. Additionally, each returned process includes the cluster name in addition to the sensor name when displaying the search results. When a specific process is selected within the result set, a global context version of the process analysis page displays.

Process Analysis Page

The Process Analysis page is similar to the regular Carbon Black process analysis page with a few notable exceptions. First, all searches are performed in the “global context” across all clusters within the federation. Second, investigations are not supported on the federated Process Analysis page. The per-cluster context page must be viewed to tag events as part of an investigation. The investigation column does not exist for events. Last, banning is not supported in federated mode.

Binary Search Page

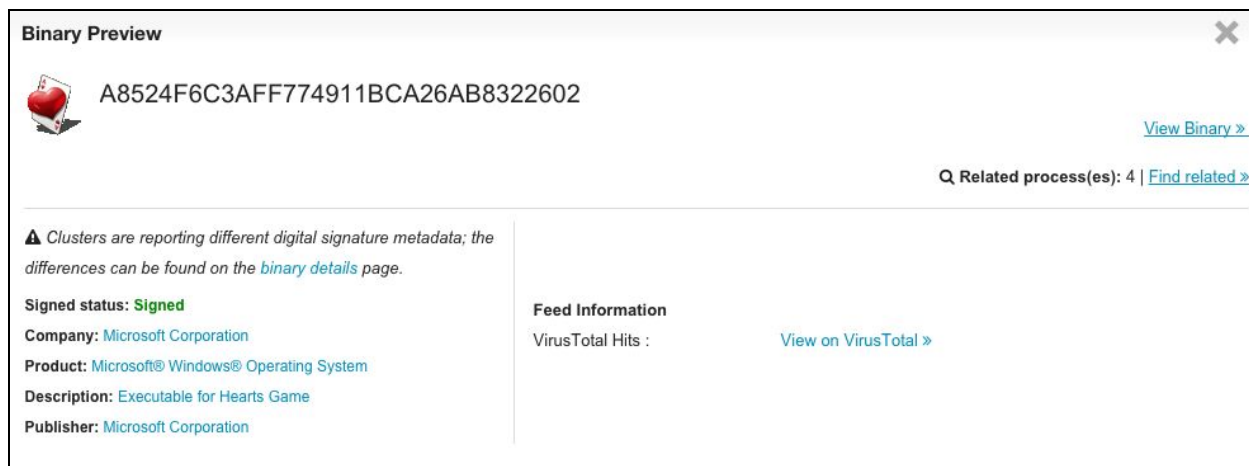
The Binary Search page is similar to the regular Carbon Black Binary Search page. One major difference in the search page that displays in the federated context is that the results returned from the binary search are not re-duplicated across the clusters. Thus, if a particular binary occurs on multiple clusters within the federation, it appears multiple times within the search results. This is due to a technical limitation for how sorting works with binary results within the Federation Server. Despite a particular binary potentially have multiple entries within the binary search results; clicking on a binary takes the user to a singular Binary Details page for that binary.

Binary Details Page

The Binary Details page within the federated context provides a singular view of the binary across the entire federation. One major change within the Binary Details page is the per-cluster breakdown of prevalence and signature information. It is possible that Carbon Black clusters could disagree on the signature status of a binary. This is because this signature status is verified on the sensor, and different environments can have different certificate trust settings. Because of this, when a conflict exists for Digital Signature Metadata, the user can click on the different signature status and see which clusters reported which status.

Binary Preview Pop-ups

Because clusters can disagree on the signature status of a binary, the Binary Preview page has been modified to reflect this disagreement. If clusters disagree on the signature status, the Binary Preview pop-up displays one status and indicates that there is a disagreement among the clusters within the federation. In this case, the summary displays “Signed” if one cluster is signed; otherwise the status displayed is one of the other statuses. See the attached screenshot for an example.



Per-Cluster Context Changes

The operation of the per-cluster context is nearly equivalent to connecting to the console for a regular Carbon Black Enterprise Server. There are some minor changes to note. First and foremost, the top level navigation bar has been modified to include the cluster name, which is given when the cluster is added to the federation. This serves as an indication that the page is in per-cluster operation.

Second, there are places in the per-cluster mode console where a “Global Search” button has been added to allow the user to apply a given search across the whole federation. When that button is clicked, a new federated context browser tab opens with the search results for the whole federation. The global search button has been applied to the Process Search page, the Binary Search Page, and the Watchlist Details Page.



Security Considerations in Per-cluster Context

Because it may not be desirable for users of the federation to modify or change settings on the cluster, some limitations are in place. First, the Profile and Logout pages have been removed. This prevents users from modifying the Carbon Black cluster account from within the per-cluster context. Users must log in directly to the Carbon Black cluster to make changes to the user account.

Second, users who don't have administrative privileges from within the Federation Server are forbidden from changing or creating new users within a Carbon Black cluster. This is true even if the user provided a per-user API token that has sufficient privileges on that cluster. Users should log into the cluster directly to make those changes.

Federated User Management

The Carbon Black Federation Server has its own user store independent of the underlying Carbon Black servers/clusters. There are two permissions levels for users within the Federation Server, administrators and regular users. Administrative users can perform the following actions:

- Adding or removing a cluster from the federation.
- Disconnecting (i.e., disabling) a cluster from searches, within the global context, for all users.
- Setting a shared token for use by all users when accessing the cluster.
- Adding or removing users from the Federation Server.
- Adding or removing the administrator permission from a user.
- Managing users in per-cluster mode. Users must have administrative rights in the Federation Server to perform user management within the cluster.

The user administration page is accessible by clicking on the administrative (GEAR) icon in the top level UI within the global context. When in the per-cluster context, the user administration functionality is performed specific to that cluster.

New users can be added by clicking on the "+ Add User" button. The users must have a first name, last name, username, and password.

Any federation user can be edited by clicking on their name in the left-hand user list. Any information about the user account can be changed at that point. User accounts can only be edited by the account owner or by an administrator. An administrator can make a user an administrator by sliding the "Administrator" slider from "No" to "Yes". Any user account can be deleted by selecting the user in the user list and selecting "Delete User".

Carbon Black Cluster/Server Management

Carbon Black servers can be added or removed from the CB Federation Server through the "Cluster Management" menu item on the administrative (GEAR) icon. The cluster management page provides the following actions:

- Adding or removing clusters to/from the federation.
- Disconnecting (or disabling) a cluster from queries for all users.
- Viewing the health status of each cluster within the federation.

- Setting authentication parameters used when accessing a cluster.
- Selecting a specific cluster for per-cluster operations.

Some cluster settings can be managed on a per-user basis. This includes setting per-user authentication credentials and excluding (i.e., disabling) clusters on a per-user basis. These actions can be taken from the “My Clusters” page of the user profile.

Cluster Health Performance

The Cluster Management page provides a high-level health overview of the existing clusters within the federation, including average query time and average heartbeat time. The health status, displayed via a red, yellow, or green health indicator, is a combination of average query time, heartbeat round trip time, and the number of errors that has occurred.

The Federation Server tracks a few key health metrics on a regular interval. By default, this interval is every 5 minutes. The Federation Server stores, by default, 5 days worth of intervals. These values are configurable. The Federation Server stores the following health statistics for each interval:

- Average Heartbeat Time: This is the average time value it takes for the Federation Server to query the info API endpoint on the cluster.
- Average Query Time: This is average time value for all non-heart beat queries made to the server. It’s possible that this value is zero if no queries were made against the server.

The Federation Server also stores the last 50 errors that have occurred when the Federation Server queries the cluster. The server also provides the error count and time of last query timeout.

More detailed information about the server health information is available by mousing over the cluster and selecting “Stats & Details”. More detailed information about the system performance including recent errors is available under the “Cluster Statistics” option button presented on each cluster. Two of the cluster health parameters tracked are the heartbeat and round-trip time. This is the time it takes the Federation Server to query the information API endpoint for a specific cluster. The second health metric is the average query time for queries performed against that cluster. Additionally, the 5 most recent errors that occurred when communicating with the server is displayed when the user selects the “Stats & Details” button on the Cluster Management page.

Cluster/Server Authentication

When a cluster is added to the federation, there are two supported mechanisms for authenticating access to it from the CB Federation Server. The first mechanism is to use a “Shared Token”, or API token, for all CB federation users when accessing the cluster. When the cluster is added to the federation, an admin user can enter a shared token for all users.

If administrators want users to use their own credentials when accessing the server, they should select “Individual user tokens” for the cluster. In this case, the user must enter an individual API token for that cluster. This can be performed on the “My Profile” section under “My Clusters”.

A single Federation Server can have some clusters configured for per-user authentication while others are configured for shared token access.

SSL Verification

When adding a cluster to the federation, the default behavior is to not validate the SSL certificate used within the SSL handshake. This allows the use of self-signed certificates, such as the ones generated by the Carbon Black server initialization script. If the Carbon Black cluster is running certificates that have been signed by a valid certificate authority, we recommend you check the “Verify SSL Certificate” box.

Disabling Clusters from Queries

The Carbon Black Federation Server does not return results from a federated (or global context) search until all the clusters have returned results for the query or have timed out. This means that the Federation Server is as slow as the slowest cluster in the federation. If a specific cluster has performance issues or down time, it can be temporarily disabled from use in federated search results. There are two ways to disable a cluster. The first is that an administrator can disconnect a cluster for all federation users within the “Stats & Details” link on each cluster. When the administrator selects “Cluster Status” slider to disabled, no queries are performed against that server.

The second is by excluding clusters from query results on a per-user basis from within the “My Clusters” section of the user profile. Users can de-select the cluster using the slider on the “My Clusters.” This allows non-administrative users to exclude a poorly performing cluster from search results.

My Profile

Profile Info
API Token
My Clusters

My Clusters

Choose which clusters are included in your searches by enabling or disabling each cluster. Enter your personal API key to enable access to clusters where required.

All available cluster

<input checked="" type="checkbox"/>	cluster_test1	Shared Token
<input checked="" type="checkbox"/>	cluster_test2	Shared Token

When a cluster is disconnected by an administrator or excluded by a user, it is still possible to access the cluster using the per-cluster context by clicking the “Browse Cluster” button on the Cluster Management page.

Logging

The Federation Server has two main sources generating logs. The first source is `nginx` that logs basic logs for access and errors. These logs are stored in `/var/log/cbfed/nginx/accsess.log` and `error.log` respectively. These logs contain the basic HTTP access made to the Federation Server for both the API and the static web content.

The second set of logs generated is from the federation service itself. The federation service generates 3 log files that are stored in `/var/log/cbfed`. They are:

- *access.log* - This file stores all HTTP accesses made against the federation API endpoints and all access requests made by the Federation Server to the back-end clusters.
- *application.log* - This is the general log file used by the federation service. It includes the status of pool workers and activity of end users.
- *error.log* - This file contains errors encountered by the federation service. This includes internal errors and exceptions as well as errors that occur when the federation services tries to access a cluster.

Log files are rotated on a daily basis. Additionally, the files are stored for 30 days. This duration can be changed by setting the “rollover” configuration setting.

Troubleshooting

If the Federation Server is having trouble querying one of the clusters, an application warning is presented to the user. An example of this warning is shown below.



When this warning is displayed, the Federation Server is unable to query results from one or more Carbon Black clusters. Clicking on “View Details” can provide the user with more information in terms of what cluster is having an error and what that error is. Generally speaking, errors are reported in terms of HTTP error codes. When an error occurs querying a single cluster, results from other clusters are still displayed.

When a cluster is not returning results, the first place to check is the Cluster Details Page within the Cluster Management interface. This page displays both the health status (*Poor*, *Fair*, or *Good*) along with the list of recent errors. This allows the user to view recent errors to understand why connectivity might not be working. In many cases, the core issue with connectivity are displayed there.

A secondary place to examine the logs is within the federation error log. This log file contains all errors and exceptions that occur when making connections to Carbon Black clusters. If there isn’t any discernible cause for the issue, the last option is to enable `debug=True` in the `federation.conf` file. This allows for verbose logging and debugging.

Command Line Tools

The Carbon Black Federation Server includes a command line tool to manage federation users and clusters within the federation. This tool allows users, running as root, to make changes without having to use the API or console. The command line tool is located in `/usr/share/cbfed/cbfed_cli`. This tool has two main modes, user management and cluster management.

User Management

To enable the user management portion of the `cbfed_cli` tool, the user must pass “user” as the first argument into the command. From there, the user can enter one of the following options:

- “get” - Queries for a given a username using the `-u (--username)` option or a user ID given the `-i (--id)` option.
- “list” - Lists all the users within the Federation Server. This command takes no options.
- “delete” - Deletes a user from the federation. This command takes a `-u (--username)` to delete by username or `-i (--id)` to delete by user ID.
- “add” - Adds a new user to the federation. A username (`-u / --username`), a first name (`-f / --first_name`), and a last name (`-l / --last_name`) are required parameters. A password is required as well. It can be provided over the command line using the `-p / --password` option, or the user is prompted for the password. Optionally, the `-a / --is_admin` parameter can be specified to create the account as federation administrator.
- “set” - Changes information about a user within the federation. Arguments for username (`-u / --username`), first name (`-f / --first_name`), and last name (`-l / --last_name`) can be used to change information for the user. The `-a / --set_admin` and `-r / --remove_admin` options can be used to set or remove administrative privileges for the user. A new password for the user can be set by passing in `-P / --prompt_password` or `-p / --password` to set the password on the command line.

Cluster Management

To enable the cluster management portion of the `cbfed_cli` tool, the user must pass “cluster” as the first argument into the command. From there the user can enter one of the following options:

- “get” - Queries for a cluster given the cluster name (`-c / --cluster_name`) or cluster id (`-i / --id`). Configuration information for the cluster is displayed.
- “list” - Lists all the users from within the Federation Server. This command takes no options.
- “delete” - Deletes a user from the federation. This command takes either the cluster name (`-c / --cluster_name`) or the cluster id (`-i / --id`).
- “add” - Adds a new cluster to the federation. A cluster name (`-c / --cluster_name`), url (`-u, --url`), and the token type (`-t / --token_type`), whether the token is shared or not (individual) are required. The user can enter an optional description with `-d (--cluster_desc)` and can chose to enable SSL verification (`-v / --verify_ssl`). Optionally, the shared token (`-s / --shared_token`) can be supplied in the command line; otherwise if the token type is shared, the user is prompted for the API token.
- “set” - Changes information about an existing cluster in the federation. The cluster name (`-c / --cluster_name`), url (`-u / --url`), description (`-d / --cluster_desc`), token type (`-t / --token_type`) arguments can be used to change the cluster information. Enabling or disabling the cluster can be done with the (`-e / --enable`) and (`-d / --disable`) respectively. Enabling or disabling SSL verification can be done using (`-v / --verifyy_ssl`) or (`-n / --no_verify_ssl`) respectively.

Other Carbon Black Documentation

You may need some or all of the following Carbon Black documentation to accomplish tasks that are not covered in this guide. These documents, as well as other technical support solutions documents, are available on the [Carbon Black Customer Portal](#) website.

The technical solutions documents are a source of information that is maintained as a knowledge base.

Some of these documents are updated with every new released build, while others are updated only for minor or major version changes. Documents of particular interest include:

- *Carbon Black v5.1.1 - Enterprise Server Sizing Guide* – This describes performance and scalability considerations in deploying Carbon Black.
- *Carbon Black v5.1.1 - User Guide* – This describes how to use, install, and configure the Carbon Black Enterprise Server and Carbon Black sensors. It also describes the features of Carbon Black, and how to use the standard Carbon Black console interface.
- *Carbon Black v5.1.1 - Release Notes* – This includes information about new and modified features, issues resolved, general improvements in this release, and known issues and limitations. It also includes required or suggested preparatory steps before installing the server. Note that different versions or patches have their titles changed accordingly.
- *Carbon Black API* – Documentation for the Carbon Black API is hosted on Github at <https://github.com/carbonblack/cbapi>
- *Carbon Black Federation API* – <https://github.com/carbonblack/cbapi>
- *Carbon Black v5.1.1 - Cb.conf* – This describes the parameters in the main Carbon Black configuration file, `cb.conf`. You may want to modify some of the settings in this file.

Contacting Carbon Black Support

For your convenience, Carbon Black Technical Support offers several means of contact:

Technical Support Contact Options
Web: www.carbonblack.com
E-mail: support@carbonblack.com
Phone: 877.248.9098 (877.BIT9.098)
Fax: 617.393.7499
Hours: 8 a.m. to 8 p.m. EST

When you call or e-mail Carbon Black technical support, please provide the following information to the support representative:

Required Information	Description
Contact	Your name, company name, telephone number, and e-mail address.
Product version	Product name and version number.

Hardware configuration	Hardware configuration of the Carbon Black Federation Server (processor, memory, and RAM).
Document version	For documentation issues, specify the date and version of the manual you are using.
Problem	Action causing the problem, error message returned, and event log output (as appropriate).
Problem severity	Critical, serious, minor, or enhancement.