CARBON
BLACK

Cb Response QRadar Connector Guide

Document Date: September 2016

Document Version: 1.0

Copyrights and Notices

Carbon Black QRadar Connector Guide
Document Version 1.0

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

Contents

Chapter 1

QRadar Integration

The Cb Response app for IBM QRadar allows administrators to leverage the industry's leading Endpoint Detection and Response (EDR) solution to view, detect, and respond to endpoint activity from directly within the QRadar console. Once installed, administrators can use QRadar to:

- Access many powerful Cb Response features, such as process searches, endpoint isolation, and system status.
- Monitor the health of the Cb Response server using the QRadar Dashboard.
- Respond to issues by quarantining endpoints using the QRadar **Offenses** and **Log Activity** tab context menus.
- Use Cb Event Forwarder, which exports Cb Response events in Log Event Extended Format (LEEF) format to the QRadar server

Sections

Overview

Three main components are involved in QRadar integration:

- **Carbon Black DSM** – Install and configure Carbon Black Device Support Module (DSM) for QRadar, which normalizes the Carbon Black data into a format that QRadar can index. The Carbon Black DSM must be installed before events forwarded via the Cb Event Forwarder can be interpreted by the QRadar console.
  - For more information, see "Install Carbon Black DSM for QRadar" on page 7.
- **Cb Event Forwarder** – The Cb Event Forwarder allows data to be ingested into QRadar from Carbon Black and allows you to build dashboards in QRadar from Cb Response data. Installing the Cb Event Forwarder is optional but recommended.
  - For information on installing and using Cb Event Forwarder, see "Cb Event Forwarder" on page 16.
  - For information on configuring Cb Event Forwarder for the QRadar connector, see "Configure Cb Event Forwarder for QRadar Integration" on page 8.
- **Cb Response App for IBM QRadar** – Download and install the Cb Response app for IBM QRadar from the IBM X-Force Security App Exchange. This app allows you to monitor the Cb Response sensor from within QRadar.
  - For more information, see "Install Cb Response App for IBM QRadar" on page 8.

Install Carbon Black DSM for QRadar

Note	*The Carbon Black DSM must be installed before the other components.*

The Carbon Black Device Support Module (DSM) for QRadar normalizes the Carbon Black data into a format that QRadar can index. The Carbon Black DSM must be installed before events forwarded via the Cb Event Forwarder can be interpreted by the QRadar console.

For instructions on installing and configuring the Carbon Black DSM on your QRadar console, see the QRadar DSM Configuration Guide.

Configure Cb Event Forwarder for QRadar Integration

This section explains how to configure Cb Event Forwarder for the QRadar integration. For more information on installing and using Cb Event Forwarder, see "Cb Event Forwarder" on page 16.

1. If you are installing Cb Event Forwarder on a machine other than the Cb Response server, copy the RabbitMQ username and password into the following variables as follows:
   - `rabbit_mq_username`
   - `rabbit_mq_password`

   a. Copy the `<username>` value from `/etc/cb/cb.conf RabbitMQUser=<username>` to `/etc/cb/integrations/event-forwarder/cb-event-forwarder.conf rabbit_mq_username=<username>`.

   b. Copy the `<password>` value from `/etc/cb/cb.conf RabbitMQPassword=<password>` to `/etc/cb/integrations/event-forwarder/cb-event-forwarder.conf rabbit_mq_password=<password>`.

2. Enter the hostname or IP address in the following variable where the Cb Response server can be reached:

   ```
   cb_server_hostname
   ```

   If the Cb Event Forwarder is forwarding events from a Cb Response cluster, this variable should be set to the hostname or IP address of the Cb Response server master node.

3. Make sure the configuration is valid by running the Cb Event Forwarder in "check mode" as "root":

   ```
   cd /usr/share/cb/integrations/event-forwarder
   ./cb-event-forwarder -check
   ```

   If everything is working properly, you will see a message starting with "Initialized output".

   If there are any problems, errors will appear on your screen.

Install Cb Response App for IBM QRadar

Install the Cb Response app for IBM QRadar via the IBM X-Force Security App Exchange. This app allows administrators to leverage Cb Response to view, detect, and take action on endpoint activity from directly within the QRadar console. Once installed, the app allows administrators to access many of the powerful features of Cb Response, such as process searches, endpoint isolation, and system status, from within and in conjunction with QRadar.

Refer to IBM documentation for instructions on how to install this app. Once the Cb Response app for IBM QRadar is installed, continue with the following sections to configure and use the app.

Requirements

The Cb Response app for IBM QRadar requires the following:

- A functional Cb Response server (version 5.1 or later)
- IBM QRadar version 7.2.6 or later

No additional hardware requirements are required to run the app above the standard requirements for Cb Response and QRadar.

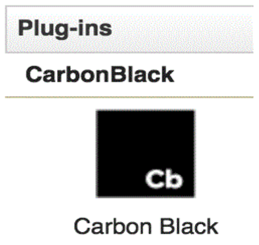
Configure the Cb Response App for IBM QRadar

Once the Cb Response app for IBM QRadar is installed, then you must configure it to connect to your Cb Response server.

Note	*The Cb Response app for IBM QRadar can connect to a single cluster (not multiple clusters).*

To configure the Cb Response app for IBM QRadar to connect to your Cb Response server:

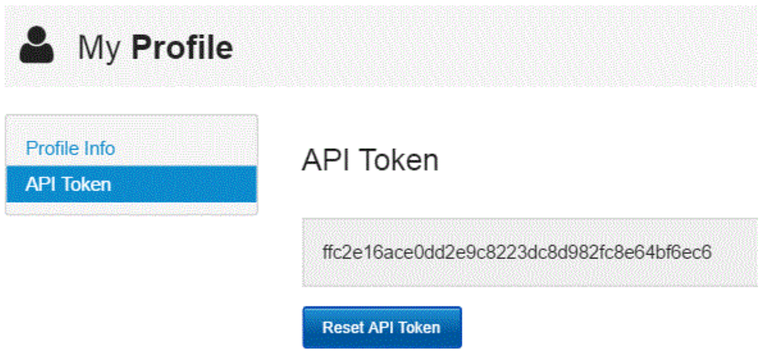
1. Navigate to the **Admin** tab of your QRadar server.
2. Scroll to the **Plug-ins** section at the bottom of the page.
3. Click the **Carbon Black** button.



4. Next, you must access the Cb Response server to retrieve the API token for the user you will use for this integration:

Note	*To enable all app features, the selected user for this integration must have Global Administrator rights on the Cb Response server.*

   a. Log into the Cb Response server with the appropriate account.
   b. When you are logged in, in the Cb Response console menu, select ***username*** > **My Profile**.
   c. On the **My Profile** page, click **API Token** in the left menu.
   d. Copy the displayed API token.



ffc2e16ace0dd2e9c8223dc8d982fc8e64bf6ec6

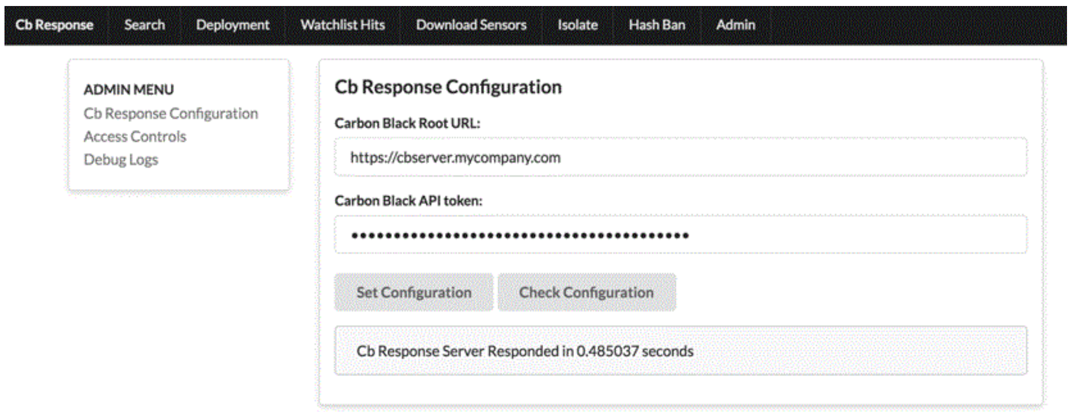
5. Return to the next Cb Response app for IBM QRadar page and do the following:
   a. Paste the API token into the **Carbon Black API Token** field.
   b. Enter the URL for your Cb Response server instance in the **Carbon Black Root URL** field. For example, enter:

      `https://cbserver.mycompany.com`

Note	*Do not place a trailing slash (/) on this URL.*

To test the configuration of the Cb Response app for IBM QRadar after setting the URL and API token:

1. Click the **Check Configuration** button.
2. If the connection succeeds, a grey status bar appears that reads "Cb Response Server Responded".
3. After the correct parameters are entered, click **Set Configuration** to save the new configuration. A grey status bar will appear that reads "Cb Response Configuration Set Successfully".



Cb Response Configuration

Carbon Black Root URL: https://cbserver.mycompany.com

Carbon Black API token: ••

Cb Response Server Responded in 0.485037 seconds

User Interface

The Cb Response app for IBM QRadar contains three major user interface components:

- **Dashboard**. For more information, see "Carbon Black Tab" **on page 12**.
- **Cb Response** tab. For more information, see "Dashboard Widget" on page 11.
- **Offenses** and **Log Activity** tab context menus. For more information, see "QRadar Context Menus" on page 15.

All users that are authorized for the Carbon Black capability have access to these components. QRadar admin users can also isolate sensors from the network and force sensors to send all queued data to the Cb Response server. For more information, see "Admin Sub-Tab" on page 14.

Dashboard Widget

To add the Carbon Black Dashboard widget to the QRadar Dashboard:

1. Click **Add item...** on the **Dashboard** action menu.
2. Select **Carbon Black** from the **Carbon Black** submenu.
3. The **Carbon Black Dashboard** should now appear.

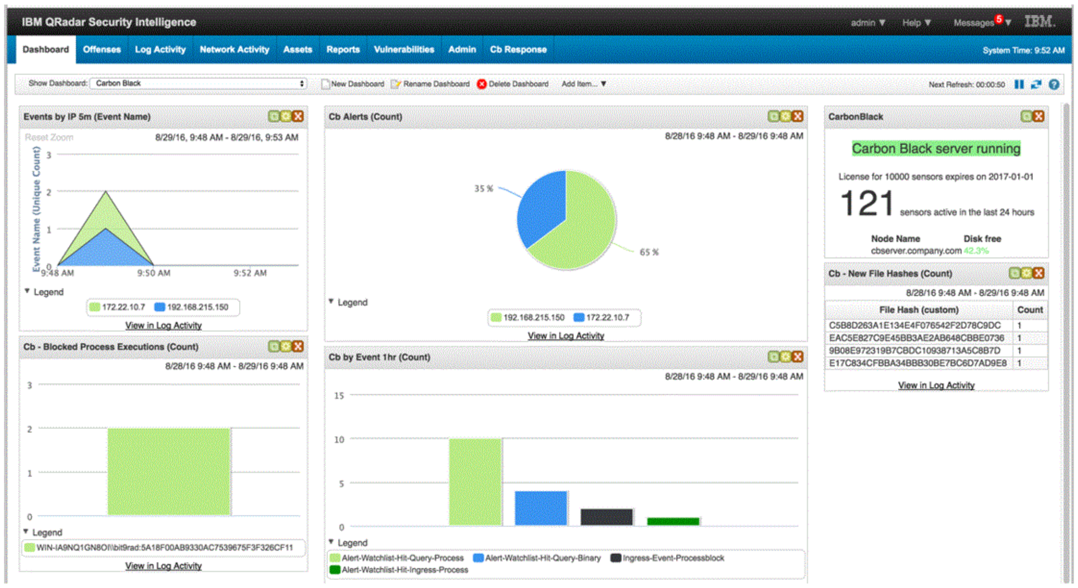
All users that are authorized for the Carbon Black capability have access to these components. QRadar users with "Admin" or "Forensics" privileges (which are configurable) can also isolate endpoints from the network, force sensors to send all queued data to the Cb Response server, and add binaries to Carbon Black's banned executable list.

Dashboard

Once the app is installed, a new dashboard named **Carbon Black** is available on your QRadar server. The **Carbon Black Dashboard** includes several panels that provide:

- An overview of your deployment status
- An overview of your alert activity
- A list of new file hashes seen on your network in the past 24 hours

To activate the dashboard, click the **Dashboard** tab in the QRadar user interface and select the **Carbon Black** dashboard in the "Show Dashboard" drop down menu at the top of the screen.


IBM QRadar Security Intelligence
admin
Help
Messages 5
IBM.
Dashboard
Offenses
Log Activity
Network Activity
Assets
Reports
Vulnerabilities
Admin
Cb Response
System Time: 9:52 AM
Show Dashboard: Carbon Black
New Dashboard
Rename Dashboard
Delete Dashboard
Add Item...
Next Refresh: 00:00:50
Events by IP 5m (Event Name)
Reset Zoom
8/29/16, 9:48 AM - 8/29/16, 9:53 AM
Event Name (Unique Count)
9:48 AM
9:50 AM
9:52 AM
Legend
172.22.10.7
192.168.215.150
View in Log Activity
Cb Alerts (Count)
8/28/16 9:48 AM - 8/29/16 9:48 AM
35 %
65 %
Legend
192.168.215.150
172.22.10.7
View in Log Activity
Cb - Blocked Process Executions (Count)
8/28/16 9:48 AM - 8/29/16 9:48 AM
Legend
WIN-IA9NQ1GN8OI\bit9rad:5A18F00AB9330AC7539675F3F326CF11
View in Log Activity
Cb by Event 1hr (Count)
8/28/16 9:48 AM - 8/29/16 9:48 AM
Legend
Alert-Watchlist-Hit-Query-Process
Alert-Watchlist-Hit-Query-Binary
Ingress-Event-Processblock
Alert-Watchlist-Hit-Ingress-Process
CarbonBlack
Cb - New File Hashes (Count)
View in Log Activity

Carbon Black server running

License for 10000 sensors expires on 2017-01-01

121 sensors active in the last 24 hours

Node Name	Disk free
cbserver.company.com	42.3%

8/28/16 9:48 AM - 8/29/16 9:48 AM

File Hash (custom)	Count
C5B8D263A1E134E4F076542F2D78C9DC	1
EAC5E827C9E45BB3AE2AB648CBBE0736	1
9B08E972319B7CBDC10938713A5C8B7D	1
E17C834CFBBA34BBB30BE7BC6D7AD9E8	1

Carbon Black Tab

The **Carbon Black** tab contains the following sub-tabs:

- **Search**
- **Deployment**
- **Watchlist Hits**
- **Download Sensors**
- **Isolate**
- **Hash Ban**
- **Admin**

Search Sub-Tab

To initiate searches across the Cb Response binary and process holdings:

1. Select the search type (process search or binary search) from the menu on the left.
2. Enter a Cb Response search query in the text field.
3. Click **Search**.
4. A new tab or window appears containing the query results in the Cb Response console.

Deployment Sub-Tab

The **Deployment** sub-tab displays the current status of the Cb Response server, including:

- License Information
- Sensor Statistics, including the number of active sensors in the past 24 hours
- Storage Statistics, summarizing the available disk space on the Cb Response server

Watchlist Hits Sub-Tab

The **Watchlist Hits** sub-tab contains the timestamp of the most recent hit per watchlist. This allows for quick viewing of which watchlists have most recently been seen..

Each watchlist is hyperlinked to the Cb Response server. Clicking on a watchlist in the QRadar interface displays a new page that contains the latest 10 hits from the watchlist in the Cb Response console. From here normal watchlist viewing and operations can be performed since the user is now looking at the Cb Response console.

Download Sensors Sub-Tab

The **Download Sensors** sub-tab allows you to download a sensor installer for any OS and sensor group configured by your Cb Response server.

To download a sensor installer:

1. Select the sensor group on the left-hand side of the page.
2. Click the operating system top to the right to download the appropriate sensor installer bundle. The download should begin shortly.

Isolate Sub-Tab

The **Isolate** sub-tab allows admin users to quarantine specific endpoints from the rest of the network and/or force endpoints to send all queued data immediately to the Cb Response server.

Isolation is performed using the Cb Live Response API. Once a sensor is isolated, it can only communicate with the Cb Response server until it is removed from isolation. You can then use Cb Live Response to retrieve additional files, kill processes, or take other remediation actions on the endpoint while it is isolated from the network.

Note	*Cb Live Response is disabled by default in the Cb Response server. To enable Cb Live Response, see the Cb Response User Guide.*

Synchronizing an endpoint signals the sensor to send all collected data to the Cb Response server upon the next check in. Be mindful of the amount of potential network traffic this action can produce.

Privileged users can find sensors to isolate and/or synchronize with the Cb Response server using one of these two methods.

To isolate and/or synchronize with the Cb Response server:

- Enter a sensor's IP address or an IP prefix (for example, enter `172.22.` to find all sensors in `172.22.0.0/16`) in the **Isolate** sub-tab of the **Cb Response** tab.
- Right-click on any IP address in the QRadar user interface and select the **Cb Response Sensor Action**.

Once the results are returned from the Cb Response server, a list of sensors that match that IP address or range in the Cb Response server are displayed in a table underneath the search box.

To isolate a sensor:

1. Click the red **Isolate** button on the row corresponding to the sensor that you want to isolate.
2. The status is updated in the sensor table when the sensor has checked in and the isolation status has been successfully applied to the endpoint.

To synchronize a sensor's state with the Cb Response server:

Click the **Sync Sensor** button on the row corresponding to the sensor you want to synchronize.

Hash Ban Sub-Tab

The **Hash Ban** sub-tab provides a view of the currently banned MD5 hashes in the Cb Response server and allows privileged users to add new hashes to or remove existing hashes from the list of banned hashes.

To view the list of banned hashes:

1. Click the **Search** button inside the **Search Hashes** box. You can also search for a subset of hashes by typing the first few characters of the MD5sum in the **Search** field and clicking **Search Hashes**.
2. The search results will appear in a new table underneath the **Search** field.

To change the status of the hash:

Click the **Ban** or **Unban** buttons on the corresponding row for the MD5 hash.

To add a new hash into the ban list:

1. Paste in the MD5 hash into the **Ban Hash** field on the right.
2. Click the large **Ban** button next to the field.

Admin Sub-Tab

The **Admin** sub-tab allows you to configure the app. Only QRadar users with "Admin" privileges can access and change settings in this tab.

The **Admin** interface provides access to three app Admin functions:

- **Cb Response Configuration** - Allows you to adjust the URL and API token associated with your Cb Response server or cluster.
- **Access Controls** - Allows you to adjust the QRadar privileges required to access the isolate endpoint, sync sensor, and hash banning functionalities in the app. You can choose to:
  - Only allow functionality through users with QRadar "Admin" privileges.
  - Only allow functionality through users in the "Forensics" group.
  - Disable the functionality entirely.
- **Debug Logs** - See "Debug Logs" on page 15.

QRadar Context Menus

The Cb Response app for IBM QRadar provides a context menu that provides you with context about specific IP addresses from a Log Activity or Offense entry. The app creates three right-click menu options for any IP address in the **Log Activity / Offense** tabs. These options are located under the **More Options** submenu:

- **Cb Response Process Search** - Search all of the Cb Response holdings for any processes that contacted this IP address. This is useful for gathering additional context around potentially malicious IP addresses that have contact with internal systems. Cb Response provides:
  - The name of the process that made the network connection
  - Details on how the process was launched
  - Details on which binary was executed
  - Details on how the binary was written to disk
- **Cb Sensor Action -** Queries the Cb Response server for any sensors that are associated with the selected IP address. If any sensors are identified, you can select one or more sensors to isolate from the network or synchronize. For more information on the network quarantine or synchronization feature, see "Isolate Sub-Tab" on page 13.

Debug Logs

Any time an error is encountered in the Cb Response QRadar app, a unique identifier is generated to track the error in the debug logs. You can view the debug logs to obtain additional detail under the **Admin** tab.

To view debug logs:

1. Click the **Cb Response** tab.
2. Select the **Admin** tab.
3. Select **Debug Logs**.
4. The most recent log messages should be displayed.
5. Optionally, search for the unique identifier given in an error message to obtain additional debug information for that error.

Chapter 2

Cb Event Forwarder

Once the Cb Event Forwarder RPM is installed on a target system, you can configure and run multiple instances of Cb Event Forwarder to push Cb Response data to several destinations concurrently. This chapter explains how to do this.

Sections

Overview

The Cb Event Forwarder is a standalone service that listens in on the Cb Response bus and exports events (watchlist/feed hits, newly seen binaries, and raw endpoint events, if configured) in a normalized JSON format. The events can be saved to a file, delivered to a network service, or archived automatically to an Amazon AWS S3 bucket. These events can be consumed by any external system that accepts JSON, including the Cb Response and Cb Protection BigFix connectors.

The list of events to collect is configurable. By default, all watchlist/feed hits, alerts, binary notifications, and raw sensor events are exported into JSON. The configuration file for the connector is stored in:

```
/etc/cb/integrations/event-forwarder/cb-event-forwarder.conf
```

Requirements

The following are Cb Event Forwarder installation requirements:

- The Cb Event Forwarder can be installed on any 64-bit Linux machine running CentOS 6.x.
- The Cb Event Forwarder can be installed on the same machine as the Cb Response server or another machine. However, if you are forwarding events from a Cb Response cluster, it is recommended you install Cb Event Forwarder on a separate machine.

Install Cb Event Forwarder RPM

To install and configure the Cb Event Forwarder RPM, perform these steps as "root" on your target Linux system:

1. Install the CbOpenSource repository if it is not already installed:

   ```
   cd /etc/yum.repos.d
   curl -O https://opensource.carbonblack.com/release/x86_64/
   CbOpenSource.repo
   ```

2. Install the Cb Event Forwarder RPM using Yum:

   ```
   yum install cb-event-forwarder
   ```

Note	*Once the Cb Event Forwarder RPM is installed on a target system, you can configure and run multiple instances of Cb Event Forwarder to push Cb Response data to several destinations concurrently. For more information, see Chapter 3, "Run Concurrent Cb Event Forwarder Instances."*

Configure Cb Event Forwarder

The Cb Event Forwarder configuration is unique for each connector. To configure Cb Event Forwarder for QRadar, see "QRadar Integration" on page 6.

Configure Cb Response

By default, Cb Response publishes `feed.*` and `watchlist.*` events over the bus. The default is acceptable for the QRadar integration. For more information on these events, see:

https://github.com/carbonblack/cb-event-forwarder/blob/master/EVENTS.md

To capture raw sensor events (network connections, file modifications, registry modifications, etc.) or the `binaryinfo.*` notifications, you must enable these features in `/etc/cb/cb.conf`:

- If you are capturing raw sensor events, then you must edit the `DatastoreBroadcastEventTypes` option in `/etc/cb/cb.conf` to enable broadcast of the raw sensor events you want to export.

Note	*Enabling this option for high-volume event types (file modifications, registry modifications, etc.) causes a performance impact on the Cb Response server. See "(Optional) Enabling the Raw Sensor Event Exchange" on page 18 to enable the Raw Sensor Event Exchange to mitigate this impact.*

- If you are capturing binary observed events, then you must edit the `EnableSolrBinaryInfoNotifications` option in `/etc/cb/cb.conf` and set it to `True`.

By default, the Message Bus listens on port 5004. Make sure firewall rules allow for incoming TCP connections to this port on the Cb Response server.

(Optional) Enabling the Raw Sensor Event Exchange

There is a performance impact when exporting all raw sensor events onto the Cb Response bus. As a result, avoid exporting all of the events. The performance impacts occur when large numbers of events are broadcasted on the bus by enabling the `DatastoreBroadcastEventTypes` setting.

The largest impact is encountered when large numbers of endpoints are checking into a server and file modification (filemod) or registry modification (regmod) events are enabled in the `DatastoreBroadcastEventTypes` setting.

To mitigate this impact, Cb Response version 5.2 introduces a new feature called the Raw Sensor Event Exchange. To enable the forwarding of all raw sensor events to the Event Forwarder, Carbon Black strongly suggests using the Raw Sensor Event Exchange instead of enabling individual event types through the `DatastoreBroadcastEventTypes` setting.

To enable the Raw Sensor Event Exchange:

1. Reset `DatastoreBroadcastEventTypes` to its initial (empty) value by commenting out the `DatastoreBroadcastEventTypes` line in the `/etc/cb/cb.conf` file.
2. Add the following line to the `/etc/cb/cb.conf` file:

   ```
   EnableRawSensorDataBroadcast=True
   ```

Apply the Changes to the Cb Response Server

You will need to restart the Cb Response server if any variables were changed in `/etc/cb/cb.conf`.

If you have a single server, log in as root and run the following:

```
service cb-enterprise restart
```

If you have a cluster, follow these steps:

1. Distribute the `DatastoreBroadcastEventTypes`, `EnableSolrBinaryInfoNotifications`, and/or `EnableRawSensorDataBroadcast` settings to the `/etc/cb/cb.conf` configuration file on all minion nodes.
2. Restart the cluster using the `/usr/share/cb/cbcluster restart` command.

Start and Stop Cb Event Forwarder Service

Once Cb Event Forwarder is installed, it is managed by the Upstart init system that comes with CentOS 6.x. You can control the Cb Event Forwarder service using the `initctl` command as follows:

- To start the service, execute this command:

  ```
  initctl start cb-event-forwarder
  ```

- To stop the service, execute this command:

  ```
  initctl stop cb-event-forwarder
  ```

The Cb Event Forwarder service is configured to start automatically on system boot.

Forward Events

The Cb Event Forwarder must be configured to forward Cb Response events in JSON or LEEF format to the Cb Response BigFix connector.

To forward Cb Response events to the connector:

1. Modify `/etc/cb/integrations/event-forwarder/cb-event-forwarder.conf` to specify the output protocol type:

   ```
   output_type=tcp
   ```

2. Change the destination network address and port to that of the connector. By default these values are `localhost` and `9999`. For more information, see "Configure Cb Event Forwarder" on page 18.

   For example, change the following:

   ```
   tcpout=<ipaddress>:<port>
   ```

   to:

   ```
   tcpout=localhost:9999
   ```

3. Change the output format to `json` or `leef` the configuration file:

   ```
   output_format=json
   output_format=leef
   ```

Logging and Diagnostics

The Cb Event Forwarder logs to the following directory:

```
/var/log/cb/integrations/cb-event-forwarder
```

The following is an example of a successful startup log:

```
2015/12/07 12:57:26 cb-event-forwarder version 3.0.0 starting
2015/12/07 12:57:26 Interface address 172.22.10.7
2015/12/07 12:57:26 Interface address fe80::20c:29ff:fe85:bcd0
2015/12/07 12:57:26 Configured to capture events:
[watchlist.hit.# watchlist.storage.hit.# feed.ingress.hit.#
feed.storage.hit.# feed.query.hit.# alert.watchlist.hit.#
ingress.event.process ingress.event.procstart
ingress.event.netconn ingress.event.procend
ingress.event.childproc ingress.event.moduleload
ingress.event.module ingress.event.filemod ingress.event.regmod
binaryinfo.# binarystore.file.added]
2015/12/07 12:57:26 Initialized output: File /var/cb/data/
event_bridge_output.json
2015/12/07 12:57:26 Diagnostics available via HTTP at http://
cbtest:33706/debug/vars
2015/12/07 12:57:26 Starting AMQP loop
2015/12/07 12:57:26 Connecting to message bus...
2015/12/07 12:57:26 Subscribed to watchlist.hit.#
2015/12/07 12:57:26 Subscribed to watchlist.storage.hit.#
2015/12/07 12:57:26 Subscribed to feed.ingress.hit.#
2015/12/07 12:57:26 Subscribed to feed.storage.hit.#
2015/12/07 12:57:26 Subscribed to feed.query.hit.#
2015/12/07 12:57:26 Subscribed to alert.watchlist.hit.#
2015/12/07 12:57:26 Subscribed to ingress.event.process
2015/12/07 12:57:26 Subscribed to ingress.event.procstart
2015/12/07 12:57:26 Subscribed to ingress.event.netconn
2015/12/07 12:57:26 Subscribed to ingress.event.procend
2015/12/07 12:57:26 Subscribed to ingress.event.childproc
2015/12/07 12:57:26 Subscribed to ingress.event.moduleload
2015/12/07 12:57:26 Subscribed to ingress.event.module
2015/12/07 12:57:26 Subscribed to ingress.event.filemod
2015/12/07 12:57:26 Subscribed to ingress.event.regmod
2015/12/07 12:57:26 Subscribed to binaryinfo.#
2015/12/07 12:57:26 Subscribed to binarystore.file.added
2015/12/07 12:57:26 Starting 4 message processors
```

In addition to the log file, the Cb Event Forwarder service starts an HTTP service for monitoring and debugging. The URL is available in the log file (see the `Diagnostics available` line above). The port is configurable through the `http_server_port` setting in the `cb-event-forwarder.conf` file. You can visit the Diagnostics page at the URL provided in the log file to track the performance and availability of the Cb Event Forwarder.

Note	*To reach the diagnostics, make sure that the port (default 33706) is open for incoming traffic on any firewalls between your host and the server running the Cb Event Forwarder.*

Chapter 3

Run Concurrent Cb Event Forwarder Instances

Once the Cb Event Forwarder RPM is installed on a target system, you can configure and run multiple instances of Cb Event Forwarder to push Cb Response data to several destinations concurrently. This chapter explains how to do this.

Sections

Check Prerequisites

Encusre that the Cb Event Forwarder (cb-event-forwarder) RPM is installed on your system:

```
[root@cbtest ~]# rpm -q -a cb-event-forwarder
cb-event-forwarder-3.2.1-1.x86_64
```

The command should return the version number of the currently installed Cb Event Forwarder (version 3.2.1 in the example above).

If the command returns no output, then follow the instructions to install the Cb Event Forwarder from the YUM repository in "Cb Event Forwarder" on page 16.

Create a New Cb Event Forwarder Configuration

1. Create a new location where the Cb Event Forwarder configuration and log files will be located. The files can stored at any location with disk space on your target machine. For example, in the following example, we use the /opt/cb-event-forwarder/analytics directory to allow this instance of the Cb Event Forwarder to push data into our internal analytics system:

   ```
   [root@cbtest ~]# mkdir -p /opt/cb-event-forwarder/analytics
   [root@cbtest ~]# mkdir -p /opt/cb-event-forwarder/analytics/log
   ```

2. Copy the configuration file into the new location:

   ```
   [root@cbtest ~]# cp /etc/cb/integrations/event-forwarder/cb-
   event-forwarder.conf /opt/cb-event-forwarder/analytics/
   ```

3. Edit the `/opt/cb-event-forwarder/analytics/cb-event-forwarder.conf` file to reflect the configuration options required for this new instance of the Cb Event Forwarder.

Note	*Make sure that you change the `http_server_port` from `33706` to another value if you want to monitor the status of the Cb Event Forwarder.*

4. Duplicate the Cb Event Forwarder startup script. In the following example, we call out the new `cb-event-forwarder-analytics` service to indicate this Cb Event Forwarder instance can connect to the analytics service:

   ```
   [root@cbtest ~]# cp /etc/init/cb-event-forwarder.conf /etc/
   init/cb-event-forwarder-analytics.conf
   ```

5. Create a startup script by editing the `/etc/init/cb-event-forwarder-analytics.conf` file to point to our new configuration location and log file location. The changes that have to be made are showcased in boldface below.

```
description "Carbon Black event forwarder - analytics service"
author "dev-support@bit9.com"

start on (started network)
stop on runlevel [!2345]

respawn

pre-start script
/usr/share/cb/integrations/event-forwarder/cb-event-forwarder -
check /opt/cb-event-forwarder/analytics/cb-event-forwarder.conf
&> /opt/cb-event-forwarder/analytics/log/cb-event-
forwarder.startup.log
end script

exec /usr/share/cb/integrations/event-forwarder/cb-event-
forwarder /opt/cb-event-forwarder/analytics/cb-event-
forwarder.conf &> /opt/cb-event-forwarder/analytics/log/cb-
event-forwarder.log
```

Start the New Service

After creating the the init script in the previous section, you should be able to start the new Cb Event Forwarder instance as follows:

```
[root@cbtest ~]# initctl start cb-event-forwarder-analytics

cb-event-forwarder-analytics start/running, process 32326
```

If you encounter errors, check the following log file for details:

```
/opt/cb-event-forwarder/analytics/log/cb-event-
forwarder.startup.log
```