CARBON
BLACK

Cb Response Connector Guide

Release 6.1

Document Date: August, 2017

Copyrights and Notices

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

Cb Response Connector Guide
Document Revision Date: August 1, 2017

Contents

About This Book

This preface describes the contents of this publication, *Cb Response Connector Guide*.

Sections

What This Document Covers

This document describes how to install, configure and maintain Carbon Black connectors. A connector enables communication between a third-party product and Cb Response server.

This document explains how to integrate with the following connectors:

- Check Point
- Fidelis
- LastLine
- WildFire
- ThreatConnect
- Cyphort
- QRadar
- Splunk
- BigFix

The following table summarizes the contents of this guide:

	Chapter	Description
1	Integrating with Third-Party Security Products	Describes how to connect a local or cloud-based third-party product to Cb Response by using a network integration.
2	Check Point Integration	Provides integration with an on-premises Check Point Software Technologies, Ltd. device for correlating Check Point alerts with data that is collected by Cb Response. For more information on Check Point, see www.checkpoint.com.
3	Fidelis Integration	Provides bi-directional integration with an on-premises device by General Dynamics Fidelis Cybersecurity Solutions for correlating Fidelis alerts in the Cb Response server and returning Cb Response metadata to Fidelis. More information about Fidelis can be found at www.fidelissecurity.com. You can also watch a short video explaining the integration at www.youtube.com/watch?v=2iwczFX162U.
5	Lastline Integration	Provides integration with the Lastline, Inc. service for checking the reputation of binary files. More information about the Lastline service can be found at www.lastline.com/platform/enterprise.
6	WildFire Integration	Provides integration with the Palo Alto Networks WildFire cloud service for checking the reputation of binary files. More information about the Palo Alto Networks WildFire service can be found at www.paloaltonetworks.com/products/technologies/wildfire.html.

	Chapter	Description
7	ThreatConnect Integration	Provides integration with ThreatConnect by retrieving IOCs from specified communities. To support this integration, Cb Response provides an out-of-band connector that communicates with the ThreatConnect API via port 6100 (default).
8	Cyphort Integration	Provides integration with Cyphort for inspection, analysis, and correlation of suspicious binaries discovered at the endpoint. Cyphort Core is a secure threat analysis engine that leverages Cyphort's multi-method behavioral detection technology and threat intelligence.
9	QRadar Integration	Provides integration with QRadar. 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.
10	Splunk Integration	Provides integration with Splunk, which allows administrators to view, detect, and respond to endpoint activity from directly within the Splunk console.
11	BigFix Integration	BigFix Integration provides integration between IBM BigFix and Cb Response. Allows administrators to deploy a full endpoint security solution to detect, contain, investigate, and remediate security threats and attacks on endpoint across the enterprise.
12	Cb Event Forwarder	Introduces Cb Event Forwarder, 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.
13	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.

Other Documentation

You may need some or all of the following documentation to accomplish tasks that are not covered in this guide. These documents, as well as other technical support solution documents, are available on the Carbon Black User eXchange.

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.

- *Cb Response User Guide* – Describes the Cb Response product and explains how to use all of its features and perform administration tasks.
- *Cb Response Integration Guide* – Provides information for administrators who are responsible for integrating Cb Response with various tools, such as Cb Protection, EMET, VDI, SSO, and more.
- *Cb Response Server/Cluster Management Guide* – Describes how to install, manage, backup/restore, etc. a Cb Response server/cluster. This guide is for on-premises Cb Response installations only.
- *Cb Response Server Sizing Guide (OER)* – Describes performance and scalability considerations in deploying a Cb Response server.
- *Cb Response Server Configuration (cb.conf) Guide* – Provides all of the `cb.conf` configuration file functions, descriptions, and parameters.
- *Cb Response Unified View Server Guide* – Describes how to install and use the Cb Response Unified View server, which is a standalone server that ties multiple Cb Response servers together and provides a single interface. The Unified View interface allows users to perform queries across multiple Cb Response servers with a unified result set. The Unified View server also allows users to manipulate the full functionality of a single Cb Response server.
- *Cb Response Release Notes* – Provides information about new and modified features, issues resolved and general improvements in this release, and known issues and limitations. It also includes required or suggested preparatory steps before installing the server.

Community Resources

The Carbon Black User eXchange website at https://community.carbonblack.com provides access to information shared by Carbon Black customers, employees and partners. It includes information and community participation for users of all Carbon Black products, including Cb Response, Cb Response, and Cb Defense.

When you login to this resource, you can:

- Ask questions and provide answers to other users' questions.
- Enter a "vote" to bump up the status of product ideas.
- Download the latest user documentation.
- Participate in the Carbon Black developer community by posting ideas and solutions or discussing those posted by others.
- View the training resources available for Carbon Black products.

You must have a login account to access the User eXchange. Contact your Technical Support representative if you need to get an account.

Contacting Support

For your convenience, Carbon Black Technical Support offers several channels for resolving support questions:

Technical Support Contact Options
Carbon Black User eXchange: https://community.carbonblack.com
Email: support@carbonblack.com
Phone: 877.248.9098
Fax: 617.393.7499

Reporting Problems

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

Required Information	Description
Contact	Your name, company name, telephone number, and email address
Product version	Product name and version number
Hardware configuration	Hardware configuration of the server or computer the product is running on (processor, memory, and RAM)
Document version	For documentation issues, specify the version of the manual you are using. The date and version of the document appear on the cover page, or for longer manuals, after the Copyrights and Notices section of the manual.
Problem	Action causing the problem, error message returned, and any other appropriate output
Problem severity	Critical, serious, minor, or enhancement

Chapter 1

Integrating with Third-Party Security Products

This chapter describes how to enable communication between a local or cloud-based third-party product to Cb Response by using a network integration called a connector, provided by Carbon Black.

Sections

Overview

Threat intelligence feeds report IOCs to the Cb Response server. This information is added to endpoint file and process data. These feeds can be added to the Cb Response server in several ways:

- **Cb Threat Intel** – Cb Threat Intel provides feeds from Cb Protection and third-party partners to the Cb Response server. For more information, see the "Threat Intelligence Feeds" chapter in the *Cb Response User Guide*.
- **Manually Added Feeds** – You can create a feed and manually add it on the Threat Intelligence Feeds page by providing a URL and configuration information. See https://developer.carbonblack.com/reference/enterprise-response/5.1/threat-intelligence-feeds/ for instructions on creating a feed.
- **Connectors** – Connectors provide a Threat Intelligence Feed based on a network integration to a local or cloud-based third-party device. Connectors are provided by Carbon Black and installed separately from the Cb Response software.

Connectors report IOCs based on the capabilities of their respective devices. For example, one connector might report suspicious network traffic activity while another one reports the results of binary detonations. Some connectors also send metadata from the Cb Response server back to the connected device or service.

Alerts from connected devices and services are received in the form of IPv4 and DNS addresses, and MD5 hashes. The listener parses the event data from the third-party source into Cb Response JSON feed format for presentation to a feed that can be made available through the Cb Response console interface.

Connectors provide log files to record key actions, including errors, that occur in the connection between the third-party device or service and the Cb Response server. These log files can be useful for troubleshooting. The names and locations of the log files are described in the device-specific sections below.

All of the Cb Response connectors use the Open API exposed by the product. More information on the API can be found on the Developer Network website at https://developer.carbonblack.com. In addition, many of the Cb Response connectors are open sourced, and more are expected to be open sourced in the future. More information on open source connectors can be found on our GitHub organization page at https://github.com/carbonblack.

Communications Requirements

Connectors require two clear communication paths:

- One to the third-party cloud or on-premises network device
- One to the Cb Response server

All connectors can be installed directly on the Cb Response server, so in most cases, only one-way data communication to the third-party network device is required.

Note If you choose to install one of the network integration connectors on a system other than the Cb Response server, you will need to open the appropriate port in the firewall to allow the Cb Response server to access to feed from that device or service.

Setting up the Open Source Connector Repository

Most connectors are available through the *CbOpenSource* Yum repository. The configuration information for the open source repository must be located in a file named as follows:

```
/etc/yum.repos.d/CbOpenSource.repo
```

Before installing any connectors, you must first set up the connector repository. You can do this automatically or manually.

To set up the open source connector repository on a Cb Response server:

1. Change directories to:

   ```
   /etc/yum.repos.d
   ```

2. Execute the following command:

   ```
   curl -O https://opensource.carbonblack.com/CbOpenSource.repo
   ```

If you are unable to use the curl command to automatically create the `CbOpenSource.repo` file, you can alternatively create this file manually.

To manually set up the open source connector repository on a Cb Response server:

1. Change directories to:

   ```
   /etc/yum.repos.d
   ```

2. Create the following file:

   ```
   CbOpenSource.repo
   ```

3. Add the following content to the file:

   ```
   [CbOpenSource] name=CbOpenSource

   baseurl=https://opensource.carbonblack.com/release/x86_64
   gpgcheck=0

   enabled=1 metadata_expire=60 sslverify=1
   ```

4. Save the file.

In either case, the `CbOpenSource.repo` file is created and you will be able to install the network integration connectors you select.

Chapter 2

Check Point Integration

Cb Response provides integration with an on-premises Check Point Software Technologies, Ltd. device for correlating Check Point alerts with data that is collected by Cb Response. For more information on Check Point, see www.checkpoint.com.

To support this integration, Cb Response provides an out-of-band connector that receives alerts from the Check Point device and communicates that information with the Cb Response server through a listener on port 7000. If the connector is installed on the Cb Response server, no special firewall changes are necessary.

Sections

Check Point Installation

Install the Cb Response out-of-band connector to receive Check Point alerts on the Cb Response server.

To install the connector:

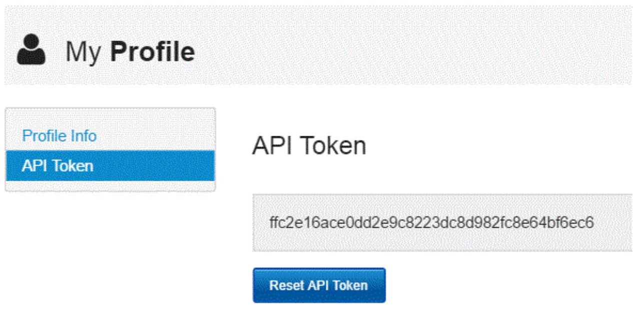
1. Ensure that the following prerequisites are met:
   - Cb Response server version 5.0 or later is installed with Internet connectivity.
   - Check Point is installed on a device.
2. The Check Point connector requires a special Yum repository configuration for installation. Create and save a new file in the following location:
   ```
   /etc/yum.repos.d/CheckPoint.repo
   ```
3. Add the following content:
   ```
   [CheckPoint]

   name=CheckPoint
   baseurl=https://yum.carbonblack.com/enterprise/integrations/
   checkpoint/x86_64

   gpgcheck=0
   enabled=1

   metadata_expire=60
   sslverify=1

   sslclientcert=/etc/cb/certs/carbonblack-alliance-client.crt
   sslclientkey=/etc/cb/certs/carbonblack-alliance-client.key
   ```
4. Verify the Yum configuration and install the Check Point connector by running these commands:
   ```
   yum info python-cb-checkpoint-bridge
   yum install python-cb-checkpoint-bridge
   ```
5. In the Cb Response console interface, retrieve the API key for the user you intend to use for this integration:
   a. On the Cb Response console, log in with the account you selected. (This user must have administrative rights on the server.)
   b. In the top-right corner of the Cb Response console, select ***Username*** **> My Profile**.
   c. On the **My Profile** page, choose **API Token** on the left menu. Leave this page open so the information is available to you.



```
ffc2e16ace0dd2e9c8223dc8d982fc8e64bf6ec6
```

6. Edit the Check Point connector configuration file. The Check Point connector configuration file is located at:

   ```
   /etc/cb/integrations/carbonblack_checkpoint_bridge/
   carbonblack_checkpoint_bridge.conf
   ```

   a. Update the `carbonblack_server_url` option to set the URL of the Cb Response server.
   b. Update the `carbonblack_server_token` option by copying and pasting the API Token string displayed in the **API Token** field in the Cb Response console (see step 5).
   c. The remainder of the options are documented in the configuration file and can be customized as needed to match specific requirements.
   d. Save the configuration file.

Important Do not start the connector yet. The initial Check Point configuration must be completed before starting the Cb Response-Check Point connector.

Check Point Configuration

The following sections describe the steps to enable the Check Point Log Export API (LEA) server and to configure the Cb Response LEA application.

Enable the LEA Server

Complete these steps if the LEA server on the Check Point device has not already been configured.

To enable the LEA server:

1. Login to the terminal access provided by the Check Point device. Depending on its configuration, expert mode might be required after logging in.
2. Edit the `*$FWDIR/conf/fwopsec.conf` file by adding the following configuration options:

   ```
   lea_server auth_port 18184
   lea_server auth_type sslca
   ```

3. Save the configuration file.
4. Restart the Check Point Firewall service:

   ```
   cpstop
   cpstart
   ```

Modify the Firewall Rule Set

Modify the firewall rule set to enable the Check Point LEA server to connect to the Cb Response host that is running the Cb Response Check Point connector.

To modify the firewall rule set:

From the host that is running the Cb Response Check Point connector, create a firewall rule to allow connections for the following services:

```
FW1_ica_pull
FW1_lea
```

Register the OPSEC LEA Client Application

Register the OPSEC (Operations Security) LEA client application to create the certificates needed for secure LEA communications.

To register the OPSEC LEA client application:

1. Using the Check Point Smart Dashboard, navigate to **Manage > Servers and OPSEC Applications**.
2. Add a new entry for the Cb Response OPSEC LEA application:
   a. Specify a name for the application registration, such as `CB_LEA_APP`.
   b. Specify a new host, such as `CB_Bridge`.
   c. Ensure that the LEA option in **Client entities** is selected.
   d. Click the **Communication** button, enter a one-time password, and then select **Initialize**. This creates the certificates that are required for secure LEA communications.
   e. Close the **Communication** window. A Domain Name (DN) should appear. If it does not appear, the DN should appear after closing the **Application** window and then reopening it.

Note Take note of this Domain Name. You will need it in later steps.

Retrieve the OPSEC LEA Application Certificate

Retrieve the OPSEC LEA application certificate from the Check Point server to add to the Cb Response Check Point connector host computer.

To retrieve the OPSEC LEA application certificate:

1. On the Cb Response Check Point connector host, use the `opsec_pull_cert` utility to pull the application certificate from the Check Point server. The utility is located at:

   ```
   /usr/share/cb/integrations/carbonblack_checkpoint_bridge/
   opsec_lea/tools/
   ```

   Run the following command to retrieve the certificate:

   ```
   opsec_pull_key -h [checkpoint server ip or name] -n [opsec lea
   application registration name] -p [one-time password used
   above]
   ```

   For example:

   ```
   opsec_pull_key -h 192.168.1.10 -n CB_LEA_APP -p mypassword
   ```

   This creates a file named `opsec.p12` in the current directory.

2. Place the `opsec.p12` file in the following directory:

   ```
   /usr/share/cb/integrations/carbonblack_checkpoint_bridge/
   opsec_lea/
   ```

Create and Retrieve the Host Authentication Key

Create a secret key to retrieve the authentication keys from the Check Point server, and then add the authentication keys to the Cb Response Check Point connector host computer.

To create and retrieve the host authentication key:

1. On the Check Point server terminal access system, run the following command to create the keys:

   ```
   fw putkey -opsec -ssl [ip address of bridge host]
   ```

   The command prompts for a secret key that is used to retrieve the authentication keys.
2. On the Cb Response Check Point connector host, from the utility location, run the following command to retrieve the keys. The utility is located here:

   ```
   /usr/share/cb/integrations/carbonblack_checkpoint_bridge/
   opsec_lea/tools/
   ```
3. Run the following command:

   ```
   opsec_putkey -ssl -port 18184 [ip address of Check Point
   server]
   ```
4. The previous command generates two files in the current directory:

   ```
   sslauthkeys.C
   ```

   ```
   sslsess.C.
   ```

   Place these files in the following directory:

   ```
   /usr/share/cb/integrations/carbonblack_checkpoint_bridge/
   opsec_lea/
   ```

Configure the Cb Response OPSEC LEA

Configure the Cb Response OPSEC LEA file on the Cb Response Check Point connector host.

To configure the Cb Response OPSEC LEA:

1. On the Cb Response Check Point connector host, edit the following file:

   ```
   /usr/share/cb/integrations/carbonblack_checkpoint_bridge/
   opsec_lea/cb_checkpoint_opsec_lea.conf
   ```

   Set the options as shown below:

   ```
   opsec_sic_name "[DN from application registration]"
   opsec_sslca_file "opsec.p12"
   opsec_shared_local_path "/usr/share/cb/integrations
   carbonblack_checkpoint_bridge/opsec_lea/"
   opsec_sic_policy_file "cb_checkpoint_opsec_lea_sic_policy.conf"
   lea_server ip [Check Point server ip address]
   lea_server auth_port 18184
   ```

```
lea_server auth_type sslca
lea_server opsec_entity_name "[DN of Check Point server]"
```

Note To obtain the server Domain Name (DN), double-click the main Check Point server object. The DN displays under **Test SIC Status**.

2. Save changes to the configuration file.
3. Start the connector service:
   ```
   /etc/init.d/cb-checkpoint-bridge start
   ```
4. Examine the Check Point connector log to verify that the service is running normally:
   ```
   /var/log/cb/integrations/carbonblack_checkpoint_bridge/
   carbonblack_checkpoint_bridge.log
   ```

Start and Stop the Check Point Connector

To start the Check Point connector:

```
# service /etc/init.d/cb-checkpoint-bridge start
```

To stop the Check Point connector:

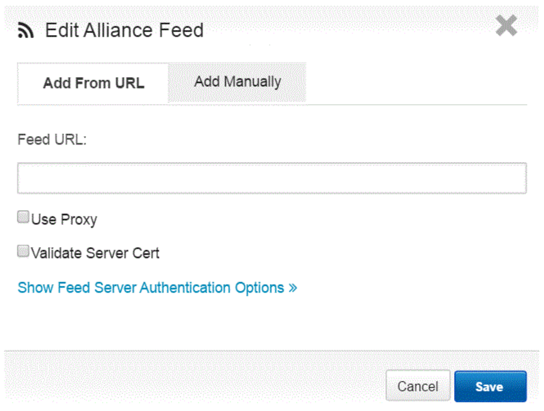
```
# service /etc/init.d/cb-checkpoint-bridge stop
```

Add the Check Point Feed to Cb Response

When the Check Point connector service is running, you can add the Check Point feed to the threat intelligence feeds on the Cb Response server. For more information, see the “Threat Intelligence Feeds” chapter in the *Cb Response User Guide*.

To add a new Check Point feed:

1. Login to the Cb Response console.
2. In the left navigation menu, select **Threat Intelligence**.
3. Click **Add New Feed**.
4. The **Edit Alliance Feed** window opens:



5. In the **Feed URL** field, enter the URL:

   ```
   http://[bridge host]:[listener_port from bridge config]/
   checkpoint/json
   ```

   For example:

   ```
   http://127.0.0.1:7000/checkpoint/json
   ```

Chapter 3

Fidelis Integration

Cb Response provides bi-directional integration with an on-premises device by General Dynamics Fidelis Cybersecurity Solutions for correlating Fidelis alerts in the Cb Response server and returning Cb Response metadata to Fidelis. More information about Fidelis can be found at www.fidelissecurity.com. You can also watch a short video explaining the integration at www.youtube.com/watch?v=2iwczFX162U.

To support this integration, Cb Response provides an out-of-band connector that communicates with the Fidelis device and the Cb Response server through a listener that uses port 8000 by default. This connector is installed on the Cb Response server.

Sections

Fidelis Installation

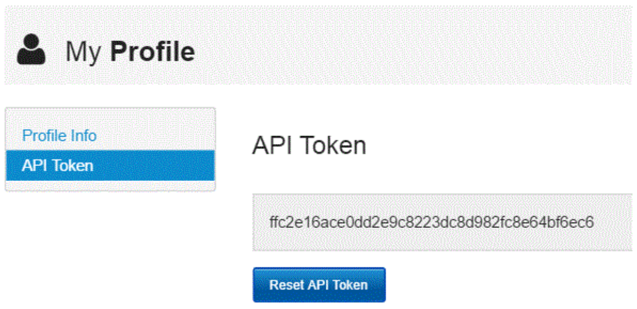
Install the Cb Response out-of-band connector to receive Fidelis alerts on the Cb Response server.

To install the connector:

1. Ensure that the following prerequisites are met:
   - Cb Response server version 5.0 or later is installed with Internet connectivity.
   - You have access to a Fidelis XPS device.
2. Configure the Cb Response open source connectors Yum repository as described in "Setting up the Open Source Connector Repository" on page 19.
3. Verify the Yum configuration and install the Fidelis connector:

```
yum info python-cb-fidelis-bridge
yum install python-cb-fidelis-bridge
```

4. In the Cb Response console interface, retrieve the API key for the user you wills use for this integration:
   a. On the Cb Response console, login with the account you selected. (This user must have administrative rights on the server.)
   b. In the top-right corner of the Cb Response console, select ***Username*** **> My Profile**.
   c. On the **My Profile** page, choose **API Token** in the left menu. Leave this page open so the information is available to you.



ffc2e16ace0dd2e9c8223dc8d982fc8e64bf6ec6

5. Edit the Fidelis connector configuration file in this location:
   `/etc/cb/integrations/carbonblack_fidelis_bridge/carbonblack_fidelis_bridge.conf`
   a. Update the `listener_api_token` option to set the shared secret token to allow Fidelis-to-connector communications. Make a note of this token -- it must also be entered into the Fidelis XPS appliance.
   b. Update the `carbonblack_server_url` option to set the URL of the Cb Response server.
   c. Update the `carbonblack_server_token` option by copying and pasting the API Token string displayed in the **API Token** box in the Cb Response console (see step 4).
   d. The remainder of the options are documented in the configuration file and can be customized as needed to match specific requirements.
   e. Save the configuration file.

6. Edit iptables on the Cb Response server (or the system running the Cb Response connector, if different) to open TCP 8000 to receive XPS alerts, and then restart it.
   a. Enter the following:
      ```
      # vi /etc/sysconfig/iptables
      ```
   b. Add the following line to iptables and then save the file:
      ```
      -A INPUT -p tcp -m conntrack --ctstate NEW -m tcp --dport
      8000 -j ACCEPT
      ```
   c. Restart `iptables` to apply your changes:
      ```
      # service iptables restart
      ```
   d. Confirm that the port is now open:
      ```
      # iptables -nL |grep 8000
      ```
7. Configure the Cb Response Nginx web server to forward requests from the `/fidelis/` context root to the connector that is now installed and listening on port 8000.
   a. Edit the `cb.server.custom` file:
      ```
      # vi /etc/cb/nginx/conf.d/cb.server.custom
      ```
   b. Add the following contents below the comment section:
      ```
      location /fidelis/ {
      expires                0;
      proxy_set_header       Host              $host;
      proxy_set_header       X-Real-IP         $remote_addr;
      proxy_set_header       X-Forwarded-For
      $proxy_add_x_forwarded_for;
      #proxy_set_header       X-Client-Cert-Id $ssl_client_serial;
      #proxy_set_header       X-Ssl-Protocol   $ssl_protocol;
      proxy_pass             http://localhost:8000/fidelis/;
      }
      ```
   c. Save and close the file.
   d. Restart the Cb Response Nginx web server:
      ```
      # service cb-nginx restart
      ```
8. Start the Fidelis connector:
   ```
   /etc/init.d/cb-fidelis-bridge start
   ```
9. Examine the Fidelis connector log to verify that the service is running normally:
   ```
   /var/log/cb/integrations/carbonblack_fidelis_bridge/
   carbonblack_fidelis_bridge.log
   ```

Start and Stop the Fidelis Connector

To start the Fidelis connector:

```
# service /etc/init.d/cb-fidelis-bridge start
```

To stop the Fidelis connector:

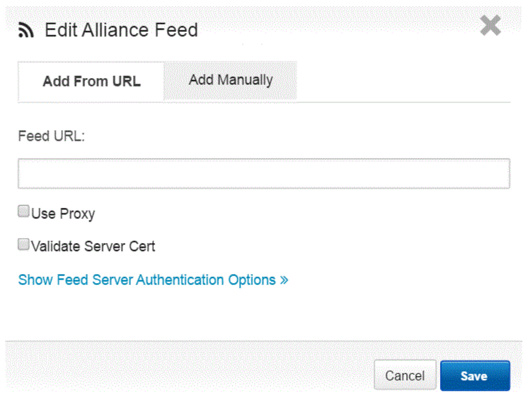
```
# service /etc/init.d/cb-fidelis-bridge stop
```

Fidelis Feed

When the Fidelis connector service is running, you can add the Fidelis feed to the threat intelligence feeds on the Cb Response server. For more information, see the “Threat Intelligence Feeds” chapter in the *Cb Response User Guide*.

To add a new Fidelis feed:

1. Login to the Cb Response console.
2. In the left navigation menu, select **Threat Intelligence**.
3. Click **Add New Feed**.
4. The **Edit Alliance Feed** window opens.



5. In the **Feed URL** field, enter the URL:

```
http://[bridge host]:[listener_port from bridge config]/
fidelis/json
```

For example:

```
http://127.0.0.1:8000/fidelis/json
```

Fidelis XPS Device Configuration

By integrating with the Cb Response server, the Fidelis XPS Host Activity Monitor can detect if malware seen on the network actually reaches the endpoint and if it is written to disk or executed.

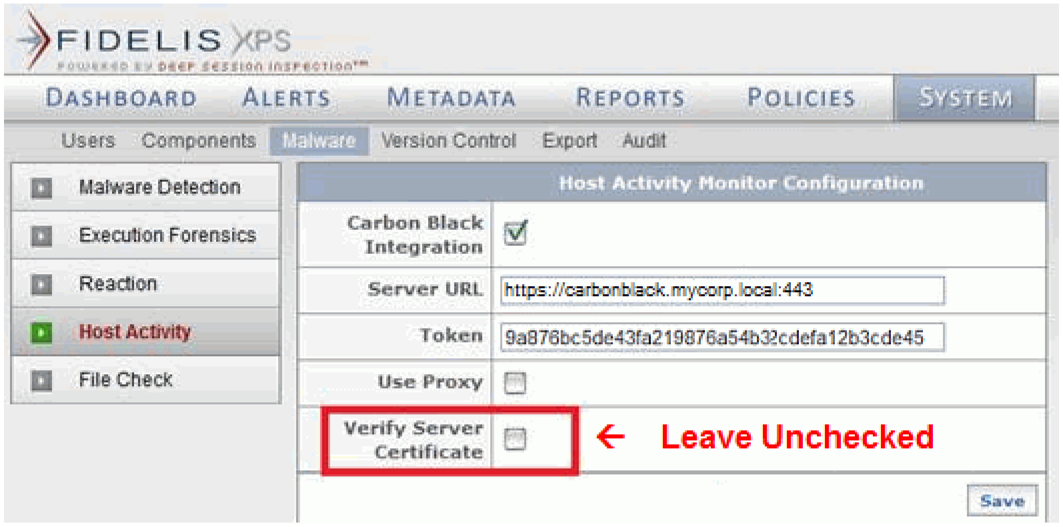
Note *The Host Activity report only indicates when actual malware (whose md5 matches the alert md5) is saved to disk or executed. If the malware is contained in a zip, tar, or other container file, saving the container file will not trigger a Host Activity report.*

Even when a report is triggered, a delay can occur in receiving a Host Activity report depending on the Cb Response client.

Within the XPS application and from the Cb Response view, ensure the following fields and options are accurate:

To configure the Cb Response view in the Fidelis XPS application:

1. In the **Host Activity Monitor Configuration** view on XPS, check the **Carbon Black Integration** checkbox.
2. In the **Server URL** text box, enter the *connector* URL. If you completed the nginx setup earlier, this will be **https://**<*cb server hostname*>**/fidelis**. Alternatively, you can point the Fidelis XPS appliance directly to the connector by entering **http://**<*cb server hostname*>**:8000** into the Server URL text box.
3. Enter the `listener_api_token` from the Fidelis connector confugration file into the **Token** text box (step 5 on page 30).
4. Check **Use Proxy** if the server is outside of your network.
5. Make sure the **Verify Server Certificate** box is *not checked.*
6. Click **Save**.



Carbon Black Integration ✓
Server URL: https://carbonblack.mycorp.local:443
Token: 9a876bc5de43fa219876a54b32cdefa12b3cde45
Use Proxy ☐
Verify Server Certificate ☐

Chapter 4

Lastline Integration

Cb Response provides integration with the Lastline, Inc. service for checking the reputation of binary files. More information about the Lastline service can be found at www.lastline.com/platform/enterprise.

To support this integration, Cb Response provides an out-of-band connector that communicates with the Lastline service and the Cb Response server through a listener on port 4000 (default). This connector is installed on the Cb Response server.

Sections

Lastline Installation

Install the out-of-band connector to enable communication between the Lastline service and the Cb Response server.

To install the connector:

1. Ensure that the following prerequisites are met:

- Cb Response server installation version 5.0 or later is installed with Internet connectivity.
- You have a Lastline service or device API key and token.

2. If you have not already done so, configure the Cb Response open source connectors Yum repository as described in "Setting up the Open Source Connector Repository" on page 19.

3. Verify the Yum configuration and install the Lastline connector:

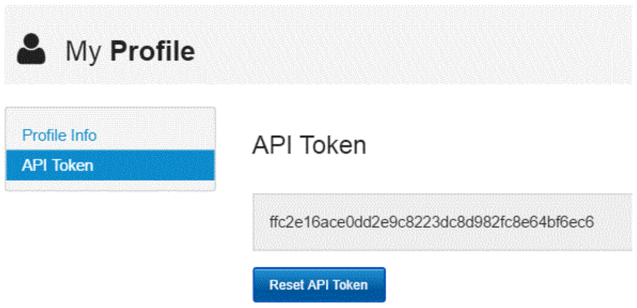
```
yum info python-cb-lastline-connector
yum install python-cb-lastline-connector
```

4. In the Cb Response console interface, retrieve the API key for the Cb Response user account you intend to use for this integration:

a. On the Cb Response console, log in with the account you selected. (This user must have administrative rights on the server.)

b. In the top-right corner of the Cb Response console, select ***Username*** **> My Profile**.

c. On the **My Profile** page, choose **API Token** in the left menu. Leave this page open so the information is available to you.



ffc2e16ace0dd2e9c8223dc8d982fc8e64bf6ec6

5. Installation of the connector includes an example configuration file for Lastline:

`/etc/cb/integrations/lastline/connector.conf.example`

Copy this file to:

`/etc/cb/integrations/lastline/connector.conf`

6. Edit `connector.conf` to provide the necessary configuration parameters:

a. Update the `lastline_api_key=` option to set the Lastline API key. Multiple keys can be separated by semicolons. This key is given to you from Lastline. You can find it by contacting your Lastline representative, the Lastline CLI, or in Lastline, on the **Admin** tab, in the **License** menu under **License Key**.

b. Update the `lastline_api_token=` option to set the Lastline service API token. This key is given to you from Lastline. You can find it by contacting your Lastline representative or the Lastline CLI.

c. Update the `lastline_url=` option to set the URL of the Lastline server. This URL specifies your local or cloud Lastline appliance.

   Cloud example: `https://analysis.lastline.com/`
   Local example: `https://lastline.companyDomain.local`

d. Update the `carbonblack_server_url=` option to set the URL of the Cb Response server. Since this is installed on the Cb Response server, you should be able to keep the default configuration, which is:

   ```
   carbonblack_server_url=https://localhost/
   ```

e. Copy the API Token string displayed in the **API Token** field in the Cb Response console (see step 4), and update the `carbonblack_server_token` value with the token you copied.

f. Edit the `lastline_server_sslverify` option to validate the SSL cert of the last line appliance (a value of 1) or not validate (a value of 0).

   For the cloud service, you should set this to **1**.
   For an on-premises appliance, the appliance most likely does not have a valid SSL certificate, and so **0** is the correct choice.

g. The remainder of the options are documented in the configuration file and can be customized as needed to match specific requirements.

h. Save the configuration file.

7. Start the Lastline connector:

   ```
   /etc/init.d/cb-lastline-connector start
   ```

8. Examine the Lastline connector log to verify that the service is running normally.

   ```
   /var/log/cb/integrations/lastline/lastline.log
   ```

Once the Lastline connector is installed and running, the Lastline feed automatically is added to the Cb Response console.

Start and Stop the Lastline Connector

To start the Lastline connector:

```
# service /etc/init.d/cb-lastline-connector start
```

To stop the Lastline connector:

```
# service /etc/init.d/cb-lastline-connector stop
```

Chapter 5

WildFire Integration

Cb Response provides integration with the Palo Alto Networks WildFire cloud service for checking the reputation of binary files. More information about the Palo Alto Networks WildFire service can be found at www.paloaltonetworks.com/products/technologies/wildfire.html.

To support this integration, Cb Response provides an out-of-band connector that communicates with the Palo Alto Networks WildFire service and the Cb Response server through port 3774 (default). This connector is installed on the Cb Response server.

The current Palo Alto Networks WildFire service can handle querying and uploading binaries that are 32-bit executables and less than 10 MB in size.

Sections

WildFire Installation

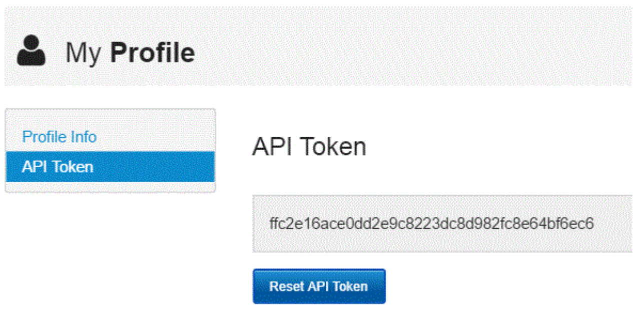
Install the Cb Response out-of-band connector to enable communication between Palo Alto Networks WildFire and the Cb Response server.

To install the connector:

1. Ensure that the following prerequisites are met:
   - Cb Response server installation version 5.0 or later is installed with Internet connectivity.
   - You have the Palo Alto Networks WildFire service API tokens.
2. If you have not already done so, configure the Cb Response open source connectors Yum repository as described in "Setting up the Open Source Connector Repository" on page 19.
3. Verify the Yum configuration and install the Palo Alto Networks WildFire connector:

   ```
   yum info python-cb-wildfire-connector
   ```

   ```
   yum install python-cb-wildfire-connector
   ```
4. In the Cb Response console interface, retrieve the API key for the Cb Response user account you intend to use for this integration:
   a. On the Cb Response console, log in with the account you selected. (This user must have administrative rights on the server.)
   b. In the top-right corner of the Cb Response console, select ***Username* > My Profile**.
   c. On the **My Profile** page, choose **API Token** in the left menu. Leave this page open so the information is available to you.



ffc2e16ace0dd2e9c8223dc8d982fc8e64bf6ec6

5. Installation of the connector includes an example configuration file for Wildfire:

   `/etc/cb/integrations/wildfire/connector.conf.example`

   Copy this file to:

   `/etc/cb/integrations/wildfire/connector.conf`
6. Edit `connector.conf` to provide the necessary configuration parameters:
   a. Update the `wildfire_api_keys` to proivde your API key(s) for the Wildfire appliance or cloud service. If you have one API key, place it here unchanged. If you have more than one API key, separate them by semicolons.

   For example:

```
apikey1;apikey2;apikey3
```

   In this example, the connector will use `apikey1` until it receives an "API retries exhausted" error for that key. It will then cycle through to `apikey2` and so on.

b. Update the `carbonblack_server_url` option to set the URL of the Cb Response server.

c. Copy the API Token string displayed in the **API Token** field in the Cb Response console (see step 4), and update the `carbonblack_server_token` value with the token you copied.

d. Edit the `wildfire_server_sslverify` option to validate the SSL cert of the last line appliance (a value of 1) or not validate (a value of 0).

For the cloud service, you should set this to **1**.

For an on-premises appliance, the appliance most likely does not have a valid SSL certificate, and so **0** is the correct choice.

e. Examine and edit the `binary_filter_query` setting if needed. This setting defines the binary types that should be sent to Wildfire to analysis. By default, only 32-bit executables (not DLLs) that are *not signed* by Microsoft are sent for analysis.

f. The remainder of the options are documented in the configuration file and can be customized as needed to match specific requirements.

g. Save the configuration file.

7. Start the Palo Alto Networks WildFire connector:

```
/etc/init.d/cb-wildfire-connector start
```

8. Examine the Palo Alto Networks WildFire connector log to verify the service is running normally:

```
/var/log/cb/integrations/wildfire/wildfire.log
```

When the WildFire connector is installed and running, the WildFire feed is automatically added to the Cb Response console.

Note *For the benefit of users upgrading from a previous version of this connector, the* `connector.conf` *file includes the* `legacy_feed_directory` *setting. This setting automatically imports feed reports from the older version of the Wildfire connector.*

Starting and Stopping the WildFire Connector

To start the WildFire connector:

```
# service /etc/init.d/cb-wildfire-connector start
```

To stop the WildFire connector:

```
# service /etc/init.d/cb-wildfire-connector stop
```

Chapter 6

ThreatConnect Integration

Cb Response provides integration with ThreatConnect by retrieving IOCs from specified communities. To support this integration, Cb Response provides an out-of-band connector that communicates with the ThreatConnect API via port 6100 (default).

The Cb Threat Intel connectivity is not required for the ThreatConnect connector.

Sections

ThreatConnect Installation

Install the out-of-band connector to receive ThreatConnect alerts on the Cb Response server.

To install the connector:

1. Ensure that the following prerequisites are met:
   - Cb Response server version 5.0 or later is installed with Internet connectivity.
   - Your Cb Response Alliance SSL certificate and key. These are usually located in:

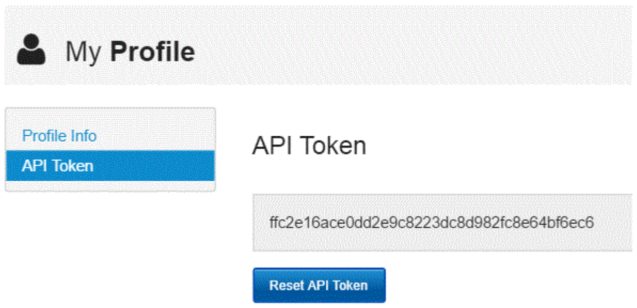
   ```
   /etc/cb/certs/carbonblack-alliance-client.crt
   /etc/cb/certs/carbonblack-alliance-client.key
   ```
   - You have a ThreatConnect API key and secret key.
2. Configure the Cb Response open source connectors Yum repository as described in "Setting up the Open Source Connector Repository" on page 19.
3. Verify the Yum configuration and install the ThreatConnect connector:

   ```
   yum info python-cb-threatconnect-connector
   yum install python-cb-threatconnect-connector
   ```
4. In the Cb Response console interface, retrieve the API key for the Cb Response user account you intend to use for this integration:

   a. On the Cb Response console, log in with the account you selected. (This user must have administrative rights on the server.)

   b. In the top-right corner of the Cb Response console, select ***Username* > My Profile**.

   c. On the **My Profile** page, choose **API Token** in the left menu. Leave this page open so the information is available to you.



ffc2e16ace0dd2e9c8223dc8d982fc8e64bf6ec6

5. Edit the ThreatConnect connector configuration file. You can find the configuration file on the Cb Response server at this location:

   ```
   /etc/cb/integrations/threatconnect/connector.conf
   ```

   a. `api_key` – The secret, numeric identifier issued by ThreatConnect for your API account.

   b. `secret_key` – The secret key for your ThreatConnect API account. Set the secret key in plaintext under `secret_key`.

   c. `feed_retrieval_minutes` – Defines how often the connector retrieves the full list of IOCs (in minutes).

   d. `listener_address` – The address that the connector uses to listen for connections. The default is `127.0.0.1`.

   e. `listener_port` – The port number that the connector uses to listen for connections. The default is `6100`.
   f. Add any additional communities to which you are subscribed under the `[sources]` section with a unique tag and a corresponding API endpoint.
6. Start the ThreatConnect connector:

   ```
   /etc/init.d/cb-threatconnect-connector start
   ```
7. Examine the ThreatConnect connector logs to verify that the service is running normally:

   ```
   /var/log/cb/integrations/cb-threatconnect-connector/cb-
   threatconnect-connector.log
   ```

When the ThreatConnect connector is installed and running, the ThreatConnect feed is automatically added to the Cb Response console.

Start and Stop the ThreatConnect Connector

To start the ThreatConnect connector:

```
# service /etc/init.d/cb-threatconnect-connector start
```

To stop the ThreatConnect connector:

```
# service /etc/init.d/cb-threatconnect-connector stop
```

View Diagnostic Page

With the ThreatConnect connector service running, you can connect to the following location on the server running the ThreatConnect connector to view diagnostic information:

http://127.0.0.1:6100/

Note *Change the IP address or port if you modified the default settings.*

View the Feed

With the ThreatConnect connector service running, you can connect to the following location on the server running the ThreatConnect connector (if you used the default settings) to view the feed:

http://127.0.0.1:6100/threatconnect/json

Change the IP address or port if you modified the default settings.

Troubleshooting

- You might have to modify iptables and your firewall rules if you want to connect from a remote system.
- The connector's log file is located in the following location:

  ```
  /var/log/cb/integrations/cb-threatconnect-connector/cb-
  threatconnect-connector.log
  ```

- To see any exceptions that get thrown, run the program directly with write mode:

  ```
  /usr/share/cb/integrations/cb-threatconnect-connector/bin/cb-
  threatconnect-connector write <filename_to_write>
  ```

- Connection issues arising from failed logins are written to the log file and appended to the `last_sync column` on the connector diagnostic page, located on the same machine at:

  ```
  http://localhost:6100/
  ```

  Sample error:

  ```
  No sync performed ({u’status’: u’Failure’, u’errorMessage’:
  u’Timestamp out of acceptable time range’})
  ```

Common Issues

- Symptom: No synchronization occurs, `“Timestamp out of acceptable time range”` reported in error logs or on the connector diagnostic page.
  - Diagnosis: The system clock is out of sync. Successful authentication with the ThreatConnect API requires that the system clock is correct.
  - Fix: Set the system clock. For example:

  ```
  date --set="2 MAR 2014 18:00:00"
  ```

- Symptom: No synchronization occurs, `“Signature data did not match expected result”` reported in error logs or on the connector diagnostic page.
  - Diagnosis: The text of the community URL is incorrect. This text is used as part of the ThreatConnect API. Below is one example of a typo that can lead to this error.

* Incorrect (note the extra percent sign):

```
CommonCommunity = /v1/indicators/?owner=Common%%20Community
```

* Correct:

```
CommonCommunity = /v1/indicators/?owner=Common%20Community
```

  - Fix: Carefully examine the URL for any typos.

- Symptom: No synchronization occurs, `{u’status’: u’Failure’, u’errorMessage’: u”}` reported in error logs or on the connector diagnostic page.
  - Diagnosis: Your API key or secret key is incorrect.
  - Fix: Ensure that your API key and secret key were entered correctly.

Chapter 7

Cyphort Integration

Cb Response integrates with Cyphort for inspection, analysis, and correlation of suspicious binaries discovered at the endpoint. Cyphort Core is a secure threat analysis engine that leverages Cyphort's multi-method behavioral detection technology and threat intelligence.

Cb Response can submit unknown or suspicious binaries to Cyphort Core to deliver threat scores used in Cb Response to enhance detection, response, and remediation efforts. More information about Cyphort Core can be found at www.cyphort.com.

To support this integration, Cb Response provides an out-of-band connector that communicates with the Cyphort Core and the Cb Response server via port 7000.

Sections

Cyphort Installation

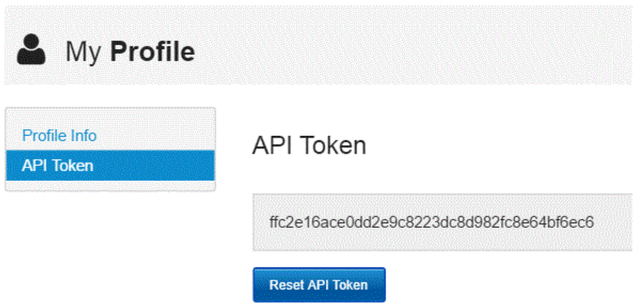
Install the out-of-band connector to enable communication between the Cyphort Core and the Cb Response server.

To install the connector:

1. Ensure that the following prerequisites are met:
   - Cb Response server version 5.1 or later is installed with Internet connectivity.
   - You have Cyphort service or device API key.
2. Configure the Cb Response open source connectors YUM repository as described in "Setting up the Open Source Connector Repository" on page 19.
3. Verify the Yum configuration and install the Cyphort connector:

   ```
   yum info python-cb-cyphort-connector
   yum install python-cb-cyphort-connector
   ```

4. In the Cb Response console interface, retrieve the API key for the Cb Response user account you intend to use for this integration:
   a. On the Cb Response console, log in with the account you selected. (This user must have administrative rights on the server.)
   b. In the top-right corner of the Cb Response console, select ***Username* > My Profile**.
   c. On the **My Profile** page, choose **API Token** in the left menu. Leave this page open so the information is available to you.



ffc2e16ace0dd2e9c8223dc8d982fc8e64bf6ec6

5. Installation of the connector included an example configuration file for Cyphort:

   `/etc/cb/integrations/cyphort/connector.conf.example`

   Copy this file to:

   `/etc/cb/integrations/cyphort/connector.conf`
6. Edit `connector.conf` to provide the necessary configuration parameters:
   a. Update the `cyphort_api_key=` option to set the Cyphort API key. This key is given to you from Cyphort. You can find it by contacting your Cyphort representative, the Cyphort CLI, or in Cyphort on the **Admin** tab, in the **License** menu under **License Key**.
   b. Update the `cyphort_url=` option to set the URL of the Cyphort server. This URL specifies the URL for the Cyphort API.
   Cloud example: `https://demo.cyphort.com/`
   Local example: `https://cyphort-appliance`

   c. Update the `carbonblack_server_url=` option to set the URL of the Cb Response server. Since this is installed on the Cb Response server, you should be able to keep the default configuration, which is:

      ```
      carbonblack_server_url=https://localhost/
      ```

   d. Copy the API Token string displayed in the **API Token** box in the Cb Response console (see step 4), and update the `carbonblack_server_token` value with the token you copied.

   e. Edit the `cyphort_server_sslverify` option to validate the SSL cert of the last line appliance (a value of 1) or not validate (a value of 0).

      If you are using the cloud service, you should set this to **1**.
      If you are using an on-premises appliance, the appliance most likely does not have a valid SSL certificate, and so **0** is the correct choice.

   f. The remainder of the options are documented in the configuration file and can be customized as needed to match specific requirements.

   g. Save the configuration file.

7. Start the Cyphort connector:

   ```
   /etc/init.d/cb-cyphort-connector start
   ```

8. Examine the Cyphort connector log to verify that the service is running normally.

   ```
   /var/log/cb/integrations/cyphort/cyphort.log
   ```

When the Cyphort connector is installed and running, the Cyphort feed is automatically added to the Cb Response interface.

Start and Stop the Cyphort Connector

To start the Cyphort connector:

```
# service /etc/init.d/cb-cyphort-connector start
```

To stop the Cyphort connector:

```
# service /etc/init.d/cb-cyphort-connector stop
```

Chapter 8

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 52.
- **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 85.
  - For information on configuring Cb Event Forwarder for the QRadar connector, see "Configure Cb Event Forwarder for QRadar Integration" on page 52.
- **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 54.

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 85.

The Cb Event Forwarder can be configured to forward Cb Response events in LEEF format to a QRadar log collector appliance.

To forward Cb Response events to a QRadar server:

1. Create a log source for the Carbon Black server. See the following IBM QRadar documentation for information on how to create a log source:

   ftp://ftp.software.ibm.com/software/security/products/qradar/documents/iTeam_addendum/b_logsource.pdf

   You can configure the log source to use the "syslog" or "TLS syslog" protocol to receive events from the Cb Event Forwarder.

2. 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 in:

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

   - `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>`.

3. 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.

4. Modify `/etc/cb/integrations/event-forwarder/cb-event-forwarder.conf` to specify the protocol and network address of the QRadar server:

   For the "syslog" protocol configuration:
   ```
   syslogout=tcp:<qradaripaddress>:514
   ```
   For the "TLS syslog" protocol configuration:
   ```
   syslogout=tcp+tls:<qradaripaddress>:6514
   ```

Note The default port when using "syslog" is 514, and the default port when using "TLS syslog" is 6514.

If using the "TLS syslog" protocol configuration, ensure that the server certificate installed on the QRadar server is signed by a legitimate Certification Authority, or turn off CA validation in the cb-event-forwarder configuration file:

```
tls_verify=false
```

5. Change the output format to LEEF in the configuration file:

   ```
   output_format=leef
   ```

6. Change the output type to syslog in the configuration file:

```
output_type=syslog
```

Note For more information on LEEF format, see https://github.com/carbonblack/cb-event-forwarder/blob/master/EVENTS.md.

7. 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.

8. Start the Cb Event Forwarder:

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

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 7.2.7 or later

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

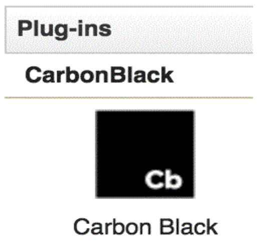
Configure the Cb Response App for IBM QRadar

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:

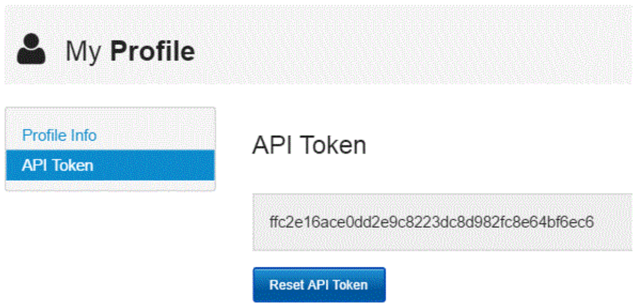
1. Navigate to the **Admin** tab of your QRadar server.
2. Scroll to the **Plug-ins** section at the bottom of the page.
3. Click the **Carbon Black** button.



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. In the top-right corner of the Cb Response console, select ***Username*** **> My Profile**.
   c. On the **My Profile** page, click **API Token** in the left menu.
   d. Copy the displayed API token.



ffc2e16ace0dd2e9c8223dc8d982fc8e64bf6ec6

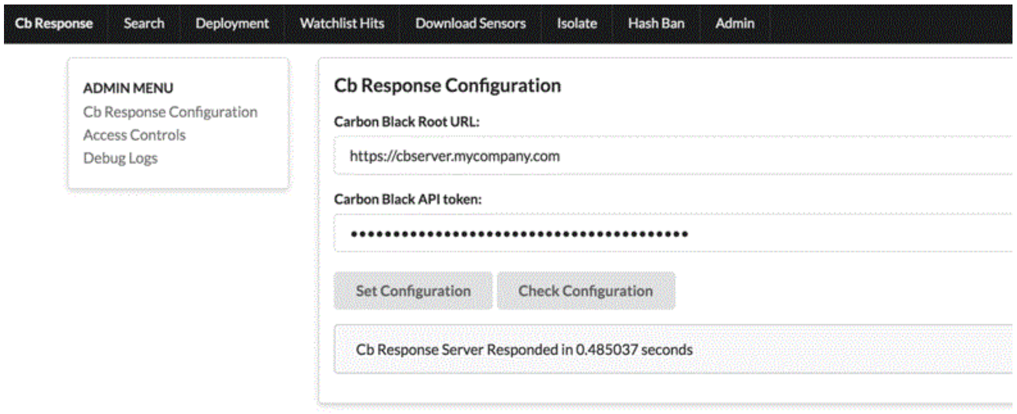
5. Return to the next Cb Response app for IBM QRadar page and do the following:
   a. Paste the API token into the **Carbon Black API Token** field.
   b. Enter the URL for your Cb Response server instance in the **Carbon Black Root URL** field. For example, enter:

   `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”.



https://cbserver.mycompany.com

••

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 57**.**
- **Cb Response** tab. For more information, see “Dashboard Widget” on page 56.
- **Offenses** and **Log Activity** tab context menus. For more information, see “QRadar Context Menus” on page 60.

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 59.

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.

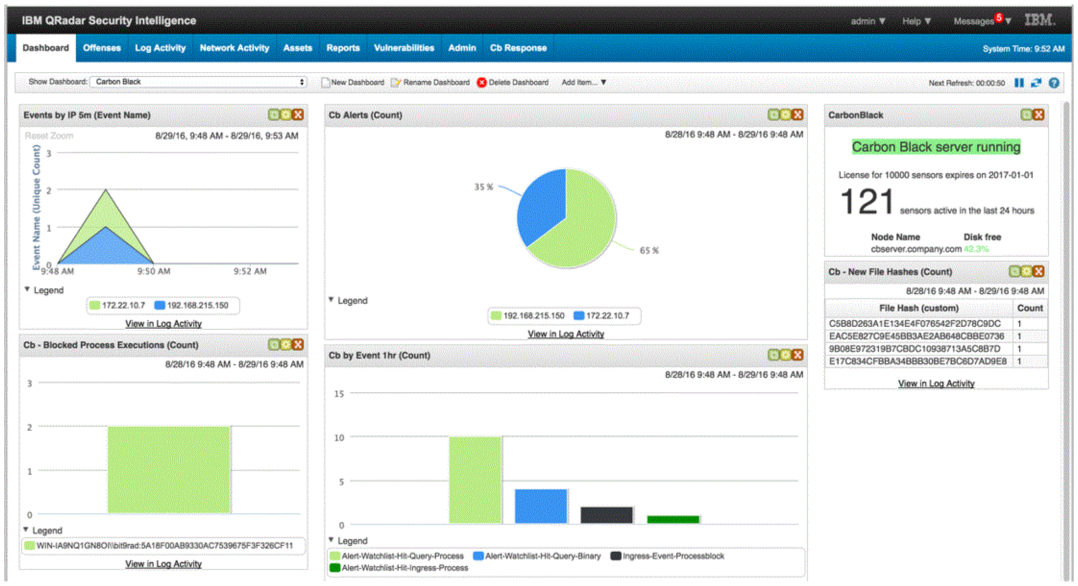
All users that are authorized for the Carbon Black capability have access to these components. QRadar users with “Admin” or “Forensics” privileges (which are configurable) can also isolate endpoints from the network, force sensors to send all queued data to the Cb Response server, and add binaries to 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
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)
8/29/16, 9:48 AM - 8/29/16, 9:53 AM
Event Name (Unique Count)
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
Cb - Blocked Process Executions (Count)
WIN-IA9NQ1GN8OI\bit9rad:5A18F00AB9330AC7539675F3F326CF11
Cb by Event 1hr (Count)
Alert-Watchlist-Hit-Query-Process
Alert-Watchlist-Hit-Query-Binary
Ingress-Event-Processblock
Alert-Watchlist-Hit-Ingress-Process
CarbonBlack
Cb - New File Hashes (Count)

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

View in Log Activity

Carbon Black Tab

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

- **Search**
- **Deployment**
- **Watchlist Hits**
- **Feed 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.

Feed Hits

The Feed Hits sub-tab contains the timestamp of the most recent hit per feed. This allows for quick viewing the most recently seen feeds. The tab also shows the total number of feed hits for both process and binary hits.

Each feed is hyperlinked to the associated Cb Response Feed query. Clicking on the feed in the QRadar interface forwards you to the Cb Response Web UI and displays the latest 10 process feed hits.

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 disabl 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 60.

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 58.

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 9

Splunk Integration

The Cb Response app for Splunk 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 Splunk console. Once installed, administrators can use Splunk to:

- Access many powerful Cb Response features, such as process searches, endpoint isolation, and system status.
- Monitor the health of the Cb Response server, deployment, and threat data using the Dashboards built into the Cb Response app.
- Use pre-build saved searches as starting points for threat hunting using Splunk's advanced analytics and search capability on visibility data collected via Cb Response.
- Respond to issues by quarantining endpoints using the new Splunk **Adaptive Response** capability.
- Use Cb Event Forwarder, which exports Cb Response events to the Splunk server for correlation, analysis, and alerting.

Sections

Overview

Three main components are involved in the Splunk integration:

- **Cb Response Splunk Technology Add-on (TA)** – Download and install the Carbon Black Technology Add-on for Splunk, which defines the Splunk sourcetype associated with Cb Response data and normalizes the Carbon Black data into the Splunk Common Information Model (CIM). The Carbon Black TA must be installed before events forwarded via the Cb Event Forwarder can be interpreted by the Splunk console.
- **Cb Event Forwarder** – Download, install and configure the Cb Event Forwarder. The Cb Event Forwarder allows data to be ingested into Splunk from Carbon Black. Installing the Cb Event Forwarder is optional but recommended.
  - For information on installing and using Cb Event Forwarder, see "Cb Event Forwarder" on page 85.
  - For information on configuring Cb Event Forwarder for the Splunk connector, see "Configure Cb Event Forwarder for Splunk Integration" on page 62.
- **Carbon Black Response App for Splunk** – Download, install, and configure the Cb Response app for Splunk from Splunkbase. This app contains dashboards, saved searches, custom commands, and Adaptive Response actions that you can use to visualize, analyze, and respond to Cb Response events from the Splunk console. For more information, see "Install Cb Response App for Splunk" on page 65 on page 42.

Install Carbon Black TA for Splunk

The Carbon Black Technology Add-on (TA) for Splunk normalizes the Carbon Black data into a format that Splunk can index. The Carbon Black TA must be installed before events forwarded via the Cb Event Forwarder can be interpreted by the Splunk console.

You can download the Carbon Black TA for Splunk from Splunkbase at https://splunkbase.splunk.com/app/2790/. More information, including detailed installation directions for both standalone and clustered Splunk installations, can be found on Splunk's website at http://docs.splunk.com/Documentation/AddOns/latest/Bit9CarbonBlack/About.

Configure Cb Event Forwarder for Splunk Integration

This section explains how to configure Cb Event Forwarder for the Splunk integration. For more information on installing and using Cb Event Forwarder, see "Cb Event Forwarder" on page 85.

You can install the Cb Event Forwarder on the same machine as your Cb Response server or on a different server. For more information on the recommended deployment scenarios for the Cb Event Forwarder, see "Cb Event Forwarder" on page 85.

To configure Cb Event Forwarder for the Splunk integration:

1. If you are installing the Cb Event Forwarder on the same machine as your Cb Response server, you can skip this step. 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. Ensure that the `output_type` is set to "`file`" (the default) and the `output_format` is set to "`json`" (also default)
4. 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 and Configure Splunk Universal Forwarder

Install and configure the Splunk Universal Forwarder to forward the Cb Event Forwarder output to the Splunk indexer. The Universal Forwarder must be installed on the same host running the Cb Event Forwarder. Detailed instructions on installing the Splunk Universal Forwarder on Linux can be found on Splunk's website:

http://docs.splunk.com/Documentation/Forwarder/6.4.3/Forwarder/Installanixuniversalforwarder

Configure a Listener on the Splunk Console

To configure a listener on the Splunk console:

1. Log into the Splunk console as an administrator.
2. Select on **Settings > Data > Forwarding** and receiving.
3. Click **Configure Receiving**.
4. Click the green **New** button and configure a listener port. The default port number is "9997".
5. Click Save.

Configure the Splunk Universal Forwarder

Next, configure the Splunk Universal Forwarder to send Cb Response event data to the Splunk listener configured in "Configure a Listener on the Splunk Console" on page 64.

To configure the Splunk Universal Forwarder:

1. SSH into the host running the Splunk Universal Forwarder and Cb Event Forwarder.
2. Ensure that no active forwards are configured by executing this command:

   ```
   /opt/splunkforwarder/bin/splunk list forward-server
   ```
3. Add an active forwarder by executing this command:

   ```
   /opt/splunkforwarder/bin/splunk add forward-server <ip address
   of splunk server> <port number for receiving port>
   ```
4. Create a new `inputs.conf` file containing the following information:

> **Note** The `outputs.conf` file is located in `/opt/splunkforwarder/etc/system/local`. See Splunkbase for more information on configuring the outputs file.

- The "host" directive should be changed to an identifier that represents this Cb Response server. It does not have to match the DNS name of the server; however Carbon Black recommends setting this to a unique identifier, since multiple Cb Response servers can forward into the same Splunk server.
- The file path in the "monitor" directive should be adjusted to match the directory containing the Cb Event Forwarder output file. The default filename in the Cb Event Forwarder configuration file is `/var/cb/data/event-forwarder/output.json`.

> **Note** Any files located within the /var/cb/data/event-forwarder directory will be sent to Splunk and interpreted as Cb Event Forwarder source files, so no other log files should be located in this directory.

```
[default]
host=cbtest
[monitor:///var/cb/data/event-forwarder]
sourcetype = bit9:carbonblack:json
```

```
initCrcLength = 2048
recursive = false
```

5. Start the Cb Event Forwarder and Splunk Universal Forwarder:

```
initctl start cb-event-forwarder
service splunk start
```

Install Cb Response App for Splunk

Install the Cb Response App for Splunk via Splunkbase. This app contains dashboards, saved searches, custom commands, and Adaptive Response actions that you can use to visualize, analyze, and respond to Cb Response events from the Splunk console. The bundled Adaptive Response actions allow administrators to take actions on endpoints, including banning hashes, isolating endpoints, and killing individual processes, based on Splunk console searches.

Requirements

The Cb Response app for Splunk requires the following:

- A functional Cb Response server (version 5.1 or later)
- Splunk version 6.3 or later

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

Configure the Cb Response App for IBM QRadar

After the Cb Response app for Splunk is installed, you must then configure it to connect to your Cb Response server.

Note The Cb Response app for Splunk can only connect to single-server or single-cluster deployments. (It does not support multiple-cluster deployments.)

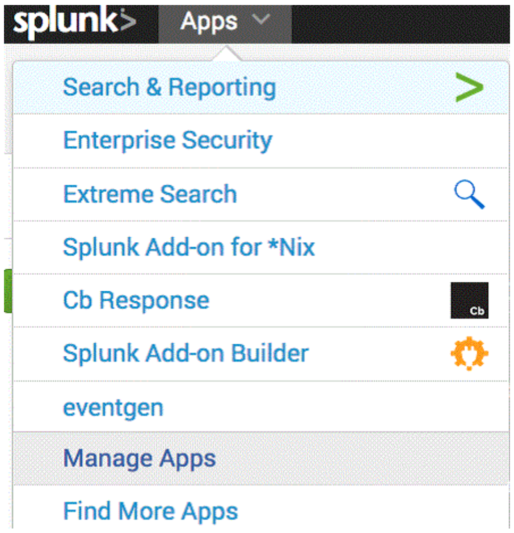
The Cb Response app for Splunk uses a Cb Response API key to:

- Power the `sensorsearch`, `processsearch`, and `binarysearch` custom commands by performing searches via the Cb Response API.
- Enable the “Endpoint Isolation” Adaptive Response Action by requesting endpoint isolation through the Cb Response API
- Enable the “Ban Hash” Adaptive Response Action by using the Cb Response API to add an MD5 hash to the list of banned hashes.
- Enable the “Kill Process” Adaptive Response Action by using the Cb Response Live Response API to kill a process on a remote endpoint.

Note Note that Live Response must be enabled on the Cb Response server for this to function. See the Cb Response User Guide for more information on Live Response.

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

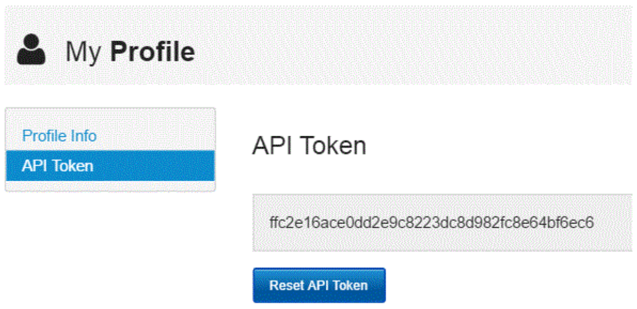
1. From the **Apps** drop-down menu (next to the Splunk icon on the top of the Splunk dashboard), select **Manage Apps**:



2. Click the **Set Up** action to the right of the Cb Response app.
3. Acquire an API key for a Global Administrator user on the Cb Response server to perform these steps:

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. In the top-right corner of the Cb Response console, select ***Username* > My Profile**.
   c. On the **My Profile** page, click **API Token** in the left menu.
   d. Copy the displayed API token.



4. Return to the Splunk Configuration page and perform these steps:

   a. Paste the API token into the **apikey** field.

   b. Enter the URL for your Cb Response server instance in the **cburl** field. For example, enter:

   https://cbserver.mycompany.com

Note Do not place a trailing slash (/) at the end of this URL.

5. Click **Save** to save the new configuration.

The Cb Response app for Splunk uses Splunk's encrypted credential storage facility to store the API token for your Cb Response server, so the API key is stored securely on the Splunk server.

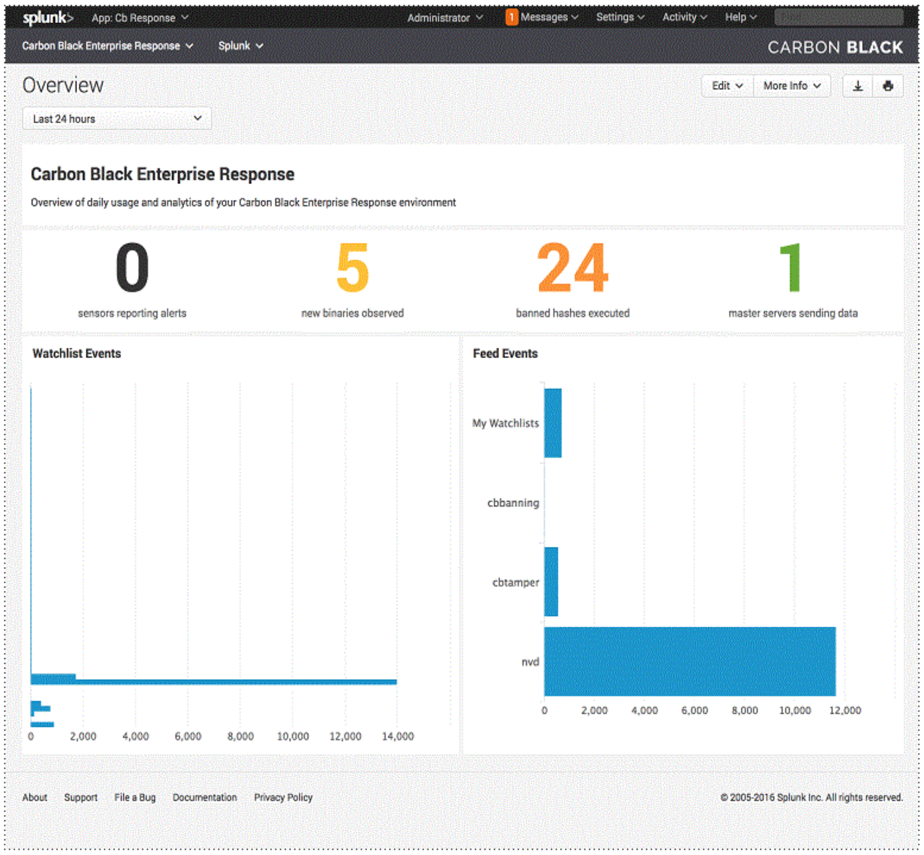
Functionality

The Cb Response app for Splunk provides five major functional categories:

- Custom Commands – These commands can be used in a Splunk search pipeline to use the power of Splunk's visualization and searching capability against Cb Response data without ingesting all of the raw endpoint data into Splunk itself.
- Dashboards – Pre-built dashboards provide you a quick check on the health of your Cb server, status of your Cb Response deployment, and an overview of the detected threats on your network. Several example dashboards are distributed with this app–not all of these can be populated with data, depending on what events are being forwarded to Splunk via the Cb Event Forwarder.
- Adaptive Response Alert Actions – Splunk's new Adaptive Response capability now allows you to take action straight from the Splunk console. The Cb Response Splunk app currently includes three Adaptive Response Alert Actions that allow you to take action either as a result of automated Correlation Searches or on an ad-hoc basis through the Splunk Enterprise Security Incident Review page.
- Saved Searches – Included in this release are 58 saved searches to jump-start Threat Hunting from within the Splunk environment, thanks to community contributions from Mike Haag and others.
- Workflow Actions – This app includes workflow actions to provide additional context from Cb Response on events originating from any product that pushes data into your Splunk server.

Dashboards

Once the app is installed, a new icon appears on the left hand side of the Splunk front page with the Cb Response logo. Clicking the logo brings you to the default dashboard of the Cb Repsonse for Splunk app, shown below. Additional dashboards include an overview of endpoint status, including a breakdown of OS and sensor versions, as well as data on the latest new binaries seen in the environment.



Overview of daily usage and analytics of your Carbon Black Enterprise Response environment

Custom Commands

The Splunk app includes three custom commands to perform searches on the Cb Response datastore from Splunk:

- `binarysearch`
- `processsearch`
- `sensorsearch`

These three commands also have corresponding views in the Cb Response app:

- **Binary Search**
- **Process Search**
- **Sensor Search**

To use the custom commands in your Splunk searches, first ensure that you're using the Cb Response context by invoking the search through the **Splunk > Search** menu inside the Cb Response app. Then, you can use any of the search commands by appending the Cb Response query as a "query" parameter.

For example:

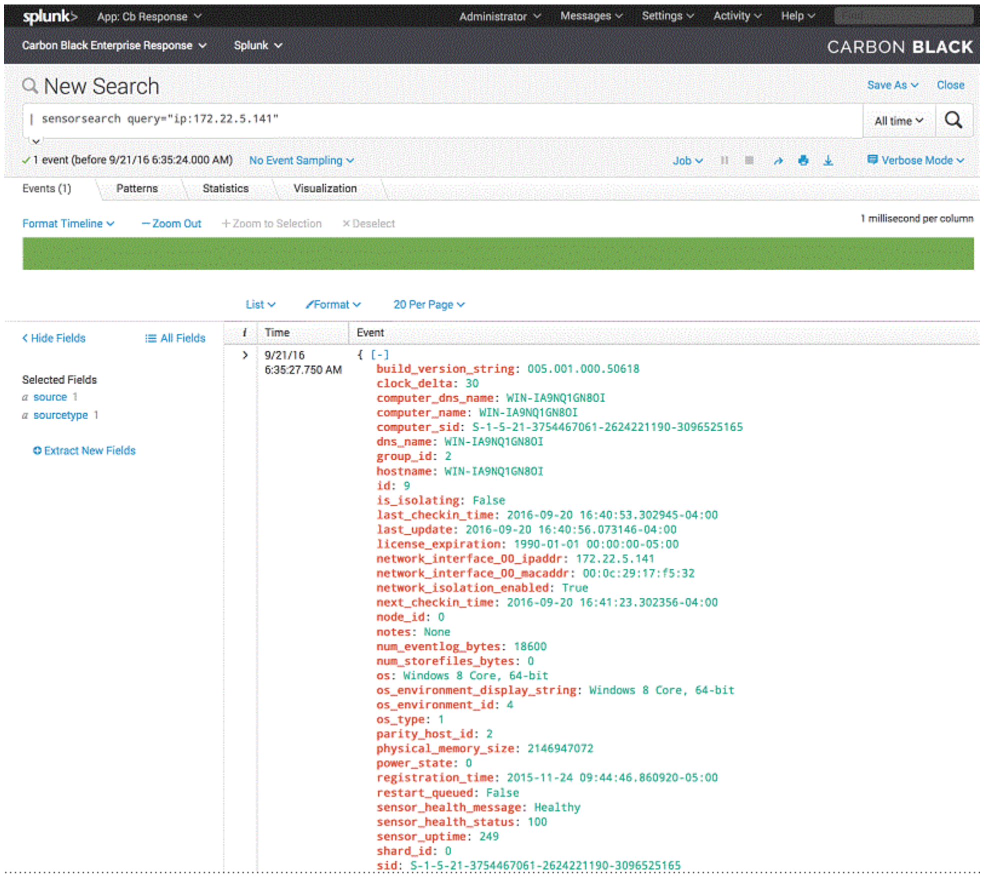
```
| sensorsearch query=”ip:172.22.5.141”
```

will send an API request to Cb Response to query for all sensors that have reported an IP address of 172.22.5.141. The result of this query can be piped through to other Splunk commands for aggregation, visualization, and correlation.

Saved Searches

Several example reports and saved searches are included in this app release. A full list of these searches can be found by the **Settings > Searches**, **Reports**, and **Alerts** menu items from the Cb Response app.

Note	None of these are run or scheduled by default, and some will not return any data unless certain data types (netconns, procstarts, and so on) are forwarded via the Cb Event Forwarder into Splunk.



```
| sensorsearch query="ip:172.22.5.141"
```

✓ 1 event (before 9/21/16 6:35:24.000 AM) No Event Sampling

```
9/21/16            { [-]
6:35:27.750 AM       build_version_string: 005.001.000.50618
                     clock_delta: 30
                     computer_dns_name: WIN-IA9NQ1GN8OI
                     computer_name: WIN-IA9NQ1GN8OI
                     computer_sid: S-1-5-21-3754467061-2624221190-3096525165
                     dns_name: WIN-IA9NQ1GN8OI
                     group_id: 2
                     hostname: WIN-IA9NQ1GN8OI
                     id: 9
                     is_isolating: False
                     last_checkin_time: 2016-09-20 16:40:53.302945-04:00
                     last_update: 2016-09-20 16:40:56.073146-04:00
                     license_expiration: 1990-01-01 00:00:00-05:00
                     network_interface_00_ipaddr: 172.22.5.141
                     network_interface_00_macaddr: 00:0c:29:17:f5:32
                     network_isolation_enabled: True
                     next_checkin_time: 2016-09-20 16:41:23.302356-04:00
                     node_id: 0
                     notes: None
                     num_eventlog_bytes: 18600
                     num_storefiles_bytes: 0
                     os: Windows 8 Core, 64-bit
                     os_environment_display_string: Windows 8 Core, 64-bit
                     os_environment_id: 4
                     os_type: 1
                     parity_host_id: 2
                     physical_memory_size: 2146947072
                     power_state: 0
                     registration_time: 2015-11-24 09:44:46.860920-05:00
                     restart_queued: False
                     sensor_health_message: Healthy
                     sensor_health_status: 100
                     sensor_uptime: 249
                     shard_id: 0
                     sid: S-1-5-21-3754467061-2624221190-3096525165
```

Splunk Workflow Actions

The Cb Response app for Splunk includes Workflow Actions that provide you with context about events in any Splunk view, including Enterprise Security's Notable Event table. The following Workflow Actions are included:

- Sensor Information by IP – Find detailed information about a Cb Response sensor given an IP address field.
- Binary Search by MD5 hash – Retrieve context around a binary given an MD5 file hash.
- Search for Processes contacting IP – Retrieve a list of processes from Cb Response that have made a connection to or received a connection from the given IP address.
- Search for Processes related to MD5 hash – Retrieve a list of processes from Cb Response that have links to the given MD5 hash (for example, a loaded module/DLL, the executable itself, or a file write to an executable with the given MD5 hash)
- Search for Processes contacting Domain – Retrieve a list of processes from Cb Response that have made a connection to or received a connection from the given domain name.
- Search for Processes related to filename – Retrieve a list of processes from Cb Response which have referred to the given filename (written/modified the file, etc.)

In addition, for events that were generated by Cb Response (forwarded into Splunk via the Cb Event Forwarder), additional Workflow Actions are enabled to provide deep links into the Cb Response console directly from the event in Splunk, where applicable. These deep links require the Cb Event Forwarder to be configured properly to generate these links at event generation time (see the Cb Event Forwarder configuration file for more details).

- Deep Link to target process's **Process Analysis** page
- Deep Link to parent process's **Process Analysis** page
- Deep Link to child process's **Process Analysis** page
- Deep Link to **Binary Analysis** page
- Deep Link to **Sensor** page

Diagnostics

The Cb Response App for Splunk writes its log files into the standard Splunk log directory. The following log files (all located under `$SPLUNK_HOME/var/log/splunk`) are used by the App:

- `da-ess-cbresponse.log` – The main log file for common Cb Response helper functions, including the Search Custom Commands.
- `isolate_modalert.log` – The log file for the Isolate Endpoint Adaptive Response Action.
- `banhash_modalert.log` – The log file for the Ban Hash Adaptive Response Action.
- `killprocess_modalert.log` – The log file for the Kill Process Adaptive Response Action.

Chapter 10

BigFix Integration

The IBM BigFix and Carbon Black integration allows administrators to deploy a full endpoint security solution to detect, contain, investigate, and remediate security threats and attacks on endpoints across the enterprise. You can leverage BigFix Fixlets to more easily deploy, monitor, manage, and troubleshoot Carbon Black sensors through BigFix. More importantly, you can now use BigFix to remediate the vulnerabilities or malware identified by Carbon Black so the security threats/attacks can be eliminated or mitigated quickly and effectively.

Sections

Getting Started

The following integration functions are available in the BigFix and Carbon Black integration. Each integration function delivers a unique value and is independent of the others, although all functions depend on the same solution infrastructure provided by BigFix and Carbon Black. You can decide which integration functions you need based on the following descriptions:

- **Cb Event Forwarder** – The Cb Response BigFix connector requires the Cb Event Forwarder standalone service, which pushes data from the Cb Response server to the Cb Response BigFix connector. For information on installing and using Cb Event Forwarder, see "Cb Event Forwarder" on page 85. For information on configuring Cb Event Forwarder for the BigFix integration, see "Installation Materials" on page 73.
- **Cb Sensor Deployment and Health Monitoring** – A number of BigFix Fixlets are provided to deploy, monitor, manage, and troubleshoot Cb Response sensors. Specially, the Fixlets can be used to perform the following tasks:
  - Deploy the Cb Response sensors.
  - Identify machines lacking Cb Response sensors.
  - Check current sensor status and configuration.
  - Create deployment reports and dashboards in IBM BigFix.
  - Use sensor debug & diagnostic health data for rapid troubleshooting.
  - Uninstall Cb Response sensors.

  You can download and install the Fixlets from the BigFix site https://bigfix.me/. For more information, see https://bigfix.me/projects/details/25. For more information on BigFix management of Carbon Black, see the *IBM BigFix and Carbon Black Integration* document on the IBM developerWorks site.
- **Removal of Malware Identified by Cb Response** – The Cb Response BigFix connector processes data that is streamed from Cb Response and sends it to BigFix. The connector also sends alerts for applications that have been detected as vulnerable, so that BigFix can suggest high-priority patches and provide the location of banned files that attempted to execute in your environment. Administrators can then decide whether to delete affected files from within their environment. The Cb Response BigFix connector requires the Cb Event Forwarder standalone service, which pushes data from the Cb Response server to the Cb Response BigFix connector. For more information on Cb Event Forwarder, see "Cb Event Forwarder" on page 56. For more information on the Cb Response BigFix connector, see "Configure Cb Response BigFix Connector" on page 78.
- **Remediation of Vulnerabilities Identified and Prioritized by Cb Response** – Using the Manage Vulnerable Computers dashboard, you can view and remediate Carbon Black vulnerability data, known as Common Vulnerability Exposures (CVEs). The Manage Vulnerable Computers dashboard lists the CVEs, associated the CVE risk score and identifies the targeted computers and impacted computers.

- The dashboard provides a list of the Fixlets that are available for CVEs. Fixlets are the BigFix packages for the corrective action to remediate CVEs and security threats. You can also quarantine/unquarantine computers from the dashboard. For more information on the dashboard, see the *IBM BigFix and Carbon Black Integration* document on the IBM developerWorks site.

Note For more information on the components required on the BigFix side of this integration, visit the IBM developerWorks site at:

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Endpoint%20Manager/page/BigFix%20and%20Carbon%20Black%20Integration

Search for the “BigFix and Carbon Black Integration” documentation.

Installation Materials

To install this integration, you need installation files that are referenced in this document. Contact your Carbon Black customer service representative to obtain these files.

Cb Sensor Deployment and Health Monitoring

Carbon Black’s BigFix Fixlet deploys, monitors, manages, and troubleshoots the CbResponse sensor from the BigFix console. This facilitates the following system administration and lifecycle management tasks:

- Deploy Cb Response sensor.
- Identify machines lacking the Carbon Black sensor.
- Check current sensor status and configuration.
- Create deployment reports and dashboards in IBM BigFix.
- Enforce sensor state on- and off-networks.
- Sensor debug and diagnostic health data for rapid troubleshooting.
- Uninstall Cb Response sensor.

Requirements

The requirements are as follows:

- Cb Response Server 5.1.*x*, 5.2.*x*, or 6.1.*x*
- IBM BigFix v9.2.x and v9.5.x

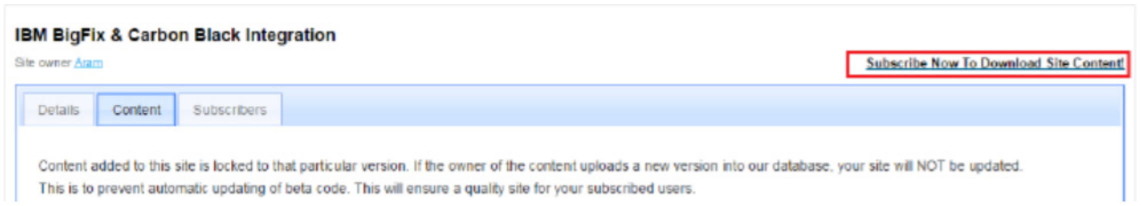
Importing the Content

The content associated with this integration is currently available at:

https://bigfix.me/site/details/8345#tabs-2

Note The content in this site is supported by Carbon Black.

On the bigfix.me site, register if necessary, and then log in. Subscribe to the site to download the content, as shown in the following example:


IBM BigFix & Carbon Black Integration

Site owner Aram

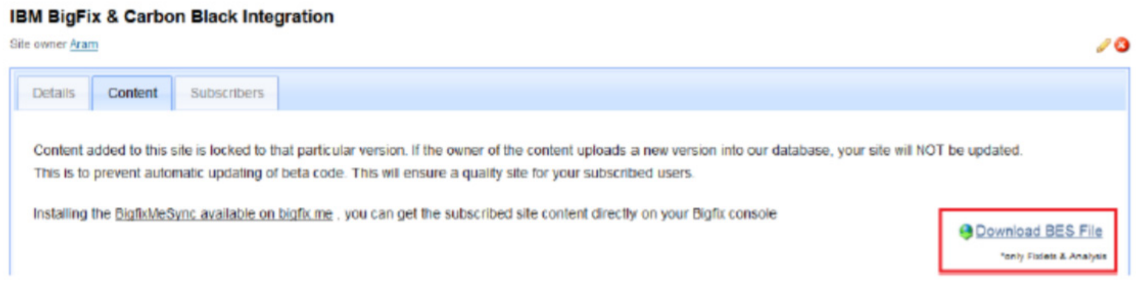
Subscribe Now To Download Site Content!

Details | Content | Subscribers

Content added to this site is locked to that particular version. If the owner of the content uploads a new version into our database, your site will NOT be updated. This is to prevent automatic updating of beta code. This will ensure a quality site for your subscribed users.

After you subscribe, you can import the content in three different ways:

- Use the BigFixMeSync Plugin at: https://bigfix.me/fixlet/details/9287.
- Download a single .BES file containing all of the site's content, and import it into the console by double-clicking the downloaded .BES file, as shown in the following example:


IBM BigFix & Carbon Black Integration

Site owner Aram

Details | Content | Subscribers

Content added to this site is locked to that particular version. If the owner of the content uploads a new version into our database, your site will NOT be updated. This is to prevent automatic updating of beta code. This will ensure a quality site for your subscribed users.

Installing the BigfixMeSync available on bigfix.me , you can get the subscribed site content directly on your Bigfix console

Download BES File

*only Fixlets & Analysis

- Download the individual .BES files associated with the content you want, and import them individually.

For information about how to import a .BES file, see: http://www.ibm.com/support/knowledgecenter/SSQL82_9.5.0/com.ibm.bigfix.doc/Platform/Console/Dialogs/import.html.

Content Details

The following sections provide information about the content that is available.

The Fixlets are organized into three categories:

- "Software Deployment" on page 74
- "Application Maintenance" on page 75
- "Troubleshooting" on page 76

Software Deployment

The software deployment Fixlets are as follows:

- **Cb Response - Deploy Windows Sensor**
- **Cb Response - Deploy Linux Sensor**

  To leverage these Fixlets, add the appropriate Cb Response Sensor installer in the root server's Uploads folder at:

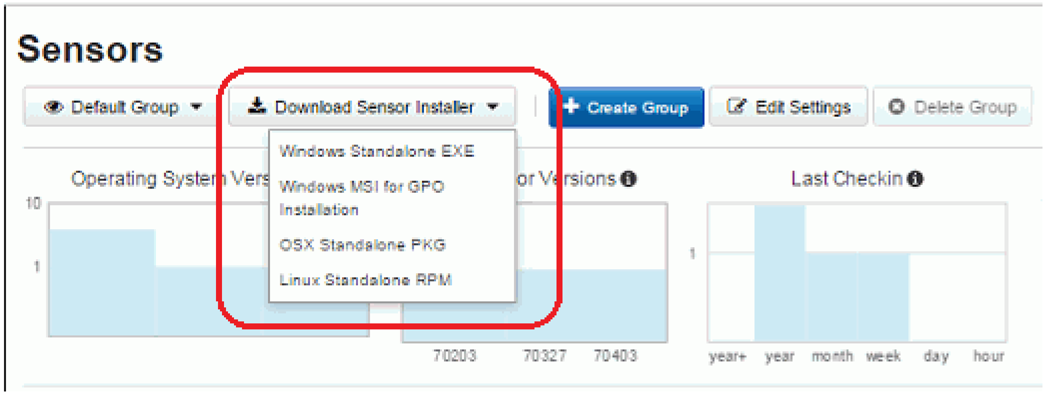
  ```
  ..\BES Server\wwwrootbes\Uploads
  ```

or

`/var/opt/BESServer/wwwrootbes/Uploads`.

Modify the action script of the Fixlet accordingly. For additional details on how to manually cache a file on the BigFix server, see http://www-01.ibm.com/support/docview.wss?uid=swg21506037.

To download the sensor installers from the Cb Response console, navigate to **Sensors** in the left navigation menu, click **Download Sensor Installer** in the top menu bar, and select the appropriate sensor:



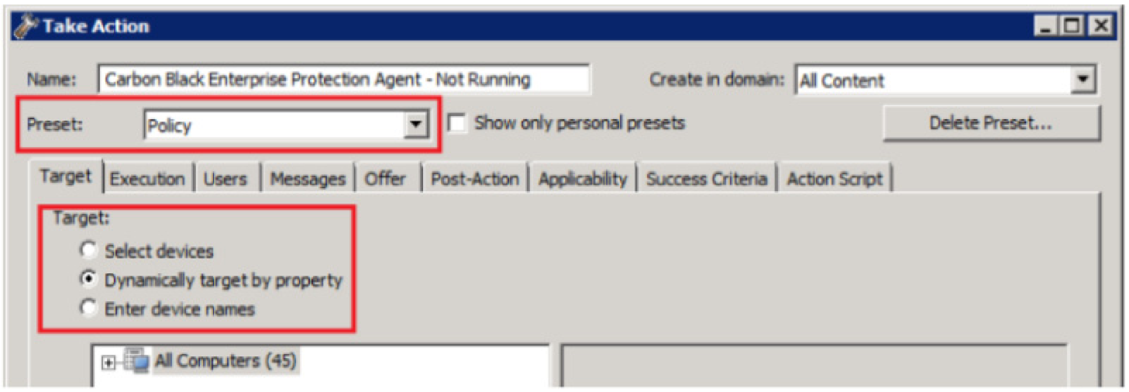
For more information about how to manually cache a file on the BigFix Server, see: http://www-01.ibm.com/support/docview.wss?uid=swg21506037.

Application Maintenance

The Cb Response Fixlet can identify endpoints with the sensor not running, and ensure that it (the sensor) continues running (even when off the network).

To deploy the Fixlet as a policy to enforce sensor compliance:

1. Click **Take Action**.
2. Select **Policy** from the **Preset** drop-down menu.
3. In the **Target** tab, select the devices in scope:



Name: Carbon Black Enterprise Protection Agent - Not Running
Create in domain: All Content
Preset: Policy
Show only personal presets
Target:
Select devices
[✓ Dynamically target by property]
Enter device names

Troubleshooting

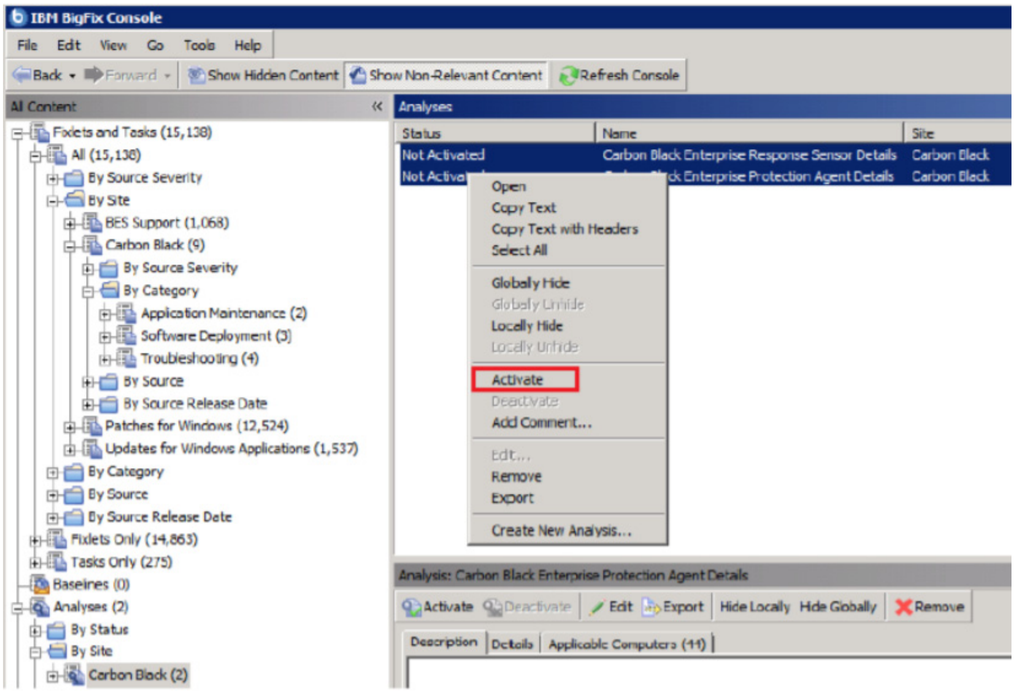
A number of tasks are provided to facilitate data collection and troubleshooting as follows:

- **Generate and Collect Sensor Diagnostics** – Directs the Cb Response sensor to generate diagnostics data and upload it to the BigFix Server. These diagnostic files are often requested by the Carbon Black team during support calls.
- **Force Sensor Check-In** – In typical operation, the Cb Response sensor checks in with the server every few minutes to upload new event data and binaries. When testing or diagnosing communications, it is useful to use this task to instruct the sensor to check in immediately and upload its data. Similarly, this could be combined with the Carbon Black server's option to force the upload of all dta on the next check in to quickly pull all information available from the sensor.
- **Uninstall Cb Sensor** – If you want to uninstall the Cb Response sensor, for example, if you need to migrate an endpoint to a different Carbon Black deployment, you can use task can automate the uninstallation procedure.

Analyses

To collect information about Cb Response, activate the following analyses from the BigFix Console:

1. Log on as a master operator.
2. Select the analyses.
3. Right-click the analyses and select **Activate**, as shown in the following example:



Status	Name	Site
Not Activated	Carbon Black Enterprise Response Sensor Details	Carbon Black
Not Activa	...k Enterprise Protection Agent Details	Carbon Black

- **Cb Response Sensor Details**: this analysis returns data for the following elements:
  - CbR Version
  - CbR Installation Date
  - CbR Service State
  - CbR Configuration/Profile Name
  - CbR Backend Server
  - CbR Sensor ID
  - CbR Sensor Modules/Collect Configuration

Configure Cb Event Forwarder for BigFix Integration

This section explains how to configure Cb Event Forwarder for the BigFix integration. For more information on installing and using Cb Event Forwarder, see "Cb Event Forwarder" on page 85.

To configure Cb Event Forwarder for the BigFix integration:

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>.`

   c. 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.

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

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

Configure Cb Response BigFix Connector

To configure the Cb Response BigFix connector:

1. Listens to data from a Cb Response server through the Cb Event Forwarder (see "Cb Event Forwarder" on page 85 and "Installation Materials" on page 73).
2. Processes the received data.
3. Sends the data over to BigFix.

This connector provides these services:

- Alerts BigFix about vulnerable applications that are running on endpoint devices.
- Suggests connections between vulnerable applications and watchlist hits.
- Automatically creates BigFix Fixlets to remove banned binaries from the endpoint systems that attempted to execute.

The connector requires an installed and properly configured Cb Event Forwarder and an API authentication information for both the Cb Response and IBM BigFix servers.

The following topics are covered in this section:

Requirements

Cb Response BigFix connector integration requires the following:

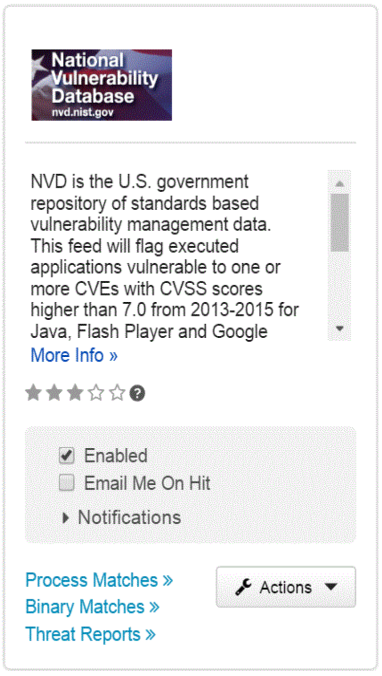
- Cb Response Server 5.1.*x*, 5.2.*x*, or 6.1.*x*. For more information, see the *Cb Response User Guide*.
- Cb Event Forwarder version 3.0 or newer. For more information, see "Cb Event Forwarder" on page 85.
- Installation on a Cb Response server (or a server running either CentOS 6 or Red Hat 6).
- Network connectivity to the BigFix API from the server where the connector is installed.

Enable the NVD Feeds in Cb Response

To enable the NVD feed:

1. Log into the Cb Response console.
2. In the left navigation menu, select **Threat Intelligence**.
3. Locate the **National Vulnerability Database (NVD)** option.

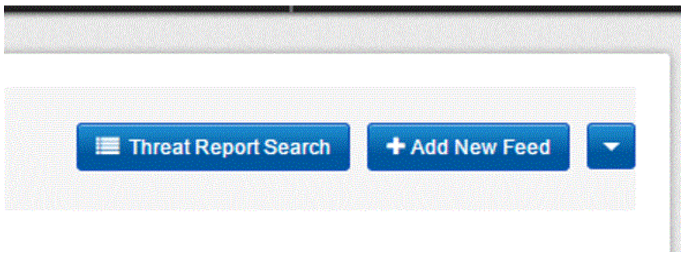
4. Select **Enabled**:


NVD is the U.S. government repository of standards based vulnerability management data. This feed will flag executed applications vulnerable to one or more CVEs with CVSS scores higher than 7.0 from 2013-2015 for Java, Flash Player and Google

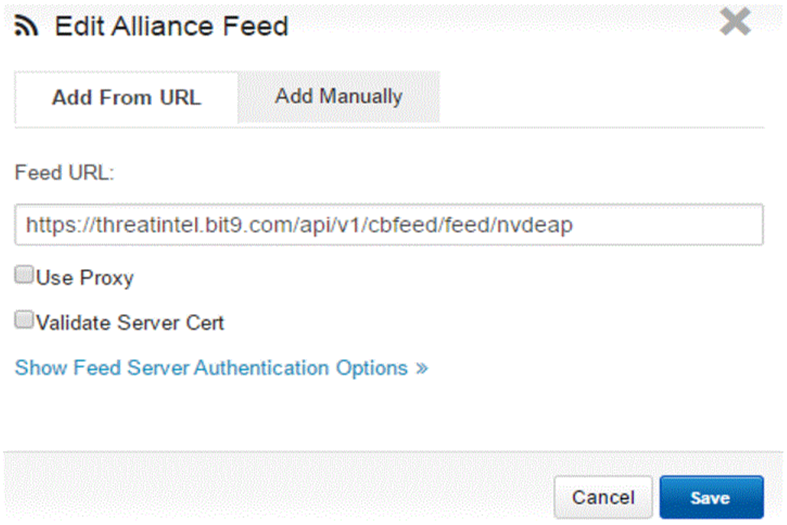
In addition to enabling the built-in NVD feed, also add in the updated NVD feed containing a much richer and personalized feed through the following process.

To add in the updated NVD feed:

1. Log into the Cb Response Console.
2. In the left navigation menu, select **Threat Intelligence**.
3. Click the **Add New Feed** button in the top-right corner of the page.



4. In the **Feed URL** field, enter the following URL:
   https://threatintel.bit9.com/api/v1/cbfeed/feed/nvdeap



Feed URL: https://threatintel.bit9.com/api/v1/cbfeed/feed/nvdeap

5. Click **Save**.
6. You should now see two NVD feeds at the bottom of the page. Ensure both are enabled.

Installation

To install the Cb Response BigFix connector:

1. Log in as “root” on the target Linux system
2. 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
```

3. Install Cb Event Forwarder. For more information, see “Cb Event Forwarder” on page 85.
4. Install the Cb Response BigFix connector RPM:

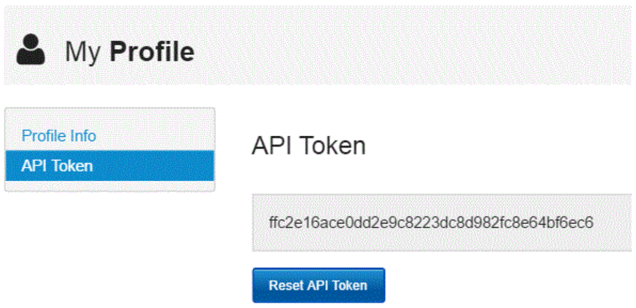
```
rpm -i cb-response-bigfix-connector.rpm
```

Obtain an API Token

To obtain a Cb Response API token:

1. Log into the Cb Response server with the appropriate account.
2. In the top-right corner of the Cb Response console, select ***Username*** **> My Profile**.
3. On the **My Profile** page, click **API Token** in the left menu.

4. Copy the displayed API token to a temporary location such as in Notepad. You will need this in the next section "Configure the Integration" on page 82.



ffc2e16ace0dd2e9c8223dc8d982fc8e64bf6ec6

Obtain a BigFix API Username and Password

Obtain a dedicated, separate BigFix operator account for this integration. This account will be used for the dashboard data variables that interface with the plugin as well as the banned files Fixlets.

Required BigFix Permissions

Master operators can access the IBM BigFix Console to change permissions. Users must have certain permissions to read/write to the BigFix Manage Vulnerable Computers dashboard and Fixlet APIs. For more information, see http://www.ibm.com/support/knowledgecenter/SS6MCG_9.5.0/com.ibm.bigfix.doc/Platform/Config/c_understanding_operator_rights.html.

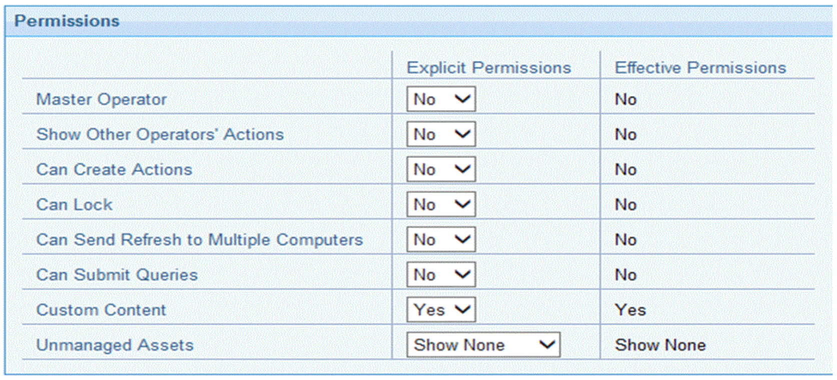
To set required BigFix permissions:

1. In the IBM BigFix Console, navigate to **All Content > Operators**.

2. Select the appropriate **Operator**

3. Navigate to the **Details** tab and set the following:

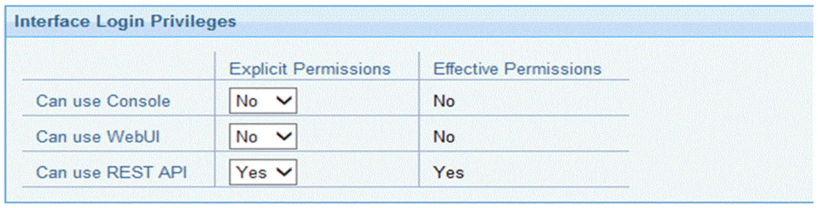
a. Under **Permissions**, set **Custom Content** to **Yes**.



Permissions

	Explicit Permissions	Effective Permissions
Master Operator	No	No
Show Other Operators' Actions	No	No
Can Create Actions	No	No
Can Lock	No	No
Can Send Refresh to Multiple Computers	No	No
Can Submit Queries	No	No
Custom Content	Yes	Yes
Unmanaged Assets	Show None	Show None

b. Under **Interface Login Privileges**, set **Can use REST API** to **Yes**:



Interface Login Privileges

	Explicit Permissions	Effective Permissions
Can use Console	No	No
Can use WebUI	No	No
Can use REST API	Yes	Yes

Note For more information on this integration from the BigFix side, visit the IBM developerWorks site at:

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Endpoint%20Manager

Search for the "BigFix and Carbon Black Integration" documentation.

Configure the Integration

Edit the following file using this procedure:

`/etc/cb/integrations/bigfix/connector.config`

1. Under the `cb-event-forwarder` heading, enter the port through which Cb Response will receive data from Cb Event Forwarder. The default value is 9999.

```
listen_port = <portnumber>
```

2. Under the `cb-enterprise-response` heading, enter the URL of the Cb Response server and the API token that you obtained in "Obtain an API Token" on page 80.

Note Make sure the API token has access permissions to all feed and watchlist configuration. A global administrator token is required for this.

```
url = https://<dns-name>:<port>/
api_token = <api_token>
```

3. Under the `ibm-bigfix` heading, enter the DNS or IP address and the network port for the server's API interface (not web console). The default port is `52311`. Also, provide the username and password for accessing the BigFix server.

Note On the BigFix side, users must have certain permissions to read/write to the BigFix dashboard and Fixlet APIs (see "Required BigFix Permissions" on page 81).

```
url = <dns_name>:<api_network_port>

username = <bigfix_api_username>

password = <bigfix_api_password>
```

4. Under the `ibm-bigfix` heading, enter the BigFix custom site name. This must be the exact site name that is used when configuring BigFix side of this integration:

```
bigfix_custom_site_name = <site name>
```

5. Under the `integration-core heading`, enter the names of the watchlists you would like the connector to use as “security events”. Cb Response uses hits on these watchlists to implicate a vulnerable process as the potential cause of the detected issue.

   You can choose which watchlists are listened to. By default, Cb Response listens to the built-in watchlist “Alliance: VirusTotal Score > 3”, but Carbon Black highly recommends inserting your own watchlists that represent your environment. Hits on these watchlists will be considered indicators of compromise for your environment and trigger this connector’s processing to indicate suspected vulnerabilities linked to the compromise within the BigFix console.

   ```
   integration_implication_watchlists = [
     “<watchlist_name>”,
     “<watchlist2_name>”,
     ...
   ]
   ```

Starting and Stopping the Connector

Once the Cb Response BigFix connector is installed, the integration automatically registers to be launched upon startup. However, immediately after the installation, the connector remains inactive to allow for user configuration.

Start Connector

When you have finished the Cb Response BigFix connector configuration, start the service as follows:

```
sudo start cb-response-bigfix-connector
```

Note When the connector starts, it creates a special watchlist for this integration. The watchlist is a product of the feed names indicated in the advanced section of the configuration file and could be updated at any time. As a result, do not rely on this watchlist for reasons beyond what is required by this connector.

Restart Connector

If you make additional configuration changes, restart the connector using this command:

```
sudo restart cb-response-bigfix-connector
```

Stop Connector

To temporarily stop the connector, use this command:

```
sudo stop cb-response-bigfix-connector
```

Note The service resumes upon reboot. If you want the service to remain powered off, follow the instructions in “Uninstall” on page 84.

Troubleshooting

If you notice problems with the Cb Response BigFix connector (or if Carbon Black support asks you for details during troubleshooting), you may be asked to provide the Cb Response BigFix connector log file. The output log file for the connector is located at:

```
/var/log/cb/bigfix/connector.log
```

Any issues during startup of the connector service may be found in:

```
/var/log/cb/bigfix/connector.errors
```

By default, the logging level is set to a minimal level. If you need more detailed logging, alter the log level within the configuration file here:

```
/etc/cb/integrations/bigfix/connector.config
default_log_level = ERROR
```

Select one of the following log levels:

- `CRITICAL`
- `ERROR`
- `WARNING`
- `INFO`
- `DEBUG`

> **Note** Remember to lower the log level after you have collected the information you need. Otherwise, the log file can become extremely large.

Uninstall

To uninstall the Cb Response BigFix connector, run this command:

```
rpm -e cb-response-bigfix-connector
```

Banned Files

When Cb Response bans files, the following occurs:

1. When a banned file is executed and blocked, the Cb Response server sends a notification through the Cb Event Forwarder.
2. The Cb Response BigFix connector (which is always listening to the Cb Event Forwarder) identifies the blocked execution event.
3. The service creates new (or updates existing) Fixlets. It creates a single Fixlet per MD5 hash (banned file) and sends these to the BigFix server with the new location.
4. The BigFix operator can run the Fixlet at any time to delete the banned files.

> **Note** Cb Response can only remove the banned files that have attempted to execute. This might not include all file instances within the enterprise.

Chapter 11

Cb Event Forwarder

This chapter introduces Cb Event Forwarder, 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.

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.

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 12, "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 51.

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` file 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 87 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 connectors.

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 86.

   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 12

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 85.

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 shown 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 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
```