Carbon Black
Syslog Integration Procedures

29-Oct-2013
Support@Bit9.com

Introduction

One of the numerous integration options available with Carbon Black is utilizing the syslog facilities for notification and data intelligence sharing. The open nature implemented within the Carbon Black server allows for a great number of possibilities, and signifies Carbon Black's leadership in integration capabilities.

Table of Contents

Carbon Black Syslog Integration

First Steps

Directing alerts to syslog files allows for a variety of integration options for numerous platforms. Specific fields might vary depending upon the watchlist parameters chosen during creation. Please refer to Search Syntax guide for specific fields to use when crafting queries, and refer to the Server Configuration guide for help creating a watchlist once the query is finalized.

Carbon Black Search Syntax Reference

Carbon Black Server Configuration Guide - Basic

Setting up the remote device

Regardless if the remote device is an instance of SPLUNK, ArcSight, or another manager-of-managers platform such as Tivoli, the basic setup requirements hold. The remote device must be configured with a new receiver to accept the rsyslog feed from Carbon Black. As this method will differ depending upon the device itself only the basics are described below and should be adapted to your particular platform.

1. Add a new UDP receiver to the remote device
2. Enable the new receiver to communicate using a new and unique UDP port number for the communication with Carbon Black.
    a. The system may require the Carbon Black IP address to be authorized prior to accepting data.
3. Verify the receiver is working and listening on the appropriate port.

Setting up the Carbon Black server to send data to the remote device

On the Carbon Black server the rsyslog feature will be used to transmit each watchlist hit to the remote device or multiple remote devices depending upon the needed configuration.

Begin by accessing the Carbon Black server either through the console or remote terminal connection using SSH. The rsyslog file below needs to be edited to allow for syslog information to be redirected.

```
/etc/rsyslog.d/cb-coreservices.conf
```

Example output from an unaltered cb-coreservices.conf file.

```
# By default the value of this directive is 'on' so that any special
character (ASCII < 32) is escaped. However,
# that causes multiline messages to be rather unreadable. While the practice
```

```
of printing multiple lines in a log
# should be discouraged, it is useful when error exception stack tracers are
being reported.  This option might
# also cause problems if other log file reader software is being used as it
may not be able to read additional
# lines as those lines wouldn't have any timestamp/source information.
#
# If this option is causing problems, it can be disabled which would make
interpreting stack traces a bit more
# difficult. However, the following command can be used when reading log
files to make stack traces readable again:
#     cat /path/to/log/file | sed 's/#012/\n\t/g'
#
$EscapeControlCharactersOnReceive off

$template AccessLogFormat,"%msg%\n"
$template CbLogFormatWithPID,"%timegenerated% [%procid%] <%syslogseverity-
text%> %msg%\n"

$template DynaFile,"/var/log/cb/notifications/%PROGRAMNAME%.log"

if $programname startswith 'process' then -?DynaFile

if $programname == 'cb-coreservices' and $syslogfacility-text == 'local0'
then /var/log/cb/coreservices/debug.log;CbLogFormatWithPID
& ~
if $programname == 'cb-coreservices' and $syslogfacility-text == 'local7'
then /var/log/cb/coreservices/access.log;AccessLogFormat
& ~
if $programname == 'cb-allianceclient' and $syslogfacility-text == 'local0'
then /var/log/cb/allianceclient/allianceclient.log;CbLogFormatWithPID
& ~
if $programname == 'cb-job-runner' then /var/log/cb/job-runner/job-
runner.log;CbLogFormatWithPID
& ~
if $programname == 'cb-notifications' then /var/log/cb/notifications/cb-all-
notifications.log;CbLogFormatWithPID
& ~
if $programname startswith 'cb-notifications-' then -
?DynaFile;CbLogFormatWithPID
& ~
if $programname == 'cb-services' then
/var/log/cb/services/init.log;CbLogFormatWithPID
& ~
```

Setting up Carbon Black to send all data to the remote device

Directing all watchlist output is a very easy configuration step. Simply add the remote device IP address to the cb-all-notifications parameter, and all watchlist syslog entries will be sent.

1. Log into the Carbon Black server
2. Edit the cb-coreservices.conf file
   ```
   vi /etc/rsyslog.d/cb-coreservices.conf
   ```
3. Add the following line to the configuration file under the cb-all-notifications line:

   ```
   if $programname == 'cb-notifications' then
   /var/log/cb/notifications/cb-all-
   notifications.log;CbLogFormatWithPID
   & @<remote device IP address>:<UDP port>;CbLogFormatWithPID
   & ~
   ```

4. Restart the rsyslog daemon so the changes take effect:
   ```
   service rsyslog restart
   ```
5. Verify the data is now present in the remote device

Setting up Carbon Black to send specific watchlist data to a remote device

Directing specific watchlist output is requires additional configuration steps to filter each watchlist independently.

1. Log into the Carbon Black server
2. Edit the cb-coreservices.conf file
   ```
   vi /etc/rsyslog.d/cb-coreservices.conf
   ```
3. Add the following line to the configuration file. Note here the specific watchlist must be specified. Verify the watchlist ID from within the Carbon Black UI prior to adding these lines to ensure the correct watchlist is forwarded.
   note this entire section must be added to the cb-coreservices.conf file

   ```
   if $programname == 'cb-notifications-watchlist-105' then
   /var/log/cb/notifications/cb-notifications-watchlist-
   105.log;CbLogFormatWithPID
   & @<remote device IP address>:<UDP port>;CbLogFormatWithPID
   & ~
   ```

4. Restart the rsyslog daemon so the changes take effect:
   ```
   service rsyslog restart
   ```
5. Verify the data is now present in the remote device

Enabling survivability of the notifications during communication interruptions

To allow for notifications to be spooled on the Carbon Black server if communication with the remote device is interrupted the following lines need the comment markup removed. All the "#" symbols need to be removed per the example below. Any "#" symbol not highlighted in red should NOT be removed. To do this, edit the file:

`/etc/rsyslog.conf`

```
//
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.
#$WorkDirectory /var/lib/rsyslog # where to place spool files
#$ActionQueueFileName fwdRule1 # unique name prefix for spool files
#$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as
possible)
#$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
#$ActionQueueType LinkedList   # run asynchronously
#$ActionResumeRetryCount -1    # infinite retries if host is down
//
```

Appendix - Carbon Black Syslog Architecture

Logging Location

Carbon Black stores all logged information in the following folder. When troubleshooting any server side activity start within this logging structure first.

```
/var/log/cb/
```

Watchlist Log Location

Specific to watchlist created in the User Interface, Carbon Black maintains two separate syslog files for watchlists. The first is a single file with all watchlist hits consolidated in one place. The second type saves each watchlist hit to it's own file. All the watchlist syslog files are stored in a single location on the Carbon Black server per below:

```
/var/log/cb/notifications
```

Each watchlist is assigned a specific number which can be viewed from the User Interface per the example https://<server name>/#/watchlist/105. In this example the watchlist number is 105.

Carbon Black creates a numbered syslog that matches the watchlist number. So in our example above the watchlist 105 syslog creates the output file:

```
cb-notifications-watchlist-105.log-20131031
```

The syslog file name format follows a standard convention for all watchlists per below:

```
cb-notifications-watchlist-<watchlist#>.log-YYYYMMDD
```

The single summary syslog with all watchlist hits in one consolidated file uses the following naming convention:

```
cb-all-notifications.log-YYYYMMDD
```