CloudPassage Halo

The CloudPassage Halo DSM for IBM® Security QRadar® can collect event logs from the CloudPassage Halo account.

The following table identifies the specifications for the CloudPassage Halo DSM:

Table 1. CloudPassage Halo DSM Specifications

Specification	**Value**
Manufacturer	CloudPassage
DSM name	CloudPassage Halo
RPM file name	DSM-CloudPassageHalo-*build_number*.noarch.rpm
Supported versions	All
Event format	Syslog, Log file
QRadar recorded event types	All events
Automatically discovered?	Yes
Included identity?	No
More information	CloudPassage website (*www.cloudpassage.com*)

To integrate CloudPassage Halo with QRadar, use the following steps:

1. If automatic updates are not enabled, download the latest versions of the following RPMs:
   - DSMCommon RPM
   - CloudPassage Halo RPM
2. Configure your CloudPassage Halo to enable communication with QRadar.
3. If QRadar does not automatically detect CloudPassage Halo as a log source, create a CloudPassage Halo log source on the QRadar Console.

Configuring CloudPassage Halo for communication with QRadar

To collect CloudPassage Halo events, download and configure the CloudPassage Halo Event Connector script to send syslog events to QRadar.

Before you begin

Before you can configure the Event Connector, you must create a read-only CloudPassage API key. To create a read-only key, log in to your CloudPassage Portal and click **Add New Key** on the Site Administration window.

About this task

The Event Connector script requires Python 2.6 or later to be installed on the host on which the Event Connector script runs. The Event Connector makes calls to the CloudPassage Events API, which is available to all Halo subscribers.

Note: You can configure the CloudPassage Halo Event Collect to write the events to file for QRadar to retrieve by using the Log File Protocol, however, this method is not recommended.

Procedure

1. Log in to the CloudPassage Portal.
2. Go to to **Settings > Site Administration**.
3. Click the **API Keys** tab.
4. Click **Show** for the key you want to use.
5. Copy the key ID and secret key into a text file.

   Ensure that the file contains only one line, with the key ID and the secret key separated by a vertical bar/pipe (|), for example, `your_key_id|your_secret_key`. If you want to retrieve events from multiple Halo accounts, add an extra line for each account.
6. Save the file as `haloEvents.auth`.
7. Download the Event Connector script and associated files from https://github.com/cloudpassage/halo-event-connector-python.
8. Copy the following files to a Linux or Windows system that has Python 2.6 (or later) installed:
   - haloEvents.py
   - cpapi.py
   - cputils.py
   - remote_syslog.py (use this script only if you deploy the Event Connector on Windows and you want to send events through syslog)
   - haloEvents.auth
9. Set the environment variables on the Linux or Windows system:
   - On Linux, include the full path to the Python interpreter in the PATH environment variable.
   - On Windows, set the following variables:
     - Set the PATH variable to include the location of haloEvents.py and the Python interpreter.
     - Set the PYTHONPATH variable to include the location of the Python libraries and the Python interpreter.
10. To send events through syslog with the Event Connector is deployed on a Windows system, run the haloEvents.py script with the **`--leefsyslog=<QRadar IP`**> switch:

    ```
    haloEvents.py --leefsyslog=1.2.3.4
    ```

    By default, the Event Connector retrieves existing events on initial connection and then retrieves onlynew events thereafter. To start event retrieval from a specific date, rather than retrieving all historical events on startup, use the **`--starting=<date>`** switch, where date is in the YYYY-MM-DD format:

    ```
    haloEvents.py --leefsyslog=1.2.3.4 --starting=2014-04-02
    ```
11. To send events through syslog and deploy the Event Connector on a Linux system, configure the local logger daemon.
    a. To check which logger the system uses, type the following command:

       ```
       ls -d /etc/*syslog*
       ```

       Depending on what Linus distribution you have, the following files might be listed:

       -

  - rsyslog.conf
  - syslog-ng.conf
  - syslog.conf

b. Edit the appropriate .conf file with relevant information for your environment.

   Example configuration for syslog-ng:

```
source s_src {
      file("/var/log/leefEvents.txt");
};
destination d_qradar {
     udp("qradar_hostname" port(514));
};
log {
     source(s_src); destination(d_qradar);
};
```

c. To run the `haloEvents.py` script with the **`leeffile=<filepath>`** switch, type the following command:

   `haloEvents.py --leeffile=/var/log/leefEvents.txt`

   You can include **`--starting=YYYY-MM-DD`** switch to specify the date from which you want events to be collected for on initial startup.

   Note: As an alternative to using syslog, you can write events to a file for QRadar to retrieve by using the Log File protocol. For Windows or Linux to write the events to a file instead, use the **`--leeffile=<filename>`** switch to specify the file to write to.

Configuring a CloudPassage Halo log source in QRadar

To collect CloudPassage Halo events, configure a log source in QRadar.

Procedure

1. Log in to QRadar.
2. Click the **Admin** tab.
3. In the navigation menu, click **Data Sources**.
4. Click the **Log Sources** icon.
5. Click **Add**.
6. From the Log Source Type list, select **CloudPassage Halo**.
7. From the Protocol Configuration list, select **Syslog** or **Log File**.
8. Configure the remaining parameters:
9. Click **Save**.
10. On the Admin tab, click **Deploy Changes**.