Idaptive Identity Services Platform (IISP)
SIEM Integration Guide

March 2019

Idaptive Corporation

Abstract

This guide describes how to configure the OAuth app and the SIEM user on a tenant, install a docked app that retrieves IISP event logs, and obtain guidelines to set up the Splunk Add-on for IISP.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal Notice</td>
<td>ii</td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Setting up the SIEM User and the OAuth App on the Tenant</td>
<td>2</td>
</tr>
<tr>
<td>Setting up the Environment for Linux</td>
<td>8</td>
</tr>
<tr>
<td>Step 1: Set up Docker on Linux</td>
<td>8</td>
</tr>
<tr>
<td>Step 2: Set up Syslog Server on Local Host Machine</td>
<td>8</td>
</tr>
<tr>
<td>Setting Up the Environment for Windows Server 2012 R2</td>
<td>9</td>
</tr>
<tr>
<td>Step 1: Set up Docker on Windows Server (Machine #1)</td>
<td>9</td>
</tr>
<tr>
<td>Step 2: Set up a Remote Syslog Server on Linux (Machine #2)</td>
<td>12</td>
</tr>
<tr>
<td>Running the Syslog Writer</td>
<td>13</td>
</tr>
<tr>
<td>Starting Syslog Writer</td>
<td>13</td>
</tr>
<tr>
<td>On-Premises Tenant</td>
<td>14</td>
</tr>
<tr>
<td>Automatic Restarting</td>
<td>17</td>
</tr>
<tr>
<td>Checking Execution Logs</td>
<td>18</td>
</tr>
<tr>
<td>Re-Running Syslog Writer After Cleanup</td>
<td>18</td>
</tr>
<tr>
<td>Restarting a Stopped Syslog Writer Container</td>
<td>19</td>
</tr>
<tr>
<td>Stopping Syslog Writer Container</td>
<td>19</td>
</tr>
<tr>
<td>Important Parameters in data/config.ini</td>
<td>19</td>
</tr>
<tr>
<td>Important note about data load</td>
<td>20</td>
</tr>
<tr>
<td>Splunk IISP Add-on</td>
<td>20</td>
</tr>
<tr>
<td>Setting Up Splunk Universal Forwarder</td>
<td>20</td>
</tr>
<tr>
<td>Installing the Splunk Add-on</td>
<td>21</td>
</tr>
<tr>
<td>Configuring Data Input</td>
<td>21</td>
</tr>
<tr>
<td>Searching for IISP Events</td>
<td>22</td>
</tr>
</tbody>
</table>
Introduction

Syslog Writer is a docked application that captures events from Idaptive Identity Services and logs them to a syslog server. This syslog then becomes the data source for a SIEM solution using Splunk.

Syslog Writer is configured to start fetching IISP events from the previous day and then run every five minutes to fetch events incrementally. Events are fetched from the IISP server using REST APIs after authenticating via OAuth client credentials.

This document provides the instructions to install and configure Docker and Syslog server. It specifically focuses on CentOS 6.9 and Windows Server 2012 R2.

The supported platforms to use Syslog Writer for Idaptive Identity Platform on cloud, include:

- CentOS 6.9, CentOS 7
- RHEL 6.5, RHEL 7
- Windows Server 2012 R2

This document supplies:

- Detailed steps to configure the OAuth app and the SIEM user on a tenant as a prerequisite for setting up Syslog Writer
- Installation procedures for Docker and an interactive configuration to set up the Syslog Writer on Linux or Windows
- Guidelines for the user to set up the Splunk add-on for IISP (for Splunk 6.5.x and above)
Setting up the SIEM User and the OAuth App on the Tenant

1. On the Idaptive Admin Portal, go to **Apps > Web Apps**.

2. Click **Add Web Apps**, then click **Custom**.

3. Locate the **OAuth2 Client** and click **Add**.

4. When prompted, add the Web App, OAuth2 Client, click **Yes**.

5. From the Settings tab, in the Application ID field, enter **oauthsiem**.

6. From the General Usage tab, leave the defaults as shown.
7. From the **Tokens** tab, for **Auth methods**, check **Client Creds** and click **Save**

![Tokens tab](image)

8. From the **Scope** tab, under Scope definitions, click **Add** to add a new scope.

9. In the Scope definitions window:
   
   a. In the Name field, enter **siem**.
   
   b. In the Allowed REST APIs section, click **Add**, and enter **Redrock/query.**

   c. Click **Save**.

10. On the Idaptive Admin Portal, go to **Core Services > Users > Add User**.

11. In the **Create Idaptive Directory User** page:
   
   a. For the Login Name, enter **siemuser**.
   
   b. For the Suffix, enter **idaptive.com** (or leave as is).
   
   c. For the Password and Confirm Password, enter the password of your choice.
12. For Status:
   
   a. Check **Password never expires**.
   
   b. Select **Is OAuth confidential client (Preview)**. This automatically also selects the options **Password never expires** and **Is Service User**.

13. On the Idaptive Admin Portal, go to **Core Services > Roles > Add Role**.

14. Once page opens, In Description tab:

   For the Name, enter **service account** and click **Save**.

   This entry serves as the role name.

15. Open the newly created role, and from the Members tab:

   a. Click **Add** and search the siemuser that you created earlier (in Step 11 a).

   b. Click **Save**.

16. Open the **Administrative Rights** tab:
a. Click **Add**.

b. In the **Add Rights** list, check **Read Only System Administration**

c. Click **Add**.

d. Click **Save**.
17. On the **Assigned Applications** tab:
   
   a. Click **Add**.
   
   b. In Add Applications list, check **OAuth2 Client**.
   
   c. Click **Add**.
   
   ![Assigned Applications Diagram]

   d. Click **Save**.
18. Make final checks:

- On the Idaptive Admin Portal, from the Core Services > Users tab:
  - The siemuser created earlier, is shown as the Idaptive Directory User. Click it to open the user’s page.
  - In Roles section for this user, the role named service account must be listed, with Read Only System Administration in Administrative Rights.

- On the Idaptive Admin Portal, on the Apps > Web Apps tab:
  - Select OAuth2 Client
  - In the Permissions tab > Name column shows the earlier created role service account with the View permission checked.
Setting up the Environment for Linux

Syslog Writer on Linux will most likely involve only one machine, as the Syslog server can be a local one. This machine will have Docker to run Syslog Writer, and the Syslog server.

Step 1: Set up Docker on Linux

The installation commands in this section are specific to CentOS 6.9. If you have another supported OS, the following link takes you to the installation instructions for Docker on your OS:

https://docs.docker.com/engine/installation/

1. Make sure that the existing yum packages are updated and that the EPEL repository is enabled:

   ```
   sudo yum update
   sudo yum install -y epel-release
   ```

2. Install Docker:

   ```
   sudo yum install -y docker-io
   ```

3. Start the Docker daemon:

   ```
   sudo service docker start
   ```

4. Make sure that the Docker service is running:

   ```
   sudo service docker status
   ```

Step 2: Set up Syslog Server on Local Host Machine

1. Allow TCP input in the Syslog server configuration.

   TCP ensures that no messages are lost even when the load is high, so it is used by Syslog Writer for logging events to the Syslog server (local or remote). The steps in this section are specific for setting up rsyslog to receive TCP input.

   **NOTE:** If you have a different syslog server, you will need to modify the syntax accordingly.
a. Open the rsyslog configuration file:

```
sudo vi /etc/rsyslog.conf
```

b. Uncomment the following two lines in the rsyslog.conf file (if they are commented out):

```
$ModLoad imtcp
$InputTCPServerRun 514
```

c. Restart the rsyslog server:

```
sudo service rsyslog restart
```

2. Monitor syslog

Before starting Syslog Writer, it is helpful to keep another terminal window open to check syslog:

```
sudo tail -f /var/log/messages
```

**NOTE:** The path for Ubuntu is: `/var/log/syslog`

---

**Setting Up the Environment for Windows Server 2012 R2**

Syslog Writer on Windows box can currently only write the IISP events to a remote syslog server on Linux, so the setup will involve two machines:

- **Machine #1** runs Windows with Docker and the Syslog Writer app.
- **Machine #2** runs Linux with Syslog server, and is in the same network.

**Step 1: Set up Docker on Windows Server (Machine #1)**

The installation commands in this section are specific to Windows 10 Professional edition, running a VMWare virtual machine of Windows Server 2012 R2.

If you are using another supported platform, you must modify the syntax for setting up the Docker Toolbox for your machine.

1. To enable virtualization, power **OFF** your Windows Server VM.
2. In setting Devices > Processors, select Virtualize Intel VT-x/EPT or AMD-V/RVI as the Virtualization Engine.

3. Power ON the Windows server VM again.
4. Install Docker Toolbox for Windows by following these instructions: https://docs.docker.com/toolbox/toolbox_install_windows/
5. Open the Docker Quickstart terminal to create a default Docker machine and provide a prompt for running the Docker commands.
6. Add a shared folder on the host machine by opening Oracle Virtualbox and click **Settings > Shared Folders** for the default Docker machine.
7. In **Shared Folders**, make sure that `c:\Users` is listed.

---

**Step 2: Set up a Remote Syslog Server on Linux (Machine #2)**

The steps in this section are specific for setting up rsyslog on CentOS 6.9.

*If you have a different syslog server, you must modify the syntax accordingly.*

1. Allow TCP input in syslog server configuration:
   
   a. Open the rsyslog configuration file:
      
      ```
      sudo vi /etc/rsyslog.conf
      ```
   
   b. Uncomment these two lines in the `conf` file (if they are commented out):
      
      ```
      $ModLoad imtcp
      $InputTCPServerRun 514
      ```
c. Restart the rsyslog server:

```bash
sudo service rsyslog restart
```

2. Allow the firewall to accept TCP input on port 514:

a. Open the `iptables` config file:

```bash
sudo vi /etc/sysconfig/iptables
```

b. Add this line before the `COMMIT` line, if it is not present already:

```
-I INPUT -p tcp --dport 514 -j ACCEPT
```

c. Restart `iptables`:

```bash
sudo service iptables restart
```

3. Monitor syslog:

Before starting syslog writer, it is helpful to keep another terminal window open to check the syslog:

```bash
sudo tail -f /var/log/messages
```

**NOTE:** The path for Ubuntu is `/var/log/syslog`

---

### Running the Syslog Writer

The commands in this section are shown in Linux format and are run in a terminal window on Linux. For Windows, open a Docker Quickstart terminal window and enter these commands without the `sudo` prefix.

### Starting Syslog Writer

1. Check the images list.
   - If the list contains `syslogwriter_image`, delete it first.

```bash
sudo docker images
```
2. Copy the zip file IISP_syslog_writer.zip from SFTP to a convenient location and extract it.

   On a Windows Server VM, make sure that the extracted folder is under the shared folder, c:\Users.
   For example, the extracted folder: C:\Users\<username>\apps\IISP_syslog_writer

3. Load the image from the tar file and make sure that syslogwriter_image is in the images list:

   ```
   cd <extracted_path>/IISP_syslog_writer
   sudo docker load < syslog-writer-img.tar.gz
   sudo docker images
   ```

4. Run the Syslog Writer container.

   For a cloud tenant, you can run this command directly. However, if you are using an On-Premise Tenant, refer to additional steps in next section before running the following run command.

   ```
   sudo docker run --name syslog-writer -it --log-driver json-file --log-opt max-size=10m --net=host -v `pwd`/data:/home/centrify-syslog-writer/data syslogwriter_image
   ```

5. When prompted, enter:
   - The Tenant URL (for example, https://aaa0056.my-dev.centrify.com/my)
   - The username and password of the SIEM user
   - The IP address of Syslog server if it is remote. Otherwise, just press enter for a local Syslog server. Note that for Windows Servers, this is the IP address of Machine #2.

6. Check IISP events on the Syslog server.

   **Congratulations! Your syslog writer is up and running!**
   The first run starts immediately. Because the default value for the frequency parameter is five minutes, the Syslog Writer will run once every five minutes.

**On-Premises Tenant**

For setting up syslog writer with an On-premises Tenant, you will need to carry out these additional steps before executing the docker run command.
Note: Below Linux commands are specific to Centos 6.9/10. You will need use an equivalent command for your specific Linux OS.

1. **Ensure that the Tenant name is reachable**
   From the Linux machine, on which you are planning to setup the Syslog Writer, you need to make sure that the tenant name is reachable (Check with ping or curl command).

   For example, if tenant URL is `https://webportal.centsiem-1.com` and the server IP address is 172.31.15.16

   For this, you need to have this below entry in `/etc/hosts` file on the Linux machine.

   ```
   172.31.15.16  webportal.centsiem-1.com
   ```

2. **Install the Server Root CA certificate in CA Bundle**
   The Syslog Writer internally makes an HTTPS REST API call to the Tenant for OAuth access and fetching events. So, for successful SSL certificate verification, the CA authority needs to be a trusted one.

   By design, for an On Premise tenant, the certificate is self-signed, so you must include its server Root CA certificate in CA store of Linux machine, on which the syslog writer will be setup,

   Please note that in case you have provided your own certificate when installing the on-premise tenant (not self-signed), you will need to add that certificate to the CA store of Linux machine, instead of the server's Root CA certificate, as detailed in the steps mentioned below. And if your certificate issuer is a universally known authority, then you don't need to add it to the CA store of the Linux machine (Machine running syslog-writer).

   For adding the Root CA certificate, first locate the certificate on Tenant Server machine, in path (similar to this), `C:\ProgramData\Centrify\Centrify Identity Platform\config\root_ca_public_certificate.cer`

   Copy the above-mentioned Root CA certificate to a Centos 6.9/10 or RHEL 6.10 machine (the host, on which you will run syslog writer app). Also extract the syslog writer zip file `IISP_syslog_writer.zip`.

   Inside the extracted folder `IISP_syslog_writer`, there is `scripts` folder that contains a script which adds the Root CA certificate to CA bundle and verifies whether the update happened successfully. It also copies the updated CA bundle to the required folder `(IISP_syslog_writer/data)` later - where the syslog writer would expect it.
The script also takes a backup of the original CA bundle in `~/ca_backup_<datestamp>` folder before updating it with Server Root CA certificate. You can restore the backed-up CA, in case of any unexpected problem.

If you are using a Centos 6.9/10 or RHEL 6.10 for syslog writer, you can run the provided script as shown below, to install the server Root CA certificate in CA bundle.

```
sudo ./scripts/update_ca_on_centos_or_rhel.sh <path_of_Root_CA_certificate>
```

On Ubuntu 16.04 or 18.04, you can use the script provided for ubuntu:

```
sudo ./scripts/update_ca_on_ubuntu.sh <path_of_Root_CA_certificate>
```

Note: If you were able to use the provided scripts to update the CA bundle, you can skip the rest of this section, and directly jump to next step (Step #3) for running the syslog writer.

For other Linux versions, please refer to this URL to manually install the Root CA certificate: [https://manuals.gfi.com/en/kerio/connect/content/server-configuration/ssl-certificates/adding-trusted-root-certificates-to-the-server-1605.html](https://manuals.gfi.com/en/kerio/connect/content/server-configuration/ssl-certificates/adding-trusted-root-certificates-to-the-server-1605.html)

If you have manually updated the CA bundle - without using the provided update_ca scripts, carry out the below 2 steps to verify whether the CA bundle was updated successfully and to copy the updated CA bundle into the expected folder with the expected name.

a. Verify whether the above update to CA bundle happened successfully. Below is an example shown on Centos 6.9.

   I. Convert from DER encoded Root CA certificate copied from windows OPIE server into a PEM encoded certificate using below command.

   ```
   openssl x509 -in root_ca_public_certificate.cer -inform der -outform pem -out rootcacert.pem
   ```

   II. Use the newly created rootcacert.pem file to verify the updated CA bundle in below command. (For Ubuntu, replace the highlighted CAfile path in below command with `/etc/ssl/certs/ca-certificates.crt`)

   ```
   openssl verify -verbose -CAfile /etc/pki/tls/certs/ca-bundle.crt rootcacert.pem
   ```
If the Root CA certificate was added successfully to the CA bundle in step 2, the output will be as below, otherwise it will show an error: In case of an error, you must not proceed to next step of starting syslog writer.

```
rootcacert.pem: OK
```

b. Copy the updated ca-bundle file to data folder, where the IISP_syslog_writer.zip was extracted.

```
.. IISP_syslog_writer/data/ca-bundle.crt
```

The CA certificates file copied into the data folder must have the same name, as used in the Docker run command that will be used to start syslog writer. (in our case, ca-bundle.crt). For CentOS and RHEL the CA certificates file name is ca-bundle.crt. For Ubuntu, rename the CA certificates file copied in data folder to ca-bundle.crt or edit the docker run command mentioned ahead to use the name ca-certificates.crt.

3. Running the Syslog Writer

Note: Successful update of the CA bundle with Root CA certificate, as mentioned in previous steps, is a must, before proceeding further to start syslog writer.

Run the Syslog Writer after making sure that your present working directory is the IISP_syslog_writer folder, using an additional parameter as highlighted in the following Docker command

```
sudo docker run --name syslog-writer -it --log-driver json-file --log-opt max-size=10m --net=host -v `pwd`/data:/home/centrify-syslog-writer/data -e REQUESTS_CA_BUNDLE=/home/centrify-syslog-writer/data/ca-bundle.crt syslogwriter_image
```

Note that this path in above command: `./home/centrify-syslog-writer/data` is the path within the Docker container that is mapped from the `pwd`/data on your Linux machine via the `-v` option in the Docker run command.

**Automatic Restarting**

If you want the container and the Syslog Writer to start automatically, if the Docker daemon restarts for some reason (like machine restart), you can use the option `--restart=always` in the `run` command, as shown below. Note that the container and the Syslog Writer will not automatically restart if the container was manually stopped.
Checking Execution Logs

You can see the current console logs of the syslog writer container by using the `logs` command in a Quickstart terminal:

```
sudo docker logs -f syslog-writer
```

All execution logs are saved in `IISP_syslog_writer/data/logs/` folder on the host machine.

Re-Running Syslog Writer After Cleanup

If there was a failure during the initial interactive configuration, or to rerun or configure the Syslog Writer with a fresh account:

1. Delete the earlier container:

   ```
sudo docker rm -f syslog-writer
```

2. Remove the local conf file in the `IISP_syslog_writer/data` folder:

   ```
sudo rm data/config.ini
```

3. Run the container again using same run command from the `IISP_syslog_writer` folder:

   ```
sudo docker run --name syslog-writer -it --log-driver json-file --log-opt max-size=10m --net=host --restart=always -v `pwd`/data:/home/centrify-syslog-writer/data syslogwriter_image
```
Restarting a Stopped Syslog Writer Container

Use this command to start the container/syslog writer, if the container stops (due to the Docker daemon or machine restart, etc.):

```
sudo docker start syslog-writer
```

When the syslog writer restarts, it fetches events beyond the last event fetch date, which is internally saved on the host machine, during previous run.

Stopping Syslog Writer Container

To stop the container/Syslog Writer:

```
sudo docker stop syslog-writer
```

Important Parameters in data/config.ini

- **rollback** (in hours, default value: 24)
  
  Only applies when syslog writer is started for first time. It is the number of hours before the current time for the syslog writer to start fetching events. By default, it will fetch for 24 hours (one day) before the current time in the UTC). If required, you can configure this in config.ini by creating a copy from the config.ini.default file, before firing the Docker run command.

- **batch_size** (in minutes, default value: 10)
  
  The number of minutes to fetch data for when the time range is large. By default, it fetches data in batches of 10 minutes. When there is a larger time range such as 24 hours for the first time run, it fetches data in batches of 10-minute sizes.

- **frequency** (in minutes, default value: 5)
  
  The frequency (in minutes) for running the Syslog Writer application. By default, it runs every five minutes. During runtime, a change to this parameter will be reflected after the pending job run is triggered.

- **debug** (under APP_LOGGER. This feature is disabled by default)
  
  This is for enabling debug level logs. In case of a problem with the behavior of syslog writer, please set the value of this property to yes (debug = yes) and reproduce the problem. You can then share an archive of the data/logs folder with Centrify support for troubleshooting.
Important note about data load

For a data load of 50 events per second and a rollback of 24 hours, it will take approximately 6.5 hours to catch up with current events after starting Syslog Writer. To avoid this delay, you might want to consider using a smaller rollback value.

Splunk IISP Add-on

The Splunk IISP Add-on is responsible for data onboarding and parsing IISP logs into Splunk events. These parsed events can be used for adhoc queries or to create visualizations. This add-on co-exists with other Splunk add-ons without conflicts.

Apart from data onboarding and parsing, the Splunk IISP Add-on takes care of the following:

- **Timestamp correction**: The timestamp in Syslog is the time when logging happened and not the actual time when the event occurred, so the timestamp of the IISP event in Splunk is corrected by using the WhenOccurred field in the event payload.

- **Custom sourcetype assignment**: A new sourcetype called `centrify_IISP_syslog` is assigned to IISP events. This ensures that IISP events and other Syslog messages are not touched unintentionally.

- **Applying Centrify headers**: Headers such as `product`, `category`, and `eventname` present in the payload are assigned to IISP events in Splunk.

- **CIM compliance**: The add-on maps IISP Authentication events to the Authentication model of CIM.

Setting Up Splunk Universal Forwarder

In a distributed Splunk environment, the Splunk Universal Forwarder must be set up on the machine with the Syslog server so that the IISP events in `syslog` get forwarded to the Indexer.

To configure Splunk Universal Forwarder for a distributed setup:

1. In a terminal, navigate to the path of Splunk Universal Forwarder:

   ```
   cd splunkuniversalforwader/bin
   ```

2. Add the Forward server in the `bin` folder, using the IP address of the Splunk Indexer as the `<ipaddress>` and the Receiver port configured on the Splunk Indexer as the `<port>` (usually 9997):

   ```
   ./splunk add forward-server <ipaddress>:<port>
   ```
3. Add syslog to the monitored files list:

```
./splunk add monitor /var/log/messages
```

### Installing the Splunk Add-on

The Splunk Add-on must be installed on the indexer, and on the search head.

To install the Splunk Add-on from the Splunk Web UI:

Go to **Apps > Browse**, Select **Centrify Identity Platform Add-on for Splunk**.

### Configuring Data Input

To configure data input:

**In a distributed Splunk environment with a Forward Server:**

1. Open Splunk Enterprise in a browser.
2. Go to **Settings > Forwarding and receiving > Configure Receiving > Add New**.
3. In the **Listen to Port** text box, enter **9997**.
4. Click **Save** to send messages from the Forward Server to port 9997.

**In a stand-alone Splunk environment with a local syslog:**

1. Go to **Settings > Data Inputs > Files and Directories**.
2. Enable **/var/log/messages** (this is disabled by default).
Searching for IISP Events

To search for IISP events, enter this command:

```
Search sourcetype = "centrify_IISP_syslog"
```