Centrify Identity Services Platform Events and ArcSight CEF Guide

September 2018

Centrify Corporation

Abstract

This guide is written for customers who use the Centrify Identity Services Platform (CISP) API for retrieving events and the ArcSight Common Event Format (CEF) to create ArcSight CEF-CISP events.

Legal Notice

This document and the software described in this document are furnished under and are subject to the terms of a license agreement or a non-disclosure agreement. Except as expressly set forth in such license agreement or non-disclosure agreement, Centrify Corporation provides this document and the software described in this document "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Some states do not allow disclaimers of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of Centrify Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non- disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of Centrify Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. Centrify Corporation may make improvements in or changes to the software described in this document at any time.

Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202 -4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the licenseagreement.

Centrify, DirectControl, DirectAuthorize, DirectAudit, DirectSecure, and DirectControl Express are registered trademarks and Centrify User Suite, Centrify Infrastructure Services, Centrify for Mobile, Centrify for SaaS, Centrify for Mac, DirectManage, Centrify Suite, Centrify Express, DirectManage Express, Centrify Identity Platform, Centrify Identity Service, and Centrify Privilege Service are trademarks of Centrify Corporation in the United States and other countries. Microsoft, Active Directory, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Centrify software is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,442,962 and 9,378,391.

The names of any other companies and products mentioned in this document may be the trademarks or registered trademarks of their respective owners. Unless otherwise noted, all of the names used as examples of companies, organizations, domain names, people and events herein are fictitious. No association with any real company, organization, domain name, person, or event is intended or should be inferred.

Contents

Introduction **4**

Overview of the Steps for Accessing CISP Events **4**

Prerequisite for Accessing CISP Events **4**

Setting up the SIEM User and the OAuth App on the Tenant 5

Generating a Basic Authorization Token **11**

Example 12

Sample Output 12

Fetching the OAuth Access Token Using the oauth2/token API **12**

Sample curl Command 12

Sample Output 13

Fetching CISP Using the Redrock/query API **13**

Sample curl Commands 13

Parsing the Response Received from Redrock/query 15

References 15

ArcSight CEF Format **15**

Using CEF Without Syslog 16

Sample Python Functions for CEF Creation 16

Using the Functions to Demonstrate Sample Usage 16

CEF Mapping of CISP Events 17

CEF Header 17

CISP ArcSight CEF Extension 18

Common Properties in CISP Events 18

Event-Specific Properties in CISP 19

EventType=Cloud.Core.MfaSummary 19

EventType=Cloud.Saas.Application.AppLaunch 20

EventType=Cloud.Saas.Application.GatewayAppLaunch 21

EventType=Cloud.Saas.Application.SelfServiceAppLaunch 21

EventType=Cloud.Server.ManualAccount.SessionStart 23

EventType= Cloud.Server.LocalAccount.SessionStart 23

EventType=Cloud.Server.LocalAccount.PasswordExport 24

EventType= Cloud.Server.DomainAccount.PasswordExport 24

EventType=Cloud.Core.Server.CpsTileLaunch 25

EventType=Cloud.Core.AdaptiveMfa.RiskAnalysis 25

Alternate Approach for Creating the Common Extension Format (CEF) 25

Introduction

The *Centrify Identity Services Platform Events and ArcSight CEF Guide* is written to provide detailed instructions for accessing events from the Centrify Identity Services Platform (CISP) using REST APIs. The guide also presents instructions for creating ArcSight Common Event Format (CEF) CISP events.

Overview of the Steps for Accessing CISP Events

The general steps that you perform to access CISP events are as follows:

1. As a prerequisite to accessing CISP events, configure the tenant for OAuth access to create:
   - SIEM user
   - OAuth app
   - SIEM scope for accessing Redrock and query
2. Generate the basic authorization token.
3. Fetch the OAuth access token using the oauth2/token API.
4. Fetch the CISP events using the Redrock/query API.
5. Parse the response that was received from the Redrock/query API.

Prerequisite for Accessing CISP Events

The first task that you must perform before accessing CISP events is to configure the OAuth tenant. For detailed steps, see Setting up the SIEM User and the OAuth App on the Tenant.

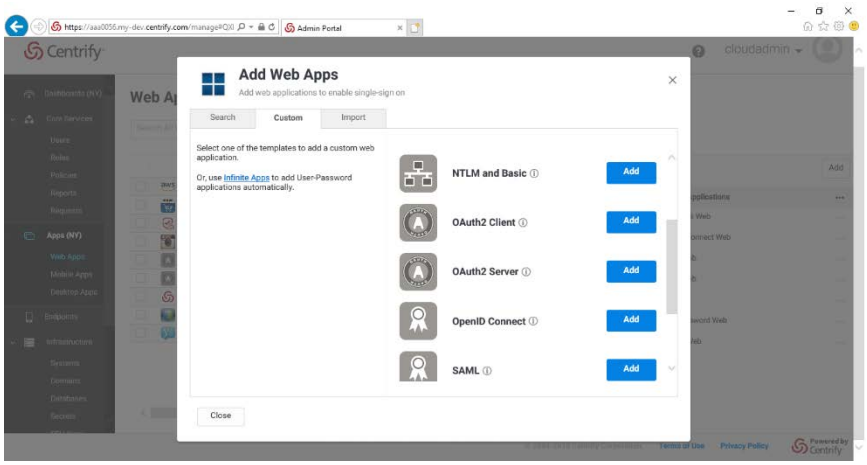
After you complete the configuration, you will have created the following:

- SIEM user
- OAuth app
- SIEM scope for accessing Redrock and query

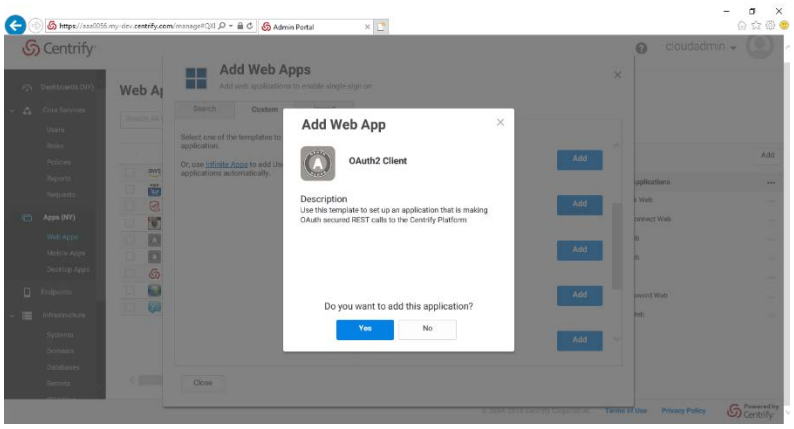
Setting up the SIEM User and the OAuth App on the Tenant

Follow these steps:

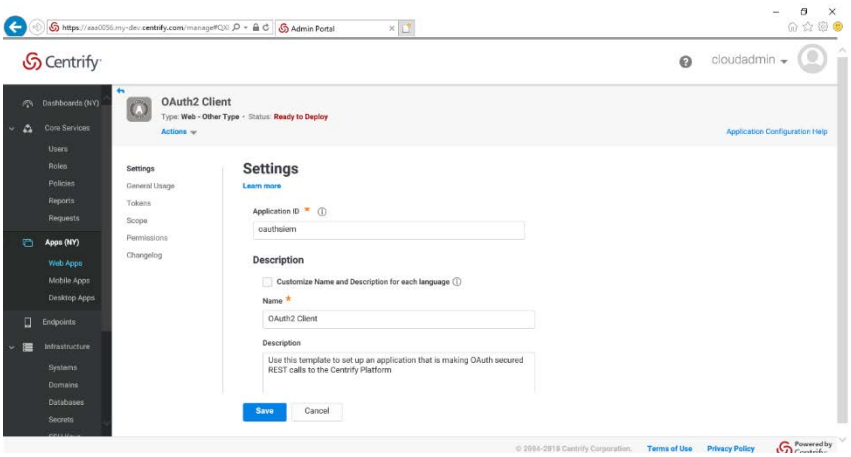
1. On the Centrify Admin Portal, click the **Apps** tab.
2. On the Add Web Apps page, click the **Custom** tab.
3. Locate the OAuth2 Client and click **Add**.



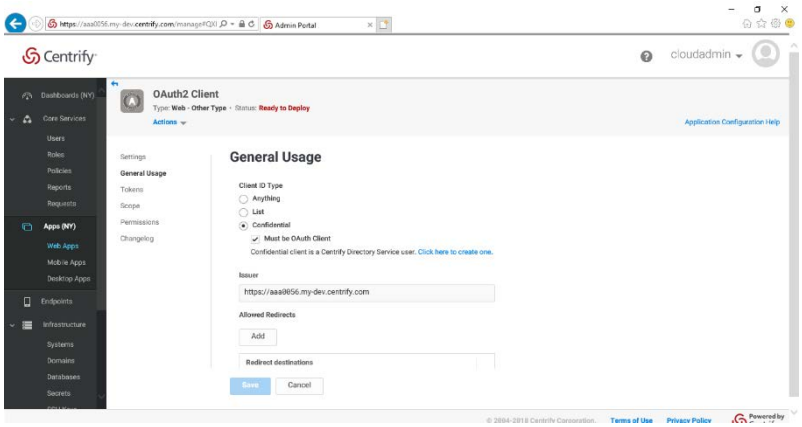
4. When prompted to add the Web App, OAuth2 Client, click **Yes**.



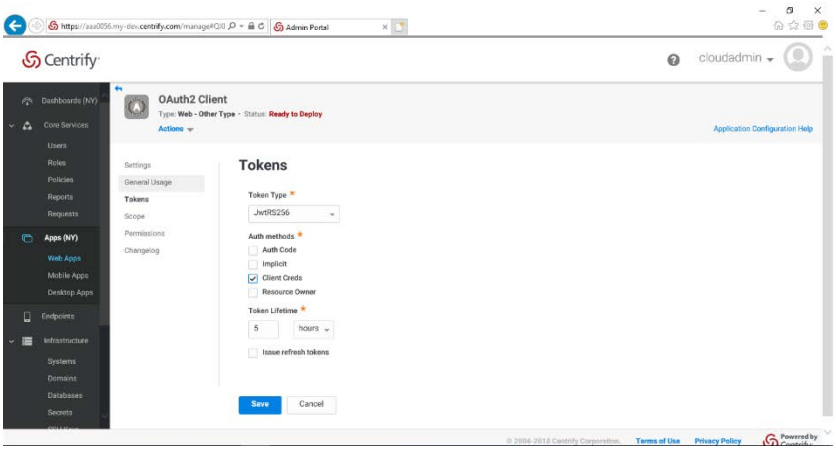
5. On the OAuth2 Client Description tab, for the Application Name, enter **oauthsiem**.



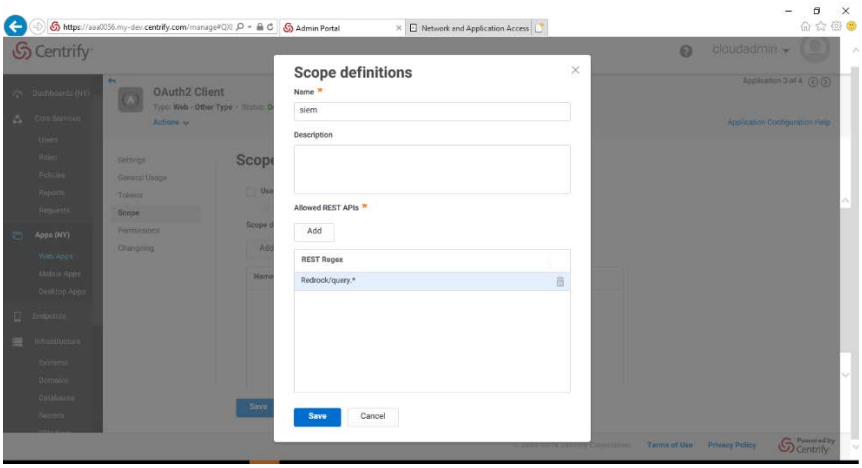
6. On the General Usage tab, leave the defaults as shown.



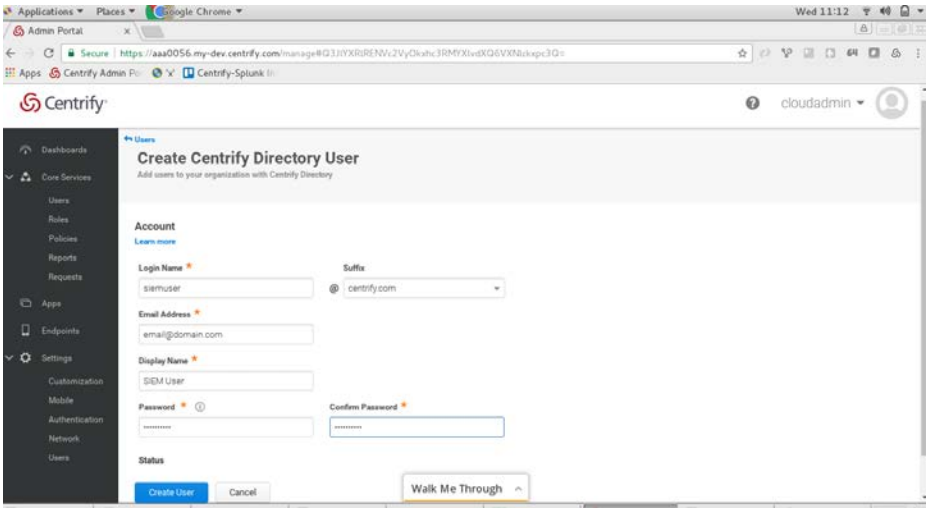
7. On the Tokens tab, for Auth methods, check **Client Creds**.



8. On the Scope tab, under Scope definitions, click **Add** to add a new scope.
9. On the Scope definitions dialog:
    a. In the Name field, enter **siem**.
    b. Under Allowed REST APIs, click **Add,** enter **Redrock/query.*,** and click **Save.**



10. On the Centrify Admin Portal, click the **Users** tab.
11. On the Create Centrify Directory User page:
    a. For the Login Name, enter: **siemuser**.
    b. For the Suffix, enter **centrify.com** (or leave as is).
    c. For the Password and Confirm Password, enter the password of your choice.

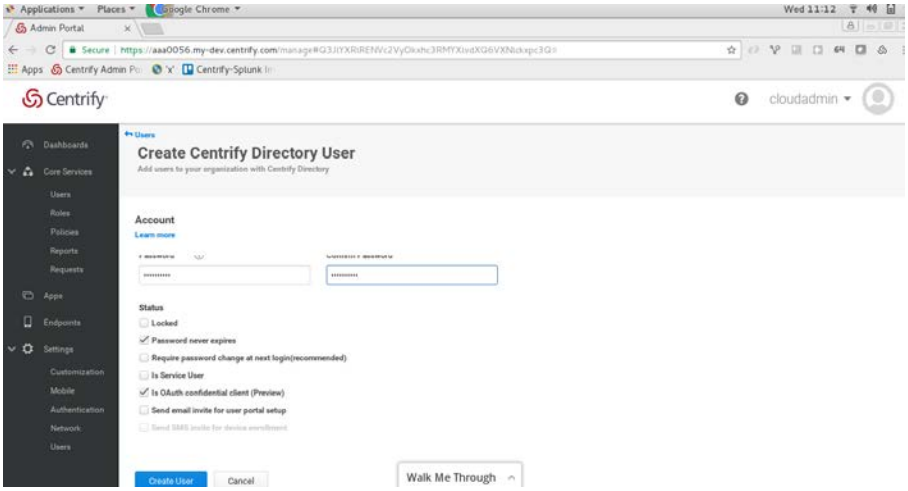


12. Under Status:

    a. Check **Password never expires**.

    b. Check **Is OAuth confidential client (Preview)**.

    c. Click the **Create User** button.

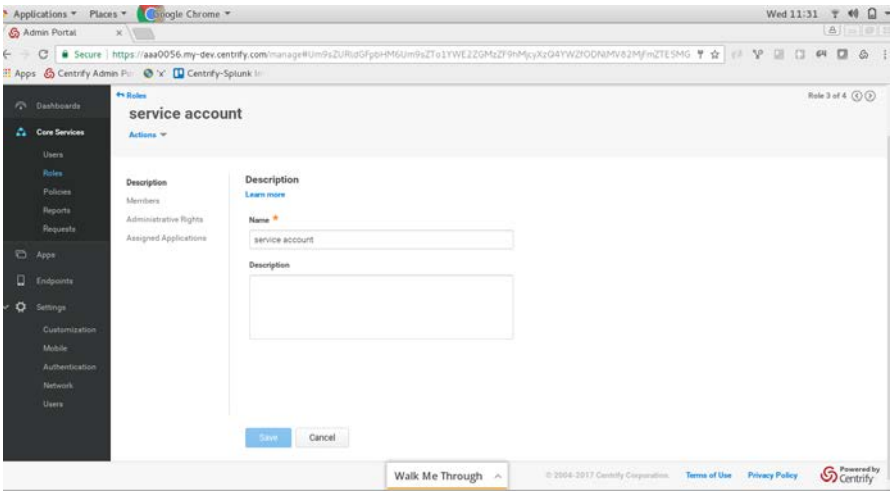


13. On the Centrify Admin Portal, click the **Roles** tab.

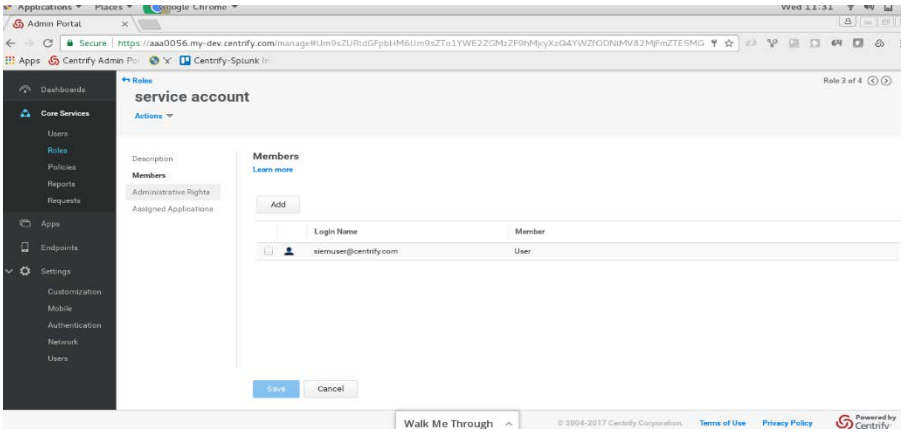
14. On the service account page:

    a. For the Name, enter: **service account**.

    This entry serves as the role name.

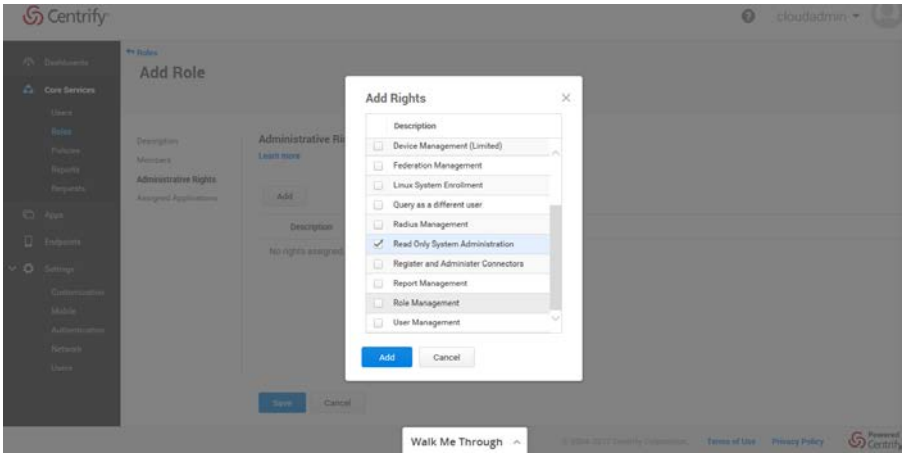


b. For the Members, add (check) the **siemuser** that you created earlier.

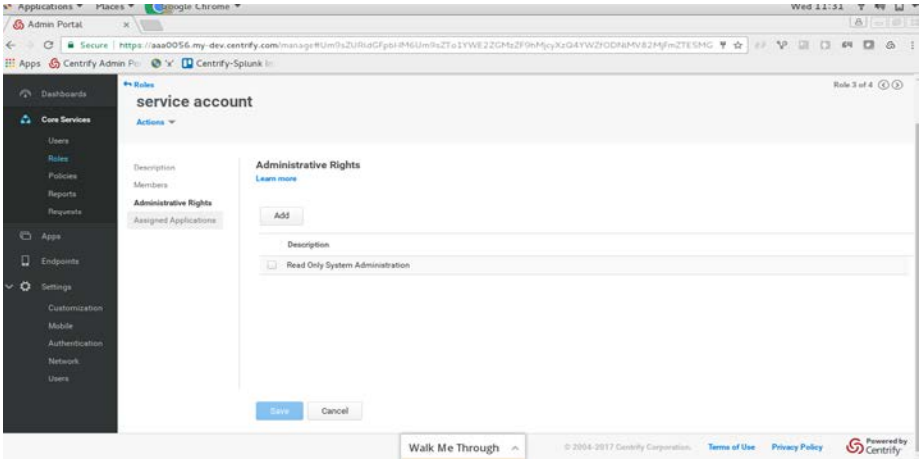


c. Click **Save**.

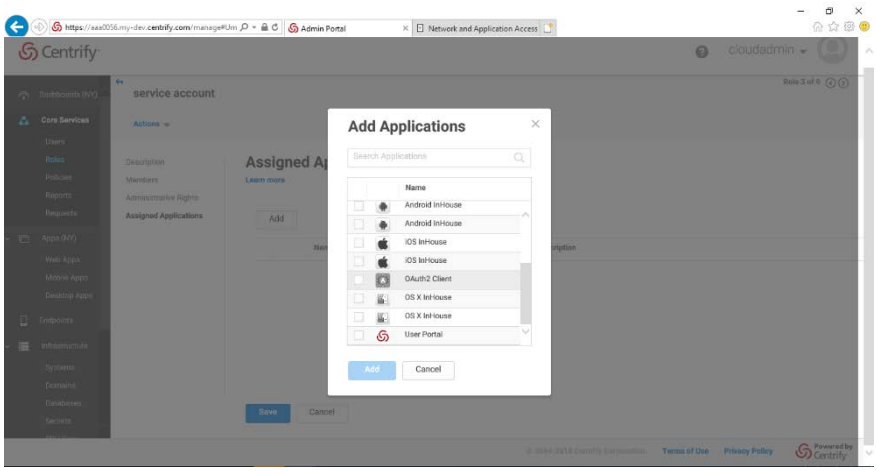
15. For the Administrative Rights, in the Add Rights list, check **Read Only System Administrator** and click **Add**.



16. Check the **Read Only System Administrator** and click **Save**.



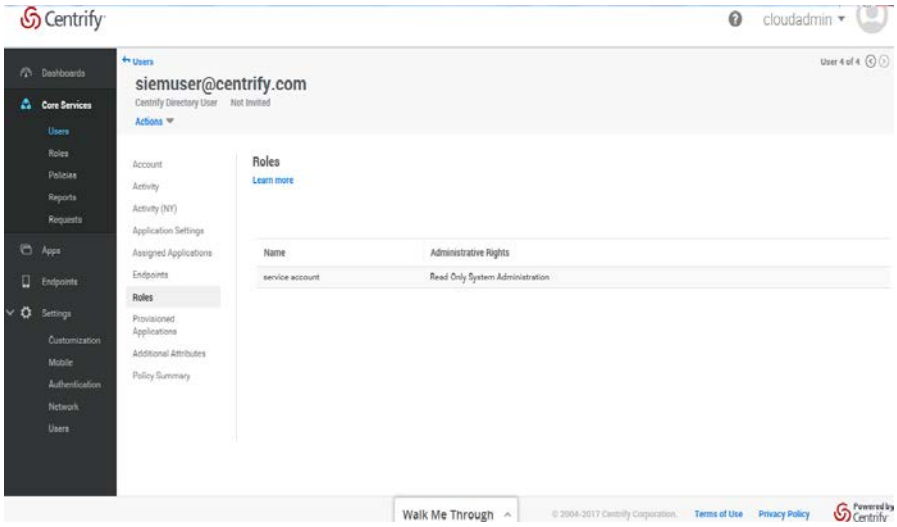
17. For the Assigned Applications, in Add Applications list, check the **OAuth2 Client** app**.**



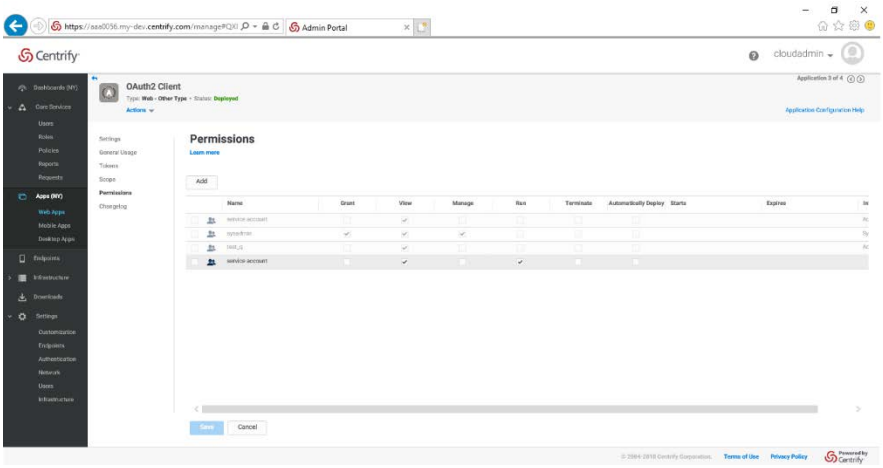
18. Click **Add**.

19. Perform final checks to make sure that:

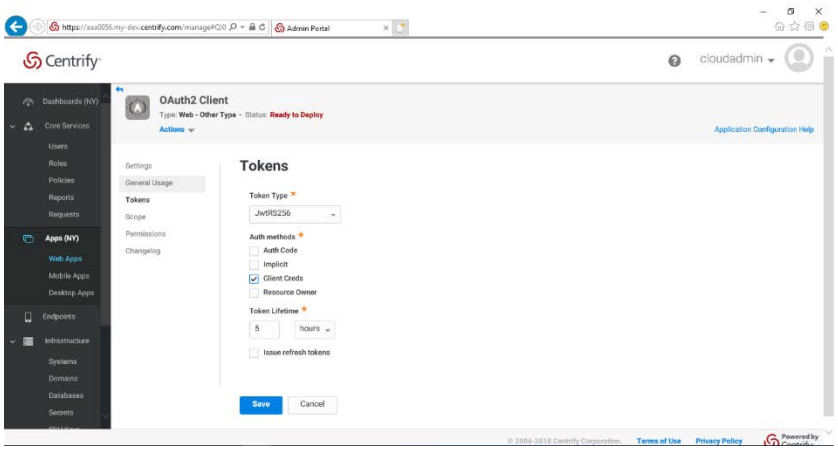
    - On the **Users** tab, the **siemuser** is shown and the Roles section lists the **Name** of the role as **service account**.



- On the **Apps** tab, In **Permissions** section of the **OAuth2 Client** app, the permissions for **service account** role looks as below.



- On the **Apps** tab, the **Tokens** section shows under **Auth methods** that **Client Creds** is checked.



Generating a Basic Authorization Token

To generate a basic authorization token, use the following command:

```
echo -ne "<siem_user>:<password>" | base64
```

Example

Review the following example:

```
echo -ne siemuser@centrify.com:Pass@2k17" | base64
```

Sample Output

The sample output looks like this:

c2llbXVzZXJAY2VudHJpZnkuY29tOkxlZW5hQDIwMTc

Fetching the OAuth Access Token Using the oauth2/token API

Use the curl command and the basic authorization token extracted in the previous step:

```
curl -H "Authorization: Basic <basic_auth_token>" -d
"grant_type=client_credentials&scope=<siem_scope>"
https://<tenant>/oauth2/token/<oauth_app_id>
```

Sample curl Command

Review the sample curl command:

```
curl -H "Authorization: Basic
c2llbXVzZXJAY2VudHJpZnkuY29tOkalZW5hQDIwMTc" -d
"grant_type=client_credentials&scope=siem" https://aaa0056.my-
dev.centrify.com/oauth2/token/oauthsiem
```

Sample Output

The sample output looks like this:

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Ijk5QzA4QjQzMjk4N0ZDQjRCN0E5MTEwMTdDMTI3QzA4NTZCMjAxQzkiLCJ4NXQiOiJtY0NMUXltSF9MUzNxUkVCZkJKOENGYXlBY2siLCJhcHBfaWQiOiJvYXV0aHNpZW0ifQ.eyJpYXQiOjE1MjE2OTkzNzgsInVuaXF1ZV9uYW1lIjoic2llbXVzZXJAY2VudHJpZnkuY29tIiwiZXhwIjoxNTIxNzE3Mzc4LCJzdWIiOiI0NDZjOTc5Ni1lOWE4LTRiMDgtYmJkZi02ZGZlNTJiOGRkOTIiLCJzY29wZSI6InNpZW0ifQ.e5oE58Cxv0qkIb1Z-nCXyhbIxcL_6Bs3znVVyBG6aFb6oHSlb_y5pPnWaLfQdmfnx6hyHtM0GGRoK6HTVJulSbrCFzqHKBHoW38YPh5M7IzTJflJ-8k0ip9we3ElWm2QiOcbR8AmULYaDR8OnvpIVtmBJ2ZBJng9oFippwoNtBi2gYFjjJsGtRClpqvlHrTytPAqe3SvM0whm8yfbq8YhIapcdk_mfJl2YEPX_pyl-Kxzyz9_nHw-_jm0LXzMazvPiAz-sFCrc8ngtzQZgvDe1wUnPqqEiB0G2Hg2-NCPYi9hcR8OUyeKD4erkgyXRq1KvvrS7G9iLHT1VrLSu0o2g

Fetching CISP Using the Redrock/query API

Use the curl command and the OAuth access token extracted in the previous step:

```
curl  -H "Authorization: Bearer <oauth_access_token>" -d
'{"Script":"<query>"}' https://<tenant>/Redrock/query
```

Sample curl Commands

This sample curl command fetches events for the last 24 hours:


```
curl  -H "Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Ijk5QzA4QjQzMjk4N0ZDQjRCN0E5MTEwMTdDMTI3QzA4NTZCMjAxQzkiLCJ4NXQiOiJtY0NMUXltSF9MUzNxUkVCZkJKOENGYXlBY2siLCJhcHBfaWQiOiJvYXV0aHNpZW0ifQ.eyJpYXQiOjE1MjE2OTkzNzgsInVuaXF1ZV9uYW1lIjoic2llbXVzZXJAY2VudHJpZnkuY29tIiwiZXhwIjoxNTIxNzE3Mzc4LCJzdWIiOiI0NDZjOTc5Ni1lOWE4LTRiMDgtYmJkZi02ZGZlNTJiOGRkOTIiLCJzY29wZSI6InNpZW0ifQ.e5oE58Cxv0qkIb1Z-nCXyhbIxcL_6Bs3znVVyBG6aFb6oHSlb_y5pPnWaLfQdmfnx6hyHtM0GGRoK6HTVJulSbrCFzqHKBHoW38YPh5M7IzTJflJ-8k0ip9we3ElWm2QiOcbR8AmULYaDR8OnvpIVtmBJ2ZBJng9oFippwoNtBi2gYFjjJsGtRClpqvlHrTytPAqe3SvM0whm8yfbq8YhIapcdk_mfJl2YEPX_pyl-Kxzyz9_nHw-_jm0LXzMazvPiAz-sFCrc8ngtzQZgvDe1wUnPqqEiB0G2Hg2-NCPYi9hcR8OUyeKD4erkgyXRq1KvvrS7G9iLHT1VrLSu0o2g" -d
```



```
'{"Script":"Select * from Event where WhenOccurred >
datefunc('\''now'\'', '\''-1'\'')"}' https://aaa0056.my-
dev.centrify.com/Redrock/query
```

This sample curl command fetches events between two timestamps:

```
curl  -H "Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Ijk5QzA4QjQzMjk4N0ZDQ
jRCN0E5MTEwMTdDMTI3QzA4NTZCMjAxQzkiLCJ4NXQiOiJtY0NMUXltSF9MUzNxUk
VCZkJKOENGYXlBY2siLCJhcHBfaWQiOiJvYXV0aHNpZW0ifQ.eyJpYXQiOjE1MjE2
OTkzNzgsInVuaXF1ZV9uYW1lIjoic2llbXVzZXJAY2VudHJpZnkuY29tIiwiZXhwI
joxNTIxNzE3Mzc4LCJzdWIiOiI0NDZjOTc5Ni1lOWE4LTRiMDgtYmJkZi02ZGZlNT
JiOGRkOTIiLCJzY29wZSI6InNpZW0ifQ.e5oE58Cxv0qkIb1Z-
nCXyhbIxcL_6Bs3znVVyBG6aFb6oHSlb_y5pPnWaLfQdmfnx6hyHtM0GGRoK6HTVJ
ulSbrCFzqHKBHoW38YPh5M7IzTJflJ-
8k0ip9we3ElWm2QiOcbR8AmULYaDR8OnvpIVtmBJ2ZBJng9oFippwoNtBi2gYFjjJ
sGtRClpqvlHrTytPAqe3SvM0whm8yfbq8YhIapcdk_mfJl2YEPX_pyl-
Kxzyz9_nHw-_jm0LXzMazvPiAz-sFCrc8ngtzQZgvDe1wUnPqqEiB0G2Hg2-
NCPYi9hcR8OUyeKD4erkgyXRq1KvvrS7G9iLHT1VrLSu0o2g" -d
'{"Script":"Select * from Event where WhenOccurred >= '\''2018-
03-15T11:33:59.273000Z'\'' and  WhenOccurred < '\''2018-03-
21T11:33:59.273000Z'\''"}' https://aaa0056.my-
dev.centrify.com/Redrock/query
```


Parsing the Response Received from Redrock/query

Refer to the following sample Python code to extract events data from a response:

```
import json
response_json = json.loads(response.text)
events = response_json['Result']['Results']
headers = []
for column in response_json['Result']['Columns']:
    headers.append(column['Name'])

for idx, event in enumerate(events):
    print('\n Row Number:' + str(idx))
    for header in headers:
        if event['Row'][header] is not None:

            print(header + "=" + str(event['Row'][header]))
```

References

For additional information, see:

https://developer.centrify.com/v1.0/docs/use-queries

https://docs.centrify.com/en/centrify/adminref/index.html#page/cloudhelp/cldAdm-DateFuncSyntax.html

ArcSight CEF Format

The Common Event Format (CEF) standard format, developed by ArcSight, enables vendors and their customers to quickly integrate their product information into ArcSight ESM.

CEF defines a syntax for log records comprised of a standard header and a variable extension, formatted as key-value pairs.

When syslog is used as a transport mechanism, CEF uses the following format, comprised of a syslog prefix, a header, and an extension:

```
Jan 18 11:07:53 host CEF:Version|Device Vendor|Device
Product|Device Version|Device Event Class
ID|Name|Severity|[Extension]
```

The following example illustrates a general CEF message using syslog transport:

```
Sep 19 08:26:10 host
CEF:0|Centrify|Centrify_Cloud|1.0|Cloud.core|Cloud.core.MfaSummar
y|5|src=10.0.0.1 dst=2.1.2.2 spt=1232
```

Using CEF Without Syslog

Syslog applies a syslog prefix to each message, no matter what device it arrives from, which contains the date and hostname:

```
Jan 18 11:07:53 host CEF:Version|…
```

However, if an event producer is unable to write syslog messages, it is still possible to write the events to a file. In this case, begin the message with the format shown below, and omit the syslog prefix:

```
CEF:Version|Device Vendor|Device Product|Device Version|Device
Event Class ID|Name|Severity|[Extension]
```

Sample Python Functions for CEF Creation

This section describes a set of sample Python functions for generating CEF-formatted CISP events.

There are three main functions in this package:

- `fetch_oauth_token()`
- `query_events()`
- `cef_generator()`

Using the Functions to Demonstrate Sample Usage

Prerequisite: Python 3.5 or above

Follow these steps:

1. Download the Python code from this location:

https://github.com/centrify/centrify-hparcsight-integration-sample/

2. Install pip packages in `requirement.txt`
3. Provide the values for `tenant`, `siem_username`, and `siem_password` in `config.ini`

4. Execute `sample_usage.py` to generate CEF-formatted CISP events for one hour:

   ```
   python3.5 sample_usage.py
   ```

The following example shows a CEF message for a Self-Service App Launch CIS Event:

```
CEF:0|Centrify|Centrify_Cloud|1.0|Cloud.Saas.Application|Cloud.Saas.Application.SelfServiceAppLaunch|5|dhost=AAA0056 duser=cloudadmin@persistent.com01 msg=User cloudadmin@persistent.com01 launched Instagram from 103.6.32.100 shost=103.6.32.100 src=103.6.32.100 rt=1525844566655 deviceProcessName=centrify-syslog-writer dvchost=dinesh-VirtualBox dtz=Africa/Abidjan requestContext=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36 Edge/15.15063 externalId=772a4a904e82da87.W00.0315.1aa20afe647f09c dpriv=WebRole destinationServiceName=CDS suid=c2c7bcc6-9560-44e0-8dff-5be221cd37ee cs1=Instagram cs1Label=applicationId cs2=Instagram cs2Label=applicationName cs3=Web cs3Label=applicationType cs4=103.6.32.100 cs4Label=clientIPAddress cs5=65f79bb1-4f91-4496-9991-d148da16cc3e cs5Label=internalSessionId cs6=0d10a24f4c57434198fb3ad4559cc48b cs6Label=azDeploymentId directoryServiceNameLocalized=Centrify Directory threadType=RestCall azRoleId=WebRole_IN_0 internalTrackingID=d3a0713b610146ca916155efca2be690 authMethod=UserPassword requestIsMobileDevice=False directoryServiceUuid=09B9A9B0-6CE8-465F-AB03-65766D33B05E requestDeviceOS=Windows level=Info
```

You can customize the usage or the APIs per your application needs.

NOTE: CEF has a predefined set of keys.

CEF Mapping of CISP Events

This section provides detailed information about how the CEF fields have been mapped from the CISP event fields in the Python application described above.

CEF Header

Header Field	CISP Event Field
Version	‘0’
Device Vendor	‘Centrify’

Header Field	CISP Event Field
Device Product	'Centrify_Cloud'
Device Version	'1.0'
Device Event Class ID	Variable — depends on the event. For example: 'Cloud.Saas.Application'
Name	Variable — depends on the event. For example: Cloud.Saas.Application.SelfServiceAppLaunch'
Severity	Variable — depends on the Level field in event. For example: '5' for Info, '10' for Error.

CISP ArcSight CEF Extension

The CEF Extension contains a collection of key-value pairs. The keys are predefined and are referred to as the *ArcSight Extension Dictionary*. (CEF Fields)

Common Properties in CISP Events

This section lists the CEF field mapping of CISP events, which are part of the CEF extension.

These properties are common to all events of the Centrify Identity Services Platform and Privilege Services.

ArcSight CEF Field	CISP Event Field
The common properties are those listed below in **bold**.	
Destination Host Name	Tenant
Destination User Name	NormalizedUser
Message	EventMessage
Source Host Name	RequestHostName
Source Address	FromIPAddress
Device Receipt Time	whenoccurred_epoch_ms (This is the event timestamp in UTC)
Device Process Name	'centrify-syslog-writer' (can be configured in cef_mapping.ini)
Device Host Name	Hostname of machine running the python app
Device Time Zone	'Africa/Abidjan' (Note: This time zone is chosen mainly to set UTC offset to 0)

ArcSight CEF Field	CISP Event Field
The keys in the common properties section below are added in the CEF message **only if** no event-specific CEF mapping is specified for an event in the mapping configuration file, which is enclosed with the Sample Python application for CEF creation.	
Device Custom String 1	AuthMethod
Device Custom String1 Label	'authMethod'
Device Custom String2	RequestIsMobileDevice
Device Custom String2 Label	'requestIsMobileDevice'
Device Custom String3	DirectoryServiceUuid
Device Custom String3 Label	'directoryServiceUuid'
Device Custom String4	RequestDeviceOS
Device Custom String4 Label	'requestDeviceOS'
Device Custom String5	Level
Device Custom String5 Label	Level

Event-Specific Properties in CISP

This section lists the event-specific properties mapped to ArcSight Fields. All events (whether they are listed below or not) will have the first nine common properties, identified in the table above, mapped in an ArcSight.CEF message. The message is generated when you use the Sample Python functions described earlier in this document.

Any CEF key appearing in event-specific mapping will override the CEF key mapping in the common properties section. For example, the Cloud.Server.ManualAccount.SessionStart event, Destination host (Dhost), and Destination User(duser) will be ‘ComputerName’ and ‘AccountName’, which will overwrite the common properties mapped for dhost and duser.

EventType=Cloud.Core.MfaSummary

ArcSight CEF Field	CISP Event Field
Reason	MfaReason
Outcome	MfaResult
RequestContext	RequestUserAgent
ExternalId	ID

ArcSight CEF Field	CISP Event Field
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Device Custom String 1	MfaInitiator
Device Custom String1 Label	'mfaInitiator'
Device Custom String2	FactorsLocalized
Device Custom String2 Label	'factorsLocalized'
Device Custom String3	ProfileName
Device Custom String3 Label	'profileName'
Device Custom String4	FailReason
Device Custom String4 Label	'failReason'
Device Custom String5	MfaUnlock
Device Custom String5 Label	'mfaUnlock'
Device Custom String6	ForgotPassword
Device Custom String6 Label	'forgotPassword'
Device Custom Number1	Factorcount
Device Custom Number1 Label	'factorCount'
Device Custom Number2	SecurityQuestionAnswerCount
Device Custom Number2 Label	'securityQuestionAnswercount'

NOTE: The remaining fields in an event that are not mapped to CEF keys will still be added in the CEF message with their CISP-event field keys. These custom non-CEF keys will not be available for reporting in ArcSight, but they can viewed as part of the raw event message.

EventType=Cloud.Saas.Application.AppLaunch

ArcSight CEF Field	CISP Event Field
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName

ArcSight CEF Field	CISP Event Field
Suid	UserGuid
Device Custom String 1	ApplicationID
Device Custom String1 Label	‘applicationId’
Device Custom String2	ApplicationName
Device Custom String2 Label	‘applicationName’
Device Custom String3	ApplicationType
Device Custom String3 Label	‘applicationType’
Device Custom String4	TemplateName
Device Custom String4 Label	‘templateName’
Device Custom String5	InternalSessionId
Device Custom String5 Label	‘internalSessionId’
Device Custom String6	AzDeploymentId
Device Custom String6 Label	azDeploymentId

EventType=Cloud.Saas.Application.GatewayAppLaunch

EventType=Cloud.Saas.Application.SelfServiceAppLaunch

ArcSight CEF Field	CISP Event Field
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Suid	UserGuid
Device Custom String 1	ApplicationID
Device Custom String1 Label	‘applicationId’
Device Custom String2	ApplicationName
Device Custom String2 Label	‘applicationName’
Device Custom String3	ApplicationType
Device Custom String3 Label	‘applicationType’
Device Custom String4	ClientIPAddress
Device Custom String4 Label	‘clientIPAddress’
Device Custom String5	InternalSessionId
Device Custom String5 Label	‘internalSessionId’

ArcSight CEF Field	CISP Event Field
Device Custom String6	AzDeploymentId
Device Custom String6 Label	azDeploymentId

EventType=Cloud.Server.ManualAccount.SessionStart

EventType= Cloud.Server.LocalAccount.SessionStart

ArcSight CEF Field	CISP Event Field
Src	FromIPAddress
Suser	NormalizedUser
Dhost	ComputerName
Duser	AccountName
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Suid	UserGuid
Device Custom String 1	UserType
Device Custom String1 Label	‘userType’
Device Custom String2	SessionType
Device Custom String2 Label	‘sessionType’
Device Custom String3	AuthorityName
Device Custom String3 Label	‘authorityName’
Device Custom String4	JumpType
Device Custom String4 Label	‘jumpType’
Device Custom String5	DirectoryServiceNameLocalized
Device Custom String5 Label	‘directoryServiceNameLocalized’
Device Custom String6	AuthoritySource
Device Custom String6 Label	‘authoritySource’

EventType=Cloud.Server.LocalAccount.PasswordExport

EventType= Cloud.Server.DomainAccount.PasswordExport

ArcSight CEF Field	CISP Event Field
Src	FromIPAddress
Suser	NormalizedUser
Dhost	ComputerName
Duser	AccountName
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Suid	UserGuid
Device Custom String 1	UserType
Device Custom String1 Label	‘userType’
Device Custom String2	AuthorityID
Device Custom String2 Label	‘authorityID’
Device Custom String3	AuthorityName
Device Custom String3 Label	‘authorityName’
Device Custom String4	AzRoleId
Device Custom String4 Label	‘azRoleId’
Device Custom String5	DirectoryServiceNameLocalized
Device Custom String5 Label	‘directoryServiceNameLocalized’
Device Custom String6	CheckedOut
Device Custom String6 Label	‘checkedOut’
Device Custom Date1	WhenDueBack
Device Custom Date1 Label	‘whenDueBack’

EventType=Cloud.Core.Server.CpsTileLaunch

ArcSight CEF Field	CISP Event Field
RequestContext	RequestUserAgent
ExternalId	ID
Dpriv	AzRoleName
DestinationServiceName	DirectoryServiceName
Suid	UserGuid
Device Custom String 1	UserType
Device Custom String1 Label	‘userType’
Device Custom String2	ApplicationType
Device Custom String2Label	‘applicationType’
Device Custom String3	ApplicationName
Device Custom String3Label	‘applicationName’
Device Custom String4	ApplicationID
Device Custom String4Label	‘applicationId’
Device Custom String5	DirectoryServiceNameLocalized
Device Custom String5Label	‘directoryServiceNameLocalized’
Device Custom String6	InternalTrackingID
Device Custom String6Label	‘internalTrackingID’

EventType=Cloud.Core.AdaptiveMfa.RiskAnalysis

Only Common properties.

Alternate Approach for Creating the Common Extension Format (CEF)

In case you are using the CISP REST APIs directly in your application and generating your own CISP syslog messages in a generic non-CEF format having key=value pairs separated by a delimiter, then ArcSight SmartConnector will need to be installed and configured to collect these CISP syslog.

These logs will need to be parsed into CEF format by creating ArcSight FlexConnector, to enable the CISP events to be usable for SIEM in ArcSight. The only downside to using a FlexConnector is that ArcSight does not officially certify it.