Cloud-Based API Request Signing

Note

The following applies to the xConnect and Asset Management APIs

https://assetmgmt-api.senecaxconnect.com

https://api.senecaxconnect.com

All HTTP API requests into the xConnect Platform must be signed by the client application and validated by the Asse
Platform to ensure they come from authenticated clients. There are 2 important pieces of information - **apiKey** and **s**
from the cloud portal and embedded into the client application for signing requests. It's recommended that these ke
client side.

For example in this document, we use the following keys

apiKey

`5501f50fdc62aee5d04dbd6a58b68b781ee2aaade8ad1eb24b1e4e77cb282ae2`

secretKey

`ARAzUzRzekFwRTNACBQYUx89LlZyImhKFVloHUVMDw8EGRxxSCckFgdFPysAAWJCLDgMdkstZzw3GGVqNHxXcno5Iz54LRBSK`

The client application needs to process the API request details using 5 steps explained below. The input parameters
the URL, HTTP method, query string, JSON payload, HTTP headers, the security keys, etc. The computation outcome
included in the HTTP header.

Step 1: Create a Canonical Request for signing

The first step is to build a string representation of your request in a pre-defined format.

Java

```
canonicalRequest =
    httpRequestMethod + '\n'  +
    canonicalURI + '\n' +
    canonicalQueryString + '\n' +
    hexEncode(hash(payload));
```

Python

```
#  url_query_params may be blank based on the endpoint being used. Example:
#  url_query_params = '_page=' + urllib.parse.quote(PAGE_TO_RETRIEVE) + '&' + '_size=' +
urllib.parse.quote(NUM_ITEMS_ON_EACH_PAGE)
if url_query_params != '':
    canonicalRequest = httpRequestMethod + '\n' + canonicalURI + '\n' + query_params_to_encrypt
else:
    canonicalRequest = httpRequestMethod + '\n' + canonicalURI + '\n' + hexEncode
```

httpRequestMethod

Put one of the following methods in a line by itself - GET, POST, PUT, PATCH

POST

canonicalURI

The canonical URI is the URI-encoded version of the absolute path component of the URI, which is everything in the U
question mark character ("?") that begins the query string parameters (if any).

```
/api/v1/kronos/gateways
```

canonicalQueryString

- Create each line containing a name/value pair in the format name=value. Name field is converted to lowercase a
- Sort all the lines in ascending order

Example QueryString:

lastName=Doe&firstName=Jane&Age=30

```
age=30
firstname=Jane
lastname=Doe
```

hexEncode(hash(payload))

1. The payload (if any) is most likely in JSON format. If there's no payload, use an empty string
2. Hash the payload content with SHA-256

3. Hex-encode the hash value
4. Convert the hex-encoded value to lowercase

Example of an empty payload:

```
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Combining all of the results above produces the following string:

```
POST
/api/v1/kronos/gateways
age=30
firstname=Jane
lastname=Doe
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Finally, hash this computed canonical request, hex-encode and lowercase it. This is the result of step 1.

```
5a2d3589ffb15fab720069fbd26fd8e8311a1c7047e5899608faff450df6d7dc
```

Java

```
stringtoSign =
     hashedCanonicalRequest + '\n' +
     apiKey + '\n' +
     requestTimestamp + '\n' +
     apiversion;
```

hashedCanonicalRequest

This is the result of step 1

`5a2d3589ffb15fab720069fbd26fd8e8311a1c7047e5899608faff450df6d7dc`

apiKey

Provided apiKey from Konexios Portal

`5501f50fdc62aee5d04dbd6a58b68b781ee2aaade8ad1eb24b1e4e77cb282ae2`

requestTimestamp

Current UTC time in ISO-8601 format - YYYY-MM-DDThh🇲🇲s.sssZ

`2016-04-12T14:28:36.218Z`

apiVersion

Version is 1

`1`

Combining all of the results above produces the following string. This is the result of step 2.

```
5a2d3589ffb15fab720069fbd26fd8e8311a1c7047e5899608faff450df6d7dc
5501f50fdc62aee5d04dbd6a58b68b781ee2aaade8ad1eb24b1e4e77cb282ae2
2016-04-12T14:28:36.218Z
1
```

Step 3: Create the signing key

In this step we're going to use the HMAC-SHA256 algorithm a few times to generate the signing key. In the pseudoc hmacSha256(key, data). Note the order of the input parameters because if you're using some third-party library for th input parameters might be reversed. The output of this function is in binary, hence note the hex-encode function bein

Java

```
signingKey = secretKey;
signingKey = hexEncode(hmacSha256(apiKey, signingKey));
signingKey = hexEncode(hmacSha256(requestTimestamp, signingKey));
signingKey = hexEncode(hmacSha256(apiVersion, signingKey));
```

secretKey

Provided secretKey from Kronos portal

apiKey

Provided apiKey from Arrow Connect portal

requestTimestamp

Must be the same value as in step 2

apiVersion

Must be the same value as in step 2

For the example in this guide, here's the output of this step. This is the result of step 3

```
signingKey =
ARAzUzRzekFwRTNACBQYUx89LlZyImhKFVloHUVMDw8EGRxxSCckFgdFPysAAWJCLDgMdkstZzw3GGVqNHxXcno5Iz54LRBS
signingKey = 3c6e85f6a719e5b8bd77fde0cbdbe19d947f38451afbc8ef6e49a083d86a9c54
signingKey = 3223bf9bc2d2180046cc40c2e1ed6f9d08261a6c4a394b23c5311e83633a8ef7
signingKey = d0d1518fc5290c22f1444d46d9c08dd03cc33c6fdad8bbcd57be65b1e2b0b493
```

Step 4: Create the final signature

The signature is computed by signing the result from step 2 and step 3

Java

```
signature = hexEncode(hmacSha256(signingKey, stringToSign))
```

signingKey

Result of step 3

stringToSign

Result of step 2

Result of step 4 is below

`28c3ab6cc82294b61e9b2855b428090e474fd1e066c4da63f9715bd2204df553`

Step 5: Add required headers to request and submit

x-arrow-apikey	Provided apiKey
x-arrow-date	requestTimestamp from step 2
x-arrow-version	apiVersion from step 2
x-arrow-signature	final signature from step 4

```
x-arrow-apikey: 5501f50fdc62aee5d04dbd6a58b68b781ee2aaade8ad1eb24b1e4e77cb282ae2
x-arrow-date: 2016-04-12T14:28:36.218Z
x-arrow-version: 1
x-arrow-signature: 28c3ab6cc82294b61e9b2855b428090e474fd1e066c4da63f9715bd2204df553
```

Complete API Request Example

Python

```
# -*- coding: utf-8 -*-
import datetime
import hashlib
import hmac
import json

import requests
import urllib.parse

dir(hashlib)

# Base URL and api method URI here:
base_url = 'https://assetmgmt-api.senecaxconnect.com'
api_method_uri = '/api/v1/kronos/telemetries/devices/{deviceHid}/latest'

# Put API key and Secret Keys (RAW) here:
# Contact support@senecaxconnect.com for this information.
application_hid = "your_applicationhid_goes_here"
apiKey = 'api_key_goes_here'
secretKey = 'secret_key_goes_here'
device_hid = 'device_hid_goes_here'

api_method_uri = api_method_uri.format(deviceHid=device_hid)

# ************************STEP-1 Create a Canonical Request for signing***********
httpRequestMethod = 'GET'

# Telemetry pull by application hid
canonicalURI = api_method_uri

# Example on how to handle the from/to timestamps for API calls
# from_timestamp_raw = datetime.datetime.now() - datetime.timedelta(hours=HOURS_TO_RETRIEVE)
# to_timestamp_raw = datetime.datetime.now()
# from_timestamp = from_timestamp_raw.strftime('%Y-%m-%dT%H:%M:%S.%fZ')
# to_timestamp = to_timestamp_raw.strftime('%Y-%m-%dT%H:%M:%S.%fZ')

# Paging variables go here.
PAGE_TO_RETRIEVE = '0'
NUM_ITEMS_ON_EACH_PAGE = '150'

# Example: URL Query params with timestamp ranges
# url_query_params = '_page=' + urllib.parse.quote(PAGE_TO_RETRIEVE) + '&' + '_size=' +
urllib.parse.quote(NUM_ITEMS_ON_EACH_PAGE) + '&' + 'fromTimestamp=' + urllib.parse.quote(from_ti
'toTimestamp=' + urllib.parse.quote(to_timestamp)
# query_params_to_encrypt = '_page=' + PAGE_TO_RETRIEVE + '\n' + '_size=' + NUM_ITEMS_ON_EACH_PA
'fromtimestamp=' + from_timestamp + '\n' + 'totimestamp=' + to_timestamp

# URL query params with paging only
url_query_params = ''
query_params_to_encrypt = ''
# url_query_params = '_page=' + urllib.parse.quote(PAGE_TO_RETRIEVE) + '&' + '_size=' +
urllib.parse.quote(NUM_ITEMS_ON_EACH_PAGE)
# query_params_to_encrypt = '_page=' + PAGE_TO_RETRIEVE + '\n' + '_size=' + NUM_ITEMS_ON_EACH_PA
# use this to handle any body payloads coming in
body_payload = ""

# print(canonicalQueryString)
# ************* REQUEST VALUES *************

hexEncode = hashlib.sha256(body_payload.encode('utf-8')).hexdigest()

if url_query_params != '':
    canonicalRequest = httpRequestMethod + '\n' + canonicalURI + '\n' + query_params_to_encrypt
else:
    canonicalRequest = httpRequestMethod + '\n' + canonicalURI + '\n' + hexEncode

print(canonicalRequest)

# ********************STEP-2  Create the string to sign*********************************
hashedCanonicalRequest = hashlib.sha256(canonicalRequest.encode('utf-8')).hexdigest()

# Set the current timestamp of the request, API version should be set to 1
t = datetime.datetime.utcnow()
requestTimestamp = t.strftime('%Y-%m-%dT%H:%M:%S.%fZ')
apiVersion = "1"

# Step-2
stringToSign = hashedCanonicalRequest + '\n' + apiKey + '\n' + requestTimestamp + '\n' + apiVer
print(stringToSign)

# ************STEP-3  Create the signing key**********************

def sign(key, msg):
    return hmac.new(key, msg.encode('utf-8'), hashlib.sha256).hexdigest()

# Create the signing key using the function defined above.
signingKey = secretKey
signingKey = sign(apiKey.encode('utf-8'), signingKey)
signingKey = sign(requestTimestamp.encode('utf-8'), signingKey)
signingKey = sign(apiVersion.encode('utf-8'), signingKey)

# ************STEP-4  Create the final signature*******************************
# Sign the string_to_sign using the signing_key
signature = sign(signingKey.encode('utf-8'), stringToSign)


# Make the GET call for this API function
# Example with application HID
# url = base_url + api_method_uri + application_hid + "?" + url_query_params
url = base_url + api_method_uri + "?" + url_query_params
headers = {
    'Content-type': 'application/json',
    'Accept': 'application/json',
    'x-arrow-apikey': apiKey,
    'x-arrow-date': requestTimestamp,
    'x-arrow-version': apiVersion,
    'x-arrow-signature': signature
}

r = requests.get(url, headers=headers)
```

```
if r and r.content:
    # format the content as dictionary objects
    returnedData = json.loads(r.content)
    for data in returnedData['data']:
        print(data)
else:
    print('No telemetry data returned.')
else:
    print('There was no data returned from the API.')
```