Hash Verification

Last Modified on 05/24/2018 2:59 pm EDT

All connector instances containing a Callback Notification Signature Key will include the `Elements-Webhook-Signature` header field. The value of this field will contain a unique SHA256 hash that corresponds to the event (e.g., `sha256=jHdbRx5EZAsOfTwAPJOGkNUzQMVVdu5VJlxcsk+G6jQ=`). This hash is calculated by creating a SHA256 hash of the payload using the Callback Notification Signature Key as the key. In order to verify Event Notifications originate from SAP Cloud Platform Open Connectors regenerate the SHA256 hash and test it against the value of the `Elements-Webhook-Signature` header.

SAP Cloud Platform Open Connectors has provided sample code for re-generating the SHA256 hash for event notifications:

Java

```java
import org.apache.commons.codec.binary.Base64;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;


public class SecurityEventKey {

    // Your Event Notification Signature

    private static final String SIGNATUREKEY = "MySecretEventSignatureKey"
;

    // Your Event Notification Payload (Raw Body)

    private static String payload = "";

    public static void main(String[] args) {
        try {
            Mac sha256_HMAC = Mac.getInstance("HmacSHA256");
            SecretKeySpec secret_key = new SecretKeySpec(SIGNATUREKEY.getB
ytes(), "HmacSHA256");
            sha256_HMAC.init(secret_key);

            String hash = Base64.encodeBase64String(sha256_HMAC.doFinal(pa
yload.getBytes()));

            //This should equal what came on the event header as "Elements
-Webhook-Signature"

            System.out.println("sha256=" + hash);
        }
        catch (Exception e){
            System.out.println("Error: " + e);
        }
    }

}

// sha256=jHdbRx5EZAsOfTwAPJOGkNUzQMVVdu5VJlxcsk+G6jQ=
```

Javascript (Node)

```
var hmac = crypto.createHmac('sha256', 'MySecretEventSignatureKey')
           .update('').digest('base64');

// shaHeader == "sha256=" + hmac
// sha256=jHdbRx5EZAsOfTwAPJOGkNUzQMVVdu5VJlxcsk+G6jQ=
```

Ruby

```
require 'openssl'
require "base64"

hash  = OpenSSL::HMAC.digest('sha256', "MySecretEventSignatureKey", "")
puts "sha256="+Base64.strict_encode64(hash)

# sha256=jHdbRx5EZAsOfTwAPJOGkNUzQMVVdu5VJlxcsk+G6jQ=
```