

Edge Fundamentals

Dealing with Secured Backend

Building Target Authentication

Building our Proxy against a target that uses Basic Authentication provides options to store credentials on the Edge Platform.

Which poses the question -

"From the options below, which is the best method to store credentials?"

- Hard Code Them
- 2. Replace Them at Build Time
- 3. Key Value Map
- 4. Encrypted Key Value Map
- 5. Node.js Vault

Proprietary + Confidential

Hard Coding using Assign Message

```
<AssignMessage name="assignSetAuthHeader">
        <Set>
            <Headers>
                 <Header name="Authorization">Basic ASDfasdfas23434radsfaasdf
    a</Header>
            </Headers>
        </Set>
        <AssignVariable>
            <Name>username</Name>
            <Value>cleartext_user</Value>
        </AssignVariable>
10.
        <AssignVariable>
11.
12.
            <Name>password</Name>
            <Value>cleartext_password</Value>
        </AssignVariable>
14.
        <IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
        <AssignTo createNew="false" transport="http" type="request"/>
16.
    </AssignMessage>
```

Build Time Replacement

```
<AssignMessage name=""assignSetAuthHeader"">
 2.
        <Set>
            <Headers>
                 <Header name="Authorization">Replace Me</Header>
            </Headers>
        </Set>
 7.
        <AssignVariable>
             <Name>username</Name>
 9.
             <Value>replace_me</Value>
10.
        </AssignVariable>
11.
        <AssignVariable>
12.
             <Name>password</Name>
             <Value>replace_me</Value>
14.
        </AssignVariable>
15.
        <IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
        <AssignTo createNew="false" transport="http" type="request"/>
16.
    </AssignMessage>
```

Build Time Replacement (contd.)

```
"configurations": [{
        "name": "dev",
        "policies": [{
            "name":"assignSetAuthHeader.xml",
            "tokens": [{
             "value":"Basic ASDfasdfas23434radsfaasdfa",
             "xpath":"/AssignMessage/Set/Headers/Header/@name=Authorization"
            Э,
11.
             "value":"cleartext-username",
             "xpath":"/AssignMessage/AssignVariable[Name="username"]/Value"
12.
            Ъ,
13.
             "value":"cleartext-password",
             "xpath":"/AssignMessage/AssignVariable[Name="password"]/Value"
17.
         "proxies": [],
22.
         "targets": []
23. 111
```

Storing Credentials in a Node.js Access Vault

Step 1: Build Vault and Vault Data Via API

```
curl https://api.enterprise.apigee.com/v1/o/{org}/vaults
-H "Content-Type: application/json"
-d '{"name": "vault" }' -X POST

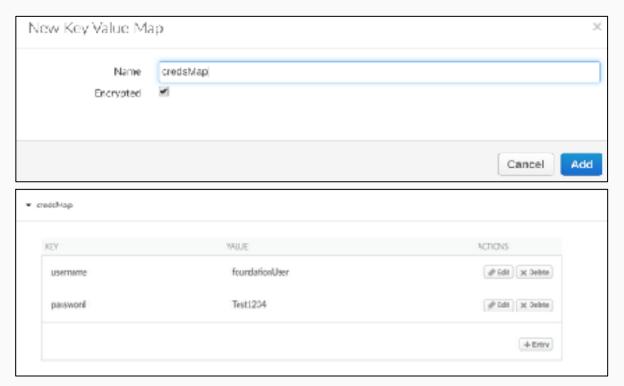
curl https://api.enterprise.apigee.com/v1/o/{org}/vaults/{vault}/entries
-H "Content-Type: application/json"
-d '{"name": "valuel", "value": "verysecret" }' -X POST
```

Step 2: Retrieve Data in Node.js App

```
var apigee = require('apigee-access');
var orgVault = apigee.getVault('vault1', 'organization');
orgVault.get('key1', function(err, secretValue) {
    // use the secret value here
};
```

Storing Credentials in a Key Value Map (encrypted)

Step 1: Create KVM and KVM Data Via UI



Storing Credentials in a Key Value Map (encrypted)

Step 2: Retrieve Data via KVM Policy

```
<KeyValueMapOperations name="getEncrypted" mapIdentifier="credsMap">
  <Scope>environment</Scope>
  <Get assignTo="private.encryptedUsername" index="1">
      <Key>
         <Parameter>username</Parameter>
      </Key>
  </Get>
  <Get assignTo="private.encryptedPassword" index="1">
      <Key>
         <Parameter>password</Parameter>
      </Key>
  </Get>
</KeyValueMapOperations>
```

Proprietary + Confidential

Lab

Please follow the steps provided here

THANK YOU