False Positive Documentation

Kindly provide the answers for the following queries regarding the vulnerability.

1. Name of the Application

Credex Backup & Recovery Solution

2. Package ID/Version ID/Listing ID

Package ID :- 04t5i0000002la3, version ID- 3D04t5i0000002la3 version :- 1.5

3. Partner Name

Credex Technology

4. Document Filing Date (YYYY-MM-DD)

2023-12-04

5. Target URL (Optional)

6. Review Date (In Case if app already undergone security review)

7. Vulnerability Name

Query: Async Future Method inside Loops

Meant for the Internal Salesforce Security Team

Final Comment from Salesforce Team	
Date of Review (YYYY-MM-DD)	
Result	

8. Detected By

- ~~*Checkmarx*~~
- PMD
- Security Review
- Others,

  If case of others Please provide details :

9. Detailed explanation for considering False Positive (Elaborate as much as possible)

This was flagged in checkmarx report and is a false positive as we know that at most 50 future method callouts are allowed in a single transaction so in our code we have added limit of 50 in for loop, so it will not hit the governor limits.

10. Evidence (screenshots of the code, implementation, Command Output)

```
@auraEnabled(cacheable=true)
Public static String createDataBackup(List<String> ObjectApiNames,String credentials,String fromDate, String toDate){

    try{
        List<String> batches = new List<String>();
        if(ObjectApiNames.size()>50)
        {
            double batchresult = (double)(ObjectApiNames.size())/50;
            double decimalPart = batchresult - (integer)batchresult;
            integer batchSize = (integer)batchresult;

            if(decimalpart>0){
                batchSize++;
            }
            for(integer i=0;i<=ObjectApiNames.size();i+=50){
                List<String> objectBatch = new List<String>();
                for(integer j=i;j<50+i && j<ObjectApiNames.size();j++)
                {
                    objectBatch.add(ObjectApiNames[j]);
                }
                batches.add('batch'+batchSize);
                DataBackupCallout.getQueryJobIds(objectBatch,'batch'+batchSize,fromDate,toDate);
                batchSize-=1;
            }
         }
```

11. Vulnerability Name

ApexSuggestUsingNamedCred

Meant for the Internal Salesforce Security Team

Final Comment from Salesforce Team	
Date of Review (YYYY-MM-DD)	
Result	

12. Detected By

- Checkmarx
- ~~*PMD*~~
- Security Review
- Others,

  If case of others Please provide details :

13. Detailed explanation for considering False Positive (Elaborate as much as possible)

This was flagged in PMD report as we are getting the credentials from the UI dynamically, this is a false positive as we cannot update the named credentials as per user input from the UI so cannot use named credentials.

14. Evidence (screenshots of the code, implementation, Command Output)

```
Http http = new Http();

HTTPResponse response = http.send(req);


if (response.getStatusCode() == 200 || response.getStatusCode()==201)
{
    Dom.Document doc = new Dom.Document();
    doc.load(response.getbody());
    Dom.XmlNode rootNode = doc.getRootElement();

    for(Dom.XmlNode childElement:rootNode.getChildElements())
    {
        for(Dom.XmlNode detailElement:childElement.getChildElements())
        {
            for(Dom.XmlNode detailElement1:detailElement.getChildElements()){
                if(detailElement1.getName()=='Name'){
                    Buckets.add(detailElement1.getText());
                }
            }
        }
    }
}
```

15. Vulnerability Name

ApexSuggestUsingNamedCred

Meant for the Internal Salesforce Security Team

Final Comment from Salesforce Team	
Date of Review (YYYY-MM-DD)	
Result	

16. Detected By

- Checkmarx
- ~~PMD~~
- Security Review
- Others,

  If case of others Please provide details :

17. Detailed explanation for considering False Positive (Elaborate as much as possible)

This was flagged in PMD report as we are getting the credentials from the UI dynamically, this is a false positive as we cannot update the named credentials as per user input from the UI so cannot use named credentials.

18. Evidence (screenshots of the code, implementation, Command Output)

```
      /*
@description => This method is use to create Http request for Bulk api
*/
    private static HttpRequest CreateHttpRequest(String method){
        HttpRequest request = new HttpRequest();
        request.setMethod(method);
        request.setHeader('Authorization', 'Bearer ' + userinfo.getSessionId());
        request.setHeader('Content-Type', 'application/json');
        //    request.setBody(requestBody);

        return request;
    }
```

19. Vulnerability Name

ApexSuggestUsingNamedCred

Meant for the Internal Salesforce Security Team

Final Comment from Salesforce Team	
Date of Review (YYYY-MM-DD)	
Result	

20. Detected By

- Checkmarx
- ~~*PMD*~~
- Security Review
- Others,

  If case of others Please provide details :

21. Detailed explanation for considering False Positive (Elaborate as much as possible)

This was flagged in PMD report as we are getting the credentials from the UI dynamically, this is a false positive as we cannot update the named credentials as per user input from the UI so cannot use named credentials.

22. Evidence (screenshots of the code, implementation, Command Output)

```
public static void createInsertJob(Map<String,Blob> objectWithData) {
    try{
        List<DataRecovery.ObjectsJobIdCSVWrapper> ObjectsJobIdCSVWrapperList = new List<DataRecovery.ObjectsJobIdCSVWrapper>();
        List<String> objectApiNames = new List<String>(objectWithData.keyset());
        for(String objectApiName:objectApiNames)
        {
            String endpoint = endpointURL;
            String requestBody = JSON.serialize(getJobRequestForInsert(objectApiName));

            HttpRequest request = new HttpRequest();
            request.setEndpoint(endpoint);
            request.setMethod('POST');
            request.setHeader('Authorization', 'Bearer ' + UserInfo.getSessionId());
            request.setHeader('Content-Type', 'application/json');
            request.setBody(requestBody);
            HttpResponse response = new Http().send(request);
            if(response.getStatusCode() == 200 || response.getStatusCode() == 201)
            {
                Map<String, Object> jobResponse = (Map<String, Object>) JSON.deserializeUntyped(response.getBody());

                DataRecovery.ObjectsJobIdCSVWrapper objWrapper = new DataRecovery.ObjectsJobIdCSVWrapper();
                objWrapper.objectName = objectApiName;
                objWrapper.jobId = (String)jobResponse.get('id');
                objWrapper.csvData = objectWithData.get(objectApiName);
                objWrapper.DmlType ='insert';
                ObjectsJobIdCSVWrapperList.add(objWrapper);
            }
```