False Positive Documentation

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

1. Name of the Application

2. Package ID/Version ID/Listing ID

3. Partner Name

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

5. Target URL (Optional)

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

Repeat Step 6 to 9 in case of multiple Vulnerability

7. Vulnerability Name

Meant for the Internal Salesforce Security Team

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

8. Detected By

- Chimera
- PMD
- Security Review
- Others,

  If case of others Please provide details :

  _____

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

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

References

1. *Name of the Application*

Hello for Salesforce

2. *Package ID/Version ID/Listing ID*

a0N4V00000HwgXYZK

3. *Partner Name*

Hari Krishna Company Ltd

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

2022-08-20

5. *Target URL (Optional)*

https://test.world.com/

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

2022-08-15

Repeat Step 7 to 10 in case of multiple Vulnerability

7. *Vulnerability Name*

CRUD Violation

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)*

CRUD Violation Issue was flagged during security review as it was mentioned under explanation that no CRUD/FLS check was performed before the DML operation. But it is a false positive as a relevant check is performed in the BaseHelper class and the ‘checkCRUDPermissionOnSobject’ object is being called before the DML operation. ‘checkCRUDPermissionOnSobject’ object necessary to check the CRUD permission along with the FLS check.

Relevant evidence provided in the next section. Hence the following vulnerability should be marked as False positive and marked closed.

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

Dummy.cls

```
public static Dummy (){

          BaseHelper.checkCRUDPermissionOnSobject('Test', 'updateable');
          Test Item = new Test();
          Item.Id = accordionItmId;
          Item.Dummy = User.Id;
          update Item;
     }
```

BaseHelper.cls

```
public static void checkCRUDPermissionOnSobject(String sobjName, Set<String>
permissionTypes){
   PermissionWrapper permissionWrapperObj = new PermissionWrapper();
   permissionWrapperObj = setPermissionWrapper(sobjName, null);
   Schema.DescribeSObjectResult sobjResult = permissionWrapperObj.sobjResult;
   Boolean isPermissionTypeIssue = isCRUDPermissionTypeIssueOnSobject(sobjName,
permissionTypes);
   if (isPermissionTypeIssue){
     throw new DataTableHelper.OtherException('Insufficient \'' + permissionTypes +
'\' permission access to sobject \'' + sobjResult.getLabel() + '\'');
   }
 }
```

Repeat Step 11 to 14 in case of multiple Vulnerability

11. Vulnerability Name

Sharing Violation

Meant for the Internal Salesforce Security Team

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

12. Detected By

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

If case of others Please provide details :

_____

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

Sharing Violation issue was flagged during the chimera scan as class was declared as without sharing. As this class has to be called by all the users whoever wants to access the application (even outside the org also) for the installation. Sharing permission is explicitly omitted and hence it is declared as Without Sharing.

As this is a business requirement this issue should be marked as false positive and closed.

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

```
global without sharing class PostInstallClass implements InstallHandler {
   global void onInstall(InstallContext context) {
       SobjectIconController.createSobjectIconMetadata();

       if(context.previousVersion() == null) {
           DataFactory.createDataFactorySetting();
       }
```