# False Positive Documentation

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

1. Name of the Application

| SetupAI |
| --- |

1. Package ID/Version ID/Listing ID

| 04tDp000000uQd2IAE |
| --- |

1. Partner Name

| SetupAI |
| --- |

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

| 2023-09-06 |
| --- |

1. Target URL (Optional)

|  |
| --- |

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

|  |
| --- |

#### Repeat Step 6 to 9 in case of multiple Vulnerability

1. Vulnerability Name

| Validate CRUD permission before SOQL/DML operation or enforce user mode |
| --- |

#### *Meant for the Internal Salesforce Security Team*

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

1. Detected By

| * Chimera * PMD * Security Review * Others,   If case of others Please provide details : Code Analyzer |
| --- |

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

| I have a custom CRUD class that contains methods for checking user CRUD access to the Chat\_\_c and Message\_\_c objects so that all field and object access can be checked more succinctly.  The Code Analyzer picked up a violation on line 25 of Controller.cls where a new Message\_\_c is inserted. This insert statement is inside an if statement on line 21 that checks CRUD access on both Chat and Message.  2 more violations are on lines 42 and 43 of Controller.cls inserting a Chat\_\_c and a Message\_\_c. These are inside an if statement on line 40 checking CRUD access.  Finally, line 52 of Function.cls creates a Message\_\_c record. This is inside a CRUD check if statement on line 51. |
| --- |

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

| **Controller.cls - lines 16-52**  //Saves user message to Salesforce  @AuraEnabled  public static String submitMessage(String content, String chatId){  try {  //Verify access to messages and chat  if(CRUD.createChat() && CRUD.createMessage()){  if(chatId == null){  return initialize(content);  }else{  insert new Message\_\_c(Thread\_\_c = chatId,  Content\_\_c = content,  Role\_\_c = 'user');  return null;  }  }else{  throw new AuraHandledException(Label.Create\_Chat\_Access);  }  } catch (Exception e) {  throw new AuraHandledException(e.getMessage());  }  }  //Creates Chat and System message before inserting first chat message  private static String initialize(String firstMessage){  if(CRUD.createChat() && CRUD.createMessage()){  Chat\_\_c chat = generateChat(firstMessage);  insert chat;  insert new Message\_\_c(  Role\_\_c = 'user',  Content\_\_c = firstMessage,  Thread\_\_c = chat.Id  );  return chat.Id;  }else{  throw new AuraHandledException(Label.Create\_Chat\_Access);  }  }  **Function.cls - lines 50-56**  //Check message create permissions  if(CRUD.createMessage()){  insert new Message\_\_c(Thread\_\_c = chatId,  Role\_\_c = 'function',  Content\_\_c = content,  Function\_Name\_\_c = name);  }  **CRUD.cls - entire class**  public with sharing class CRUD {  //Validates creation of Chat\_\_c records  public static Boolean createChat(){  return (  Chat\_\_c.SObjectType.getDescribe().isCreateable() &&  Schema.SObjectType.Chat\_\_c.fields.Name.isCreateable() &&  Schema.SObjectType.Chat\_\_c.fields.Running\_\_c.isCreateable()  );  }  //Validates creation of Message\_\_c records  public static Boolean createMessage(){  return (  Message\_\_c.SObjectType.getDescribe().isCreateable() &&  Schema.SObjectType.Message\_\_c.fields.Completion\_Tokens\_\_c.isCreateable() &&  Schema.SObjectType.Message\_\_c.fields.Content\_\_c.isCreateable() &&  Schema.SObjectType.Message\_\_c.fields.Function\_Arguments\_\_c.isCreateable() &&  Schema.SObjectType.Message\_\_c.fields.Function\_Name\_\_c.isCreateable() &&  Schema.SObjectType.Message\_\_c.fields.Model\_\_c.isCreateable() &&  Schema.SObjectType.Message\_\_c.fields.Prompt\_Tokens\_\_c.isCreateable() &&  Schema.SObjectType.Message\_\_c.fields.Role\_\_c.isCreateable() &&  Schema.SObjectType.Message\_\_c.fields.Thread\_\_c.isCreateable()  );  }  } |
| --- |