# 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

Type to enter text

|  |
| --- |
| 04tDp000000qToeIAE |

1. Partner Name

|  |
| --- |
| SetupAI |

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

|  |
| --- |
| 2024-03-04 |

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 |  |

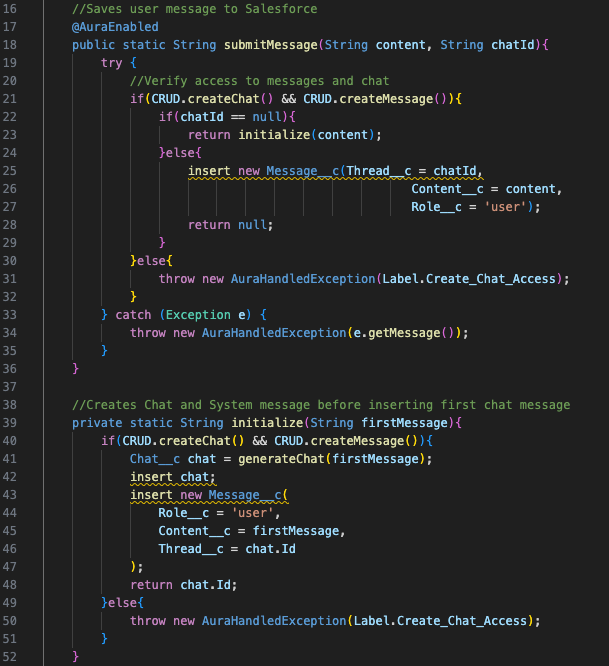
1. Detected By

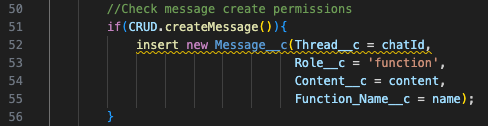
|  |
| --- |
| * Chimera * PMD * Security Review * Others,   If case of others Please provide details : Code Analyzer |

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

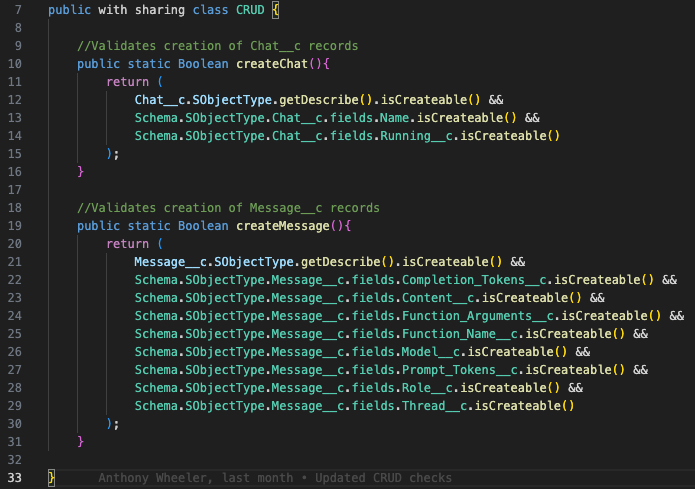
|  |
| --- |
| I have a custom CRUD class that contains methods for checking user CRUD access to the Chat\_\_c and Message\_\_c objects so that all field and object access can be checked more efficiently.  The Code Analyzer picked up a violation on line 25 of Controller.cls where a new Message\_\_c is 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.  Line 52 of Function.cls creates a Message\_\_c record. This is inside a CRUD check if statement on line 51.  Lastly, on line 35 of the CalloutSecurity class, there is an update on a SetupAI\_\_Token\_Usage\_\_c record. This is a protected custom setting that the end user isn’t supposed to be able to access in the UI, but updates to this custom setting by end users through Apex are required for the app to function. I’ve encapsulated this update statement inside an if statement on line 32 checking that the user has the SetupAI\_\_SetupAI custom permission set on their user.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  There were also a number of false positives in the PMD scan.  All but one mistook variables with ‘password’ or ‘apikey’ in the name to be hardcoded credentials, when in reality all credentials are saved in Salesforce setup before being retrieved in Apex.  The other remaining callout is on using getSessionId in Apex. This is required in the app as access to the Tooling API is needed to inform the AI. This call is only done on line 49 of the Callout class and is directly placed into a header map containing other headers. All the headers are then immediately placed into the header of the Tooling API request. |

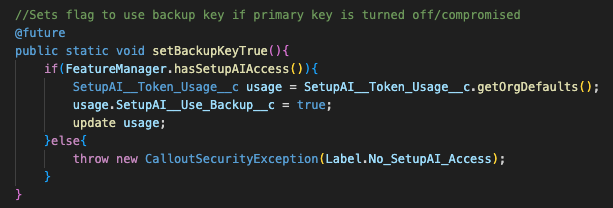
1. Evidence (screenshots of the code, implementation, Command Output)

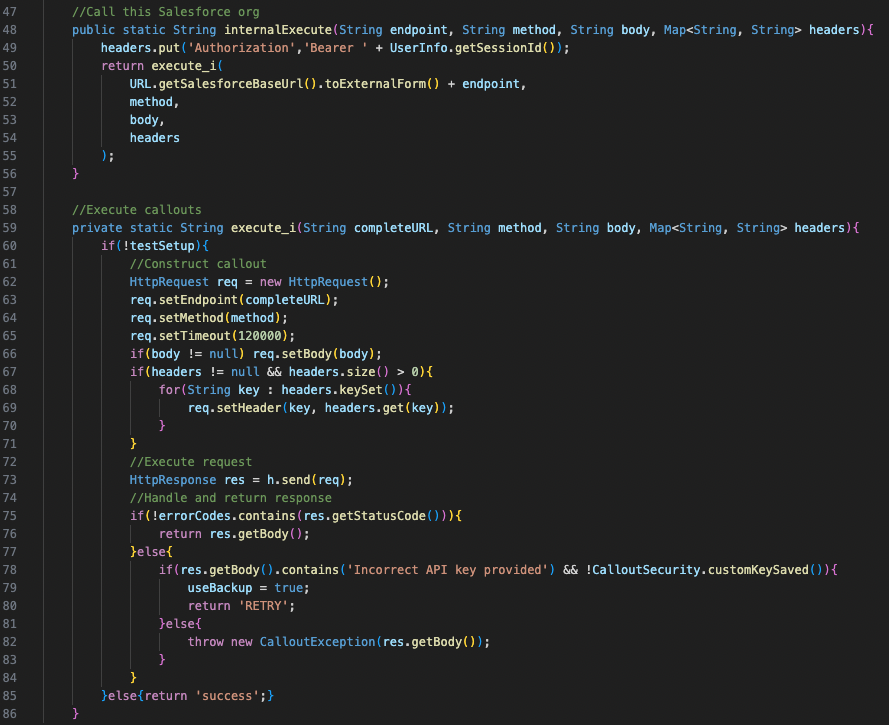
**Controller.cls - lines 16-52**

**Function.cls - lines 50-56**

**CRUD.cls - Entire Class**

****

**CalloutSecurity.cls**

**Callout.cls - 47-86**