DEREF\_OF\_NULL.RET.ASSERT

Детектор предназначен для ситуаций проверки возвращаемого значения функций на ноль. Указывает на что ошибка возникает в специальной конструкции assert.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Может привести к срыву выполнения программы или выбросу исключения (NullPointerException) в случае, если условие внутри assert не выполняется. |
| Комментарий | Использование конструкции assert для проверки на null не всегда безопасно, так как в продакшн-режиме утверждения могут быть отключены. Рекомендуется использовать явные проверки на null для обеспечения надежности программы. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  String value = getMaybeNull();  // попытка разыменования null в assert  assert value != null : value.length();  } | public static void correctTest() {  String value = getMaybeNull();  assert value != null : "value is null";  } |

Методы устранения уязвимости при срабатывании детектора:

1. Избегайте разыменования потенциально нулевых значений непосредственно в выражении assert. assert предназначен для проверки предположений, которые всегда должны быть истинными.
2. Если вам нужно проверить длину или другой атрибут возвращаемого значения, сделайте это *перед* assert. Сохраните результат проверки в булевой переменной и используйте эту переменную в assert.
3. Используйте assert только для проверки того, что возвращаемое значение не null. Сообщение об ошибке в assert должно быть простым и информативным о том, что предположение о ненулевом значении нарушено.
4. Для обработки случая, когда возвращаемое значение может быть null в обычном ходе выполнения программы, используйте явные операторы if (value == null) и соответствующую логику обработки (например, возврат, выброс исключения, использование значения по умолчанию). Не полагайтесь на assert для обработки таких ситуаций, так как assert может быть отключен в production.