DEREF\_AFTER\_NULL.UNCLEAR

Детектор соответствует передаче указателя в неизвестную функцию.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Функция может разыменовать null-указатель, что приведёт к исключению или аварийному завершению работы программы. |
| Комментарий | Необходимо обеспечить, чтобы передаваемое значение гарантированно не было null, либо использовать защитные обёртки и дополнительные проверки перед вызовом неизвестных функций. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  String value = getNullable();  // возможно value == null, а метод будет делать value.length()  unknownMethod(value);  } | public static void correctTest() {  String value = getNullable();  if (value != null) {  unknownMethod(value); // безопасно  } else {  System.out.println("Null value, method not called");  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Добавьте явную проверку на null перед вызовом неизвестной функции. Если указатель может быть нулевым, убедитесь, что вы обрабатываете этот случай до передачи его в функцию.
2. Если null является недопустимым значением для неизвестной функции, выбросьте исключение или верните ошибку до вызова функции. Это предотвратит передачу некорректных данных.
3. Используйте защитные обертки (например, Optional в Java) для передачи потенциально отсутствующих значений. Это обязывает вызывающий код явно обрабатывать случай отсутствия значения перед его использованием в неизвестной функции (если функция ожидает Optional).