UNREACHABLE\_CODE.SWITCH

Означает то, что в блоке кода switch-case есть код, который никогда не будет выполнен или достигнут. Это может произойти по различным причинам. Например, если у вас есть case без соответствующего break, и следующий case никогда не будет выполнен. Или это может быть связано с логическими ошибками в условиях case.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Ошибка может привести к тому, что блоки кода останутся неиспользуемыми, что ухудшит читаемость программы и повысит вероятность других ошибок. |
| Комментарий | Необходимо тщательно проверять логику условий в конструкции switch-case, чтобы убедиться, что все возможные случаи корректно обрабатываются, а недостижимый код удаляется. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(int value) {  int x = value % 2;  switch (x) {  case 0:  System.out.println("x is 0");  break;  case 1:  System.out.println("x is 1");  break;  case 2: // Этот case всегда будет выполнен  System.out.println("x is 2");  break;  }  } | public static void correctTest(int value) {  int x = value % 2;  switch (x) {  case 0:  System.out.println("x is 0");  break;  case 1:  System.out.println("x is 1");  break;  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Проверьте логику условий case. Убедитесь, что условия case в конструкции switch логически исключают друг друга или что поток выполнения корректно завершается (например, с помощью break, return, throw) после каждого блока case, если это необходимо.
2. Удалите недостижимые блоки case. Если определенные блоки case никогда не будут выполнены из-за логики условий или отсутствия соответствующих значений для выражения switch, удалите эти блоки.