REDUNDANT\_COMPARISON

Результат определенных сравнений иногда можно вывести из их контекста и результатов других сравнений. Это может указывать на ошибочную логику и может привести к неработающему коду или бесконечным циклам, если, например, условие цикла никогда не меняет своего значения.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | Goa;SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Избыточные сравнения увеличивают сложность и неоптимальность кода, снижая его читаемость и производительность. Они могут также скрывать ошибки логики, такие как бесконечные циклы или недостижимые условия, что приводит к неправильному поведению программы. |
| Комментарий | Рекомендуется удалять лишние сравнения, чтобы улучшить читаемость и поддерживаемость кода. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(int b) {  int a = 10;  // Избыточное сравнение: b == 10, так как a == b  if (a == b && b == 10) {  System.out.println("a and b are both 10");  }  } | public static void correctTest(int b) {  int a = 10;  if (a == b) {  System.out.println("a and b are both 10");  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Выявите избыточные сравнения. Проанализируйте логику условных операторов и циклов, чтобы определить сравнения, результат которых можно определить на основе других сравнений в том же контексте.
2. Удалите избыточное сравнение. Упростите условие, удалив ту часть сравнения, которая является избыточной и не добавляет новой информации к логике.
3. Пересмотрите логику (при необходимости). Если удаление избыточного сравнения приводит к изменению предполагаемой логики, пересмотрите условие целиком, чтобы убедиться, что оно по-прежнему корректно отражает требуемое поведение программы.