HANDLE\_LEAK.CLOSEABLE

Указывает на утечку ресурса, в частности, на то что объект, который реализует интерфейс не был правильно закрыт.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Неправильно закрытые ресурсы могут привести к утечке памяти или системных ресурсов, таких как файловые дескрипторы, что в долгосрочной перспективе может вызвать сбои приложения или замедление работы. |
| Комментарий | Следует строго соблюдать правило закрытия всех ресурсов, реализующих интерфейс Closeable, особенно при работе с потоками, файлами и сетевыми соединениями. Оптимальным решением является использование конструкции try-with-resources, которая обеспечивает автоматическое закрытие ресурсов. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  BufferedReader reader = null;  try {  reader = new BufferedReader(new FileReader("test.txt"));  System.out.println(reader.readLine());  } catch (IOException e) {  throw new RuntimeException(e);  }  } | public static void correctTest() {  try (BufferedReader reader = new BufferedReader(new FileReader("test.txt"))) {  System.out.println(reader.readLine());  } catch (IOException e) {  throw new RuntimeException(e);  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Используйте конструкцию try-with-resources (начиная с Java 7) для всех объектов, реализующих интерфейс AutoCloseable (который расширяет Closeable). Это гарантирует автоматическое закрытие ресурса после завершения блока try (обычным образом или из-за исключения).
2. Если try-with-resources использовать невозможно, вручную закройте ресурс в блоке finally. Это гарантирует, что ресурс будет закрыт независимо от того, возникло исключение или нет.
3. Перед вызовом метода close() в блоке finally добавьте проверку на null. Это предотвратит NullPointerException, если ресурс не был успешно инициализирован.