HANDLE\_LEAK.FRUGAL

Относится к ситуациям, когда дескриптор (например, файловый дескриптор или дескриптор сокета), полученный из системы, не освобождается в тех местах кода, где это возможно.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | Roslyn;SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Хотя в краткосрочной перспективе это может не вызвать утечку ресурсов, накопление незакрытых дескрипторов может привести к исчерпанию доступных системных ресурсов, что может привести к сбоям программы, блокировкам или проблемам с производительностью, особенно при высоком объеме операций с ресурсами. |
| Комментарий | Рекомендуется всегда освобождать дескрипторы сразу после завершения их использования, используя подходы управления ресурсами. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  FileInputStream fis;  try {  fis = new FileInputStream("test.txt");  System.out.println(fis.read());  } catch (IOException e) {  throw new RuntimeException(e);  }  // Файл не закрывается здесь или в блоке finally, утечка дескриптора  } | public static void correctTest() {  try (FileInputStream fis = new FileInputStream("test.txt")) {  System.out.println(fis.read());  } catch (IOException e) {  throw new RuntimeException(e);  }  } |