FB.STI\_INTERRUPTED\_ON\_UNKNOWNTHREAD

Детектор определяет ошибку, в ходе которой метод вызывает метод Thread.interrupted() для объекта Thread, который выглядит как объект Thread, который не является текущим потоком. Поскольку метод прерывания() является статическим, метод прерывания будет вызываться для другого объекта, отличного от того, который задумал автор.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Некорректное использование Thread.interrupted() может привести к неожиданному поведению программы. Например, поток, для которого ожидалось сбросить флаг прерывания, останется без изменений, а флаг прерывания может сбрасываться у совершенно другого потока, что затруднит обработку прерываний и приведёт к трудноотлавливаемым ошибкам. |
| Комментарий | Следует использовать Thread.currentThread().isInterrupted() или Thread.currentThread().interrupt(), если требуется корректно управлять состоянием прерывания текущего потока. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  Thread thread = new Thread(() -> {  // ...  });  thread.start();  if (thread.interrupted()) {  // Thread was interrupted ...  }  } | public static void correctTest() {  Thread thread = new Thread(() -> {  // ...  });  if (Thread.interrupted()) {  // Thread was interrupted ...  }  thread.start();  // Прерываем поток  thread.interrupt();  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Использование Thread.currentThread().isInterrupted() для проверки состояния прерывания текущего потока:** Если вы хотите проверить, был ли прерван *текущий* поток, используйте нестатический метод isInterrupted() экземпляра текущего потока, полученного через Thread.currentThread(). Этот метод возвращает текущее состояние флага прерывания, не сбрасывая его.
2. **Использование Thread.currentThread().interrupt() для установки флага прерывания текущего потока:** Если вам необходимо установить флаг прерывания для *текущего* потока (например, чтобы сигнализировать о необходимости завершения работы), используйте нестатический метод interrupt() экземпляра текущего потока, полученного через Thread.currentThread().
3. **Для проверки состояния прерывания *другого* потока (что обычно не рекомендуется делать напрямую):** Если по какой-то причине вам необходимо проверить состояние прерывания другого потока, вы должны использовать нестатический метод isInterrupted() *экземпляра* этого потока. Однако следует помнить, что прямое управление флагом прерывания другого потока может привести к проблемам с синхронизацией и непредсказуемому поведению.