FB.TQ\_UNKNOWN\_VALUE\_USED\_WHERE\_ALWAYS\_STRICTLY\_

REQUIRED

Значение используется таким образом, что требуется аннотация значения квалификатором типа. Квалификатор типа является строгим, поэтому инструмент отклоняет любые значения, не имеющие соответствующей аннотации.

Чтобы заставить значение иметь строгую аннотацию, определите функцию идентификации, в которой возвращаемое значение будет помечено строгой аннотацией. Это единственный способ превратить неаннотированное значение в значение со строгой аннотацией квалификатора типа.

Таблица №1 – Основные характеристики детектора FB.TQ\_UNKNOWN\_VALUE\_USED\_WHERE\_ALWAYS\_STRICTLY\_

REQUIRED

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Unverified Value in Strict Context |
| CWE | --- |
| CVE | --- |
| Возможные последствия | Нарушение инвариантов программы, критические сбои при выполнении, уязвимости безопасности при обработке непредусмотренных значений, неопределённое поведение системы. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  // 1. Непроверенный URL  String userUrl = getUserInput();  openSecureConnection(userUrl); // @UnknownValue в строгом контексте  // 2. Невалидированный размер файла  int fileSize = getUploadedFileSize();  processFile(fileSize); // Возможное превышение лимита  // 3. Непроверенный объект  Object rawData = parseUserData();  saveToSecureStorage(rawData); // Нарушение типовой безопасности  } | public static void correctTest() {  // 1. Строгая валидация URL  String userUrl = validateUrl(getUserInput());  if (userUrl != null) {  openSecureConnection(userUrl);  }  // 2. Контроль размера файла  int fileSize = getUploadedFileSize();  if (fileSize > 0 && fileSize <= MAX\_FILE\_SIZE) {  processFile(fileSize);  }  // 3. Гарантированная типизация  UserData data = safelyParseUserData();  if (data != null && data.isValid()) {  saveToSecureStorage(data);  }  } |