UNCHECKED\_FUNC\_RES.USER

Предупреждение UNCHECKED\_FUNC\_RES.USER указывает на то, что результат функции не проверяется перед использованием. Это может привести к непредсказуемому поведению программы, если функция возвращает ошибку или неожиданное значение. Например, если функция должна возвращать указатель на объект, но происходит ошибка и возвращается NULL, а затем этот NULL используется без проверки - это приведет к ошибке.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Может привести к разыменованию null-указателя, логическим ошибкам или сбоям выполнения, если функция возвращает ошибку или недопустимое значение. |
| Комментарий | Рекомендуется всегда проверять возвращаемые значения функций перед использованием, особенно если они могут сигнализировать об ошибках или аномальных ситуациях. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  String username = getUsernameById(-1);  // Не проверили на null — и сразу вызываем метод (NullPointerException)  System.out.println("Username length: " + username.length());  }  public static String getUsernameById(int id) {  if (id < 0) {  return null;  }  return "user\_" + id;  } | public static void correctTest() {  String username = getUsernameById(-1);  if (username != null) {  System.out.println("Username length: " + username.length());  } else {  System.out.println("User not found");  }  }  public static String getUsernameById(int id) {  if (id < 0) {  return null;  }  return "user\_" + id;  } |

Методы устранения уязвимости при срабатывании детектора:

1. Проверьте возвращаемое значение на наличие ошибок. Сразу после вызова функции проверьте, не вернула ли она значение, сигнализирующее об ошибке (например, null, -1, false, исключение).
2. Обработайте возможные ошибки. Если функция вернула ошибку, реализуйте соответствующую логику для обработки этой ситуации: вывод сообщения об ошибке, повторная попытка выполнения, использование значения по умолчанию или завершение программы.
3. Проверьте возвращаемое значение на допустимость. Убедитесь, что возвращаемое значение находится в ожидаемом диапазоне или является допустимым для дальнейшего использования, особенно если функция может возвращать неожиданные, но формально не ошибочные значения.