FB.ICAST\_QUESTIONABLE\_UNSIGNED\_RIGHT\_SHIFT

Детектор определяет ошибку, в ходе которой код выполняет беззнаковый сдвиг вправо, результат которого затем преобразуется в короткое число или байт, при этом старшие биты результата отбрасываются. Поскольку старшие биты отбрасываются, разницы между знаковым и беззнаковым сдвигом вправо может не быть (в зависимости от размера сдвига).

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Беззнаковый сдвиг вправо (>>>), примененный к int или long, с последующим приведением к short или byte, может привести к непредсказуемым результатам, поскольку старшие биты отбрасываются. |
| Комментарий | Необходимо убедиться, что беззнаковый сдвиг вправо действительно необходим. Если старшие биты все равно отбрасываются, возможно, стоит использовать обычный (>>) сдвиг или применить явное побитовое и (&) для маскирования нужных битов. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class MyClass {  void shiftExample(int value) {  short result = (short) (value >>> 8);  System.out.println("Incorrect result: " + result);  }  }  MyClass myClass = new MyClass();  myClass.shiftExample(-1024);  } | public static void correctTest() {  class MyClass {  void shiftExample(int value) {  // явно обрезаем до 16 бит перед приведением  short result = (short) ((value >>> 8) & 0xFFFF);  System.out.println("Correct result: " + result);  }  }  MyClass myClass = new MyClass();  myClass.shiftExample(-1024);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Проанализируйте необходимость беззнакового сдвига:** Убедитесь, что вы действительно нуждаетесь в поведении беззнакового сдвига (заполнение старших битов нулями) и что последующее приведение к short или byte не нивелирует этот эффект из-за отбрасывания старших битов.
2. **Примените побитовое И (&) с маской:** Если вам нужны только определенные младшие биты после беззнакового сдвига, явно примените побитовое И с соответствующей маской (например, 0xFFFF для short, 0xFF для byte) перед приведением типа.
3. **Рассмотрите использование знакового сдвига (>>):** Если отбрасывание старших битов является приемлемым, и вам не требуется заполнение старших битов нулями (например, при арифметическом сдвиге), возможно, обычный знаковый сдвиг (>>) будет более подходящим.
4. **Проверьте логику программы:** Убедитесь, что выбранный метод сдвига и приведения типа соответствует ожидаемому поведению программы и не приводит к ошибкам.
5. **Будьте внимательны к размерам типов:** Помните о различиях в размерах типов int, long, short и byte и о том, как приведение типов может влиять на значения.