FB.DM\_BOOLEAN\_CTOR

Детектор определяет ошибку, в ходе которой создание новых экземпляров java.lang.Boolean тратит впустую память, поскольку логические объекты неизменяемы и существует только два полезных значения этого типа. Вместо этого используйте метод Boolean.valueOf() (или автоупаковку Java 5) для создания логических объектов.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Создание новых экземпляров Boolean с помощью конструктора (new Boolean(true) или new Boolean(false)) ведет к избыточному расходованию памяти, поскольку существует только два возможных значения этого типа. |
| Комментарий | Рекомендуется использовать метод Boolean.valueOf() или автоупаковку (например, Boolean true или Boolean false) вместо конструктора для логических объектов, что позволит избежать ненужного выделения памяти и повысить производительность. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  // deprecated since version 9  Boolean bool1 = new Boolean(true);  Boolean bool2 = new Boolean(false);  System.out.println(bool1);  System.out.println(bool2);  } | public static void correctTest() {  // Правильное использование Boolean.valueOf()  Boolean bool1 = Boolean.valueOf(true);  // Автоупаковка true  Boolean bool2 = true;  System.out.println(bool1);  System.out.println(bool2);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Замените new Boolean(true) на Boolean.valueOf(true):** Используйте статический фабричный метод valueOf() для получения экземпляра Boolean, представляющего true.
2. **Замените new Boolean(false) на Boolean.valueOf(false):** Аналогично, используйте Boolean.valueOf() для получения экземпляра Boolean, представляющего false.
3. **Используйте автоупаковку:** Если вы присваиваете примитивное значение boolean переменной типа Boolean, Java автоматически выполнит упаковку. Например, Boolean b = true; или Boolean b = someBooleanVariable;.
4. **Проверьте весь код:** Найдите все места, где используется конструктор new Boolean() и замените его на рекомендуемые способы.
5. **Обратите внимание на сравнение:** При сравнении объектов Boolean рекомендуется использовать метод equals() вместо оператора ==, чтобы сравнивать значения, а не ссылки на объекты (хотя Boolean.valueOf() и автоупаковка способствуют повторному использованию одних и тех же объектов Boolean.TRUE и Boolean.FALSE).