FB.RV\_RETURN\_VALUE\_OF\_PUTIFABSENT\_IGNORED

Детектор определяет уязвимость, для исправления которой требуется знать, что метод putIfAbsent обычно используется, чтобы гарантировать, что с данным ключом связано одно значение (первое значение, для которого put, если оно отсутствует, завершается успешно). Если вы проигнорируете возвращаемое значение и сохраните ссылку на переданное значение, вы рискуете сохранить значение, отличное от того, которое связано с ключом в карте. Если имеет значение, какой из них вы используете, и вы используете тот, который не сохранен в карте, ваша программа будет вести себя неправильно.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE440 |
| CVE | — |
| Возможные последствия | Игнорирование возвращаемого значения putIfAbsent() может привести к тому, что используемое значение не будет соответствовать фактически сохраненному в карте. |
| Комментарий | При использовании putIfAbsent() необходимо проверять возвращаемое значение: если оно не null, значит, в карте уже было сохранено другое значение, и переданное значение не было записано. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  Map<String, String> map = new ConcurrentHashMap<>();  // Игнорируется возвращаемое значение  map.putIfAbsent("key", "value1");  // Не перезапишет значение  map.putIfAbsent("key", "value2");  } | public static void correctTest() {  Map<String, String> map = new ConcurrentHashMap<>();  String s = map.putIfAbsent("key", "value1");  if (s == null) {  // Значение действительно установилось  s = "value1";  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Сохраните возвращаемое значение putIfAbsent() в переменную.** Это позволит вам определить, было ли ваше значение фактически добавлено в карту или же там уже существовало другое значение.
2. **Проверьте возвращаемое значение на null.** Если putIfAbsent() возвращает null, это означает, что переданное вами значение было успешно добавлено, так как ключа ранее не существовало в карте.
3. **Если возвращаемое значение не null, используйте его (существующее значение из карты) вместо переданного вами.** Это гарантирует, что вы работаете с тем значением, которое фактически связано с ключом в карте.
4. **Пересмотрите логику, если вам критично, какое именно значение используется.** В зависимости от вашей задачи, вам может потребоваться предпринять различные действия в случае, если значение уже присутствует в карте (например, использовать существующее, попытаться обновить его другим способом или сообщить об ошибке).
5. **Избегайте предположений о том, какое значение находится в карте, основываясь только на вызовах putIfAbsent() без проверки возвращаемого значения.** Состояние карты может быть изменено другими потоками в многопоточной среде.
6. **Убедитесь, что последующий код, который использует значение, связанное с ключом, оперирует с правильной ссылкой (либо на только что добавленное значение, либо на уже существовавшее).** Игнорирование возвращаемого значения putIfAbsent() может привести к использованию "потерянного" значения, которое не было фактически сохранено в карте для данного ключа.