NO\_BASE\_CALL.LIB

При переопределении метода не вызван метод базового класса, который должен быть вызван (либо потому, что остальные переопределения его вызывают, либо согласно документации). Но детектор с подкатегорией LIB проверяет библиотечные функции.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | Roslyn;SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Отсутствие вызова метода базового класса может нарушить работу библиотеки, привести к некорректной инициализации или освобождению ресурсов, а также к нарушению контракта библиотеки. |
| Комментарий | Рекомендуется внимательно изучать документацию библиотеки и всегда вызывать родительский метод, если это требуется для корректного функционирования библиотеки. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(HttpServletRequest req, HttpServletResponse resp) throws IOException {  class MyServlet extends HttpServlet {  @Override  protected void service(HttpServletRequest req, HttpServletResponse resp) throws IOException {  // Не вызван super.service — может сломать поведение HTTP методов (GET/POST)  resp.getWriter().write("...");  }  }  new MyServlet().service(req, resp);  } | public static void correctTest(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {  class MyServlet extends HttpServlet {  @Override  protected void service(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {  super.service(req, resp); // Вызываем родительский метод  // Обычно переопределяется doGet/doPost, а не service напрямую  }  @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {  resp.getWriter().write("...");  }  }  new MyServlet().service(req, resp);  } |

Методы устранения уязвимости при срабатывании детектора:

1. Изучите документацию переопределяемого метода базового класса (библиотечной функции). Определите, является ли вызов super.methodName() обязательным или рекомендованным для корректной работы библиотеки и вашего кода.
2. Если документация явно указывает на необходимость вызова метода базового класса, добавьте super.methodName(arguments) в начало или конец переопределенного метода. Порядок вызова может быть важен и также должен быть определен документацией.