Санкт-Петербургский государственный университет

Кафедра системного программирования

Статический анализ программ на платформе .NET

Бучин Вячеслав Андреевич

Научный руководитель: к.ф.-м.н. Д.А.Мордвинов, доцент кафедры системного программирования
Консультант: М.П.Костицын, ведущий разработчик ООО «ИИнтелКод»
Рецензент: В.Соболь, старший академический консультант ООО «КоулмэнТех»

Санкт-Петербург
2025

Статический анализ

- не требует запуска кода
- имеет широкую область применения
- сильно зависит от особенностей языка программирования
- ГОСТ Р 56939-2024

Анализ потока данных

Анализ потока данных — техника, позволяющая собирать информацию о возможном наборе значений, вычисляемых в различных точках программы

Анализ потока данных

Анализ потока данных — техника, позволяющая собирать информацию о возможном наборе значений, вычисляемых в различных точках программы

Taint-анализ:

- «загрязнения» данных
- source, sink
- существует эффективный алгоритм

Алгоритм IFDS

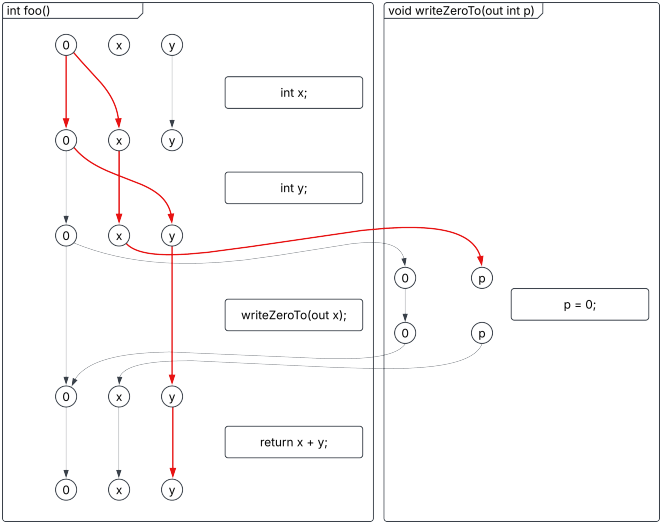
- факт — локация + метка
- вершины графа — факты $\times$ инструкции
- нулевые факты — достижимость
- ребра определяются flow-функциями

Алгоритм IFDS

```
void writeZeroTo(out int p) {
    p = 0;
}

int foo() {
    int x;
    int y;
    writeZeroTo(x);
    return x + y;
}
```

- метка — uninitialized



Цель и задачи

Целью данной работы является повышение безопасности ПО, разработанного на платформе .NET, с помощью обнаружения потенциальных уязвимостей статическим анализом во время разработки

Задачи

- провести обзор подходов анализа потока данных, а также инструментов статического анализа для платформы .NET
- спроектировать решение, учитывающее особенности платформы .NET
- реализовать спроектированное решение в инфраструктуре проекта USVM
- провести экспериментальное сравнение разработанной системы с другими инструментами статического анализа для .NET

Инструменты статического анализа .NET

- CodeQL
  - taint-анализ
  - промежуточное представление в виде БД
- Semgrep
  - taint-анализ
  - использует ИИ для фильтрации результатов

Особенности .NET

- структуры
- передача value-типов по ссылке
- виртуальные вызовы
- делегаты
- система типов
- небезопасный контекст

Flow-фукнции

$$[\![\text{return } x]\!](\langle z.g, \tau\rangle) = \{\langle z.g, \tau\rangle, \langle \text{return}, \tau\rangle\}, \text{если } x = z$$

$$[\![x \leftarrow y]\!](\langle z.g, \tau\rangle) = \begin{cases} \{\langle x.g, \tau\rangle, \langle y.g, \tau\rangle\}, \text{если } y = z \\ \emptyset, \text{если } x = z \\ \{\langle z.g, \tau\rangle\}, \text{иначе} \end{cases}$$

$$[\![x \leftarrow y.f]\!](\langle z.g, \tau\rangle) = \begin{cases} \{\langle x.h, \tau\rangle, \langle y.g, \tau\rangle\}, \text{если } y = z, g = fh \\ \emptyset, \text{если } x = z \\ \{\langle z.g, \tau\rangle\}, \text{иначе} \end{cases}$$

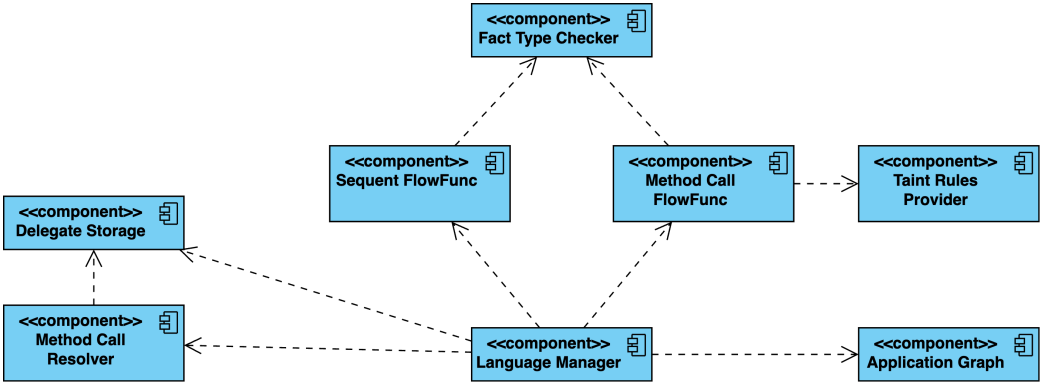
$$[\![x.f \leftarrow y]\!](\langle z.g, \tau\rangle) = \begin{cases} \{\langle x.fg, \tau\rangle, \langle y.g, \tau\rangle\}, \text{если } y = z \\ \emptyset, \text{если } x = z, g = fh \\ \{\langle z.g, \tau\rangle\}, \text{иначе} \end{cases}$$

$$[\![x = source^{\tau}()]\!](\mathbf{0}) = \{\langle x, \tau\rangle\}$$

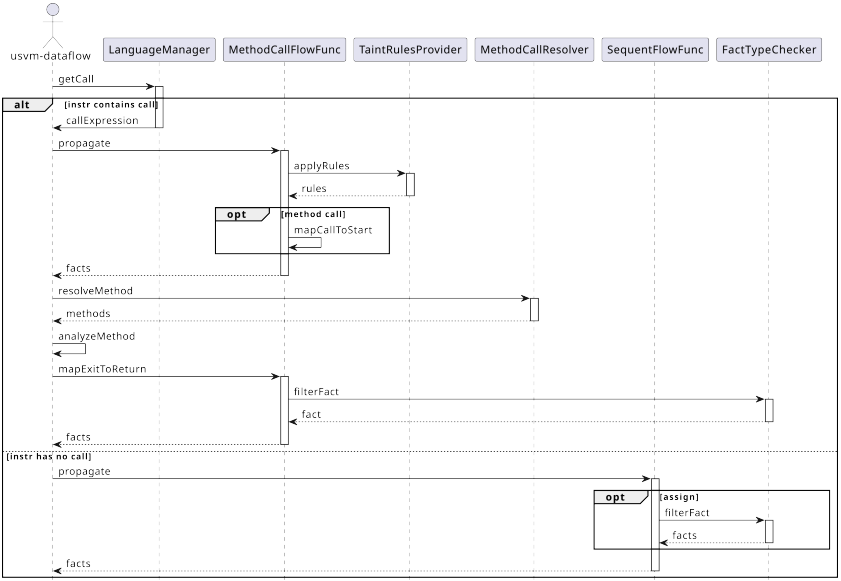
$$[\![x = c.m(y)]\!]_{call}(\langle z.g, \tau\rangle) = \begin{cases} \{\langle \text{this}.g, \tau\rangle\}, \text{если } c = z \\ \{\langle \text{arg}.g, \tau\rangle\}, \text{если } y = z \\ \{\langle \text{static}(A).g, \tau\rangle\}, \text{если } z = \text{static}(A) \\ \emptyset, \text{иначе} \end{cases}$$

$$[\![x = c.m(y)]\!]_{exit}(\langle z.g, \tau\rangle) = \begin{cases} \{\langle c.g, \tau\rangle\}, \text{если } z = \text{this} \\ \{\langle y.g, \tau\rangle\}, \text{если } z = \text{arg}, y = \text{ref}(r) \\ \{\langle x.g, \tau\rangle\}, \text{если } z = \text{return} \\ \{\langle \text{static}(A).g, \tau\rangle\}, \text{если } z = \text{static}(A) \\ \emptyset, \text{иначе} \end{cases}$$

Архитектура



Архитектура



Эксперимент

Метрики:

- precision
- recall
- false positive rate
- f1 score

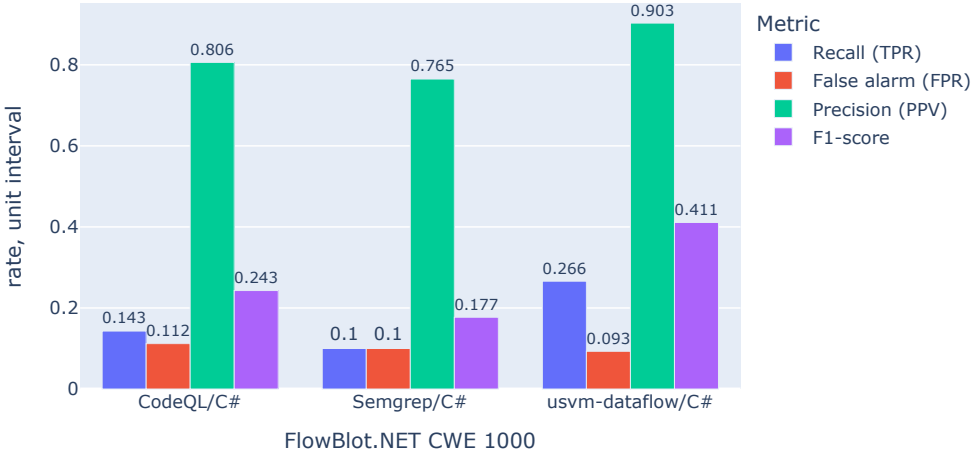
Данные:

- FlowBlot.NET

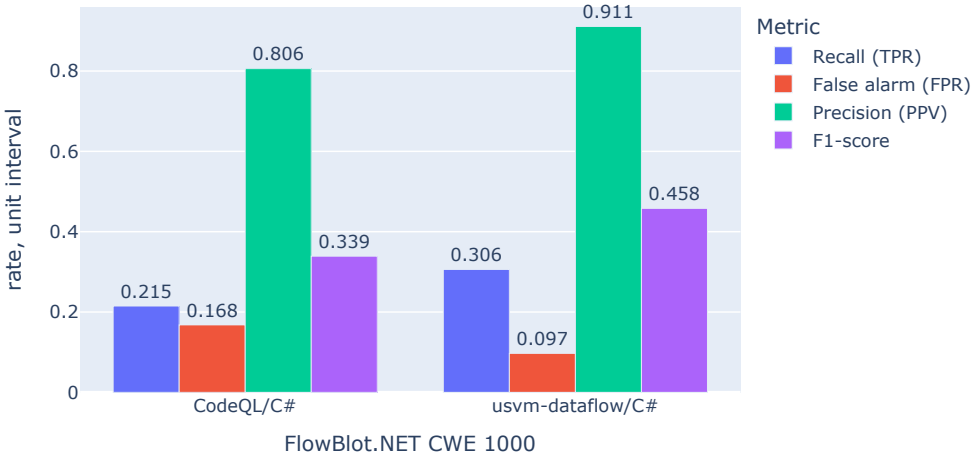
Инструменты:

- CodeQL
- Semgrep
- usvm-dataflow

Эксперимент



Результаты

- Проведен обзор анализа потока данных, а также следующих инструментов статического анализа для .NET: CodeQL, Semgrep
- Разработана архитектура системы статического анализа, учитывающая такие особенности платформы .NET, как структуры, передача аргументов value-типа по ссылке, виртуальные вызовы, делегаты
- Спроектированная система была реализована в инфраструктуре проекта USVM
- Проведено сравнительное исследование реализованной системы с CodeQL и Semgrep, показавшее, что разработанная система демонстрирует более высокие метрики качества

Эксперимент с учетом замечаний