**Сравнительный анализ методов обратного проектирования алгоритмов**

| **Критерий** | **Дизассемблирование** | **Декомпиляция** | **Анализ на основе поведения** | **Анализ на основе данных (data-flow)** |
| --- | --- | --- | --- | --- |
| **Описание** | Преобразование исполняемого кода в ассемблер | Преобразование байт-кода в псевдокод высокого уровня | Наблюдение за выполнением программы в реальном времени | Анализ перемещений и изменений данных в алгоритме |
| **Уровень абстракции** | Низкий | Средний | Высокий | Средний–высокий |
| **Точность восстановления алгоритма** | Высокая (при ручном анализе) | Средняя–высокая | Зависит от сценариев поведения | Высокая при наличии всех данных |
| **Требования к знаниям** | Знание ассемблера, архитектуры ЦП | Знание языков высокого уровня | Навыки анализа исполнения и мониторинга | Понимание логики обработки данных |
| **Инструменты** | IDA Pro, Ghidra (в режиме ASM), Radare2 | Ghidra, JD-GUI, JADX, ILSpy | Procmon, Wireshark, Sandbox'ы | Data Flow Graph tools, Taint Analysis |
| **Применение** | Анализ вредоносных программ, reverse engineering ПО | Восстановление бизнес-логики, изучение исходного кода | Анализ поведения вирусов и ПО | Анализ алгоритмов обработки данных |
| **Сложность автоматизации** | Высокая | Средняя | Средняя–низкая | Высокая |
| **Риски / ограничения** | Большие объёмы кода, ручная работа | Потеря точной логики, ошибки преобразования | Поверхностный анализ | Требует полной трассировки и данных |