**ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**им. Т.Г.ШЕВЧЕНКО**

**БЕНДЕРСКИЙ ПОЛИТЕХНИЧЕСКИЙ ФИЛИАЛ**

**Практическая работа №7**

по УП 03.01

Тема: «Обратное проектирование алгоритма»

Выполнил:

студент 3 курса, БК22АР52ИС2 группы,

**«**Информационные системы и программирования**»**

Николаев Валерий Евгеньевич

Проверил:

Лозовский А.В

2025 год.

**Обратное проектирование алгоритма** — это процесс анализа входных и выходных данных, а также поведения программы с целью восстановления логики работы алгоритма без доступа к его исходному коду. Этот метод часто используется в следующих сферах:

**1.** Анализ вредоносного ПО — изучение алгоритмов вирусов.

**2.** Восстановление утерянного кода — если исходный код недоступен, но есть исполняемый файл.

**3.** Исследование проприетарных алгоритмов — например, разбор алгоритмов сжатия или шифрования в закрытых программах.

**4.** Оптимизация и поиск уязвимостей — анализ кода конкурентов или проверка безопасности своих систем.

**Основные этапы обратного проектирования алгоритма**

**1. Сбор информации**

* Изучение входных и выходных данных.
* Наблюдение за поведением программы (например, с помощью отладчика).
* Анализ сетевого трафика (если алгоритм взаимодействует с сервером).

**2. Декомпиляция и дизассемблирование**

* Если есть исполняемый файл (EXE, ELF, Mach-O), можно использовать:

Дизассемблеры: IDA Pro, Ghidra, Binary Ninja.

Декомпиляторы: Ghidra, Hex-Rays (IDA), RetDec.

* Анализ ассемблерного кода и восстановление псевдокода.

**3. Динамический анализ**

* Запуск программы в отладчике (OllyDbg, x64dbg, GDB, WinDbg).
* Трассировка вызовов функций.
* Модификация памяти и регистров для проверки гипотез.

**4. Восстановление логики**

* Построение графа вызовов функций.
* Анализ структур данных (например, поиск хеш-таблиц, деревьев).
* Поиск математических операций (шифрование, хеширование, сжатие).

**5. Верификация алгоритма**

* Написание собственной реализации и сравнение результатов.
* Проверка на разных входных данных.

**Инструменты для Reverse Engineering**

|  |  |
| --- | --- |
| **Категория** | **Инструменты** |
| Дизассемблеры | IDA Pro, Ghidra, Binary Ninja |
| Декомпиляторы | Ghidra, Hex-Rays, RetDec |
| Отладчики | x64dbg, OllyDbg, GDB, WinDbg |
| Анализаторы | Radare2, Hopper, JADX (для Android) |
| Анализ трафика | Wireshark, Fiddler, Burp Suite |

**Проблемы и ограничения**

Обфускация кода (запутывание логики).

Антиотладочные техники (детект отладчика, VM-детекция).

Динамическое генерация кода (например, unpacking malware).

Сложные криптоалгоритмы (AES, RSA, собственные схемы).

**Вывод:**

Обратное проектирование алгоритмов требует глубоких знаний в низкоуровневом программировании, ассемблере и методах анализа ПО. Это мощный инструмент для анализа безопасности, восстановления кода и исследования конкурентных решений.