# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

# 1. Наименование

Reflective DLL injection - это технология, которая позволяет внедрить DLL в целевой процесс из памяти.

**2. Основание для разработки**

В системном программировании внедрение кода в целевой процесс играет важную роль для решения прикладных задач по поддержке существующих приложений. Процесс внедрения стороннего кода, ожидает запуска конкретного приложения и вводит дополнительный код в запущенный процесс, чтобы изменить или дополнить его функциональные возможности. Например: программная платформа сценариев, такая как «AutoHotkey», с помощью добавления небольшого кода, предоставляет пользователям возможность создавать сценарии, которые запускаются в фоновом режиме с помощью комбинаций, предварительно заданных, горячих клавиш. Для того чтобы внедрить dll в целевой процесс нужно изучить методы, которые различаются по технологическому процессу, исходным данным, формату полезной нагрузки, эффективности внедрения, типу работы в адресном пространстве целевого процесса.

Для того, чтобы выбрать наиболее эффективный и работоспособный метод, необходимо произвести исследование и осуществить анализ методов внедрения кода в целевой процесс.

**3. Исполнитель**

Студент группы ИУК5-42Б Фролов Кирилл Дмитриевич

**4. Цель разработки**

Целью курсовой работы является формирование практических навыков по разработке и реализации программного приложения с использованием интерфейса прикладного программирования (АРI) операционных систем.

Задачи проектирования:

1. овладение первичными навыками ведения научно-исследовательской, проектной и производственно-технологической деятельности, развитие творческих способностей индивидуально для каждого студента;
2. подготовка к выполнению выпускной квалификационной работы;
3. усвоение методов грамотного ведения, оформления и редактирования технической документации.

Целью разработки является исследование и анализ методов внедрения программного кода в целевой процесс для определения наиболее безопасных и актуальных методов.

**5. Содержание работы**

**5.1. Задачи, подлежащие решению:**

исследование существующих методов внедрения программного кода в целевой процесс;

реализация актуальных методов внедрения программного кода в целевой процесс;

проведение на практике реализованных методов;

1. анализ полученных результатов;
2. подготовка расчетно-пояснительной записки и графических листов;
3. подготовка презентации и речи для защиты курсовой работы;
4. защита курсовой работы.

**5.2. Требования к архитектуре АСОИ**

К архитектуре предъявляются следующие требования:

модульная архитектура программного обеспечения, в которой каждый модуль представляет собой один из методов внедрения программного кода;

каждый метод должен быть представлен в виде отдельного класса;

классы должны быть реализованы в виде иерархии классов, в которой классы, реализующие методы внедрения программного кода   
в сторонний процесс, наследуется от абстрактного виртуального класса.

**5.3. Требования к составу программных компонентов**

Программный комплекс должен состоять из следующих программных компонентов:

1. исполняемый файл PE формата с расширением .exe, реализующий методы внедрения программного кода в сторонний процесс;
2. полезная нагрузка в виде динамической библиотеки .dll

**5.4. Требования к прикладным программам**

Для работы программного комплекса необходимы:

1. Microsoft Windows 8/10;
2. Microsoft Visual Studio 2019;
3. минимальный набор драйверов, обеспечивающих   
   работоспособность ПК.

**5.5. Требования к входным/выходным данным**

Входные данные:

1. PID процесса, в который будет производится внедрение;
2. путь к полезной нагрузке.

Выходные данные:

1. всплывающее окно типа «MessageBox» с сообщением об успешном внедрении.

**5.6. Требования к временным характеристикам**

Требования к временным характеристикам программы не предъявляются

**5.7. Требования к составу технических средств**

Для функционирования системы необходимы:

* процессор: 1 ГГц и быстрее с поддержкой PAE, NX и SSE2;
* RAM: 1 Гбайт (32 бит) или 2 Гбайт (64 бит);
* HDD: 500 Мбайт (32 бит) или 700 Мбайт (64 бит);
* видеокарта: поддержка Microsoft DirectX 9 с драйвером WDDM;
* монитор;
* клавиатура;
* мышь.

**6. Этапы разработки**

исследование существующих методов внедрения программного кода в целевой процесс;

реализация актуальных методов внедрения программного кода в сторонний процесс;

Проверка на практике реализованных методов

1. анализ полученных результатов.
2. Создание интерфейса

**7. Техническая документация, предъявляемая по окончании работы**

По окончанию работы предъявлена расчетно-пояснительная записка в состав которой входят:

* техническое задание;
* научно-исследовательская часть;
* проектно-конструкторская часть;
* проектно-технологическая часть.

Также должна быть предоставлена графическая часть работы, выполненная формате А1 на 2 листах, в которую входят:

* демонстрационные чертежи;
* алгоритмические схемы.

**8. Дополнительные условия**

язык программирования С/С++;

использование Windows API функций для реализации методов внедрения программного кода в сторонний процесс;

среда разработки Visual Studio 2019;

тип приложения – оконное;

интерфейс должен предоставлять пользователю текстовые поля для ввода PID в который необходимо внедриться и путь к полезной нагрузке, и кнопку “внедриться”. По завершении работы должен появиться MessegeBox с информацией что внедрение произошло успешно