Защита от отладки

Основными инструментами для исследования программ являются дисассемблеры и отладчики.

- Дизассемблирование - это получение из исполняемого кода программы код на языке ассемблера.
- Дизассемблер - программа, осуществляющая дисассемблирование.
- Интерактивный дизассемблер - программа, тесно взаимодействующая с пользователем в процессе дисассемблирования.
- Отладчик - программа, предназначенная для анализа поведения другой программы, обеспечивающая остановку в указанных точках и позволяющая просматривать (редактировать) содержимое ячеек памяти, регистров процессора и команды программы.
- Эмулирующий отладчик - отладчик, который самостоятельно интерпретирует и выполняет команды программы (без использования реального процессора).

Существует также множество программ-утилит, предназначенных для вспомогательных операций по изучению логики работы механизма защиты. Широко используются

- Шестнадцатеричные просмотрщики - редакторы;
- Редакторы таблиц экспорта/импорта;
- Файловые мониторы, позволяющие отслеживать операции работы с файлами;
- Мониторы реестра, создающие протокол обращений к реестру;
- И многие другие.

Универсальным методом противодействия дизассемблированию программы является шифрование.

Рекомендуется использовать шифрование с открытым ключом (алгоритм RSA, шифр Эль-Гамаля и др.)

Также усилению защиты способствует динамическое шифрование.

Широко распространены на практике методы, основанные на динамическом изменении кода программы в процессе выполнения.

Предлагают различные преобразования, например,

- Перемещения участков кода;
- Всевозможные функции от истинного кода (контрольной суммы истинного кода);
- Или для генерации кода одного участка используют коды предыдущего (или какого-нибудь другого) участка программы (так называемая обратная связь).

Существует два отладочных механизма

1. Контрольные точки останова.
2. Трассировка программы.

Для обнаружения модифицированного кода традиционно применялись следующие методы:

- Подсчет контрольных сумм критических участков
- Использование контрольной суммы всего кода для расшифровки некоторого фрагмента
- Многопроходная расшифровка кода с ключом, вычисляемым на основе контрольной суммы всего кода либо критического участка
- Использование корректирующих кодов, позволяющих определить местоположение контрольного байта
- Контроль времени выполнения критического участка по сравнению с эталонным временем
- контроль относительного времени выполнения участка программы (относительно другого участка)
- и другие