

Принцип работы CodeProtect.exe

Защита кода позволяет автоматически убрать его часть из защищенного приложения и обращаться к нему только по требованию в момент выполнения. Любое обращение к защищенному хранилищу кода инициирует сначала проверку наличия электронного ключа, и лишь при его соответствии тому, на котором была произведена защита приложения, производится дальнейшее расшифровывание защищенного кода.

Код каждого метода из защищенного приложения имеет свою уникальную и независимую от других участков кода привязку к электронному ключу. Такой подход существенно затрудняет обратный анализ и реализует уникальную последовательность обращений, опирающуюся на граф потока выполнения защищаемого приложения.

Защита EXE

Защищенный исполняемый файл остается **Managed-сборкой**, но большая часть его кода переносится в **Native-DLL** специального вида, которая представляет собой защищенное хранилище кода:

1. В процессе выполнения происходит вызов кода защищенного метода, после чего вызывается код заглушки.
2. Заглушка инициирует выполнение кода из защищенного хранилища относительно вызванного метода.
3. Происходит выполнение защищенного кода.

Защита DLL

При защите динамических **DLL-библиотек** для платформы **.NET** использован аналогичный подход, что и при защите исполняемых файлов, но с учетом специфики использования библиотечного кода. В процессе защиты из оригинальной **DLL-библиотеки** удаляется **MSIL-код**, и на его место вставляются вызовы специальных методов-заклушек. Т. о., после защиты от оригинальной **DLL-библиотеки** остается только код с вызовом методов-заклушек. Данный подход к защите позволяет сохранить работоспособность принципов взаимодействия с защищенной библиотекой: на нее по-прежнему можно добавлять ссылки (**references**) из других **.NET** сборок, и она по-прежнему имеет статус **Managed-сборки**. После защиты **DLL** сборки нет необходимости вносить какие-либо изменения в код, который уже использует защищенную библиотеку, так как методы-заклушки возьмут на себя всю работу по перенаправлению вызовов к защищенному хранилищу кода.

1. Исполняемый файл обращается к методу из защищенной библиотеки
2. После обращения защищенному методу вызывается код заглушки
3. Заглушка инициирует выполнение кода из защищенного хранилища относительно вызванного метода
4. Происходит выполнение защищенного кода.