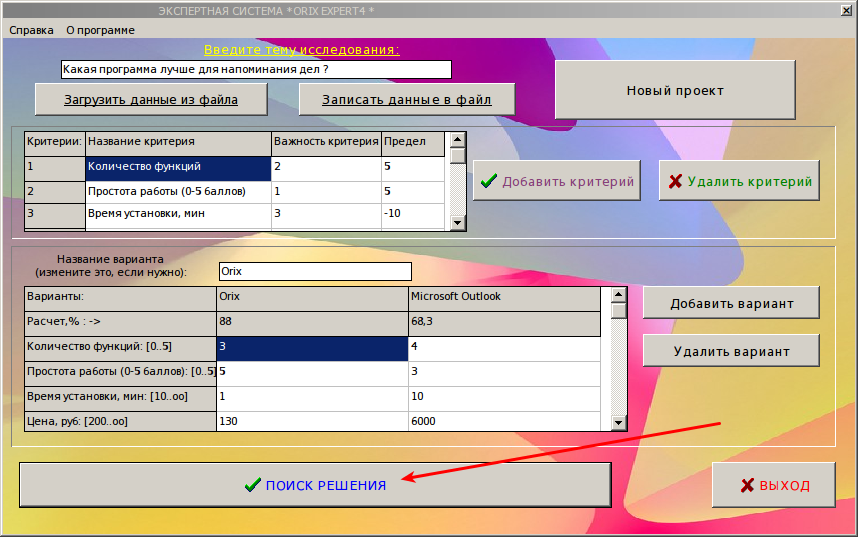
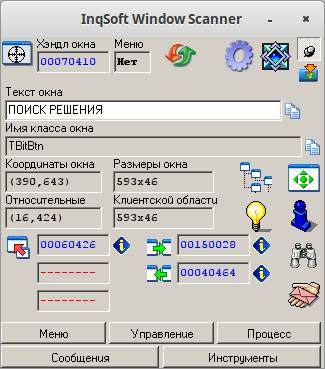
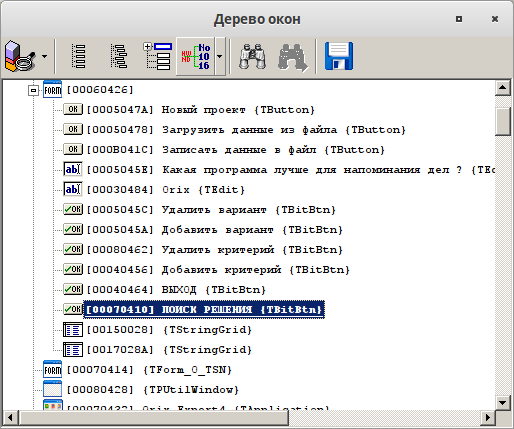
# ЛЕКЦИЯ 15. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РЕВЕРС-ИНЖЕНЕРИНГА ПРОГРАММ

Обратная разработка (обратный инжиниринг, реверс-инжиниринг) - исследование некоторого готового устройства или программы, а также документации на него с целью понять принцип его работы; например, чтобы обнаружить недокументированные возможности (в том числе программные закладки) или извлечь секретную информацию, сделать изменение, или воспроизвести устройство, программу или иной объект с аналогичными функциями, но без копирования как такового.

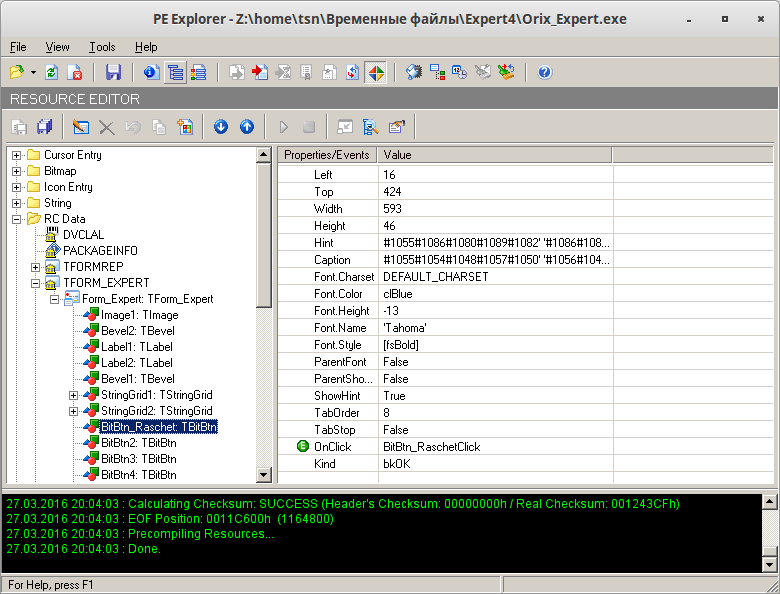
Рассмотрим пример извлечения алгоритма программы в виде кода на ассемблере из ее программного кода (exe-файла) на примере одной авторской программы.

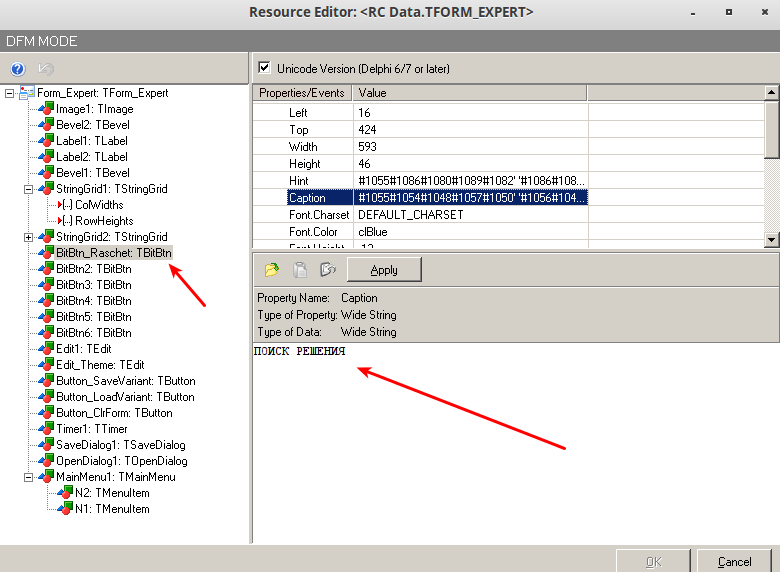


15.1. Сканеры окон программ

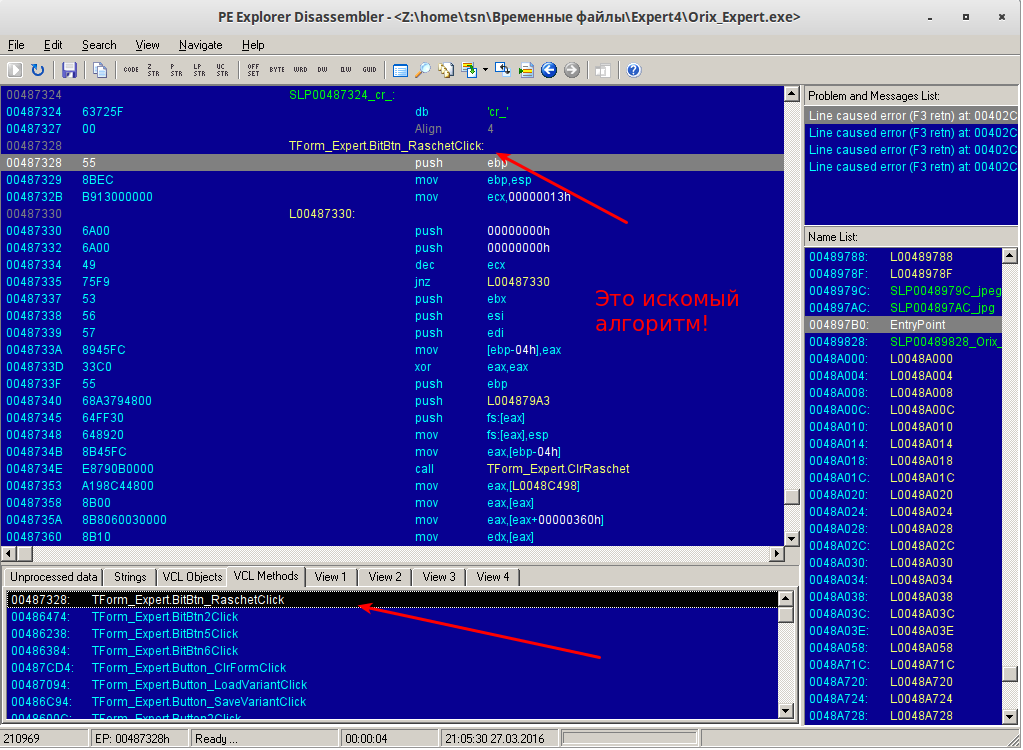
С помощью сканера окон определяется внутренний тип взламываемого элемента программы, в данном случае это кнопка «ПОИСК РЕШЕНИЯ». Сканер определил, что тип искомого объекта «TBitBtn».

15.2. Редакторы ресурсов и декомпиляторы

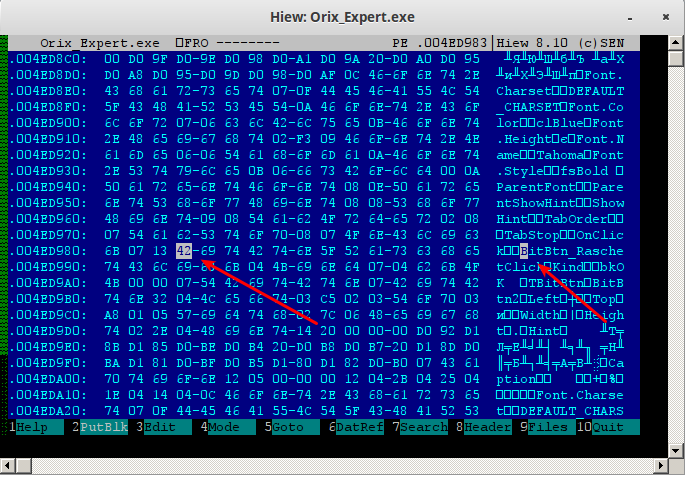
С помощью редактора ресурсов находится полное имя искомого элемента: «BitBtn\_Rascet». В данном редакторе злоумышленник может поменять надпись на кнопке, ее видимость и др.

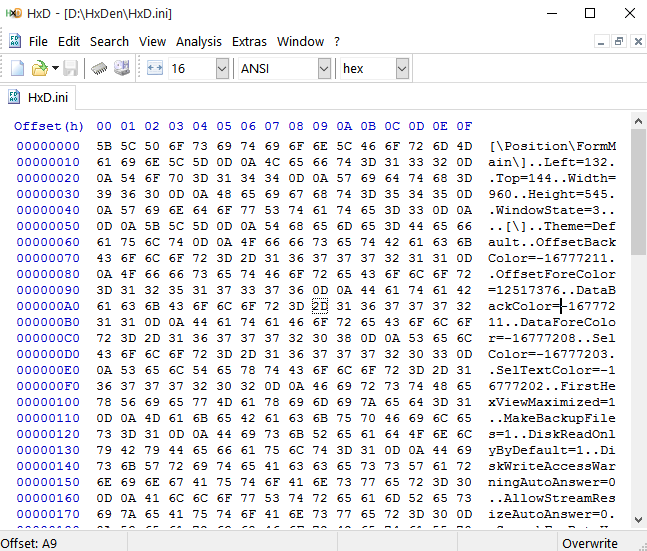


Далее с помощью декомпилятора находится исполняемый код для кнопки «BitBtn\_Rascet»…



15.3. Редакторы бинарного кода

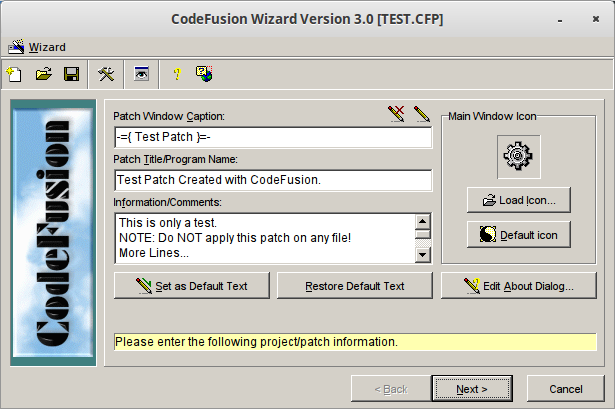
С помощью редактора бинарного кода злоумышленник может не только увидеть данные программы, но и поменять несколько нужных байт в программе, произведя тем самым ее «взлом».



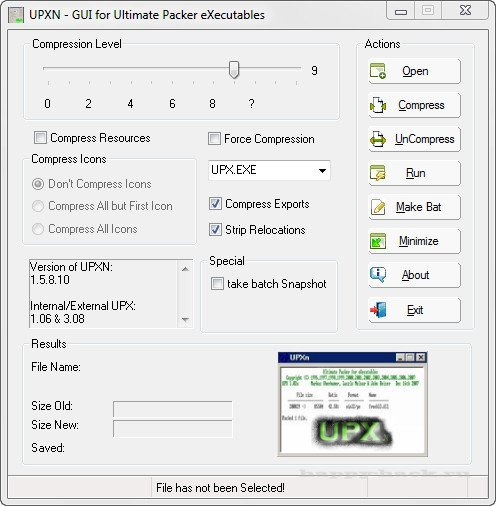
15.4. Патчеры

С помощью специальных утилит-патчеров хакер может сделать небольшую программу, которая сможет автоматически взламывать данную программу у других пользователей.

Данные утилиты используются также в «мирных» целях разработчиками ПО для обновления их программ.



15.5. Динамические компрессоры программного кода

UPX (the Ultimate Packer for eXecutables) - упаковщик исполняемых файлов, поддерживающий несколько различных платформ и форматов файлов. Это свободное и открытое программное обеспечение, распространяемое по лицензии GNU GPL.

Сжатый исполняемый файл состоит из модуля распаковки/инициализации и блока данных, содержащего в себе исходный файл в сжатом виде. При запуске модуль распаковки выделяет память и распаковывает туда содержимое блока данных. Для некоторых платформ распаковка в память невозможна, в этом случае используется распаковка во временный файл.

15.6. Обфускаторы

Обфускация (от лат. obfuscare - затенять, затемнять; и англ. obfuscate - делать неочевидным, запутанным, сбивать с толку) или запутывание кода - приведение исходного текста или исполняемого кода программы к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции.

«Запутывание» кода может осуществляться на уровне алгоритма, исходного текста и/или ассемблерного текста. Для создания запутанного ассемблерного текста могут использоваться специализированные компиляторы, использующие неочевидные или недокументированные возможности среды исполнения программы.

Существуют специальные программы, производящие обфускацию, называемые обфускаторами (англ. obfuscator). Рассмотрим пример обфусификации:

Исходный текст:

int COUNT = 100; float TAX\_RATE = 0.2;

for (int i=0; i<COUNT; i++)

{

tax[i] = orig\_price[i] \* TAX\_RATE;

price[i] = orig\_price[i] + tax[i];

}

Код после обфускации:

for(int a=0;a<100;a++){b[a]=c[a]\*0.2;d[a]=c[a]+b[a];}

