# ЛАБОРАТОРНАЯ РАБОТА №8

# ИССЛЕДОВАНИЕ БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

# ИНФОРМАЦИОННЫХ СИСТЕМ В СРЕДЕ ОТЛАДЧИКА OLLYDBG

Цель работы

Углубление знаний архитектуры 32-разрядных процессоров и системы команд языка ассемблера. Исследование методов защиты программного обеспечения информационных систем и ее нейтрализации, приобретение практических навыков исследования и отладки программ с помощью пакета OllyDbg.

Задачи

1. Повторить теоретический материал, касающийся архитектуры 32-разрядных микропроцессоров, программно доступных регистров и системы команд языка ассемблера;
2. Исследовать способы парольной защиты в программе CRAСKME1.EXE. Для этого выполнить последовательность действий, описных методических указаниях. Изменить программу таким образом, чтобы принимался любой вводимый пароль, независимо от того, верный он или неверный;
3. С помощью отладчика OllyDbg исследовать способы парольной защиты программ CRAСKME2.EXE, CRAСKME3.EXE и CRAСKME4.EXE, которые расположены в папке лабораторных работ. Определить на каких языках написаны программы. Изменить программы таким образом, чтобы принимался любой вводимый пароль, независимо от того, верный он или неверный;
4. С помощью отладчика OllyDbg исследовать способ защиты программы CRAСKME5.EXE. Определите на каком языке написана программа. В данной программе ключ генерируется по введённому в первом поле имени;
5. Разработать рекомендации по усилению защиты вскрытия пароля;

Ход работы

Была открыта программа CRACKME1.exe. При попытке ввести пароль программа проверила его и выдала сообщение о том, что он неверный (Рисунок 1).

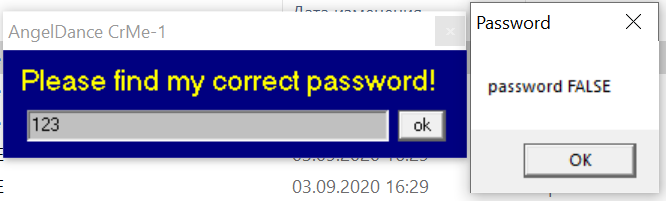


Рисунок 1 – CRACKME1 до взлома (неверный пароль)

С помощью анализатора исполняемых файлов PEiD было определено, что CRACKME1 написана на языке C++ фирмы Borland и ничем не зашифрована и не запакована.

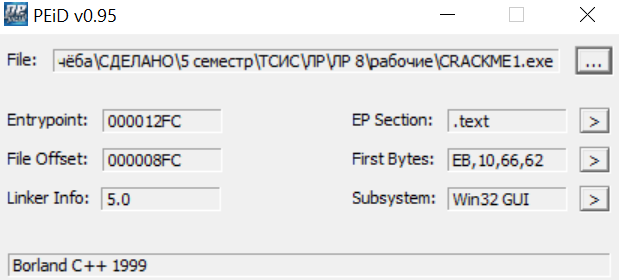


Рисунок 2 – Анализ файла CRACKME1.exe

Программа была открыта в отладчике OllyDbg. В нём были найдены блоки кода, в которых происходит вывод сообщений о правильности/неправильности пароля (Рисунок 3). Было замечено, что переход к сообщению о неправильном пароле осуществляется через проверку JZ SHORT 00401C77 (ячейка 00401C58).

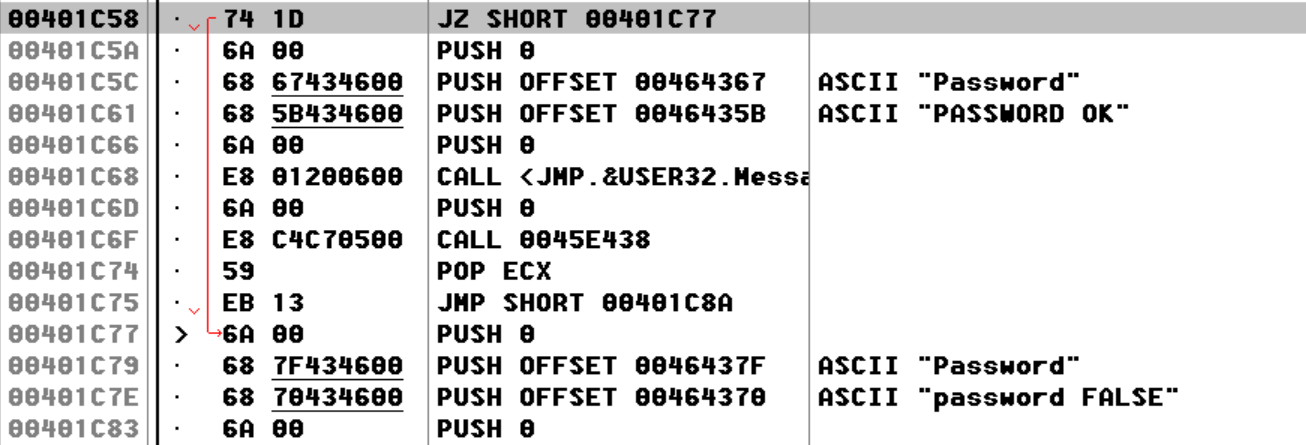


Рисунок 3 – CRACKME1 в отладчике

JZ осуществляет переход относительно результата некоторой проверки. Так была найдена написанная выше команда CMP (0401C0D). На ней была поставлена точка останова. При трассировании программы отладчик проанализировал, что она сравнивает строку со значением «S4K6n37fE».

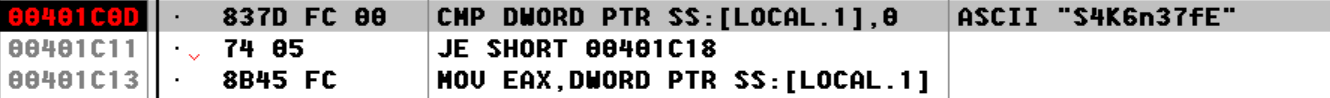


Рисунок 4 – Функция проверки строки

При попытке ввести эту строку в CRACKME1 выяснилось, что она действительно являтся паролём программы (Рисунок 5).

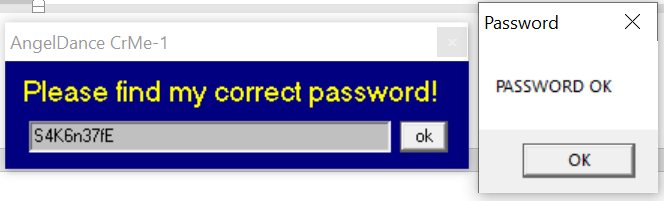


Рисунок 5 – Верный пароль от CRACKME1

Проверка JZ, совершающая переход на вывод сообщения о неверном пароле, была заменена на NOP. Переписанная программа была сохранена в исполняемом файле. Теперь она считает верным любой пароль (Рисунок 6).

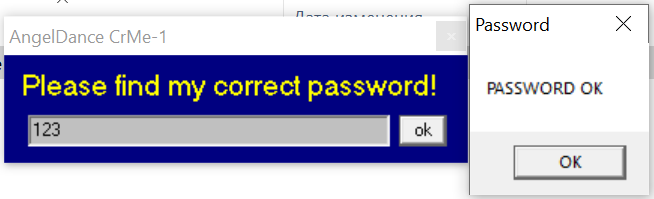


Рисунок 6 – CRACKME после взлома (верный пароль)

Была открыта программа CRACKME2.exe (Рисунок 7). Она также проверяет правильность пароля и написана на Borland C++.

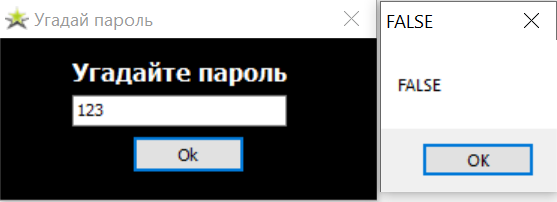


Рисунок 7 – CRACKME 2 до взлома (неверный пароль)

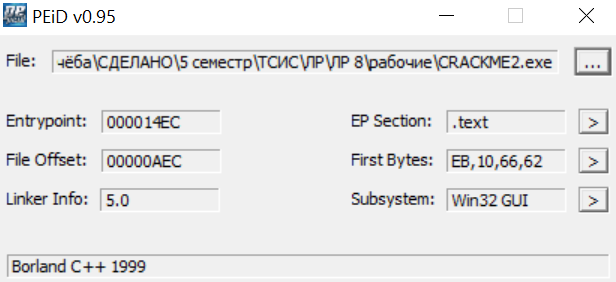


Рисунок 8 – Анализ файла CRACKME2.exe

CRACKME2 была открыта в отладчике. Через поиск по ключевому слову ‘FALSE’ был найден блок вывода сообщения об ошибке. Здесь проверку также осуществляет команда JZ, однако переход в блок ошибки осуществляется, наоборот, при Z=0, поэтому проверка JZ была замена на безусловный переход JMP (Рисунок 9).

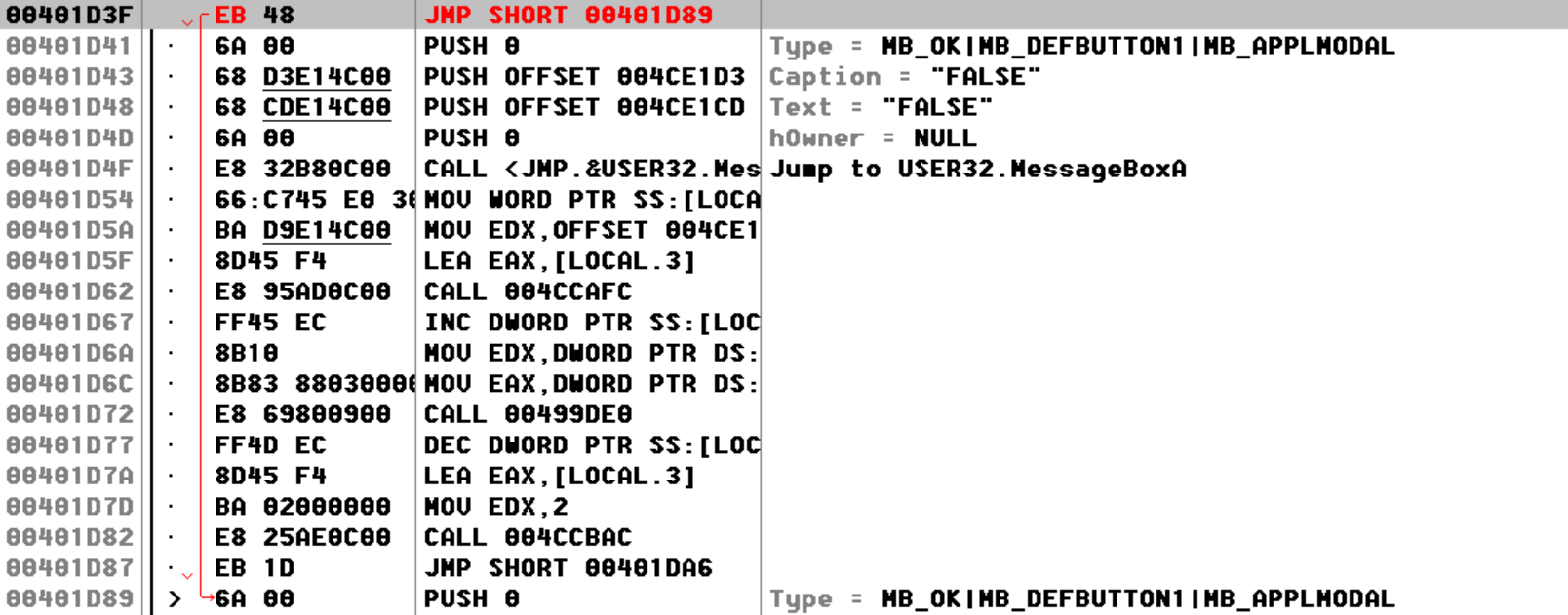


Рисунок 9 – Переписанный код CRACKME2.exe

При трассировании обновлённой программы верным стал считаться любой пароль (Рисунок 10). Также в коде программы был найден истинно верный пароль – Pass123 (ячейка 00401D0A).

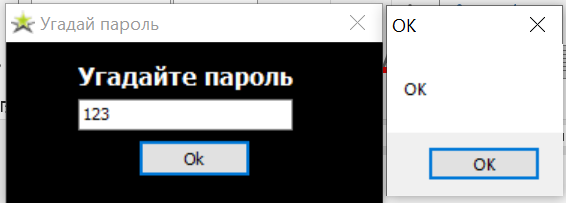


Рисунок 10 – CRACKME2 после взлома (неверный пароль)

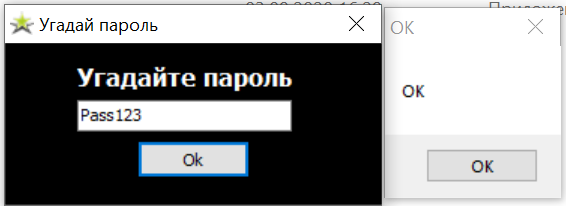


Рисунок 11 – Верный пароль от CRACKME2

Файл CRACKME3.exe оказался более сложным для анализа. Через анализатор Exeinfo PE было выяснено, что он написан на языке Delphi компании Borland (Рисунок 13).

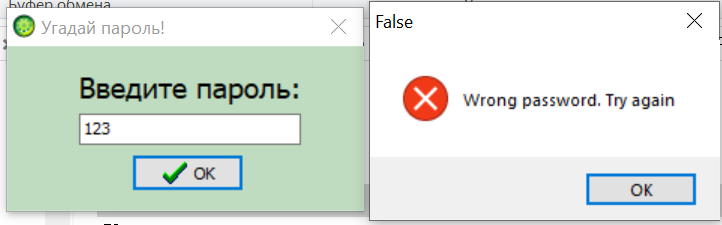


Рисунок 12 – CRACKME3 до взлома (неверный пароль)



Рисунок 13 – Анализ файла CRACKME3.exe

Программа была открыта в OllyDbg. По ключевому слову “Wrong” был найден блок кода, где проверяется пароль. Выше него в ячейке 004AAB55 был найден верный пароль программы − «Dh789rTyU78» (Рисунок 14).

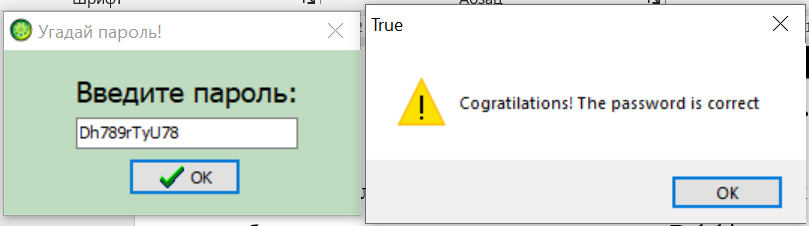


Рисунок 14 – Верный пароль от CRACKME3

Инструкция JNZ, осуществляющая переход в блок вывода сообщения о неверном пароле в случае отрицательного результата проверки, была заменена на NOP (Рисунок 15). Полученная в результате программа считает верным любой пароль (Рисунок 16).

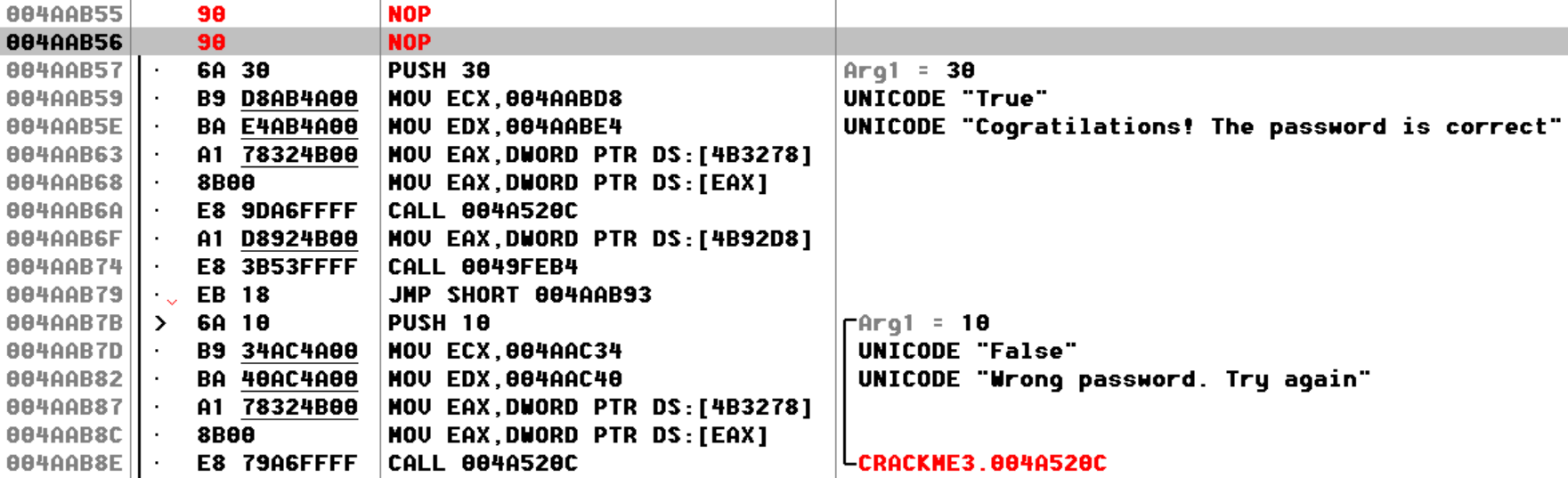


Рисунок 16 – Переписанный код CRACKME3.exe

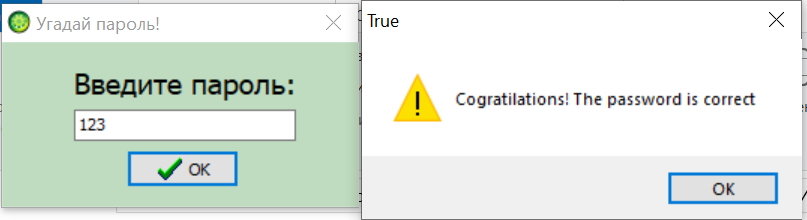


Рисунок 17 – CRACKME3 после взлома (неверный пароль)

Программа CRACKME4 написана на ассемблере и выполняет ту же функцию, что и предыдущие.

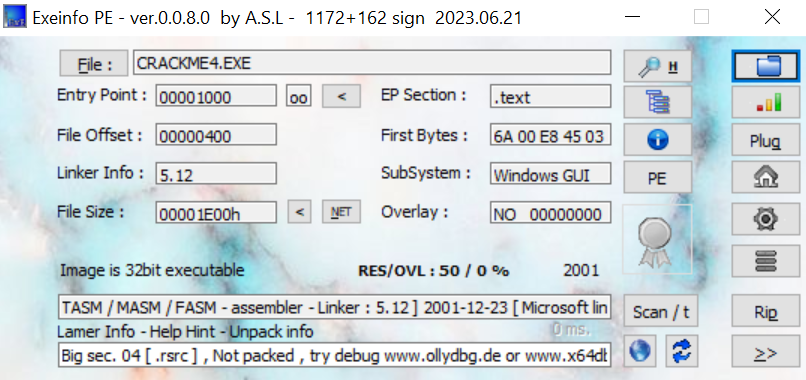


Рисунок 18 – Анализ файла CRACKME4.exe

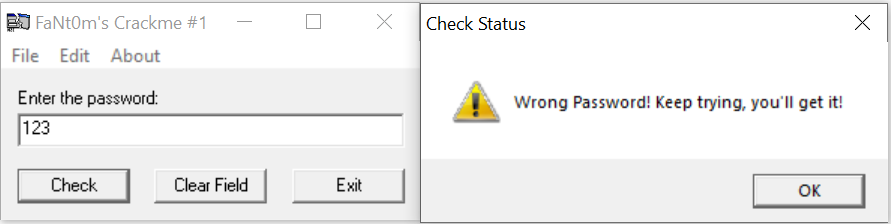


Рисунок 19 – CRACKME4 до взлома (неверный пароль)

Программа была открыта OllyDbg. В её коде был найден истинный пароль – «m0tNaF-EmKCARc» (Рисунок 20). Для того, чтобы программа воспринимала верным любой пароль, условный переход JE после операции CMP был заменён на безусловный JMP (Рисунок 21).

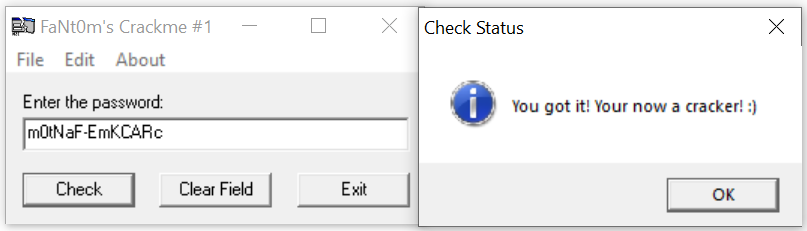


Рисунок 20 – Верный пароль от CRACKME4

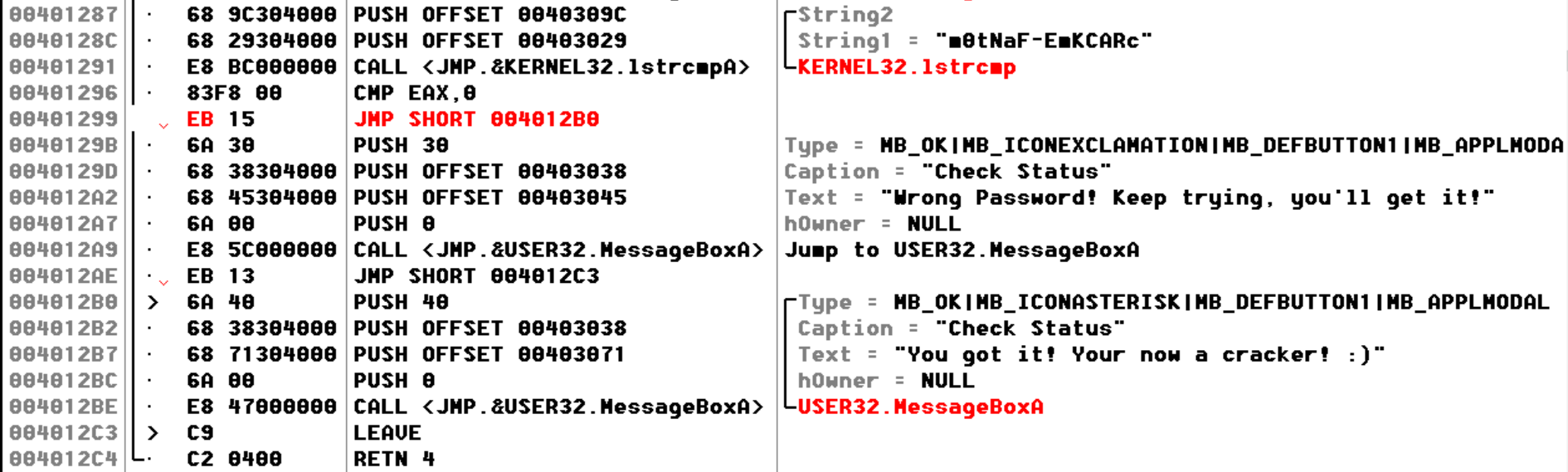


Рисунок 21 – Переписанный код CRACKME4.exe

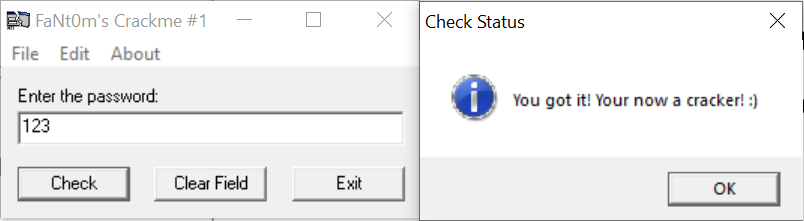


Рисунок 22 – CRACKME4 после взлома (неверный пароль)

Программа CRACKME5 отличается от прочих. Она также написана на ассемблере, однако верный пароль формируется относительно ключа – введённого пользователем имени.

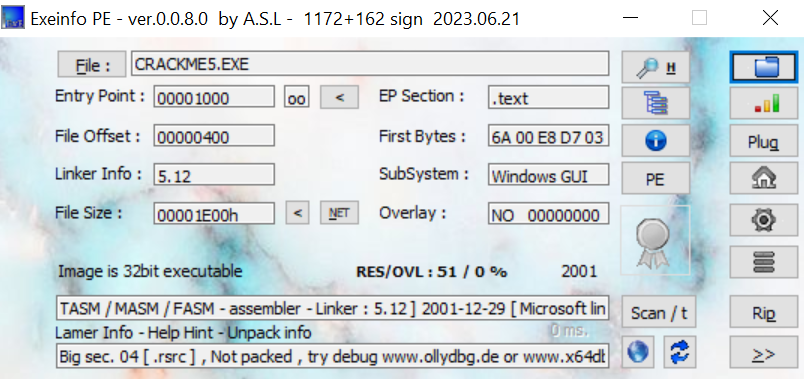


Рисунок 23 – Анализ файла CRACKME5.exe

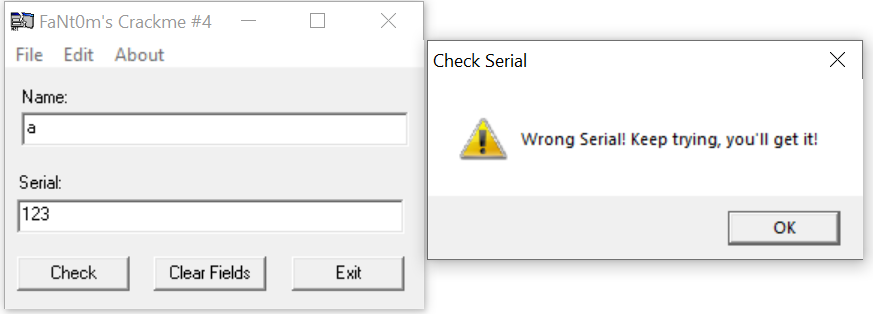


Рисунок 24 – CRACKME5 до взлома (неверный пароль)

Программа была открыта в OllyDbg. В ходе её анализа был найден цикл формирования верного пароля путём шифрования заданного ключа (Рисунок 25). Через просмотр значений регистров было определено, что для ключа “a” (лат.) паролём будет символ T.



Рисунок 25 – Цикл шифрования строки

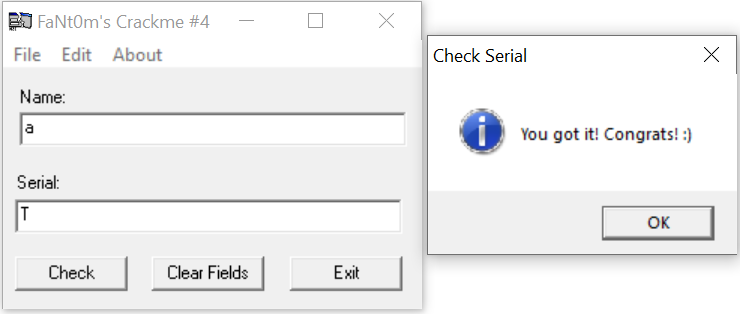


Рисунок 26 – Пример ввода верного пароля CRACKME5

Для того, чтобы программа считала любой пароль верным, вызов функции шифрования и проверки пароля (CALL 00401298) был заменён на инструкцию MOV EAX, 1. Таким образом проверка пароля всегда выдаёт верный результат (Рисунок 28).

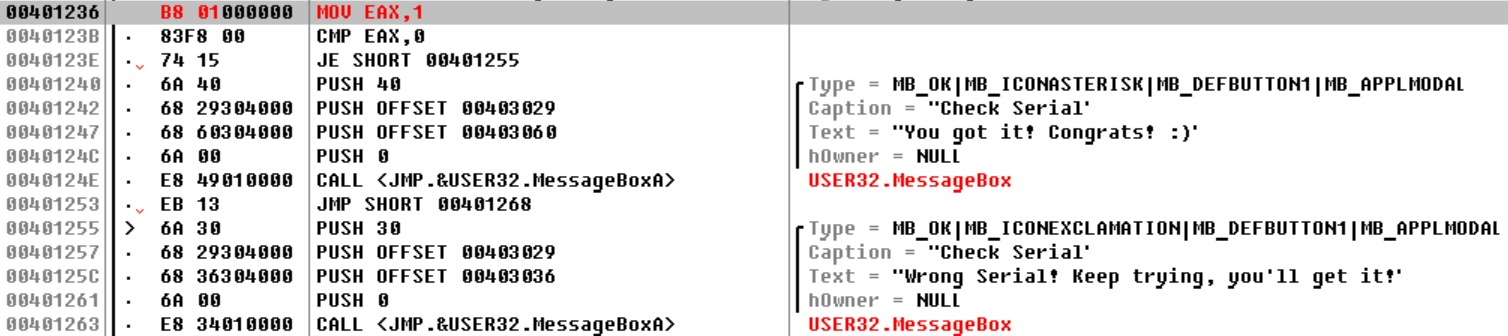


Рисунок 27 – Переписанный код CRACKME5.exe

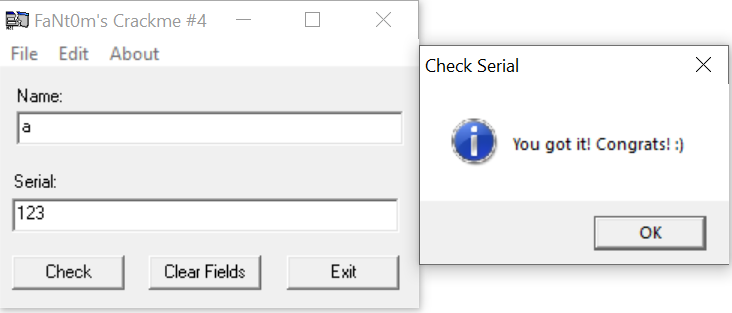


Рисунок 28 – CRACKME5 после взлома (неверный пароль)

Вывод

В ходе работы были углублены знания архитектуры 32-разрядных процессоров и системы команд языка ассемблера. Исследованы методы защиты программного обеспечения информационных систем и её нейтрализации, приобретены практические навыки исследования и отладки программ с помощью пакета OllyDbg.