

Внедрение дополнительного кода в исполняемый файл

1. Модификация заголовка и структуры исполняемого кода
2. Вставка дополнительного кода

Требуемый инструментарий

ОТЛАДЧИК ИСПОЛНЯЕМОГО КОДА
OLLY DEBUGGER

РЕДАКТОР ЗАГОЛОВКА ИСПОЛНЯЕМОГО ФАЙЛА ФОРМАТА
PE
LORD PE, PE TOOLS И ДР.

ШЕСТНАДЦАТИРИЧНЫЙ РЕДАКТОР
FLEX HEX, WIN HEX И ДР.

ОБЪЕКТ ВОЗДЕЙСТВИЯ
ИСПОЛНЯЕМЫЙ ФАЙЛ

Алгоритм действий

1. Создаем проект в среде программирования с пустой графической формой.
2. Внедряем свой код в созданный проект. Код должен исполняться при запуске графического пользовательского приложения (пустая форма).

Варианты внедрения кода

Открытие созданного приложения в Olly Debugger.

The screenshot shows the OllyDbg interface with the following components:

- Assembly Window:** Displays assembly instructions for the CPU - main thread, module target. The instruction at address 0044CA98 is highlighted in yellow. The instruction at 0044D068 is also highlighted in yellow.
- Registers (FPU) Window:** Shows the state of various registers, including EAX (00000000), ECX (0012FFB0), EDX (7C90EB94), EBX (7FFD9000), ESP (0012FFC4), EBP (0012FFF0), ESI (FFFFFFFF), EDI (7C910738), and EIP (0044CA98).
- Hex Dump Window:** Shows the memory dump at address 0044D000, with the instruction at 0044D068 highlighted in yellow.
- Command Window:** Shows the program entry point and the current command.

| Address | Hex dump | ASCII | Comment |
|--------------|-------------------------|------------|--|
| 0044CA98 | 55 | | PUSH EBP |
| 0044CA99 | 8BEC | | MOV EBP, ESP |
| 0044CA9B | 83C4 F0 | | ADD ESP, -10 |
| 0044CA9E | B8 B8C84400 | | MOV EAX, target.0044C8B8 |
| 0044CAA3 | E8 2091FBFF | | CALL target.00405BC8 |
| 0044CAAB | A1 B8DF4400 | | MOV EAX, DWORD PTR DS:[44DFB8] |
| 0044CAAD | 8B00 | | MOV EAX, DWORD PTR DS:[EAX] |
| 0044CAAF | E8 9CE6FFFF | | CALL target.0044B150 |
| 0044CAB4 | 8B0D 94E04400 | | MOV ECX, DWORD PTR DS:[44E094] target.0044FB00 |
| 0044CABA | A1 B8DF4400 | | MOV EAX, DWORD PTR DS:[44DFB8] |
| 0044CABF | 8B00 | | MOV EAX, DWORD PTR DS:[EAX] |
| 0044CAC1 | 8B15 F0C64400 | | MOV EDX, DWORD PTR DS:[44C6F0] target.0044C73C |
| 0044CAC7 | E8 9CE6FFFF | | CALL target.0044B168 |
| 0044CACC | A1 B8DF4400 | | MOV EAX, DWORD PTR DS:[44DFB8] |
| 0044CAD1 | 8B00 | | MOV EAX, DWORD PTR DS:[EAX] |
| 0044CAD3 | E8 10E7FFFF | | CALL target.0044B1E8 |
| 0044CAD8 | E8 4372FBFF | | CALL target.00403D20 |
| 0044CAD9 | 8D40 00 | | LEA EAX, DWORD PTR DS:[EAX] |
| EBP=0012FFF0 | | | |
| 0044D000 | 00 00 00 00 00 00 00 00 | | |
| 0044D008 | 02 8D 40 00 00 00 00 00 | ък@..... | |
| 0044D010 | 00 00 00 00 00 00 00 00 | | |
| 0044D018 | 00 00 00 00 00 00 00 00 | | |
| 0044D020 | 32 13 8B C0 02 00 8B C0 | 2Ъ&АЪ.<А | |
| 0044D028 | 00 8D 40 00 00 8D 40 00 | .к@.к@. | |
| 0044D030 | 00 8D 40 00 00 00 00 00 | .к@..... | |
| 0044D038 | 00 00 00 00 E8 20 40 00 | ...и @. | |
| 0044D040 | 78 22 40 00 F8 25 40 00 | х" @.ш% @. | |
| 0044D048 | 00 CB CC C8 C9 D7 CF C8 | .пмийчпи | |
| 0044D050 | CD CE DB D8 DA D9 CA DC | Нобъшъщкь | |
| 0044D058 | DD DE DF E0 E1 E3 00 E4 | эюяабг.д | |
| 0044D060 | E5 8D 40 00 45 72 72 6F | еК@.Erro | |
| 0044D068 | 72 00 8B C0 52 75 6E 74 | р.<Аrunт | |
| 0012FFC4 | 7C816D4F | | RETURN to kernel32.7C816D4F |
| 0012FFC8 | 7C910738 | | ntdll.7C910738 |
| 0012FFCC | FFFFFFFF | | |
| 0012FFD0 | 7FFD9000 | | |
| 0012FFD4 | 805522FA | | |
| 0012FFD8 | 0012FFC8 | | |
| 0012FFDC | 81ABBDAB | | |
| 0012FFE0 | FFFFFFFF | | End of SEH chain |
| 0012FFE4 | 7C8399F3 | | SE handler |
| 0012FFE8 | 7C816D58 | | kernel32.7C816D58 |
| 0012FFEC | 00000000 | | |
| 0012FFF0 | 00000000 | | |
| 0012FFF4 | 00000000 | | |
| 0012FFF8 | 0044CA98 | | target.<ModuleEntryPoint> |
| 0012FFFC | 00000000 | | |

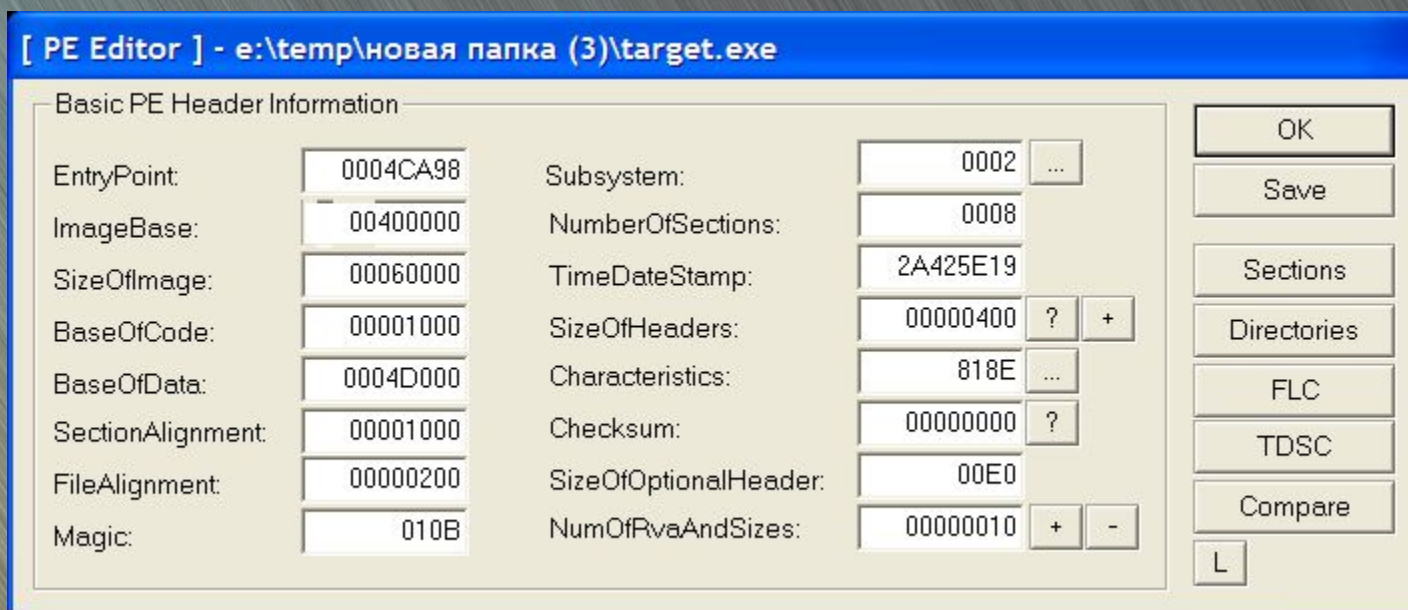
Варианты внедрения кода

1. Запись кода в память процесса.
2. Запись кода в новую секцию (данных или кода), которая создается дополнительно.

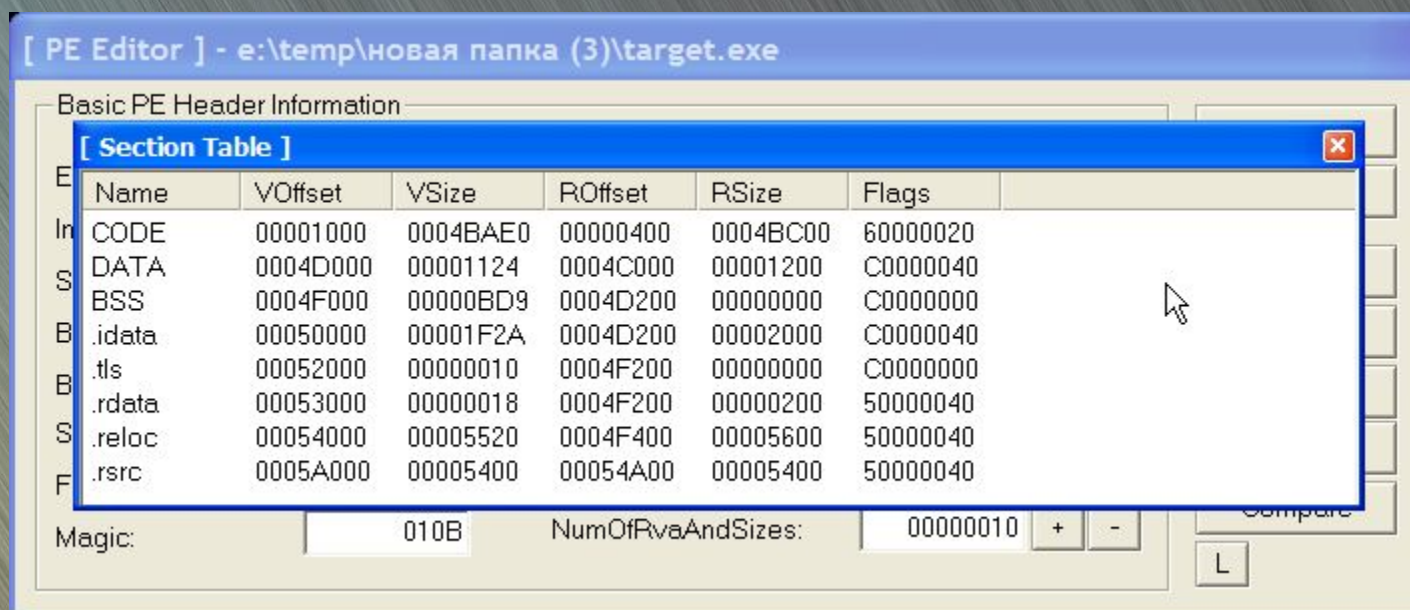
Обоснование:

Места в существующей секции кода мало (`RawSize` – размер в исполняемом файле), а размер процесса в памяти гораздо больше (`VirtualSize`) => `VirtualSize` >> `RawSize`.

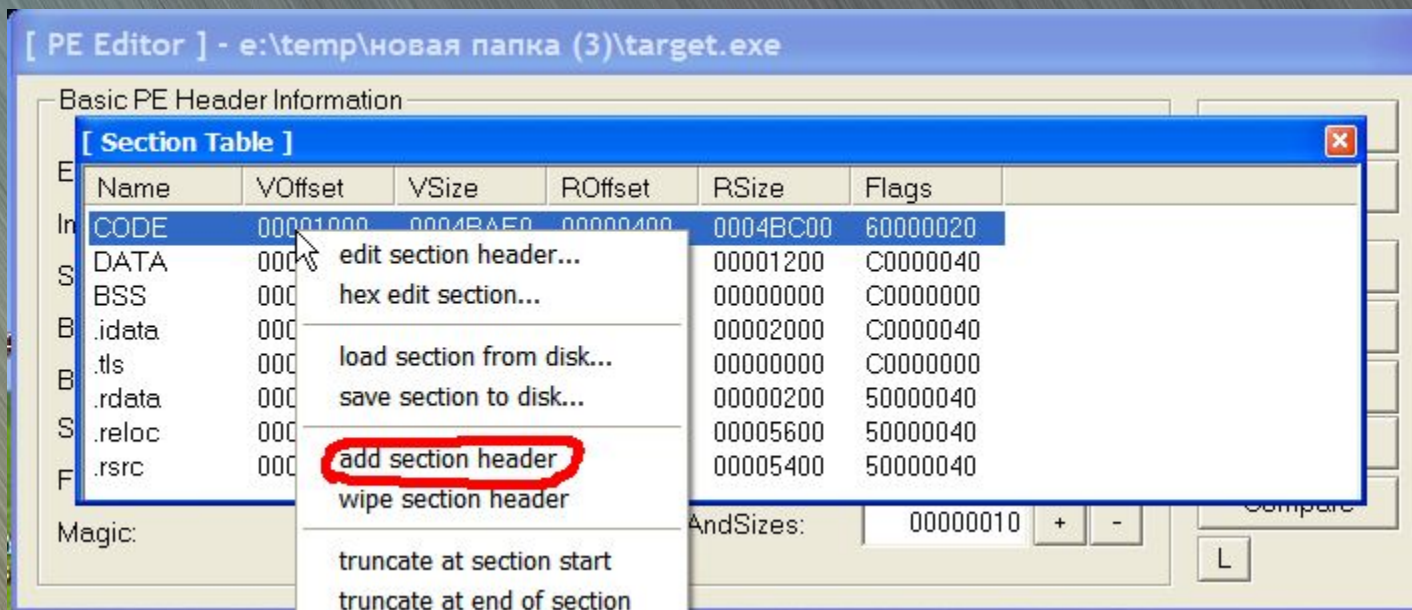
1. Открываем модифицируемый файл в Lord PE или PE tools.



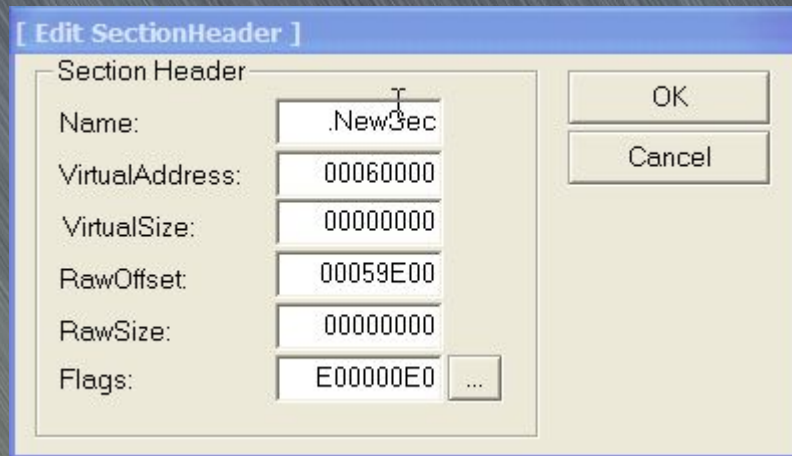
2. Открываем раздел Section.



3. Добавляем заголовок секции (Section).



4. Редактируем заголовок секции (Section).



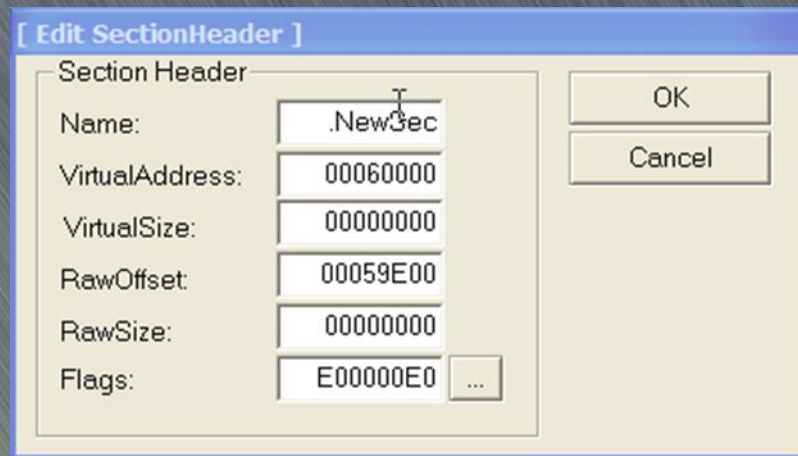
- a) В поле Name вводим какое-нибудь название секции, например “.code”;
- b) Поля VirtualAddress и RawOffset не трогаем – это адреса секции в памяти и в файле соответственно, вычисляются как

$$VirtualAddress = ((VirtualAddress(предыдущая секция) + VirtualSize(предыдущая секция) - 1) \div VirtualAlign) + 1) * VirtualAlign;$$

$$RawOffset = ((VirtualAddress(предыдущая секция) + VirtualSize(предыдущая секция) - 1) \div FileAlign) + 1) * FileAlign;$$

Их LordPE считает автоматом, и менять их не надо!

4. Редактируем заголовок секции (Section).

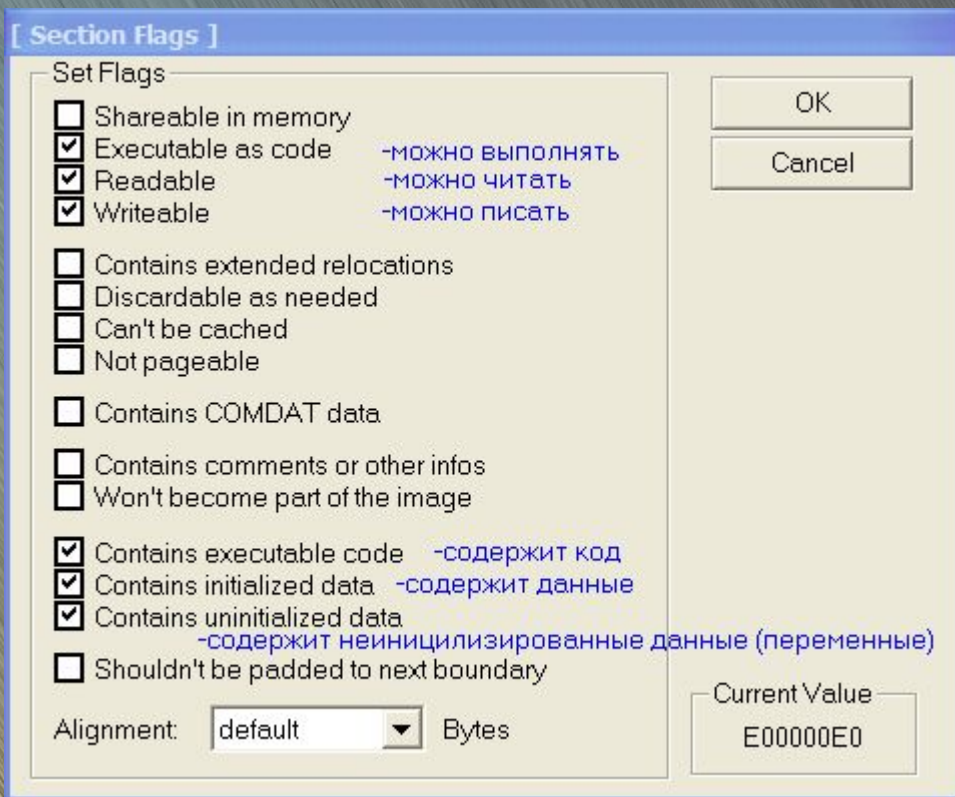


с) Рассчитываем поля **VirtualSize** и **RawSize**

RawSize - размер секции в файле = 1000 (4кб – достаточно для нашего кода).

VirtualSize - размер секции в памяти = 4000 (16кб – писать в память не только код, но и данные, чтобы не создавать новую секцию).

5. Редактируем флаги секции. Нажимаем на кнопку рядом с полем Flags:

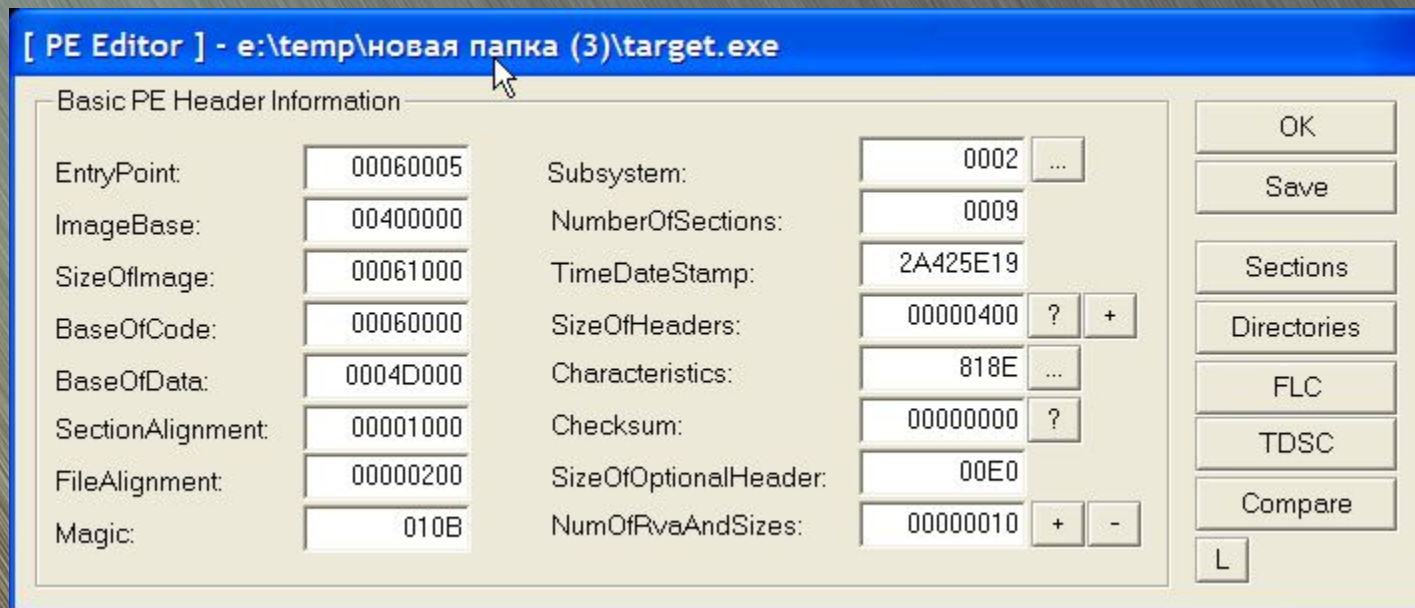


Отмечаем (ставим галочку):

- Что-то в секции можно выполнять
- Из неё можно читать
- В неё можно писать
- В ней есть код
- В ней есть инициализированные данные
- В ней есть неинициализированные данные

6. Запомнить и записать значения VirtualOffset (RVA) и RawOffset, EntryPoint, BaseOfCode

7. Пишем вместо BaseOfCode значение RVA и вместо EntryPoint значение RVA нашей секции + 5.



8. Нажимаем Save и «ОК». Запускаем файл. Он не запускается...

Секцию объявили, а записать – не записали!

9. Открываем файл во FlexHex и переходим в самый конец файла. Теперь ищем то место, где будет новая секция. А она начинается с запомненного нами RawOffset (назовём его Raw = 59E00).

Если последний существующий байт файла имеет адрес, отличный от (Raw-1), то забиваем нулями всё место от конца файла до этого числа (Щёлкаем после последнего байта в файле, далее Edit->Insert Zero Block; Block Size = Raw - адрес последнего байта - 1.

Иначе щёлкаем на пустой квадратик по адресу Raw, далее Edit->Insert Zero Block; Block Size = размер нашей секции, RawSize, то есть 1000h.

