

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение
высшего образования

«Московский технический университет связи и информатики»
(МТУСИ)

Отчет
по лабораторной работе №1
по дисциплине «Защита информации в глобальных сетях»
на тему: Составление PE файла и внедрение сигнатуры

Выполнил: студент группы МБД2031
Морозов М.Е.

Проверил: Барков В.В.

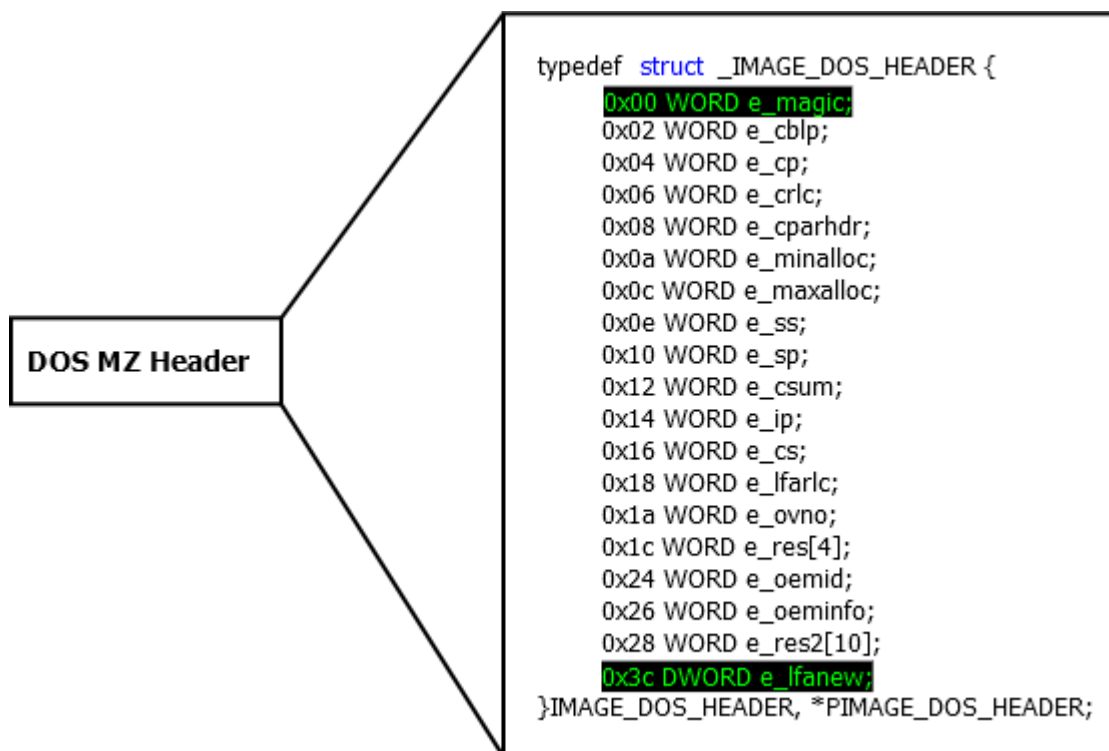
Москва
2022

Цель работы: изучить принципы работы и получить навыки составления PE файла.

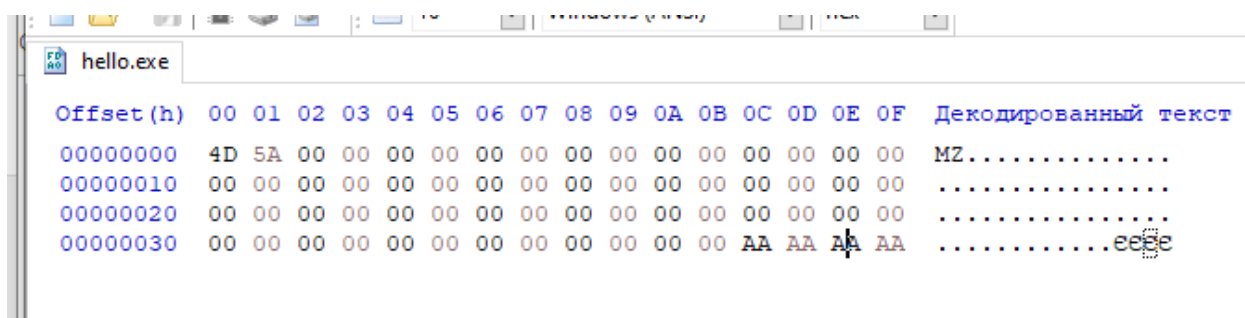
Инструментарий:

HxD Hex Editor.

PE Explorer.



e_magic: WORD — сигнатура находящаяся по смещению 0 от начала файла и равная “MZ”. Если данная сигнатура не равна MZ, то файл не загрузится.



e_lfanew: DWORD — смещение PE заголовка относительно начала файла. PE заголовок должен начинаться с сигнатуры PE\x0\x0. PE заголовок может располагаться в любом месте файла. Если посмотреть на структуру, то можно увидеть, что *e_lfanew* находится по смещению 0x3C (60 байт). То есть чтобы прочитать это значение, мы должны от указателя на начало файла

прибавить 60 байт и тогда мы встанем перед `e_lfnw`. Читаем это значение и плюсуем к указателю начала файла это значение. Это должен быть PE заголовок.

В данной лабораторной работе задаём смещение в `0x40`, пусть PE заголовок следует сразу за `e_lfnw`. Его длина 4 байта.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
00000000	4D	5A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	MZ.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00@...
00000040	50	45	00	00													PE..

Так как Dos-Header я заполнил, далее заполняю File-Header, где начиная с адреса `0x44` по порядку буду использовать следующие значения:

- Архитектура процессора 32 бита (2 байта) – `0x014c`
- Количество секций – 3 (2 байта) – `0x03`
- Временная метка (4 байта) – `NULL`
- Указатель на таблицу символов (4 байта) – `NULL`
- Количество символов (4 байта) – `NULL`
- Размер дополнительного заголовка (2 байта) - ?? `AAAA`
- Характеристика файла (2 байта) – `0x0102` где `0x0002` – исполняемый файл, а `0x0100` – 32х битная поддержка

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
00000000	4D	5A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	MZ.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00@...
00000040	50	45	00	00	4C	01	03	00	00	00	00	00	00	00	00	00	PE..L.....
00000050	00	00	00	00	AA	AA	02	01								EE..

Затем заполняю опциональный заголовок, он будет состоять из:

- Magic (2 байта) – `0x01b` указывает на 32 битность программы
- AddressOfEntryPoint (4 байта) - ?? `AAAAAAAA`
- ImageBase (4 байта) – `0x004` значение по умолчанию
- SectionAlignment (4 байта) - ?? `AAAAAAAA`
- FileAlignment (4 байта) – `0x200` значение по умолчанию
- MajorSubsystemVersion – `0x0004` Windows NT
- SizeOfImage – `AAAAAAAA` кратно `SectionAlignment`
- SizeOfHeaders – `BBBBBBBB` кратно `FileAlignment`

- Subsystem – 0x0002 т.к. программа является графической программой Windows
- NumberOfRvaAndSizes – 0x10 значение по умолчанию

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
00000000	4D	5A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	MZ.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00@...
00000040	50	45	00	00	4C	01	03	00	00	00	00	00	00	00	00	00	PE..L.....
00000050	00	00	00	00	AA	AA	02	01	0B	01	00	00	00	00	00	00ee.....
00000060	00	00	00	00	00	00	00	00	AA	AA	AA	AA	00	00	00	00eeee....
00000070	00	00	00	00	00	00	40	00	AA	AA	AA	AA	00	02	00	00@.eeee....
00000080	00	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00
00000090	AA	AA	AA	AA	BB	BB	BB	BB	00	00	00	00	02	00	00	00	eeee»»»».....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	10	00	00	00								

Далее в DataDirectory возьмём IMAGE_DIRECTORY_ENTRY_IMPORT где секцию VirtualAddress заполним AAAAAAAAAA, а Size заполним NULL

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
00000000	4D	5A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	MZ.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00@...
00000040	50	45	00	00	4C	01	03	00	00	00	00	00	00	00	00	00	PE..L.....
00000050	00	00	00	00	AA	AA	02	01	0B	01	00	00	00	00	00	00ee.....
00000060	00	00	00	00	00	00	00	00	AA	AA	AA	AA	00	00	00	00eeee....
00000070	00	00	00	00	00	00	40	00	AA	AA	AA	AA	00	02	00	00@.eeee....
00000080	00	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00
00000090	AA	AA	AA	AA	BB	BB	BB	BB	00	00	00	00	02	00	00	00	eeee»»»».....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
000000C0	AA	AA	AA	AA	00	00	00	00	00	00	00	00	00	00	00	00	eeee.....
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00								

После заполнения опционального заголовка, вычисляем его размер и он составил:

$$0x137 - 0x58 + 0x1 = 0xE0$$

Запишем это значение в SizeOfOptionalHeader:

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
00000000	4D	5A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	MZ.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00@...
00000040	50	45	00	00	4C	01	03	00	00	00	00	00	00	00	00	00	PE..L.....
00000050	00	00	00	00	E0	00	02	01	0B	01	00	00	00	00	00	00a.....
00000060	00	00	00	00	00	00	00	00	AA	AA	AA	AA	00	00	00	00eeee....
00000070	00	00	00	00	00	00	40	00	AA	AA	AA	AA	00	02	00	00@.eeee....
00000080	00	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00
00000090	AA	AA	AA	AA	BB	BB	BB	BB	00	00	00	00	02	00	00	00	eeee»»»».....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
000000C0	AA	AA	AA	AA	00	00	00	00	00	00	00	00	00	00	00	00	eeee.....
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Далее заполняем IMAGE_SECTION_HEADER, где:

- Name - .text
- VirtualSize – 0x1000
- VirtualAddress – 0x1000
- SizeOfRawData – 0x200
- PointerToRawData – 0x200
- Characteristics – 0x60000020

Остальные заполним NULL.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
00000000	4D	5A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	MZ.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00@...
00000040	50	45	00	00	4C	01	03	00	00	00	00	00	00	00	00	00	PE..L.....
00000050	00	00	00	00	E0	00	02	01	0B	01	00	00	00	00	00	00a.....
00000060	00	00	00	00	00	00	00	00	AA	AA	AA	AA	00	00	00	00eeee....
00000070	00	00	00	00	00	00	40	00	AA	AA	AA	AA	00	02	00	00@.eeee....
00000080	00	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00
00000090	AA	AA	AA	AA	BB	BB	BB	BB	00	00	00	00	02	00	00	00	eeee»»»».....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
000000C0	AA	AA	AA	AA	00	00	00	00	00	00	00	00	00	00	00	00	eeee.....
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	2E	74	65	78	74	00	00	00text...
00000140	00	10	00	00	00	10	00	00	00	02	00	00	00	02	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	20	00	00	60

Аналогично заполним секции .idata и .data

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
00000000	4D	5A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	MZ.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00@...
00000040	50	45	00	00	4C	01	03	00	00	00	00	00	00	00	00	00	PE..L.....
00000050	00	00	00	00	E0	00	02	01	0B	01	00	00	00	00	00	00a.....
00000060	00	00	00	00	00	00	00	00	AA	AA	AA	AA	00	00	00	00eeee....
00000070	00	00	00	00	00	00	40	00	AA	AA	AA	AA	00	02	00	00@.eeee....
00000080	00	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00
00000090	AA	AA	AA	AA	BB	BB	BB	BB	00	00	00	00	02	00	00	00	eeee»»»».....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
000000C0	AA	AA	AA	AA	00	00	00	00	00	00	00	00	00	00	00	00	eeee.....
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	2E	74	65	78	74	00	00	00text...
00000140	00	10	00	00	00	10	00	00	00	02	00	00	00	02	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	20	00	00	60`
00000160	2E	69	64	61	74	61	00	00	00	10	00	00	00	20	00	00	.idata.....
00000170	00	02	00	00	00	04	00	00	00	00	00	00	00	00	00	00@..@.data...
00000180	00	00	00	00	40	00	00	40	2E	64	61	74	61	00	00	000.....
00000190	00	10	00	00	00	30	00	00	00	02	00	00	00	06	00	00@...
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00

Далее открываем получившийся файл в программе PE Explorer, где уже можно определить виртуальный адрес загрузки секции кода .text:

PE Explorer - C:\Users\Max\Desktop\hello.exe						
File View Tools Help						
SECTION HEADERS						
Name	Virtual Size	Virtual Address	Size of Raw Data	Pointer to Raw Data	Characteristics	Pointing Directories
<input checked="" type="checkbox"/> .text	00001000h	00401000h	00000200h	00000200h	60000020h	
<input checked="" type="checkbox"/> .idata	00001000h	00402000h	00000200h	00000400h	40000040h	
<input checked="" type="checkbox"/> .data	00001000h	00403000h	00000200h	00000600h	00000040h	

После чего можем заменить значения в AddressOfEntryPoint, SectionAlignment, SizeOfImage, SizeOfHeaders и виртуальный адрес таблицы импорта.

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
00000000	4D	5A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	MZ.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00@...
00000040	50	45	00	00	4C	01	03	00	00	00	00	00	00	00	00	00	PE..L.....
00000050	00	00	00	00	E0	00	02	01	0B	01	00	00	00	00	00	00a.....
00000060	00	00	00	00	00	00	00	00	00	10	00	00	00	00	00	00
00000070	00	00	00	00	00	00	40	00	00	10	00	00	00	02	00	00@.....
00000080	00	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00
00000090	00	40	00	00	00	02	00	00	00	00	00	00	02	00	00	00	.@.....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
000000C0	00	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

PE Explorer - C:\Users\Max\Desktop\hello.exe			File View Tools Help		
HEADERS INFO			Address of Entry Point: 00401000 ✓ Real Image Checksum: 00005EC1h		
Field Name	Data Value	Description	Field Name	Data Value	Description
Machine	014Ch	i386@	Section Alignment	00001000h	
Number of Sections	0003h		File Alignment	00000200h	
Time Date Stamp	00000000h	01/01/1970 00:00:00	Operating System Version	00000000h	0.0
Pointer to Symbol Table	00000000h		Image Version	00000000h	0.0
Number of Symbols	00000000h		Subsystem Version	00000004h	4.0
Size of Optional Header	00E0h		Win32 Version Value	00000000h	Reserved
Characteristics	0102h		Size of Image	00004000h	16384 bytes
Magic	010Bh	PE32	Size of Headers	00000200h	
Linker Version	0000h	0.0	Checksum	00000000h	
Size of Code	00000000h		Subsystem	0002h	Win32 GUI
Size of Initialized Data	00000000h		Dll Characteristics	0000h	
Size of Uninitialized Data	00000000h		Size of Stack Reserve	00000000h	
Address of Entry Point	00401000h		Size of Stack Commit	00000000h	
Base of Code	00000000h		Size of Heap Reserve	00000000h	
Base of Data	00000000h		Size of Heap Commit	00000000h	
Image Base	00400000h		Loader Flags	00000000h	Obsolete
			Number of Data Directories	00000010h	

Для секции кода наша процедура отображения сообщения выглядит так:

Assemble

Enter your assembly code using Intel syntax below.

```
show_message:
push 0;
push 0xAAAAAAAA;
push 0xAAAAAAAA;
push 0;
call [0xAAAAAAAA];
jmp Show_message
```

В коде выглядит так:

Assembly

Raw Hex (zero bytes in bold):

6A0068AAAAAAAA68AAAAAAAA6A00FF15AAAAAAAAEBEA

String Literal:

"\x6A\x00\x68\xAA\xAA\xAA\xAA\x68\xAA\xAA\xAA\xAA\x6A\x00\xff\x15\xAA\xAA\xAA\xAA\xEB\xEA"

Array Literal:

```
{ 0x6A, 0x00, 0x68, 0xAA, 0xAA, 0xAA, 0xAA, 0x68, 0xAA, 0xAA, 0xAA, 0xAA, 0x6A, 0x00,
0xFF, 0x15, 0xAA, 0xAA, 0xAA, 0xAA, 0xEB, 0xEA }
```

Внесем в секцию .text с учетом корректировки адреса:

00000200	6A 00 68 AA AA AA AA 68 AA AA AA AA 6A 00 FF 15	j.hEEEEhEEEEj.я.
00000210	AA AA AA AA EB EA 00 00 00 00 00 00 00 00 00 00	EEEEлк.....
00000220	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000230	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000240	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000250	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000260	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000270	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000280	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000290	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000300	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000310	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000320	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000330	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000340	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000350	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000360	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000370	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000380	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000390	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

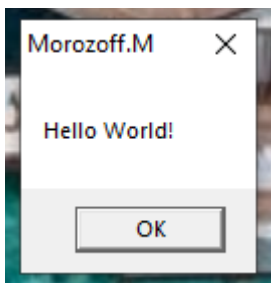
В секции данных ввел Morozoff.M Hello World! и выровнял размер секции до 0x200:

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
000005A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000600	4D	6F	72	6F	7A	6F	66	66	2E	4D	00	48	65	6C	6C	6F	Morozoff.M.Hello
00000610	20	57	6F	72	6C	64	21	00	00	00	00	00	00	00	00	00	World!.....
00000620	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Далее в секцию Import List внес функцию и имя библиотеки:

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
000003A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000003F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000400	28	20	00	00	00	00	00	00	00	00	00	00	46	20	00	00	(.....F ..
00000410	3E	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	>
00000420	00	00	00	00	00	00	00	00	30	20	00	00	00	00	00	000
00000430	00	00	4D	65	73	73	61	67	65	42	6F	78	41	00	30	20	..MessageBoxA.0
00000440	00	00	00	00	00	00	75	73	65	72	33	32	2E	64	6C	6Cuser32.dll
00000450	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000460	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000470	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000480	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Сохранил. Программа готова и работает:



Выводы

В данной лабораторной работе я изучил принципы работы и составления PE файла, получил навыки построения PE файла с помощью HEX - редактора.