Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе
Курс: Системное программирование
Тема: Мониторинг обращений к заданному файлу

Выполнил студент группы 13541/3 _____ Д.В. Круминьш
(подпись)

Преподаватель _____ Е.В. Душутина
(подпись)

Санкт-Петербург
2017 г.

Содержание

Постановка задачи

В данной работе необходимо на уровне API windows перехватывать системные функции связанные с файлами и директориями. К таким функциям относятся:

- создание;
- сохранение/пересохранение;
- открытие;
- закрытие;
- удаления.

Подобные действия необходимо логировать. А также дополнить информацией о пользователе, вызвавшем функцию, и времени вызова.

Сведения о системе

Работа производилась на реальной системы, с параметрами представленными ниже.

Элемент	Значение
Имя ОС	Майкрософт Windows 10 Pro (Registered Trademark)
Версия	10.0.14393 Сборка 14393
Дополнительное описание ОС	Недоступно
Изготовитель ОС	Microsoft Corporation
Имя системы	USER-PC
Изготовитель	HP
Модель	OMEN by HP Laptop 15-ce0xx
Тип	Компьютер на базе x64
SKU системы	1ZB00EA#ACB
Процессор	Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz, 2496 МГц, ядер: 4, логических процессоров: 4
Версия BIOS	American Megatrends Inc. F.04, 10.05.2017
Версия SMBIOS	3.0
Версия встроенного контроллера	40.20
Режим BIOS	Устаревший
Изготовитель основной платы	HP
Модель основной платы	Недоступно
Имя основной платы	Основная плата
Роль платформы	Мобильный
Состояние безопасной загрузки	Не поддерживается
Конфигурация PCR7	Привязка невозможна
Папка Windows	C:\Windows
Системная папка	C:\Windows\system32
Устройство загрузки	\Device\HarddiskVolume1
Язык системы	Россия
Аппаратно-зависимый уровень (HAL)	Версия = ”10.0.14393.1378”
Имя пользователя	USER-PC\Tom

Часовой пояс	RTZ 2 (зима)
Установленная оперативная память (RAM)	8,00 ГБ
Полный объем физической памяти	7,87 ГБ
Доступно физической памяти	3,54 ГБ
Всего виртуальной памяти	12,6 ГБ
Доступно виртуальной памяти	6,82 ГБ
Размер файла подкачки	4,75 ГБ
Файл подкачки	C:\pagefile.sys

Таблица 2.1: Информация об используемой системе

Для разработки использовалась Microsoft Visual Studio Enterprise 2017 (Версия 15.3.0).

Перехват Windows api

Рассмотрим принципы вызова функций, методы их перехвата, а также то как именно происходит вызов функции перехватчика.

3.1 Принцип вызова функции

Для вызова функций, размещенных в DLL, имеются два способа:

1. Раннее связывание;
2. Позднее связывание.

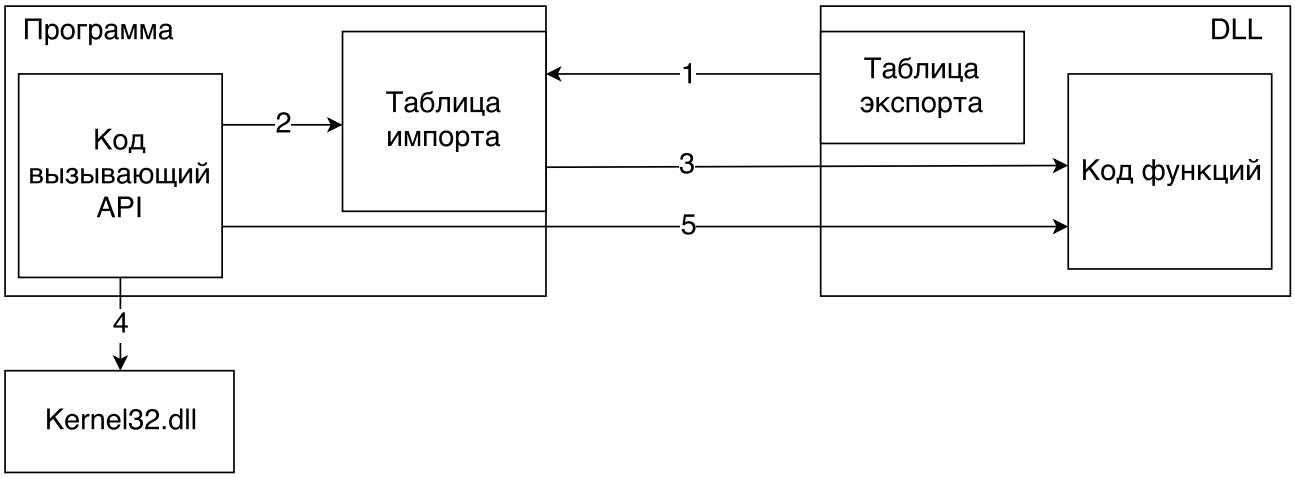

Рис. 3.1: Схема вызовов функций

3.1.1 Раннее связывание

Компилятору известен список импортируемых программой функций, на основе этого формируется таблица импорта. В этой таблице хранятся данные об используемых библиотеках и импортируемых из них функциях. Изначально адреса этих функций пусты.

На **шаге 1**, в момент загрузки программы, операционная система анализирует таблицу импорта, заполняя пустые адреса функций - реальными. Отсутствие какой-либо библиотеки приведет к ошибке загрузки программы.

При успешной загрузки, на **шаге 2** код программы будет обращаться к соответствующей записи из таблицы импорта, и на **шаге 3** будет вызвана необходимая функция DLL, адрес которой известен.

3.1.2 Позднее связывание

Данный метод отличается от раннего связывания тем, что что загрузка DLL производится динамически, обращаясь к библиотеке **kernel32.dll**, которая является ядром windows, и предоставляет многие базовые функции API. Для получения адреса функции применяется функция **GetProcAddress**.

В данной работе, будет производится перехват функций этого типа.

3.1.3 DLL библиотеки

Стоит отметить что у каждой библиотеки DLL, имеется таблица экспорта, в которой перечислены экспортируемые DLL функции, и их относительные адреса.

3.2 Способы перехвата

3.2.1 Модификация таблицы импорта

Суть метода заключается в том, что в памяти системы находится таблица импорта, и в ней исправляются адреса необходимых функций на адреса написанных перехватчиков.

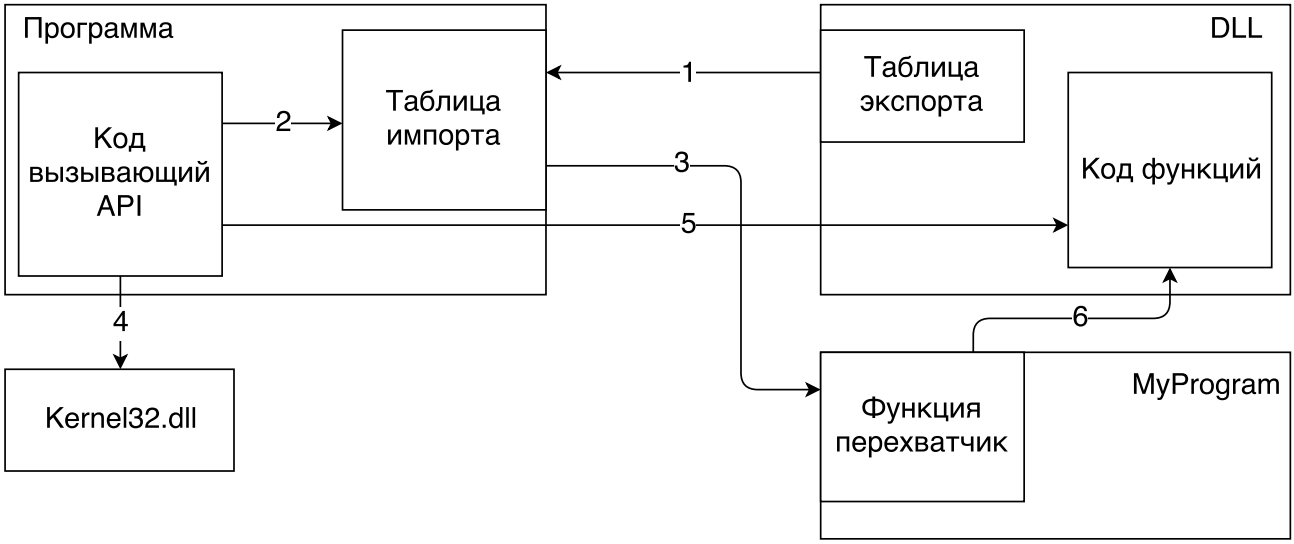

Рис. 3.2: Модификация таблицы импорта

Теперь **шаг 3** направлен в функцию перехватчик, из которой, в случае сохранения, вызывается оригинальная функция.

Используя данный метод, можно перехватить лишь статически импортируемые функции из таблицы импорта.

3.2.2 Перехват функций kernel32.dll

В данном случае, с помощью функции **GetProcAddress**, происходит получение реального адреса необходимой функции.

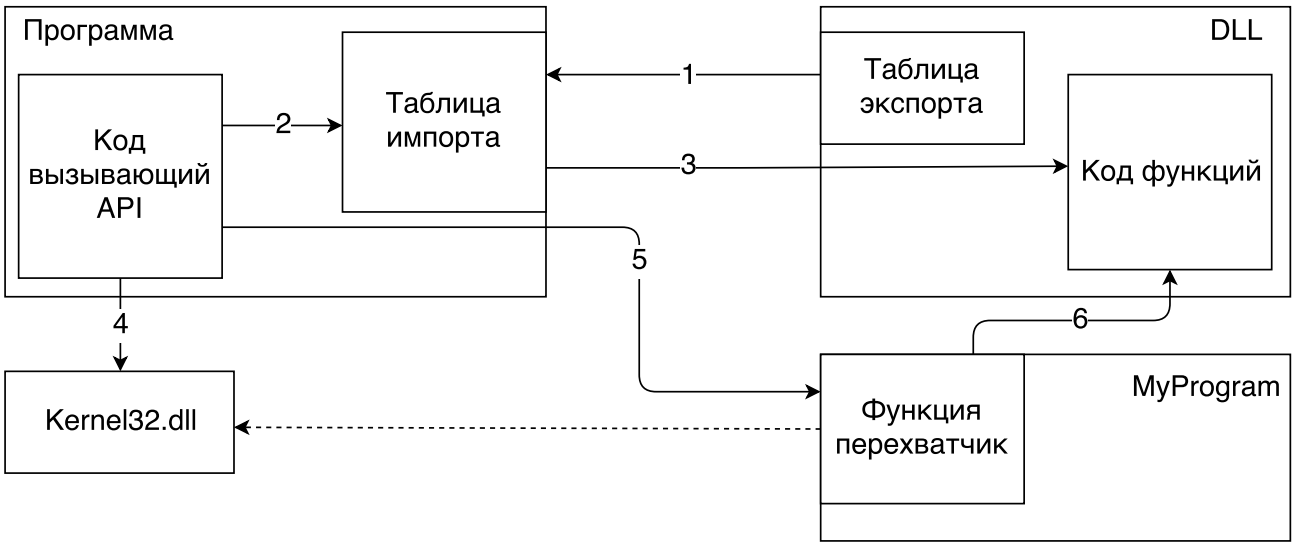

Рис. 3.3: Перехват функций kernel32.dll

Теперь **шаг 5** направлен в функцию перехватчик, из которой, в случае сохранения, вызывается оригинальная функция.

3.3 Изменение адресов

Используемый способ перехвата - вставка инструкции перехода (jump). Инструкция безусловного перехода имеет длину 5 байтов. Один байт представляет опкод, оставшиеся четыре представляют относительное 32-битное смещение.

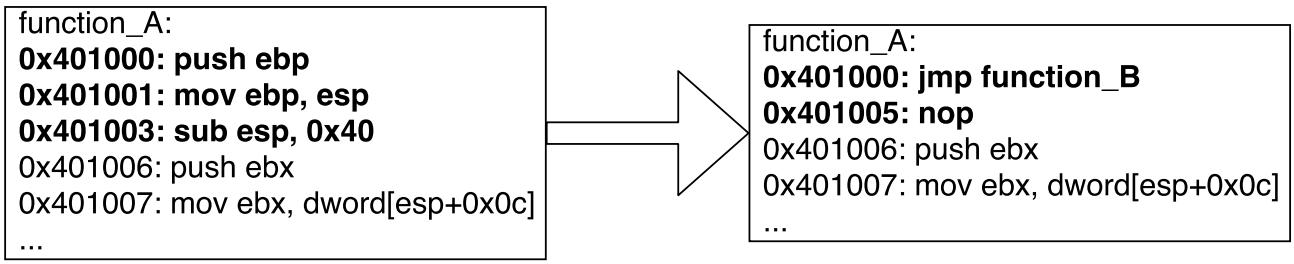

Рис. 3.4: Вызов функции перехватчика

С помощью данной инструкции перехода имеется возможность выполнить перенаправление на функцию перехватчик. На рисунке 3.4 исходные команды(6 байт) были заменены на функцию jump и команду nop(No OPeration). В случае чего, перенаправление было успешно выполнено.

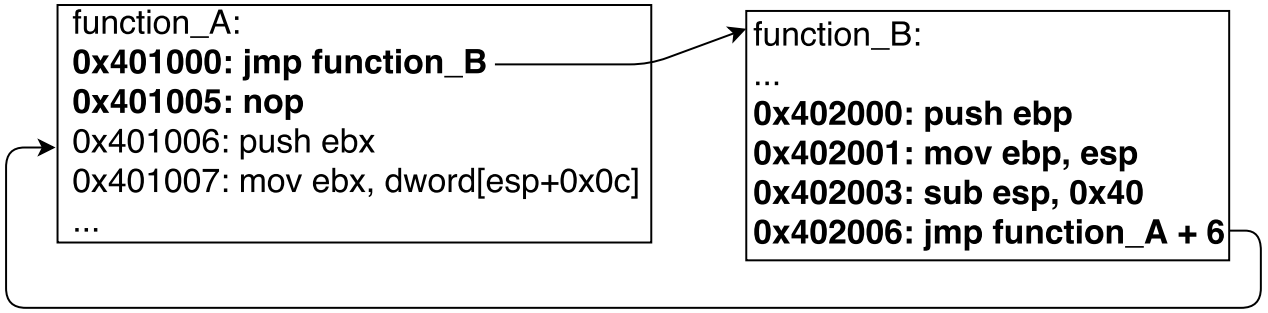

Рис. 3.5: Возвращение в реальную функцию

В функции перехватчике выполняются необходимые действия, по завершению которых, выполняется исходный код функции, который был заменен на переход. Далее выполняется переход на исходную функцию, с необходимым смещением, для того чтобы снова не попасть в функцию перехватчик.

Более правильным вариантом является использование **трамплина**. Трамплин состоит из двух частей: исходных инструкций и перехода на ту часть исходной функции, которая следует за перехватчиком.

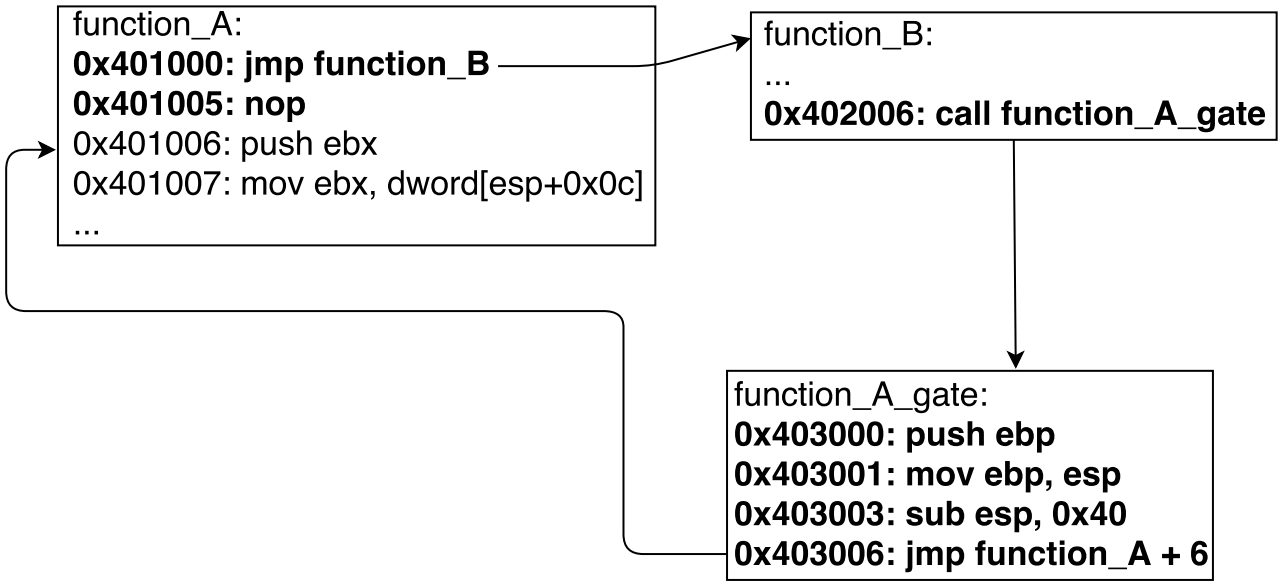

Рис. 3.6: Использование трамплина

Это решение позволяет более четко отделить команды перехватчика от восстановленных команд исходной функции.

Функция jump длиной 5 байт, имеет недостаток в том, что функция на которую ссылается jump должна находится в диапазоне ограниченном 32 битами. И если используется 64-битная система, то функция перехватчик может оказаться за пределами данного диапазона.

В данном случаем следует использовать команду jump длиной 14 байт. Выглядит она следующим образом:

FF 25 00 00 00 00 12 34 56 78 12 34 56 78

- FF 25 - опкод для идентификации команды;
- 00 00 00 00 - смещение на сегмент, в котором расположен 64-битный указатель;
- 12 34 56 78 12 34 56 78 - указатель на 64-битный адрес для перехода.

Если прежде для функции jump в 5 байт можно с вероятностью в 99% гарантировать что для неё хватит места в переписываемой функции, то для 14 байт такого сказать нельзя.

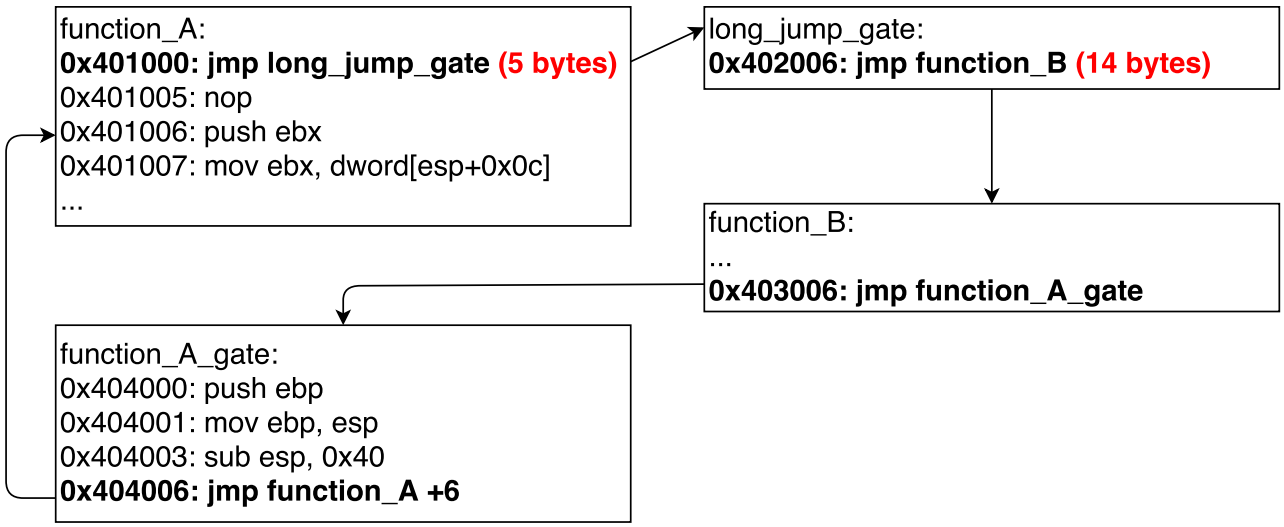

Рис. 3.7: Использование дополнительного трамплина

Для решения этой проблемы следует использовать дополнительный трамплин. Трамплин располагается рядом(в адресном пространстве(в диапазоне 32 бит)) от начальной функции, для того чтобы вызвать команду jmp на 14 байт.

Инъекция стороннего кода

Существуют различные способы для перехвата системных функций. В данной работе будет рассмотрен способ, с инъекцией DLL библиотеки с перехватчиками, используя реестр.

Чтобы внедрить DLL в процессы, которые связаны с USER32.DLL, нужно добавить имя DLL в значение следующего раздела реестра:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs

Его значение содержит одно имя DLL или группу DLL, разделенных запятой или пробелами. Согласно документации MSDN[1], все DLL, указанные значением этого ключа, загружаются каждым приложением Windows, запущенным в текущем сеансе входа в систему. Стоит отметить, что фактическая загрузка этих DLL происходит как часть инициализации USER32.

USER32 считывает значение данного раздела реестра и вызывает LoadLibrary () для этих DLL в своем коде DllMain. Однако данная инъекция применяется только к приложениям, использующим USER32.DLL.

После установки пути к необходимому для внедрения DLL, необходимо изменить следующий раздел реестра:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\LoadAppInit_DLLs

Данный раздел отвечает за включение или отключения AppInit_DLLs, и принимает следующий значения:

- 0x0 – AppInit_DLLs отключено;
- 0x1 – AppInit_DLLs включено.

Необходимо выставить значение **0x1**.
После проделанных действий следует перезагрузить компьютер.

Разработка

5.1 Настройка проекта

В настройке проекта, в первую очередь следует изменить платформу на x64, по умолчанию x86. Также необходимо изменить тип конфигурации - динамическая библиотека dll.

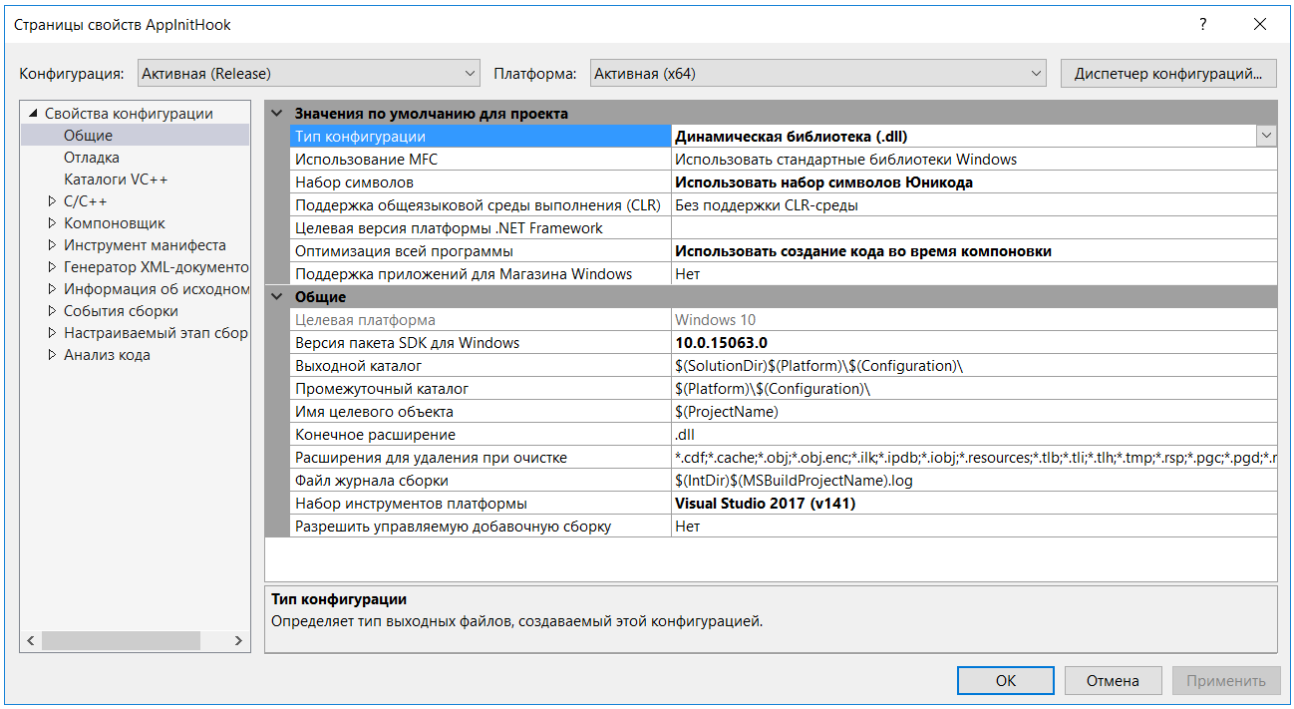

Рис. 5.1: Параметры проекта

5.2 Точка входа

У выбранного типа конфигурации, несколько иная входная точка в отличии от обычного приложения. Точкой входа в данном случае является функция **DllMain**.

```
1 BOOL WINAPI DllMain(
2   HINSTANCE hinstDLL,   // дескриптор модуля DLL
3   DWORD fdwReason,      // причина вызова функции
4   LPVOID lpvReserved    // зарезервированный
5 );
```

Листинг 5.1: Прототип функции DllMain

Функция точки входа вызывается при каждой загрузке и выгрузке библиотеки и получает три параметра - дескриптор библиотеки (HINSTANCE **hinstDLL**), флаг причины вызова(DWORD **fdwReason**) и детализация вызова (LPVOID **lpvReserved**).

Флаг причины вызова может принимать следующие значения:

1. **DLL_PROCESS_ATTACH** - При присоединении к адресному пространству текущего процесса, в результате его запуска или вызова функции LoadLibrary;
2. **DLL_THREAD_ATTACH** - Данный процесс создает новый поток;
3. **DLL_PROCESS_DETACH** - Завершение процесса либо вызов FreeLibrary;
4. **DLL_THREAD_DETACH** - Завершение потока.

Параметр lpvReserved позволяет определить каким образом загружена библиотека - статически или динамически. Если параметр имеет значение отличное от NULL - библиотека загружена статически, если же NULL - то динамически (с использованием LoadLibrary).

```
1  //////////////////////////
2  // Entry point
3
4  BOOL WINAPI DllMain(
5      __in HINSTANCE  hInstance,
6      __in DWORD      Reason,
7      __in LPVOID     Reserved
8  ){
9      switch (Reason)
10     {
11     case DLL_PROCESS_ATTACH:
12         Mhook_SetHook((PVOID*)&RealCreateDirectory, HookCreateDirectoryW);
13         break;
14
15     case DLL_PROCESS_DETACH:
16         Mhook_Unhook((PVOID*)&RealCreateDirectory);
17         break;
18     }
19     return TRUE;
20 }
```

Листинг 5.2: Точка входа

В листинге выше представлена, не полностью заполненная точка входа библиотеки. Так в строке 12 с помощью библиотеки mhook идет установка перехватчика для функции **создания директории**.

5.3 Функции перехватчики

Целью работы функций перехватчиков в данном случае является мониторинг действий в указанной директории. Для этого все действия будут логироваться в соответствующий файл, а каждая запись лога будет иметь следующий формат:

Время | Имя функции | Имя пользователя | Дополнительные параметры, зависящие от функции

Папка для мониторинга - **E:/testFolder**.

Рассмотрим реализацию функций перехватчиков.

5.3.1 Вспомогательные функции

Для сокращение дубликации кода, из тел перехватчиков были вынесены следующие функции:

1. определение времени вызова;
2. от имени какого пользователя был вызов;
3. получение имени файла, по его дескриптору.

```
1  // Для определения времени
2  time_t rawtime;
3  struct tm * timeinfo;
4  char buffer[80];Для
5
6  // определения пользователя
7  TCHAR name[UNLEN + 1];
8
9  // для определения имени файла
10 #define BUFSIZE 512
11 TCHAR pszFilename[MAX_PATH + 1];
12
13 ////////////////////////
14 // вспомогательные функции
15 void SetupTime() {
16     time(&rawtime);
17     timeinfo = localtime(&rawtime);
18     strftime(buffer, sizeof(buffer), "%d-%m-%Y %H:%M:%S", timeinfo);
19 }
20
21 void SetupUserName() {
22     DWORD size = UNLEN + 1;
23     int result = GetUserName((TCHAR*)name, &size);
24     if(result == 0)
25         _tcscpy(name, _T("Error to get user name"));
26 }
27
28 // получение имени по дескриптору файла
29 BOOL GetFileNameFromHandle(HANDLE hFile)
30 {
31     // переменная для оповещения успеха
32     BOOL bSuccess = FALSE;
33     HANDLE hFileMap;
34
35     // получение размера файла
36     DWORD dwFileSizeHi = 0;
37     DWORD dwFileSizeLo = GetFileSize(hFile, &dwFileSizeHi);
38
39     // если размер равен 0
40     if (dwFileSizeLo == 0 && dwFileSizeHi == 0)
41     {
42         return FALSE;
43     }
44
45     // создание отображения
46     hFileMap = CreateFileMapping(hFile,
47         NULL,
48         PAGE_READONLY,
49         0,
```

```
50          1,
51          NULL);
52
53      if (hFileMap)
54      {
55          // получаем указатель на участок памяти с отображением
56          void* pMem = MapViewOfFile(hFileMap, FILE_MAP_READ, 0, 0, 1);
57
58          if (pMem)
59          {
60              if (GetMappedFileName(GetCurrentProcess(),
61                  pMem,
62                  pszFilename,
63                  MAX_PATH))
64              {
65
66                  // Нормализуем название
67                  TCHAR szTemp[BUFSIZE];
68                  szTemp[0] = '\0';
69
70                  if (GetLogicalDriveStrings(BUFSIZE - 1, szTemp))
71                  {
72                      TCHAR szName[MAX_PATH];
73                      TCHAR szDrive[3] = TEXT(" :");
74                      BOOL bFound = FALSE;
75                      TCHAR* p = szTemp;
76
77                      do
78                      {
79                          // Копируем имя диска во временную переменную
80                          *szDrive = *p;
81
82                          // Ищем необходимое устройство - диск
83                          if (QueryDosDevice(szDrive, szName, MAX_PATH))
84                          {
85                              size_t uNameLen = _tcslen(szName);
86
87                              if (uNameLen < MAX_PATH)
88                              {
89                                  bFound = _tcsnicmp(pszFilename, szName,
    ↪ uNameLen) == 0
90                                      && *(pszFilename + uNameLen) == _T('\\');
91
92                                  if (bFound)
93                                  {
94                                      // чтение имени файла
95                                      TCHAR szTempFile[MAX_PATH];
96                                      StringCchPrintf(szTempFile,
97                                          MAX_PATH,
98                                          TEXT("%s%s"),
99                                          szDrive,
100                                         pszFilename + uNameLen);
101                                     StringCchCopyN(pszFilename, MAX_PATH + 1,
    ↪ szTempFile, _tcslen(szTempFile));
102                                 }
103                             }
104                         }
105
106                         // Пока есть указатель
107                         while (*p++);
```

```
108                     } while (!bFound && *p); // конец имени
109                 }
110             }
111             bSuccess = TRUE;
112             UnmapViewOfFile(pMem);
113         }
114
115         CloseHandle(hFileMap);
116     }
117     _tprintf(TEXT("File name is %s\n"), pszFilename);
118     return(bSuccess);
119 }
```

Листинг 5.3: Вспомогательные функции

Так как многие функции в своих параметрах имеют дескриптор файла, и никакой более информации о файле с которым происходит работа, появилась необходимость в функции по получению имени файла, имея лишь дескриптор. Функция[2] имеет следующий алгоритм:

1. получаем размер файла, и проверяем не равен ли он 0;
2. отображаем файл и получаем указатель на его отображение;
3. код функции предполагает также получения названия файла с удаленного компьютера, поэтому происходят некоторые преобразования пути файла;
4. чтение имени файла в глобальную переменную, которая используется в функциях перехватчиках.

5.3.2 CreateDirectoryW

Функция[3] создает новую папку. Если файловая система поддерживает безопасность файлов и каталогов, эта функция применяет указанный дескриптор безопасности к новому каталогу.

```
1 BOOL WINAPI CreateDirectory(
2   _In_     LPCTSTR                 lpPathName,
3   _In_opt_ LPSECURITY_ATTRIBUTES lpSecurityAttributes
4 );
```

Листинг 5.4: Прототип функции CreateDirectoryW

Параметры:

- **lpPathName** - путь, по которому должна быть создана директория;
- **lpSecurityAttributes** - указатель на структуру с настройками безопасности.

В случае успеха, возвращается не нулевое значение. Рассмотрим реализацию.

```
1 ////////////////////////
2 // Defines and typedefs
3
4 typedef bool(WINAPI* _CreateDirectory) (LPCTSTR, LPSECURITY_ATTRIBUTES);
5
6 ////////////////////////
7 // Original function
8
```

```
9  _CreateDirectory RealCreateDirectory = (_CreateDirectory)GetProcAddress(
       ↪ GetModuleHandle(L"Kernel32"), "CreateDirectoryW");
10
11 ////////////////////////
12 // Hooked function
13
14 bool WINAPI HookCreateDirectoryW(LPCTSTR lpPathName, LPSECURITY_ATTRIBUTES
       ↪ lpSecurityAttributes) {
15     SetupTime();
16     SetupUserName();
17
18     CStringA stringLpPathName(lpPathName);
19     const char* charLpPathName = stringLpPathName;
20
21     if (strstr(charLpPathName, PATH)) {
22         wofstream myfile;
23         myfile.open("E:\\log.txt", ios::app);
24         myfile << "Time: " << buffer << " | CreateDirectoryW | User: " << name
       ↪ << " | Path: " << charLpPathName << endl;
25         myfile.close();
26     }
27
28     return RealCreateDirectory(lpPathName, lpSecurityAttributes);
29 }
```

Листинг 5.5: Перехватчик CreateDirectoryW

В листинге выше сперва был объявлен тип с которым будет производиться работа, далее происходит получение оригинальной функции, с помощью **GetProcAddress**, в которой идет получение дескриптора библиотеки **Kernel32** и указание какой необходимой функции.

После этого следует сама функция перехватчик, в ней происходят следующие действия:

1. вызов функций **SetupTime** и **SetupUserName**;
2. преобразование к читабельному виду параметра, в котором указан путь директории;
3. запись полученной информации в файл лога;
4. возвращение реальной функции.

Важным требованием является то, чтобы тип и параметры передаваемые в функцию перехватчик были идентичны оригинальной функции.

Особенности

После нажатия на пункт **Создать->Папку**, сразу же создается папка с именем **Новая папка** или **Новая папка (n)**, где n - целочисленное число, значение которого зависит от количества папок со схожим названием. То что система предлагает ввести название папки, является уже совсем другой функцией по переименованию.

5.3.3 RemoveDirectoryW

Функция[4] удаляет папку.

```
BOOL WINAPI RemoveDirectory(
  _In_ LPCTSTR lpPathName
);
```

Листинг 5.6: Прототип функции RemoveDirectoryW

Параметры:

- **lpPathName** - путь, по которому должна быть удалена директория.

Листинг данной функции практически идентичен функции **CreateDirectoryW**, поэтому будет приведен лишь листинге со всем кодом написанной программы.

Особенности

Если для корзины включена функция удалять сразу, без перемещения в корзину, то сразу после удаления в файл с логом попадает строка примерно следующего вида:

```
Time: 19-11-2017 22:26:37 | RemoveDirectoryW | User: Tom | Path: E:\testFolder
    ↪ \000
```

Листинг 5.7: Удаление без перемещения в корзину

Но если, такая функция включена, то разумеется после нажатия на **удалить** файл лишь попадает в корзину, уже с измененным названием. Теперь перехватчик сработает если удалять файл из корзины, и лог будет выглядеть следующим образом:

```
Time: 19-11-2017 22:27:51 | RemoveDirectoryW | User: Tom | Path: E:\$RECYCLE.
    ↪ BIN\S-1-5-21-2936131933-2071197896-2896244546-1000\$RMPR9WH
```

Листинг 5.8: Удаление с перемещением в корзину

5.3.4 ReadFile

Функция[5] читает данные файла с определенного места его представления в памяти.

```
BOOL WINAPI ReadFile(
  _In_        HANDLE       hFile,
  _Out_       LPVOID       lpBuffer,
  _In_        DWORD        nNumberOfBytesToRead,
  _Out_opt_   LPDWORD      lpNumberOfBytesRead,
  _Inout_opt_ LPOVERLAPPED lpOverlapped
);
```

Листинг 5.9: Прототип функции ReadFile

Параметры:

- **hFile** - дескриптор файла;
- **lpBuffer** - буффер, в который считываются данные;
- **nNumberOfBytesToRead** - количество байт оставшихся для чтения;
- **lpNumberOfBytesRead** - количество считанных байт;
- **lpOverlapped** - указатель на структуру **OVERLAPPED**(используется при сихнронной и асинхронной работе с файлами, в данном случае скорее всего используется для определения отступа).

Сама функция перехватчик немного отличается от прочих.

```
BOOL WINAPI HookReadFile(
    _In_        HANDLE       hFile,
    _Out_       LPVOID       lpBuffer,
    _In_        DWORD        nNumberOfBytesToRead,
    _Out_opt_   LPDWORD      lpNumberOfBytesRead,
    _Inout_opt_ LPOVERLAPPED lpOverlapped) {

    SetupTime();
    SetupUserName();

    if (GetFileNameFromHandle(hFile)) {
        CStringA stringLpFileName(pszFilename);
        const char* charLpFileName = stringLpFileName;
        if (strstr(charLpFileName, PATH)) {
            wofstream myfile;
            myfile.open("E:\\log.txt", ios::app);
            myfile << "Time: " << buffer << " | ReadFile | User: " << name << "
    ↪  | Path: " << charLpFileName << endl;
            myfile.close();
        }
    }
    return RealReadFile(hFile, lpBuffer, nNumberOfBytesToRead,
    ↪ lpNumberOfBytesRead, lpOverlapped);
}
```

Листинг 5.10: Функция перехватчик ReadFile

Дело в том, что в параметрах функции отсутствует имя или путь файла с которым идет работа, имеется лишь его дескриптор. Поэтому используя вспомогательную функцию **GetFileNameFromHandle** и определяется путь с именем файла, и если путь относится к папке выбранной для мониторинга, то в лог файл добавляется новая запись.

Особенности

По пути **E:/testFolder/logFolder/getFileAttr_log.txt** находился файл размером в 21 796 Кб, полученный в ходе экспериментов с функцией **GetFileAttributesW**. При заходе в проводнике в папку **testFolder**, в лог файл было добавлено 683 строчки вида:

```
Time: 19-11-2017 18:20:51 | ReadFile | User: Tom | Path: E:\testFolder\
    ↪ logFolder\getFileAttr_log.txt
```

Листинг 5.11: Фрагмент лога

Видимо система еще до захода в папку с файлом, заранее загрузила его в память. Проведем некоторые вычисления.

$$21796/683 = 31.91$$

Отсюда следует, что функция ReadFile за одну операцию считывает 32 кБ файла, для последующей работы.

5.3.5 GetFileAttributesW

Функция возвращает аттрибуты файла или директории. Возвращаемые аттрибуты по большей части являются системными:

- **FILE_ATTRIBUTE_ARCHIVE** - файл является архивом, прочие приложения используют этот флаг для того чтобы включить в список специфичных файлов для сохранения или удаления;

- **FILE_ATTRIBUTE_HIDDEN** - файл или папка имеют скрытый тип, поэтому они не включаются в список для отображения;
- **FILE_ATTRIBUTE_NOT_CONTENT_INDEXED** - файл или папка, не были проиндексированы сервисом индексации.
- и т.д.

```
1 DWORD WINAPI GetFileAttributes(
2   _In_ LPCTSTR lpFileName
3 );
```

Листинг 5.12: Прототип функции GetFileAttributesW

Параметры:

- **lpFileName** - полный путь к файлу или папке.

```
1 Time: 20-11-2017 20:37:18 | GetFileAttributesW | User: Tom | File: E:\
    ↪ testFolder\123.txt
```

Листинг 5.13: Фрагмент лога

Особенности
Из-за обилия лога, перехватчик логирует действия этой функции в отдельный лог-файл по пути **E:/testFolder2/log.txt**. Первоначально лог производился в общий файл лога. В конце эксперимента данный файл был открыт в редакторе **Sublime text**, для просмотра результатов. Однако с этим возникли проблемы, так как в открытом окне проводника находился данный лога, и видимо система вызывала функцию **GetFileAttributesW** для него, это привело к вызову функции-перехватчика и добавление новой записи в лог, что породило замкнутый круг.

То есть:

1. В проводнике система вызывает **GetFileAttributesW** для обновленного файла лога;
2. Перехватчик перехватывает данную функцию и добавляет в лог новую запись;
3. Повторяем шаг 1.

Буквально за 40 секунд, файл с логом разросся до размера в 21 796 Кб и 282510 строк.

5.3.6 DeleteFileW

Функция[6] удаляет существующий файл.

```
1 BOOL WINAPI DeleteFile(
2   _In_ LPCTSTR lpFileName
3 );
```

Листинг 5.14: Прототип функции DeleteFileW

Параметры:

- **lpFileName** - полный путь к файлу.

```
1 Time: 20-11-2017 20:37:38 | DeleteFileW | User: Tom | File: E:\$RECYCLE.BIN\S
    ↪ -1-5-21-2936131933-2071197896-2896244546-1000\$IF873NA.txt
```

Листинг 5.15: Фрагмент лога

Особенности
Функция перехватчик успешно срабатывает, если файл удаляется из корзины. В случае если файл не помещается в корзину, а сразу удаляется перехватчик не сработает, так как видимо используется иная функция. Также были написаны перехватчики для аналогичных функций удаления - **DeleteFileA** и **DeleteFileTransactedW**, но и они не дали результата.

5.3.7 Остальные функции

Помимо вышеописанных функций, были написаны перехватчики следующих функций:

1. **CreateFileW**[7];
2. **MoveFileW**[8];
3. **CopyFileW**[9];
4. **ReplaceFileW**[10];
5. **LZOpenFileW**[11];
6. **DeleteFileA**[12];
7. **DeleteFileTransactedW**[13].

Практически все из них не были вызваны или результат отличался от ожидаемого.

Например включение перехватчика **CreateFileW** после перезагрузки системы выдавало ошибку **critical process died** и бесконечную перезагрузку, в данном случае помогло восстановить работу системы - восстановление системы на ранее созданную точку восстановление.

5.4 Результат действия перехватчиков

Для двух вариантов работы корзины, были выполнены следующие действия:

1. Открытие диска **E** в проводнике;
2. Переход в папку **testFolder**;
3. Создание новой папки, с именем **qwerty**;
4. Переход в созданную папку;
5. Создание нового текстового файла **123.txt**;
6. Открытие созданного файла в блокноте;
7. Запись произвольных символов;

8. Сохранение и закрытие файла;
9. Снова открытие файла **123.txt**;
10. Закрытие файла;
11. Удаление файла **123.txt**;
12. Удаление папки **testFolder**;

5.4.1 Без перемещения в корзину

```
Time: 21-11-2017 20:44:24 | CreateDirectoryW | User: Tom | Path: E:\testFolder\ Новая папка
Time: 21-11-2017 20:44:44 | ReadFile | User: Tom | Path: E:\testFolder\qwerty\123.txt
Time: 21-11-2017 20:44:45 | ReadFile | User: Tom | Path: E:\testFolder\qwerty\123.txt
Time: 21-11-2017 20:44:53 | RemoveDirectoryW | User: Tom | Path: E:\testFolder\qwerty
```

Листинг 5.16: Лог

```
Time: 21-11-2017 20:44:10 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:10 | GetFileAttributesW | User: Tom | File: E:\testFolder2
Time: 21-11-2017 20:44:10 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:10 | GetFileAttributesW | User: Tom | File: E:\testFolder2
Time: 21-11-2017 20:44:10 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:10 | GetFileAttributesW | User: Tom | File: E:\testFolder2
Time: 21-11-2017 20:44:11 | GetFileAttributesW | User: Tom | File: E:\testFolder2\log.txt
Time: 21-11-2017 20:44:11 | GetFileAttributesW | User: Tom | File: E:\testFolder2\log.txt
Time: 21-11-2017 20:44:11 | GetFileAttributesW | User: Tom | File: E:\testFolder2\log.txt
Time: 21-11-2017 20:44:11 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder\
    ↪ ReadFile_log.txt
Time: 21-11-2017 20:44:11 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder\
    ↪ ReadFile_log.txt
Time: 21-11-2017 20:44:11 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder\
    ↪ ReadFile_log.txt
Time: 21-11-2017 20:44:11 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder\
    ↪ getFileAttr_log.txt
Time: 21-11-2017 20:44:11 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder\
    ↪ getFileAttr_log.txt
Time: 21-11-2017 20:44:11 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder\
    ↪ getFileAttr_log.txt
Time: 21-11-2017 20:44:12 | GetFileAttributesW | User: Tom | File: E:\testFolder2
Time: 21-11-2017 20:44:12 | GetFileAttributesW | User: Tom | File: E:\testFolder2
Time: 21-11-2017 20:44:12 | GetFileAttributesW | User: Tom | File: E:\testFolder2
Time: 21-11-2017 20:44:12 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:12 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:12 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:17 | GetFileAttributesW | User: Tom | File: E:\testFolder2
Time: 21-11-2017 20:44:17 | GetFileAttributesW | User: Tom | File: E:\testFolder2
Time: 21-11-2017 20:44:17 | GetFileAttributesW | User: Tom | File: E:\testFolder2
Time: 21-11-2017 20:44:17 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:17 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:17 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:19 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:19 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:19 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:19 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
Time: 21-11-2017 20:44:19 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
Time: 21-11-2017 20:44:19 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
Time: 21-11-2017 20:44:24 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:24 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
Time: 21-11-2017 20:44:24 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
Time: 21-11-2017 20:44:24 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
Time: 21-11-2017 20:44:24 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
Time: 21-11-2017 20:44:24 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
Time: 21-11-2017 20:44:24 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
Time: 21-11-2017 20:44:27 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:27 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
```

```
 53 |Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder                    |
 54 |Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder                    |
 55 |Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder                    |
 56 |Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
 57 |Time: 21-11-2017 20:44:28 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 58 |Time: 21-11-2017 20:44:29 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
 59 |Time: 21-11-2017 20:44:29 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
 60 |Time: 21-11-2017 20:44:35 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
 61 |Time: 21-11-2017 20:44:35 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 62 |Time: 21-11-2017 20:44:35 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
 63 |Time: 21-11-2017 20:44:35 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 64 |Time: 21-11-2017 20:44:35 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 65 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    |    ↪ текстовый документ.txt                                                                                 |
 66 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    |    ↪ текстовый документ.txt                                                                                 |
 67 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    |    ↪ текстовый документ.txt                                                                                 |
 68 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 69 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
 70 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    |    ↪ текстовый документ.txt                                                                                 |
 71 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 72 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
 73 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    |    ↪ текстовый документ.txt                                                                                 |
 74 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    |    ↪ текстовый документ.txt                                                                                 |
 75 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    |    ↪ текстовый документ.txt                                                                                 |
 76 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    |    ↪ текстовый документ.txt                                                                                 |
 77 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 78 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 79 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 80 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 81 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 82 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
 83 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 84 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 85 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 86 |Time: 21-11-2017 20:44:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 87 |Time: 21-11-2017 20:44:40 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 88 |Time: 21-11-2017 20:44:40 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 89 |Time: 21-11-2017 20:44:40 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 90 |Time: 21-11-2017 20:44:40 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt:
    |    ↪ Zone.Identifier                                                                                        |
 91 |Time: 21-11-2017 20:44:40 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
 92 |Time: 21-11-2017 20:44:40 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 93 |Time: 21-11-2017 20:44:40 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 94 |Time: 21-11-2017 20:44:43 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 95 |Time: 21-11-2017 20:44:44 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 96 |Time: 21-11-2017 20:44:44 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 97 |Time: 21-11-2017 20:44:44 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 98 |Time: 21-11-2017 20:44:45 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
 99 |Time: 21-11-2017 20:44:45 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
100 |Time: 21-11-2017 20:44:45 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt:
    |    ↪ Zone.Identifier                                                                                        |
101 |Time: 21-11-2017 20:44:45 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
102 |Time: 21-11-2017 20:44:45 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
103 |Time: 21-11-2017 20:44:45 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
104 |Time: 21-11-2017 20:44:48 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
105 |Time: 21-11-2017 20:44:48 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
106 |Time: 21-11-2017 20:44:48 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt               |
107 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
108 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
109 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
110 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
111 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
112 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
113 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
114 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
115 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder                    |
116 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder                    |
117 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder                    |
118 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
119 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
120 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder2                             |
121 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
122 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder2                             |
123 |Time: 21-11-2017 20:44:51 | GetFileAttributesW | User: Tom | File: E:\testFolder2                             |
124 |Time: 21-11-2017 20:44:52 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
125 |Time: 21-11-2017 20:44:52 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
126 |Time: 21-11-2017 20:44:52 | GetFileAttributesW | User: Tom | File: E:\testFolder                              |
127 |Time: 21-11-2017 20:44:52 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
128 |Time: 21-11-2017 20:44:53 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty                       |
```

```
129 Time: 21-11-2017 20:44:53 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
130 Time: 21-11-2017 20:44:53 | GetFileAttributesW | User: Tom | File: E:\testFolder
131 Time: 21-11-2017 20:44:53 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
132 Time: 21-11-2017 20:44:53 | GetFileAttributesW | User: Tom | File: E:\testFolder
133 Time: 21-11-2017 20:44:53 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
134 Time: 21-11-2017 20:44:53 | GetFileAttributesW | User: Tom | File: E:\testFolder
135 Time: 21-11-2017 20:44:53 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
136 Time: 21-11-2017 20:44:53 | GetFileAttributesW | User: Tom | File: E:\testFolder
```

Листинг 5.17: Лог

Как и ожидалось были перехвачены функции, перехват которых успешно работал, перехватчик для **GetFileAttributesW** был вынесен в отдельный лог из-за обилия вывода.

5.4.2 С перемещением в корзину

После перемещения файла и папки в корзину, она была очищена.

```
1 Time: 21-11-2017 20:45:22 | CreateDirectoryW | User: Tom | Path: E:\testFolder\ Новая папка
2 Time: 21-11-2017 20:45:34 | ReadFile | User: Tom | Path: E:\testFolder\qwerty\123.txt
3 Time: 21-11-2017 20:45:43 | DeleteFileW | User: Tom | File: E:\$RECYCLE.BIN\S
    ↪ -1-5-21-2936131933-2071197896-2896244546-1000\$IG5ZRT1.txt
4 Time: 21-11-2017 20:45:43 | RemoveDirectoryW | User: Tom | Path: E:\$RECYCLE.BIN\S
    ↪ -1-5-21-2936131933-2071197896-2896244546-1000\$RIF9MT8
5 Time: 21-11-2017 20:45:43 | DeleteFileW | User: Tom | File: E:\$RECYCLE.BIN\S
    ↪ -1-5-21-2936131933-2071197896-2896244546-1000\$IIF9MT8
```

Листинг 5.18: Лог

```
1 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder
2 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder
3 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
4 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
5 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
6 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
7 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
8 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
9 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
10 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder
11 Time: 21-11-2017 20:45:22 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
12 Time: 21-11-2017 20:45:24 | GetFileAttributesW | User: Tom | File: E:\testFolder
13 Time: 21-11-2017 20:45:24 | GetFileAttributesW | User: Tom | File: E:\testFolder
14 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder
15 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder
16 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
17 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder
18 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\ Новая папка
19 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
20 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
21 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder
22 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
23 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
24 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
25 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
26 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
27 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
28 Time: 21-11-2017 20:45:25 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
29 Time: 21-11-2017 20:45:26 | GetFileAttributesW | User: Tom | File: E:\testFolder
30 Time: 21-11-2017 20:45:26 | GetFileAttributesW | User: Tom | File: E:\testFolder
31 Time: 21-11-2017 20:45:26 | GetFileAttributesW | User: Tom | File: E:\testFolder
32 Time: 21-11-2017 20:45:26 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
33 Time: 21-11-2017 20:45:26 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
34 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    ↪ текстовый документ.txt
35 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    ↪ текстовый документ.txt
36 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    ↪ текстовый документ.txt
37 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
38 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder
39 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    ↪ текстовый документ.txt
40 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
41 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder
42 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    ↪ текстовый документ.txt
43 Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
    ↪ текстовый документ.txt
```

```
44 |Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
   |    ↪ текстовый документ.txt
45 |Time: 21-11-2017 20:45:29 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\ Новый
   |    ↪ текстовый документ.txt
46 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
47 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
48 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
49 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
50 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
51 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
52 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
53 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
54 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder
55 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
56 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
57 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
58 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt:
   |    ↪ Zone.Identifier
59 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
60 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
61 |Time: 21-11-2017 20:45:31 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
62 |Time: 21-11-2017 20:45:34 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
63 |Time: 21-11-2017 20:45:34 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
64 |Time: 21-11-2017 20:45:34 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
65 |Time: 21-11-2017 20:45:34 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
66 |Time: 21-11-2017 20:45:35 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
67 |Time: 21-11-2017 20:45:35 | GetFileAttributesW | User: Tom | File: E:\testFolder
68 |Time: 21-11-2017 20:45:35 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
69 |Time: 21-11-2017 20:45:36 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty\123.txt
70 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder
71 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
72 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
73 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
74 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
75 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
76 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
77 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder
78 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
79 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder
80 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
81 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder
82 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder
83 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder
84 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder2
85 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder2
86 |Time: 21-11-2017 20:45:37 | GetFileAttributesW | User: Tom | File: E:\testFolder2
87 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
88 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
89 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
90 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder
91 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
92 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder
93 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
94 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder
95 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
96 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder
97 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
98 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder
99 |Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
100|Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder
101|Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
102|Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
103|Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
104|Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder
105|Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
106|Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
107|Time: 21-11-2017 20:45:38 | GetFileAttributesW | User: Tom | File: E:\testFolder
108|Time: 21-11-2017 20:45:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
109|Time: 21-11-2017 20:45:39 | GetFileAttributesW | User: Tom | File: E:\testFolder
110|Time: 21-11-2017 20:45:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
111|Time: 21-11-2017 20:45:39 | GetFileAttributesW | User: Tom | File: E:\testFolder
112|Time: 21-11-2017 20:45:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\qwerty
113|Time: 21-11-2017 20:45:39 | GetFileAttributesW | User: Tom | File: E:\testFolder
114|Time: 21-11-2017 20:45:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
115|Time: 21-11-2017 20:45:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
116|Time: 21-11-2017 20:45:39 | GetFileAttributesW | User: Tom | File: E:\testFolder\logFolder
117|Time: 21-11-2017 20:45:40 | GetFileAttributesW | User: Tom | File: E:\testFolder2
118|Time: 21-11-2017 20:45:40 | GetFileAttributesW | User: Tom | File: E:\testFolder2
119|Time: 21-11-2017 20:45:40 | GetFileAttributesW | User: Tom | File: E:\testFolder2
120|Time: 21-11-2017 20:45:40 | GetFileAttributesW | User: Tom | File: E:\testFolder
121|Time: 21-11-2017 20:45:47 | GetFileAttributesW | User: Tom | File: E:\testFolder
122|Time: 21-11-2017 20:45:47 | GetFileAttributesW | User: Tom | File: E:\testFolder2
123|Time: 21-11-2017 20:45:47 | GetFileAttributesW | User: Tom | File: E:\testFolder2
124|Time: 21-11-2017 20:45:47 | GetFileAttributesW | User: Tom | File: E:\testFolder2
125|Time: 21-11-2017 20:45:47 | GetFileAttributesW | User: Tom | File: E:\testFolder
126|Time: 21-11-2017 20:45:47 | GetFileAttributesW | User: Tom | File: E:\testFolder
```

Листинг 5.19: Лог

В данном случае удаление файла также было успешно перехвачено. При удалении, имена удаляемых папок и корзины были измененными.

5.5 Итоговый код

Всю структуру кода можно разделить на следующие части:

1. Подключение файлов заголовков;
2. Вспомогательные функции и переменные;
3. определение типов и параметров функций для перехвата;
4. Получение указателей на оригинальные функции;
5. Функции перехватчики, задача которых логировать действия с файлами и папками;
6. Точка входа, где происходит инъекция функций перехватчиков в **kernel32.dll**.

```
1  #include "stdafx.h"
2  #include <stdio.h>
3  #include <fstream>
4  #include "mhook/mhook-lib/mhook.h"
5  #include <atlstr.h>
6  #include <ctime>
7  #include <Lmcons.h>
8  #include <psapi.h>
9  #include <strsafe.h>
10 using namespace std;
11
12 #define PATH "E:\\testFolder"
13 #define PATH_RECYCLE "E:\\"
14
15 // Для определения времени
16 time_t rawtime;
17 struct tm * timeinfo;
18 char buffer[80];Для
19
20 // определения пользователя
21 TCHAR name[UNLEN + 1];
22
23 // для определения имени файла
24 #define BUFSIZE 512
25 TCHAR pszFilename[MAX_PATH + 1];
26
27 ////////////////////////
28 // вспомогательные функции
29 void SetupTime() {
30     time(&rawtime);
31     timeinfo = localtime(&rawtime);
32     strftime(buffer, sizeof(buffer), "%d-%m-%Y %H:%M:%S", timeinfo);
33 }
34
```

```
35 void SetupUserName() {
36     DWORD size = UNLEN + 1;
37     int result = GetUserName((TCHAR*)name, &size);
38     if(result == 0)
39         _tcscpy(name, _T("Error to get user name"));
40 }
41
42 // получение имени по дескриптору файла
43 BOOL GetFileNameFromHandle(HANDLE hFile)
44 {
45     // переменная для оповещения успеха
46     BOOL bSuccess = FALSE;
47     HANDLE hFileMap;
48
49     // получение размера файла
50     DWORD dwFileSizeHi = 0;
51     DWORD dwFileSizeLo = GetFileSize(hFile, &dwFileSizeHi);
52
53     // если размер равен 0
54     if (dwFileSizeLo == 0 && dwFileSizeHi == 0)
55     {
56         return FALSE;
57     }
58
59     // создание отображения
60     hFileMap = CreateFileMapping(hFile,
61         NULL,
62         PAGE_READONLY,
63         0,
64         1,
65         NULL);
66
67     if (hFileMap)
68     {
69         // получаем указатель на участок памяти с отображением
70         void* pMem = MapViewOfFile(hFileMap, FILE_MAP_READ, 0, 0, 1);
71
72         if (pMem)
73         {
74             if (GetMappedFileName(GetCurrentProcess(),
75                 pMem,
76                 pszFilename,
77                 MAX_PATH))
78             {
79
80                 // Нормализуем название
81                 TCHAR szTemp[BUFSIZE];
82                 szTemp[0] = '\0';
83
84                 if (GetLogicalDriveStrings(BUFSIZE - 1, szTemp))
85                 {
86                     TCHAR szName[MAX_PATH];
87                     TCHAR szDrive[3] = TEXT(" :");
88                     BOOL bFound = FALSE;
89                     TCHAR* p = szTemp;
90
91                     do
92                     {
93                         // Копируем имя диска во временную переменную
94                         *szDrive = *p;
```

```
95  |
96  |                           // Ищем необходимое устройство – диск
97  |                           if (QueryDosDevice(szDrive, szName, MAX_PATH))
98  |                           {
99  |                               size_t uNameLen = _tcslen(szName);
100 |
101 |                               if (uNameLen < MAX_PATH)
102 |                               {
103 |                                   bFound = _tcsnicmp(pszFilename, szName,
    |   ↪ uNameLen) == 0
104 |                                       && *(pszFilename + uNameLen) == _T('\\');
105 |
106 |                                   if (bFound)
107 |                                   {
108 |                                       // чтение имени файла
109 |                                       TCHAR szTempFile[MAX_PATH];
110 |                                       StringCchPrintf(szTempFile,
111 |                                           MAX_PATH,
112 |                                           TEXT("%s%s"),
113 |                                           szDrive,
114 |                                           pszFilename + uNameLen);
115 |                                       StringCchCopyN(pszFilename, MAX_PATH + 1,
    |   ↪ szTempFile, _tcslen(szTempFile));
116 |                                   }
117 |                               }
118 |                           }
119 |
120 |                           // Пока есть указатель
121 |                           while (*p++);
122 |                       } while (!bFound && *p); // конец имени
123 |                   }
124 |               }
125 |               bSuccess = TRUE;
126 |               UnmapViewOfFile(pMem);
127 |           }
128 |
129 |           CloseHandle(hFileMap);
130 |       }
131 |       _tprintf(TEXT("File name is %s\n"), pszFilename);
132 |       return(bSuccess);
133 | }
134 |
135 |
136 | ////////////////////////
137 | // определения функций
138 |
139 | typedef bool(WINAPI* _CreateDirectory) (LPCTSTR, LPSECURITY_ATTRIBUTES);
140 | typedef HANDLE (WINAPI  *_CreateFile)(
141 |     _In_     LPCTSTR               lpFileName,
142 |     _In_     DWORD                 dwDesiredAccess,
143 |     _In_     DWORD                 dwShareMode,
144 |     _In_opt_ LPSECURITY_ATTRIBUTES lpSecurityAttributes,
145 |     _In_     DWORD                 dwCreationDisposition,
146 |     _In_     DWORD                 dwFlagsAndAttributes,
147 |     _In_opt_ HANDLE                hTemplateFile
148 |     );
149 |
150 | typedef BOOL(WINAPI* _MoveFile)(
151 |     _In_ LPCTSTR lpExistingFileName,
152 |     _In_ LPCTSTR lpNewFileName
```

```
153 );
154 typedef BOOL (WINAPI* _RemoveDirectory)(
155     _In_ LPCTSTR lpPathName
156 );
157 typedef BOOL(WINAPI* _DeleteFile)(
158     _In_ LPCTSTR lpFileName
159 );
160 typedef BOOL (WINAPI* _CopyFile)(
161     _In_ LPCTSTR lpExistingFileName,
162     _In_ LPCTSTR lpNewFileName,
163     _In_ BOOL    bFailIfExists
164 );
165 typedef DWORD(WINAPI* _GetFileAttributes)(
166     _In_ LPCTSTR lpFileName
167 );
168 typedef BOOL(WINAPI* _ReplaceFile)(
169     _In_       LPCTSTR lpReplacedFileName,
170     _In_       LPCTSTR lpReplacementFileName,
171     _In_opt_   LPCTSTR lpBackupFileName,
172     _In_       DWORD   dwReplaceFlags,
173     _Reserved_ LPVOID  lpExclude,
174     _Reserved_ LPVOID  lpReserved
175 );
176
177 typedef INT(WINAPI* _LZOpenFile)(
178     _In_  LPTSTR     lpFileName,
179     _Out_ LPOFSTRUCT lpReOpenBuf,
180     _In_  WORD       wStyle
181 );
182
183 typedef BOOL (WINAPI* _ReadFile)(
184     _In_         HANDLE       hFile,
185     _Out_        LPVOID       lpBuffer,
186     _In_         DWORD        nNumberOfBytesToRead,
187     _Out_opt_    LPDWORD      lpNumberOfBytesRead,
188     _Inout_opt_ LPOVERLAPPED lpOverlapped
189 );
190
191 typedef HFILE (WINAPI* _OpenFile)(
192     _In_  LPCSTR     lpFileName,
193     _Out_ LPOFSTRUCT lpReOpenBuff,
194     _In_  UINT       uStyle
195 );
196
197 typedef BOOL (WINAPI* _DeleteFileTransacted)(
198     _In_ LPCTSTR lpFileName,
199     _In_ HANDLE  hTransaction
200 );
201
202
203 ////////////////////////
204 // оригинальные функции
205
206 _CreateDirectory RealCreateDirectory = (_CreateDirectory)GetProcAddress(
    ↪ GetModuleHandle(L"Kernel32"), "CreateDirectoryW");
207 _CreateFile RealCreateFile = (_CreateFile)GetProcAddress(GetModuleHandle(L"
    ↪ Kernel32"), "CreateFileW");
208 _MoveFile RealMoveFile = (_MoveFile)GetProcAddress(GetModuleHandle(L"Kernel32")
    ↪ , "MoveFileW");
209 _RemoveDirectory RealRemoveDirectory = (_RemoveDirectory)GetProcAddress(
```

```
    ↪ GetModuleHandle(L"Kernel32"), "RemoveDirectoryW");
_DeleteFile RealDeleteFileW = (_DeleteFile)GetProcAddress(GetModuleHandle(L"
    ↪ Kernel32"), "DeleteFileW");
_DeleteFile RealDeleteFileA = (_DeleteFile)GetProcAddress(GetModuleHandle(L"
    ↪ Kernel32"), "DeleteFileA");
_CopyFile RealCopyFile = (_CopyFile)GetProcAddress(GetModuleHandle(L"Kernel32")
    ↪ , "CopyFileW");
_GetFileAttributes RealGetFileAttributes = (_GetFileAttributes)GetProcAddress(
    ↪ GetModuleHandle(L"Kernel32"), "GetFileAttributesW");
_ReplaceFile RealReplaceFile = (_ReplaceFile)GetProcAddress(GetModuleHandle(L"
    ↪ Kernel32"), "ReplaceFileW");
_LZOpenFile RealLZOpenFileW = (_LZOpenFile)GetProcAddress(GetModuleHandle(L"
    ↪ Kernel32"), "LZOpenFileW");
_ReadFile RealReadFile = (_ReadFile)GetProcAddress(GetModuleHandle(L"Kernel32")
    ↪ , "ReadFile");
_OpenFile RealOpenFile = (_OpenFile)GetProcAddress(GetModuleHandle(L"Kernel32")
    ↪ , "OpenFile");
_DeleteFileTransacted RealDeleteFileTransactededW = (_DeleteFileTransacted)
    ↪ GetProcAddress(GetModuleHandle(L"Kernel32"), "DeleteFileTransactedW");

////////////////////////
// функции перехватчики

BOOL WINAPI HookDeleteFileTransactededW(
    _In_ LPCTSTR lpFileName,
    _In_ HANDLE  hTransaction) {

    // установка времени и определение имени пользователя
    SetupTime();
    SetupUserName();

    // преобразование формата
    CStringA stringLpFileName(lpFileName);
    const char* charLpFileName = stringLpFileName;

    // проверка пути отлючена намеренно, для большего лога
    //if (strstr(charLpPathName, PATH)) {
    wofstream myfile;
    myfile.open("E:\\log.txt", ios::app);
    myfile << "Time: " << buffer << " | DeleteFileTransactedeW | User: " <<
    ↪ name << " | Path: " << charLpFileName << endl;
    myfile.close();
    //}

    return RealDeleteFileTransactededW(lpFileName, hTransaction);
}

BOOL WINAPI HookOpenFile(
    _In_  LPCSTR     lpFileName,
    _Out_ LPOFSTRUCT lpReOpenBuff,
    _In_  UINT       uStyle) {

    SetupTime();
    SetupUserName();

    CStringA stringLpFileName(lpFileName);
    const char* charLpFileName = stringLpFileName;

    // проверка пути отлючена намеренно, для большего лога
```

```
259     //if (strstr(charLpPathName, PATH)) {
260     wofstream myfile;
261     myfile.open("E:\\log.txt", ios::app);
262     myfile << "Time: " << buffer << " | OpenFile | User: " << name << " | Path:
    ↪  "<< charLpFileName << endl;
263     myfile.close();
264     //}
265
266     return RealOpenFile(lpFileName, lpReOpenBuff, uStyle);
267 }
268
269 BOOL WINAPI HookReadFile(
270     _In_        HANDLE       hFile,
271     _Out_       LPVOID       lpBuffer,
272     _In_        DWORD        nNumberOfBytesToRead,
273     _Out_opt_   LPDWORD      lpNumberOfBytesRead,
274     _Inout_opt_ LPOVERLAPPED lpOverlapped) {
275
276     SetupTime();
277     SetupUserName();
278
279     // проверка на успешное получение имени файла по его дескриптору
280     if (GetFileNameFromHandle(hFile)) {
281         CStringA stringLpFileName(pszFilename);
282         const char* charLpFileName = stringLpFileName;
283         if (strstr(charLpFileName, PATH)) {
284             wofstream myfile;
285             myfile.open("E:\\log.txt", ios::app);
286             myfile << "Time: " << buffer << " | ReadFile | User: " << name << "
    ↪  | Path: " << charLpFileName << endl;
287             myfile.close();
288         }
289     }
290     return RealReadFile(hFile, lpBuffer, nNumberOfBytesToRead,
    ↪ lpNumberOfBytesRead, lpOverlapped);
291 }
292
293 INT WINAPI HookLZOpenFileW(
294     _In_  LPTSTR     lpFileName,
295     _Out_ LPOFSTRUCT lpReOpenBuf,
296     _In_  WORD       wStyle) {
297     SetupTime();
298     SetupUserName();
299
300     CStringA stringLpFileName(lpFileName);
301     const char* charLpFileName = stringLpFileName;
302
303     // проверка пути отлючена намеренно, для большего лога
304     //if (strstr(charLpPathName, PATH)) {
305         wofstream myfile;
306         myfile.open("E:\\log.txt", ios::app);
307         myfile << "Time: " << buffer << " | LZOpenFileW | User: " << name << "
    ↪ | Path: " << charLpFileName << " | style: "<< wStyle << endl;
308         myfile.close();
309     //}
310
311     return RealLZOpenFileW(lpFileName, lpReOpenBuf, wStyle);
312 }
313
314 bool WINAPI HookCreateDirectoryW(LPCTSTR lpPathName, LPSECURITY_ATTRIBUTES
```

```
    ↪ lpSecurityAttributes) {
315     SetupTime();
316     SetupUserName();
317
318     CStringA stringLpPathName(lpPathName);
319     const char* charLpPathName = stringLpPathName;
320
321     if (strstr(charLpPathName, PATH)) {
322         wofstream myfile;
323         myfile.open("E:\\log.txt", ios::app);
324         myfile << "Time: " << buffer << " | CreateDirectoryW | User: " << name
    ↪ << " | Path: " << charLpPathName << endl;
325         myfile.close();
326     }
327
328     return RealCreateDirectory(lpPathName, lpSecurityAttributes);
329 }
330
331 HANDLE WINAPI HookCreateFileW(
332     _In_     LPCTSTR               lpFileName,
333     _In_     DWORD                 dwDesiredAccess,
334     _In_     DWORD                 dwShareMode,
335     _In_opt_ LPSECURITY_ATTRIBUTES lpSecurityAttributes,
336     _In_     DWORD                 dwCreationDisposition,
337     _In_     DWORD                 dwFlagsAndAttributes,
338     _In_opt_ HANDLE                hTemplateFile
339 ){
340     SetupTime();
341     SetupUserName();
342
343     CStringA stringLpFileName(lpFileName);
344     const char* charLpFileName = stringLpFileName;
345
346     if (strstr(charLpFileName, PATH)) {
347         wofstream myfile;
348         myfile.open("E:\\log.txt", ios::app);
349         myfile << "Time: " << buffer << " | CreateFileW | User: " << " | Path:
    ↪ " << endl;
350         myfile.close();
351     }
352
353     return RealCreateFile(lpFileName, dwDesiredAccess, dwShareMode,
    ↪ lpSecurityAttributes,
354         dwCreationDisposition, dwFlagsAndAttributes, hTemplateFile);
355 }
356
357 bool WINAPI HookMoveFile(LPCTSTR lpExistingFileName, LPCTSTR lpNewFileName) {
358     SetupTime();
359     SetupUserName();
360
361     CStringA stringLpExistingFileName(lpExistingFileName);
362     const char* charLpExistingFileName = stringLpExistingFileName;
363
364     CStringA stringLpNewFileName(lpNewFileName);
365     const char* charLpNewFileName = stringLpNewFileName;
366
367     if (strstr(charLpExistingFileName, PATH)) {
368         wofstream myfile;
369         myfile.open("E:\\log.txt", ios::app);
370         myfile << "Time: " << buffer << " | MoveFileW | USER: "<< name<<" | Old
```

```
    ↪ path: " << charLpExistingFileName << " | New path: "<< charLpNewFileName
    ↪ << endl;
371         myfile.close();
372     }
373
374     return RealMoveFile(lpExistingFileName, lpNewFileName);
375 }
376
377 bool WINAPI HookRemoveDirectoryW(LPCTSTR lpPathName) {
378     SetupTime();
379     SetupUserName();
380
381     CStringA stringLpPathName(lpPathName);
382     const char* charLpPathName = stringLpPathName;
383
384     if (strstr(charLpPathName, PATH_RECYCLE)) {
385         wofstream myfile;
386         myfile.open("E:\\log.txt", ios::app);
387         myfile << "Time: " << buffer << " | RemoveDirectoryW | User: " << name
    ↪ << " | Path: " << charLpPathName << endl;
388         myfile.close();
389     }
390
391     return RealRemoveDirectory(lpPathName);
392 }
393
394 bool WINAPI HookDeleteFileW(LPCTSTR lpFileName) {
395     SetupTime();
396     SetupUserName();
397
398     CStringA stringLpFileName(lpFileName);
399     const char* charLpFileName = stringLpFileName;
400
401     if (strstr(charLpFileName, PATH_RECYCLE)) {
402         wofstream myfile;
403         myfile.open("E:\\log.txt", ios::app);
404         myfile << "Time: " << buffer << " | DeleteFileW | User: " << name << "
    ↪ | File: " << charLpFileName << endl;
405         myfile.close();
406     }
407
408     return RealDeleteFileW(lpFileName);
409 }
410
411 bool WINAPI HookDeleteFileA(LPCTSTR lpFileName) {
412     SetupTime();
413     SetupUserName();
414
415     CStringA stringLpFileName(lpFileName);
416     const char* charLpFileName = stringLpFileName;
417
418     if (strstr(charLpFileName, PATH)) {
419         wofstream myfile;
420         myfile.open("E:\\log.txt", ios::app);
421         myfile << "Time: " << buffer << " | DeleteFileA | User: " << name << "
    ↪ | File: " << charLpFileName << endl;
422         myfile.close();
423     }
424
425     return RealDeleteFileA(lpFileName);
```

```
426 }
427
428 bool WINAPI HookCopyFileW(_In_ LPCTSTR lpExistingFileName,
429     _In_ LPCTSTR lpNewFileName,
430     _In_ BOOL    bFailIfExists) {
431     SetupTime();
432     SetupUserName();
433
434     CStringA stringLpExistingFileName(lpExistingFileName);
435     const char* charLpExistingFileName = stringLpExistingFileName;
436
437     CStringA stringLpNewFileName(lpNewFileName);
438     const char* charLpNewFileName = stringLpNewFileName;
439
440     if (strstr(charLpExistingFileName, PATH)) {
441         wofstream myfile;
442         myfile.open("E:\\log.txt", ios::app);
443         myfile << "Time: " << buffer << " | CopyFileW | User: " << name << " |
    ↪ Old name: " << charLpExistingFileName <<" | New name: "<<
    ↪ charLpNewFileName << endl;
444         myfile.close();
445     }
446
447     return RealCopyFile(lpExistingFileName, lpNewFileName, bFailIfExists);
448 }
449
450 DWORD WINAPI HookGetFileAttributes(LPCTSTR lpFileName) {
451     SetupTime();
452     SetupUserName();
453
454     CStringA stringLpFileName(lpFileName);
455     const char* charLpFileName = stringLpFileName;
456
457     if (strstr(charLpFileName, PATH)) {
458         wofstream myfile;
459         myfile.open("E:/testFolder2/log.txt", ios::app);
460         myfile << "Time: " << buffer << " | GetFileAttributesW | User: " <<
    ↪ name << " | File: " << charLpFileName << endl;
461         myfile.close();
462     }
463
464     return RealGetFileAttributes(lpFileName);
465 }
466
467 bool WINAPI HookReplaceFile(
468     _In_       LPCTSTR lpReplacedFileName,
469     _In_       LPCTSTR lpReplacementFileName,
470     _In_opt_   LPCTSTR lpBackupFileName,
471     _In_       DWORD   dwReplaceFlags,
472     _Reserved_ LPVOID  lpExclude,
473     _Reserved_ LPVOID  lpReserved) {
474     SetupTime();
475     SetupUserName();
476
477     CStringA stringLpReplacedFileName(lpReplacedFileName);
478     const char* charLpReplacedFileName = stringLpReplacedFileName;
479
480     CStringA stringLpReplacementFileName(lpReplacementFileName);
481     const char* charLpReplacementFileName = stringLpReplacementFileName;
482
```

```
483     if (strstr(charLpReplacedFileName, PATH)) {
484         wofstream myfile;
485         myfile.open("E:\\log.txt", ios::app);
486         myfile << "Time: " << buffer << " | ReplaceFileW | User: " << name << "
    ↪  | Old file name: " << charLpReplacedFileName
487             << " | New file name: "<< charLpReplacementFileName << endl;
488         myfile.close();
489     }
490
491     return RealReplaceFile(lpReplacedFileName, lpReplacementFileName,
    ↪ lpBackupFileName,
492         dwReplaceFlags, lpExclude, lpReserved);
493 }
494
495 ////////////////////////
496 // точка входа
497
498 BOOL WINAPI DllMain(
499     __in HINSTANCE  hInstance,
500     __in DWORD      Reason,
501     __in LPVOID     Reserved
502 ){
503     switch (Reason)
504     {
505     case DLL_PROCESS_ATTACH:
506         Mhook_SetHook((PVOID*)&RealCreateDirectory, HookCreateDirectoryW);
507         //Mhook_SetHook((PVOID*)&RealCreateFile, HookCreateFileW); // система
    ↪ не запустится
508         Mhook_SetHook((PVOID*)&RealMoveFile, HookMoveFile);
509         Mhook_SetHook((PVOID*)&RealRemoveDirectory, HookRemoveDirectoryW);
510         Mhook_SetHook((PVOID*)&RealDeleteFileW, HookDeleteFileW);
511         Mhook_SetHook((PVOID*)&RealDeleteFileA, HookDeleteFileA);
512         Mhook_SetHook((PVOID*)&RealCopyFile, HookCopyFileW);
513         Mhook_SetHook((PVOID*)&RealGetFileAttributes, HookGetFileAttributes);
    ↪ // слишком большой лог
514         Mhook_SetHook((PVOID*)&RealReplaceFile, HookReplaceFile);
515         Mhook_SetHook((PVOID*)&RealLZOpenFileW, HookLZOpenFileW);
516         Mhook_SetHook((PVOID*)&RealReadFile, HookReadFile);
517         Mhook_SetHook((PVOID*)&RealOpenFile, HookOpenFile);
518         Mhook_SetHook((PVOID*)&RealDeleteFileTransactededW,
    ↪ HookDeleteFileTransactededW);
519         break;
520
521     case DLL_PROCESS_DETACH:
522         Mhook_Unhook((PVOID*)&RealCreateDirectory);
523         //Mhook_Unhook((PVOID*)&RealCreateFile);
524         Mhook_Unhook((PVOID*)&RealMoveFile);
525         Mhook_Unhook((PVOID*)&RealRemoveDirectory);
526         Mhook_Unhook((PVOID*)&RealDeleteFileW);
527         Mhook_Unhook((PVOID*)&RealDeleteFileA);
528         Mhook_Unhook((PVOID*)&RealCopyFile);
529         Mhook_Unhook((PVOID*)&RealGetFileAttributes);
530         Mhook_Unhook((PVOID*)&RealReplaceFile);
531         Mhook_Unhook((PVOID*)&RealLZOpenFileW);
532         Mhook_Unhook((PVOID*)&RealReadFile);
533         Mhook_Unhook((PVOID*)&RealOpenFile);
534         Mhook_Unhook((PVOID*)&RealDeleteFileTransactededW);
535         break;
536     }
537     return TRUE;
```

```
538 }
```

Листинг 5.20: main.cpp

Инъекция кода производилась посредством вызова функции **Mhook_Unhook** библиотеки **Mhook library**.

5.6 Mhook library

Для действий связанных с изменением исходного кода, его хранением, создании трамплинов, была использована библиотека **Mhook library**[14].

Mhook library - бесплатная библиотека с открытым исходным кодом для перехвата api, поддерживаются платформы x86 и x64.

В работе были использованы следующие функции библиотеки:

BOOL Mhook_SetHook(PVOID *ppSystemFunction, PVOID pHookFunction);
BOOL Mhook_Unhook(PVOID *ppHookedFunction);

Рассмотрим их работу более подробно.

5.6.1 Алгоритм работы Mhook_SetHook

Рассмотрим поэтапно алгоритм работы метода.

1. **EnterCritSec();**
    - Так как конечная система скорее всего будет многопоточной, необходимо обезопасить работу. Для этого используются функции **InitializeCriticalSection** и **EnterCriticalSection** соответственно инициализация и вход в критическую секцию. Это позволяет работать в данной секции лишь одному потоку.
2. **pSystemFunction = SkipJumps((PBYTE)pSystemFunction);**
3. **pHookFunction = SkipJumps((PBYTE)pHookFunction);**
    - Еще до установки перехватчика, адреса функций полученные с помощью вызова **GetProcAddress** ведут в таблицу прыжков на функции. Поэтому необходимо пропустить эту таблицу и перейти непосредственно к реальной функции.
4. **DWORD dwInstructionLength = DisassembleAndSkip(pSystemFunction, MHOOK_JMPSIZE, &patchdata);**
    - Изучение исходного кода функции, получение количества байт возможных для написание прыжка в функцию перехватчик. Изучение кода происходит путем побайтового чтения кода функции, затем каждый байт сравнивается с предварительно написанной таблицой опкодов, для последующего определения длины команды.
5. **if (dwInstructionLength >= MHOOK_JMPSIZE)**
    - Проверка на минимальную возможность прыжка(5 байт), если функция менее 5 байт, то прыжок выполнить невозможно.

6. **SuspendOtherThreads((PBYTE)pSystemFunction, dwInstructionLength);**

    - Перед изменением кода, необходимо приостановить выполнение прочих потоков, работающих в пределах изменяемого кода. Для этого используется системный вызов **CreateToolhelp32Snapshot**. Далее идет сравнение полученных id процессов с текущим. И наконец приостановка идет с помощью системного вызова **SuspendThread**.

7. **pTrampoline = TrampolineAlloc((PBYTE)pSystemFunction, patchdata.nLimitUp, patchdata.nLimitDown);**

    - Инициализация структуры трамплина. Инициализация происходит с помощью системного вызова **virtualalloc**, получение виртуальной памяти происходит в пределах 32 бит от оригинальной функции.

8. Снятие защиты с региона памяти с помощью функции **VirtualProtectEx**.

9. Сохранение в трамплин начальных инструкций оригинальной функции.

10. Определение адреса начала оригинальной функции, после функции прыжка.

11. **FixupIPRelativeAddressing(pTrampoline->codeTrampoline, (PBYTE)pSystemFunction, &patchdata);**

    - Фиксация относительной адресации(когда отступ идет от некоторого базового адреса).

12. Проверка на длину прыжка.

    - Если требуется большой прыжок(14 байт), создается дополнительный трамплин;
    - Иначе пишем прыжок в 5 байт.

13. Обновляем указатель на системную функцию;

14. Для оригинальной функции и перехватчика выполняется:

    (a) Обновление кеша инструкций с помощью **FlushInstructionCache**.

    (b) Установка защиты региона памяти с помощью **VirtualProtectEx**.

15. Восстановление работы остальных потоков и выход из критической секции.

Далее приведен сам исходный код функции, с добавлением некоторых комментариев.

```
1  BOOL Mhook_SetHook(PVOID *ppSystemFunction, PVOID pHookFunction) {
2      MHOOKS_TRAMPOLINE* pTrampoline = NULL;
3      PVOID pSystemFunction = *ppSystemFunction;
4      // обеспечение безопасности работы с многопоточностью
5      EnterCritSec();
6      ODPRINTF((L"mhooks: Mhook_SetHook: Started on the job: %p / %p", pSystemFunction,
       ↪ pHookFunction));
7      // поиск адреса реальной функции, пропуская таблицу прыжков
8      pSystemFunction = SkipJumps((PBYTE)pSystemFunction);
9      pHookFunction   = SkipJumps((PBYTE)pHookFunction);
10     ODPRINTF((L"mhooks: Mhook_SetHook: Started on the job: %p / %p", pSystemFunction,
       ↪ pHookFunction));
11     // получение длины зоны для записи прыжка на перехватчик
12     MHOOKS_PATCHDATA patchdata = {0};
13     DWORD dwInstructionLength = DisassembleAndSkip(pSystemFunction, MHOOK_JMPSIZE, &
       ↪ patchdata);
14     if (dwInstructionLength >= MHOOK_JMPSIZE) {
```

```
15 |         ODPRINTF((L"mhooks: Mhook_SetHook: disassembly signals %d bytes",
   |     ↪ dwInstructionLength));
16 |         // приостановка выполнения других процессов
17 |         // для того чтобы убедиться что небыло других процессов в перезаписанном коде
18 |         SuspendOtherThreads((PBYTE)pSystemFunction, dwInstructionLength);
19 |         // инициализации структуры трамплина (TODO: это весьма расточительно использовать
20 |         // VirtualAlloc для получения памяти менее чем в 100 байт)
21 |         pTrampoline = TrampolineAlloc((PBYTE)pSystemFunction, patchdata.nLimitUp,
   |     ↪ patchdata.nLimitDown);
22 |         if (pTrampoline) {
23 |             ODPRINTF((L"mhooks: Mhook_SetHook: allocated structure at %p", pTrampoline))
   |     ↪ ;
24 |             // open ourselves so we can VirtualProtectEx
25 |             HANDLE hProc = GetCurrentProcess();
26 |             DWORD dwOldProtectSystemFunction = 0;
27 |             DWORD dwOldProtectTrampolineFunction = 0;
28 |             // установка доступа к оригинальной функции на PAGE_EXECUTE_READWRITE
29 |             if (VirtualProtectEx(hProc, pSystemFunction, dwInstructionLength,
   |     ↪ PAGE_EXECUTE_READWRITE, &dwOldProtectSystemFunction)) {
30 |                 ODPRINTF((L"mhooks: Mhook_SetHook: readwrite set on system function"));
31 |                 // установка доступа к трамплину на PAGE_EXECUTE_READWRITE
32 |                 if (VirtualProtectEx(hProc, pTrampoline, sizeof(MHOOKS_TRAMPOLINE),
   |     ↪ PAGE_EXECUTE_READWRITE, &dwOldProtectTrampolineFunction)) {
33 |                     ODPRINTF((L"mhooks: Mhook_SetHook: readwrite set on trampoline
   |     ↪ structure"));
34 |
35 |                     // создание функциитрамплина—
36 |                     PBYTE pbCode = pTrampoline->codeTrampoline;
37 |                     // save original code..
38 |                     for (DWORD i = 0; i<dwInstructionLength; i++) {
39 |                         pTrampoline->codeUntouched[i] = pbCode[i] = ((PBYTE)
   |     ↪ pSystemFunction)[i];
40 |                     }
41 |                     pbCode += dwInstructionLength;
42 |                     // определение адреса начала оригинальной функции
43 |                     pbCode = EmitJump(pbCode, ((PBYTE)pSystemFunction) +
   |     ↪ dwInstructionLength);
44 |                     ODPRINTF((L"mhooks: Mhook_SetHook: updated the trampoline"));
45 |
46 |                     // фиксация относительной адресации
47 |                     FixupIPRelativeAddressing(pTrampoline->codeTrampoline, (PBYTE)
   |     ↪ pSystemFunction, &patchdata);
48 |
49 |                     DWORD_PTR dwDistance = (PBYTE)pHookFunction < (PBYTE)pSystemFunction
   |     ↪  ?
50 |                         (PBYTE)pSystemFunction - (PBYTE)pHookFunction : (PBYTE)
   |     ↪ pHookFunction - (PBYTE)pSystemFunction;
51 |                     if (dwDistance > 0x7fff0000) {
52 |                         // создание дополнительного трамплина, с большим прыжком
53 |                         // на 14 байтдля( x64 систем)
54 |                         pbCode = pTrampoline->codeJumpToHookFunction;
55 |                         pbCode = EmitJump(pbCode, (PBYTE)pHookFunction);
56 |                         ODPRINTF((L"mhooks: Mhook_SetHook: created reverse trampoline"))
   |     ↪ ;
57 |                         FlushInstructionCache(hProc, pTrampoline->codeJumpToHookFunction
   |     ↪ ,
58 |                             pbCode - pTrampoline->codeJumpToHookFunction);
59 |
60 |                         // обновляем указатель на системную функцию
61 |                         pbCode = (PBYTE)pSystemFunction;
62 |                         pbCode = EmitJump(pbCode, pTrampoline->codeJumpToHookFunction);
63 |                         ODPRINTF((L"mhooks: Mhook_SetHook: Hooked the function!"));
64 |                     } else {
65 |                         // прыжок займет лишь 5 байт
66 |                         ODPRINTF((L"mhooks: Mhook_SetHook: no need for reverse
   |     ↪ trampoline"));
67 |                         // обновляем указатель на системную функцию
68 |                         pbCode = (PBYTE)pSystemFunction;
69 |                         pbCode = EmitJump(pbCode, (PBYTE)pHookFunction);
70 |                         ODPRINTF((L"mhooks: Mhook_SetHook: Hooked the function!"));
71 |                     }
72 |
73 |                     // обновляем структуру трамплина
74 |                     pTrampoline->cbOverwrittenCode = dwInstructionLength;
75 |                     pTrampoline->pSystemFunction = (PBYTE)pSystemFunction;
```

```
76                      pTrampoline->pHookFunction = (PBYTE)pHookFunction;
77
78                      // обновление кеша инструкций и установка защиты региона памяти
79                      FlushInstructionCache(hProc, pTrampoline->codeTrampoline,
    ↪ dwInstructionLength);
80                      VirtualProtectEx(hProc, pTrampoline, sizeof(MHOOKS_TRAMPOLINE),
    ↪ dwOldProtectTrampolineFunction, &dwOldProtectTrampolineFunction);
81                  } else {
82                      ODPRINTF((L"mhooks: Mhook_SetHook: failed VirtualProtectEx 2: %d",
    ↪ gle()));
83                  }
84                  // обновление кеша инструкций и установка защиты региона памяти
85                  FlushInstructionCache(hProc, pSystemFunction, dwInstructionLength);
86                  VirtualProtectEx(hProc, pSystemFunction, dwInstructionLength,
    ↪ dwOldProtectSystemFunction, &dwOldProtectSystemFunction);
87              } else {
88                  ODPRINTF((L"mhooks: Mhook_SetHook: failed VirtualProtectEx 1: %d", gle()
    ↪ ));
89              }
90              if (pTrampoline->pSystemFunction) {
91                  // обновление указателя на функцию трамплин
92                  *ppSystemFunction = pTrampoline->codeTrampoline;
93              } else {
94                  // если ошибка то обнуляем трамплин
95                  TrampolineFree(pTrampoline, TRUE);
96                  pTrampoline = NULL;
97              }
98          }
99          // восстановка работы остальных потоков
100         ResumeOtherThreads();
101     } else {
102         ODPRINTF((L"mhooks: disassembly signals %d bytes (unacceptable)",
    ↪ dwInstructionLength));
103     }
104     // выход из критической секции
105     LeaveCritSec();
106     return (pTrampoline != NULL);
107 }
```

Листинг 5.21: Mhook_SetHook

5.6.2 Алгоритм работы Mhook_Unhook

Рассмотрим поэтапно алгоритм работы метода.

1. получение структуры трамплина по переданному адресу;
2. приостановка других потоков;
3. получаем дескриптор текущего процесса;
4. снимаем защиту памяти;
5. берем из структуры трамплина код исходной функции;
6. обновление кеша инструкций и установка защиты памяти;
7. возвращаем оригинальный указатель;
8. освобождаем из памяти трамплин;
9. восстанавливаем работу других потоков.

```
1  BOOL Mhook_Unhook(PVOID *ppHookedFunction) {
2      ODPRINTF((L"mhooks: Mhook_Unhook: %p", *ppHookedFunction));
3      BOOL bRet = FALSE;
4      EnterCritSec();
5      // получение структуры трамплина по переданному адресу
6      MHOOKS_TRAMPOLINE* pTrampoline = TrampolineGet((PBYTE)*ppHookedFunction);
7      if (pTrampoline) {
8          // приостановка других потоков
9          SuspendOtherThreads(pTrampoline->pSystemFunction, pTrampoline->cbOverwrittenCode
       ↪ );
10         ODPRINTF((L"mhooks: Mhook_Unhook: found struct at %p", pTrampoline));
11         // получаем дескриптор текущего процесса
12         HANDLE hProc = GetCurrentProcess();
13         DWORD dwOldProtectSystemFunction = 0;
14         // снимаем защиту памяти
15         if (VirtualProtectEx(hProc, pTrampoline->pSystemFunction, pTrampoline->
       ↪ cbOverwrittenCode, PAGE_EXECUTE_READWRITE, &dwOldProtectSystemFunction)) {
16             ODPRINTF((L"mhooks: Mhook_Unhook: readwrite set on system function"));
17             // берем из структуры трамплина код исходной функции
18             PBYTE pbCode = (PBYTE)pTrampoline->pSystemFunction;
19             for (DWORD i = 0; i<pTrampoline->cbOverwrittenCode; i++) {
20                 pbCode[i] = pTrampoline->codeUntouched[i];
21             }
22             // обновление кеша инструкций и установка защиты памяти
23             FlushInstructionCache(hProc, pTrampoline->pSystemFunction, pTrampoline->
       ↪ cbOverwrittenCode);
24             VirtualProtectEx(hProc, pTrampoline->pSystemFunction, pTrampoline->
       ↪ cbOverwrittenCode, dwOldProtectSystemFunction, &dwOldProtectSystemFunction);
25             // возращаем оригинальный указатель
26             *ppHookedFunction = pTrampoline->pSystemFunction;
27             bRet = TRUE;
28             ODPRINTF((L"mhooks: Mhook_Unhook: sysfunc: %p", *ppHookedFunction));
29             // освобождаем из памяти трамплин
30             TrampolineFree(pTrampoline, FALSE);
31             ODPRINTF((L"mhooks: Mhook_Unhook: unhook successful"));
32         } else {
33             ODPRINTF((L"mhooks: Mhook_Unhook: failed VirtualProtectEx 1: %d", gle()));
34         }
35         // восстанавливаем работу других потоков
36         ResumeOtherThreads();
37     }
38     LeaveCritSec();
39     return bRet;
40 }
```

Листинг 5.22: Mhook_Unhook

Вывод

Данная работа оказалась весьма познавательной, так как тема перехвата действий мне всегда была интересной.

Во первых я узнал про базовые подходы перехвата Windows api, способы инъекций, а также проблемы при реализации.

Большую часть времени заняло написание функций-перехватчиков, так как:

- Некоторые функции(**CreateFileW**, приводили к неспособности системы загрузиться в нормальном режиме;
- У одного действия(например удалить файл), имеется множество функций с похожими именами. Как минимум имелась **ANCI**(DeleteFileA) функция и **UNICODE**(DeleteFileW) функция;
- Не всегда функции работали как предполагалось, например функция-перехватчик для DeleteFileW успешно залогировала удаление из корзины, но не смогла при удалении без корзины.

В целом сама тема с перехватом системных функций весьма перспективна, так как с одной стороны, на основе этих знаний пишутся хакерские программы, а с другой стороны антивирусное ПО.

Литература

[1] AppInit_DLLs [Электронный ресурс]. — URL: https://msdn.microsoft.com/en-us/library/windows/desktop/dd744762(v=vs.85).aspx (дата обращения: 2017-11-12).

[2] Obtaining a File Name From a File Handle [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa366789(v=vs.85).aspx (дата обращения: 2017-11-19).

[3] CreateDirectory function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa363855(v=vs.85).aspx (дата обращения: 2017-11-14).

[4] RemoveDirectory function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa365488(v=vs.85).aspx (дата обращения: 2017-11-19).

[5] ReadFile function function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa365467(v=vs.85).aspx (дата обращения: 2017-11-19).

[6] DeleteFile function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa363915(v=vs.85).aspx (дата обращения: 2017-11-20).

[7] CreateFile function [Электронный ресурс]. — URL: https://msdn.microsoft.com/en-us/library/windows/desktop/aa363858(v=vs.85).aspx (дата обращения: 2017-11-21).

[8] MoveFile function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa365239(v=vs.85).aspx (дата обращения: 2017-11-21).

[9] CopyFile function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa363851(v=vs.85).aspx (дата обращения: 2017-11-21).

[10] ReplaceFile function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa365512(v=vs.85).aspx (дата обращения: 2017-11-21).

[11] LZOpenFile function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa365225(v=vs.85).aspx (дата обращения: 2017-11-21).

[12] DeleteFile function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa363915(v=vs.85).aspx (дата обращения: 2017-11-21).

[13] DeleteFileTransacted function [Электронный ресурс]. — URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa363916(v=vs.85).aspx (дата обращения: 2017-11-21).

[14] A Windows API hooking library [Электронный ресурс]. — URL: https://github.com/martona/mhook (дата обращения: 2017-11-12).

[15] x86 Instruction Set Reference [Электронный ресурс]. — URL: http://x86.renejeschke.de/html/file_module_x86_id_147.html (дата обращения: 2017-11-12).

[16] Использование DllEntryPoint [Электронный ресурс]. — URL: http://www.codenet.ru/progr/cpp/DllEntryPoint.php (дата обращения: 2017-11-13).