

[WINDOWS] МОНИТОРИНГ ОБРАЩЕНИЙ К ЗАДАННОМУ ФАЙЛУ

Студент: **Д.В. Круминьш**

Группа: **13541/3**

Санкт-Петербургский Политехнический Университет Петра Великого

Необходимые функции мониторинга файла:

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

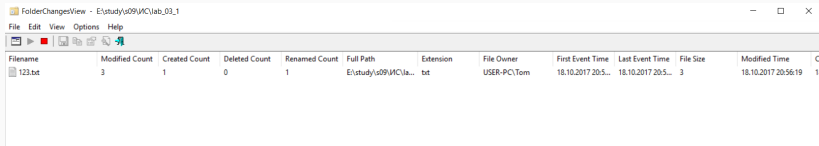
Утилита Process Monitor

Process Monitor - SystemTools.com

File Edit View Filter Tools Options Help

</

Утилита FolderChangesView



The screenshot shows the FolderChangesView application window. The title bar indicates the path E:\study\st09\WC\lab_03_1. The menu bar includes File, Edit, View, Options, and Help. The toolbar contains icons for file operations. The main display area shows a table with columns for file metadata and change counts.

Filename	Modified Count	Created Count	Deleted Count	Renamed Count	Full Path	Extension	File Owner	First Event Time	Last Event Time	File Size	Modified Time	Cr
123.bt	3	1	0	1	E:\study\st09\WC\la...	bt	USER-PC\Tom	18.10.2017 20:5...	18.10.2017 20:5...	3	18.10.2017 20:56:19	18

ВАРИАНТЫ РЕАЛИЗАЦИИ

Мониторинг папки с ее файлами, на предмет каких-либо изменений.

Плюсы:

- быстрота реализации;
- не требует прав администратора;
- мониторинг создания, изменения, удаления файлов в папке.

Минусы:

- невозможность мониторинга открытия/закрытия файлов.

Перехват API Windows. DLL инъекции.

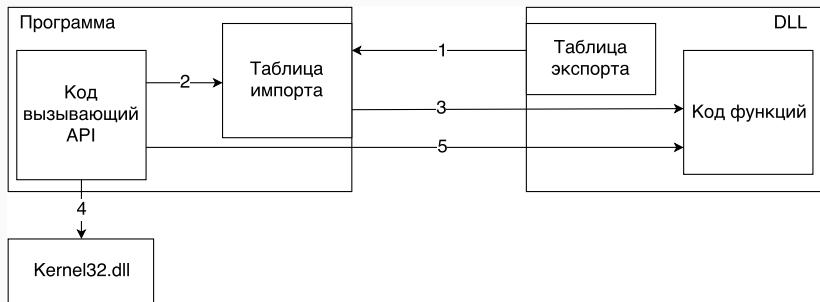
Плюсы:

- максимальные возможности в реализации.

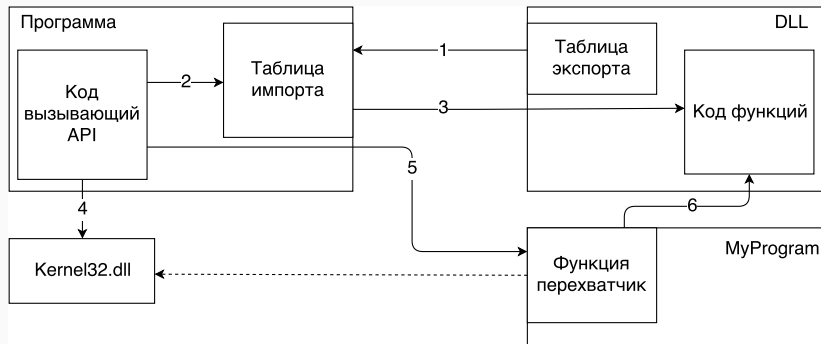
Минусы:

- более сложная реализация, по сравнению с вариантом 1;
- требуются права администратора.

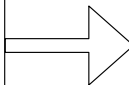
[ПЕРЕХВАТ WinAPI] ПРИНЦИП ВЫЗОВА ФУНКЦИИ



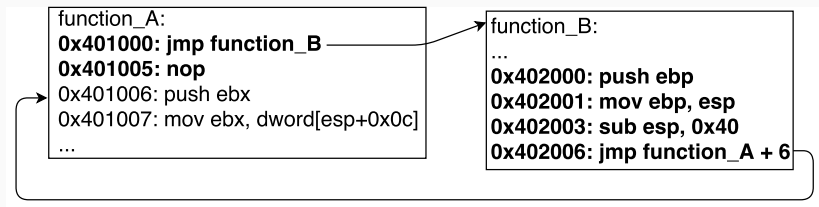
[ПЕРЕХВАТ WinAPI] ПРИНЦИП ПЕРЕХВАТА ФУНКЦИИ

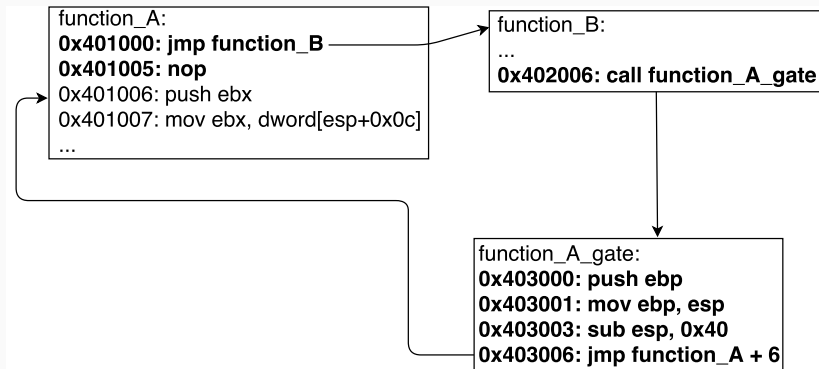


```
function_A:  
0x401000: push ebp  
0x401001: mov ebp, esp  
0x401003: sub esp, 0x40  
0x401006: push ebx  
0x401007: mov ebx, dword[esp+0x0c]  
...
```

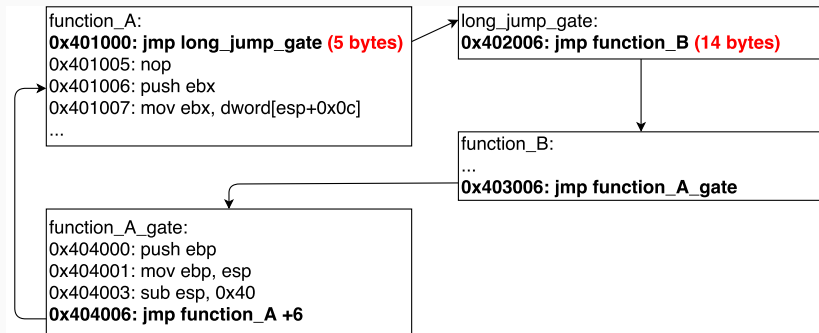


```
function_A:  
0x401000: jmp function_B  
0x401005: nop  
0x401006: push ebx  
0x401007: mov ebx, dword[esp+0x0c]  
...
```





[ПЕРЕХВАТ WINAPI] ПЕРЕХВАТ С ДВУМЯ ТРАМПЛИНАМИ



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;
- 12 34 56 78 12 34 56 78 - указатель на 64-битный адрес для перехода.

Чтобы внедрить DLL в процессы, которые связаны с kernel32.DLL, нужно добавить имя DLL в значение следующего раздела реестра:

**HKEY_LOCAL_MACHINE\Software\Microsoft\Windows
NT\CurrentVersion\Windows\ApplInit_DLLs**

Включение инъекции.

**HKEY_LOCAL_MACHINE\Software\Microsoft\Windows
NT\CurrentVersion\Windows\LoadApplInit_DLLs**

Принимает следующие значения:

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

```
1 HANDLE WINAPI CreateRemoteThread(  
2     _In_   HANDLE          hProcess ,  
3     _In_   LPSECURITY_ATTRIBUTES lpThreadAttributes ,  
4     _In_   SIZE_T          dwStackSize ,  
5     _In_   LPTHREAD_START_ROUTINE lpStartAddress ,  
6     _In_   LPVOID          lpParameter ,  
7     _In_   DWORD           dwCreationFlags ,  
8     _Out_  LPDWORD         lpThreadId  
9 );
```

- **hProcess** - дескриптор процесса, в котором будет создан поток;
- **lpThreadAttributes** - параметры безопасности;
- **dwStackSize** - начальный размер стека. Если передан 0, то используется стандартный размер(1 MB);
- **lpStartAddress** - адрес функции которую необходимо выполнить;
- **lpParameter** - параметр передаваемый в функцию;
- **dwCreationFlags** - дополнительные флаги;
- **lpThreadId** - для возврата идентификатора процесса.

```
1 // для загрузки библиотеки
2 LPVOID LoadLibAddr = (LPVOID)GetProcAddress(GetModuleHandleA("kernel32.
    ↪ dll"), "LoadLibraryA");
```

```
1 // для выгрузки библиотеки
2 LPVOID UnloadLibAddr = (LPVOID)GetProcAddress(GetModuleHandleA("
    ↪ kernel32.dll"), "FreeLibrary");
```



```
1 HANDLE WINAPI OpenProcess(  
2     _In_ DWORD dwDesiredAccess,  
3     _In_ BOOL bInheritHandle,  
4     _In_ DWORD dwProcessId  
5 );
```

```
1 HANDLE WINAPI CreateToolhelp32Snapshot(  
2     _In_ DWORD dwFlags,  
3     _In_ DWORD th32ProcessID  
4 );
```

```
1  ////////////////////////////////////
2  // точка входа
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         ...
13     case DLL_PROCESS_DETACH:
14         ...
15     }
16     return TRUE;
17 }
```

Прототип функции GetModuleHandle

```
1 HMODULE WINAPI GetModuleHandle(  
2     _In_opt_ LPCTSTR lpModuleName  
3 );
```

Получение дескриптора **kernel32.dll**

```
1 HMODULE hKernel32 = GetModuleHandle(L"kernel32.dll");
```

Прототип функции GetProcAddress

```
1 FARPROC WINAPI GetProcAddress(  
2     _In_ HMODULE hModule,  
3     _In_ LPCSTR lpProcName  
4 );
```

Получение дескриптора **kernel32.dll**

```
1 void* lpFunc = GetProcAddress(hKernel32, "CreateFileW");
```

В ходе разработки возникают следующие проблемы:

- отказ windows загружаться;
- сложности в определении ошибок;

РЕЗУЛЬТАТЫ РАБОТЫ

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

Listing 1: Удаление без перемещения в корзину

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

Listing 2: Удаление с перемещением в корзину

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

```
1  BOOL WINAPI ReadFile(  
2      _In_          HANDLE          hFile ,  
3      _Out_         LPVOID          lpBuffer ,  
4      _In_          DWORD           nNumberOfBytesToRead ,  
5      _Out_opt_     LPDWORD         lpNumberOfBytesRead ,  
6      _Inout_opt_   LPOVERLAPPED   lpOverlapped  
7  );
```

Listing 3: Прототип функции ReadFile


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

Listing 4: Фрагмент лога

По пути **E:/testFolder/logFolder/getFileAttr_log.txt** находился файл размером в 21 796 Кб.

$$21796/683 = 31.91 \text{ Кб.}$$

Функция возвращает атрибуты файла или директории.

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

Listing 5: Прототип функции GetFileAttributesW

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

Listing 6: Фрагмент лога

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

Listing 7: Прототип функции DeleteFileW

```
1 Time: 20-11-2017 20:37:38 | DeleteFileW | User: Tom | File: E:\$RECYCLE  
   ↳ .BIN\S-1-5-21-2936131933-2071197896-2896244546-1000\51F873NA.  
   ↳ txt
```

Listing 8: Фрагмент лога

Помимо вышеописанных функций, были написаны перехватчики следующих функций:

1. `CreateFileW`;
2. `MoveFileW`;
3. `CopyFileW`;
4. `ReplaceFileW`;
5. `LZOpenFileW`;
6. `DeleteFileA`;
7. `DeleteFileTransactedW`.

- [https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms683212\(v=vs.85\).aspx](https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms683212(v=vs.85).aspx)
- [https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms683199\(v=vs.85\).aspx](https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms683199(v=vs.85).aspx)
- <https://habrahabr.ru/post/90377/>
- <http://www.securitylab.ru/analytics/428735.php>
- <http://compress.ru/article.aspx?id=10835>