15 November 2022

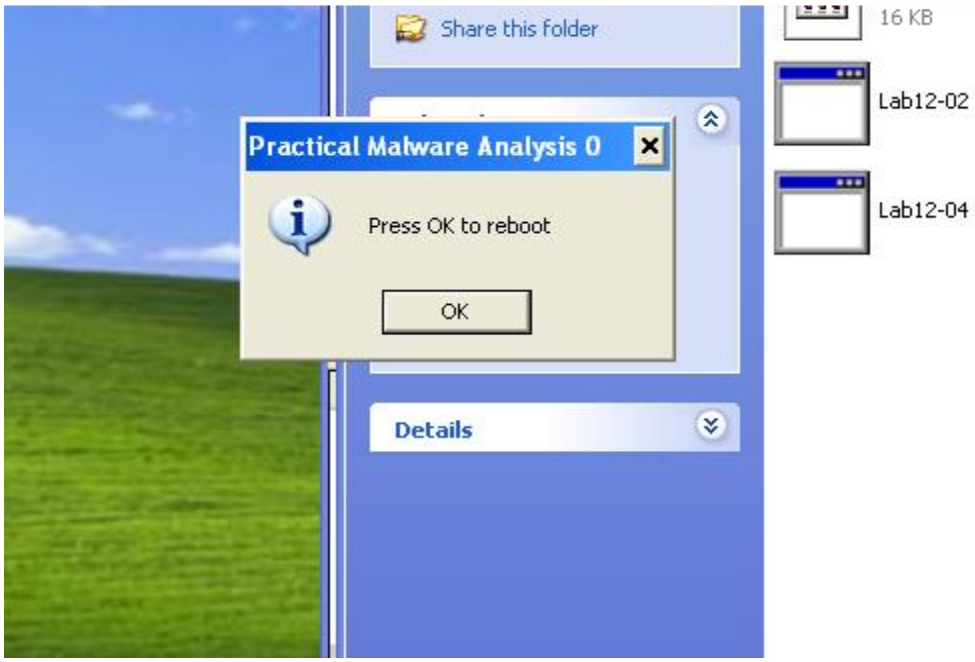
ASSIGNMENT 9

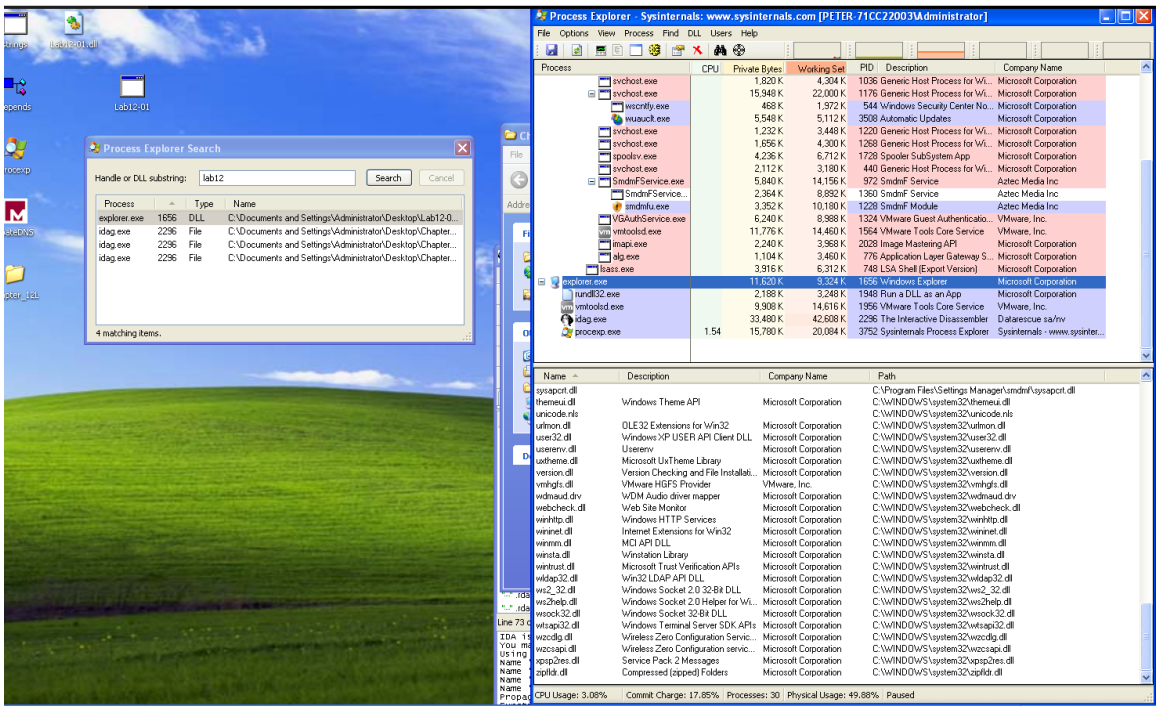
LAB 12-1

Analyze the malware found in the file Lab12-01.exe and Lab12-01.dll. Make sure that these files are in the same directory when performing the analysis.

Questions

i. What happens when you run the malware executable?





When opening the application, the executable file gives the following pop up in the first screenshot. When opening the application one can also view ProcessExplorer to analyze the functions and calls that the application makes within the system of the machine. While analyzing in ProcessExplorer one is able to conclude that the exexcutable calls for a C++ library, a Net Win32 API, and many more .dll's.

ii. What process is being injected?

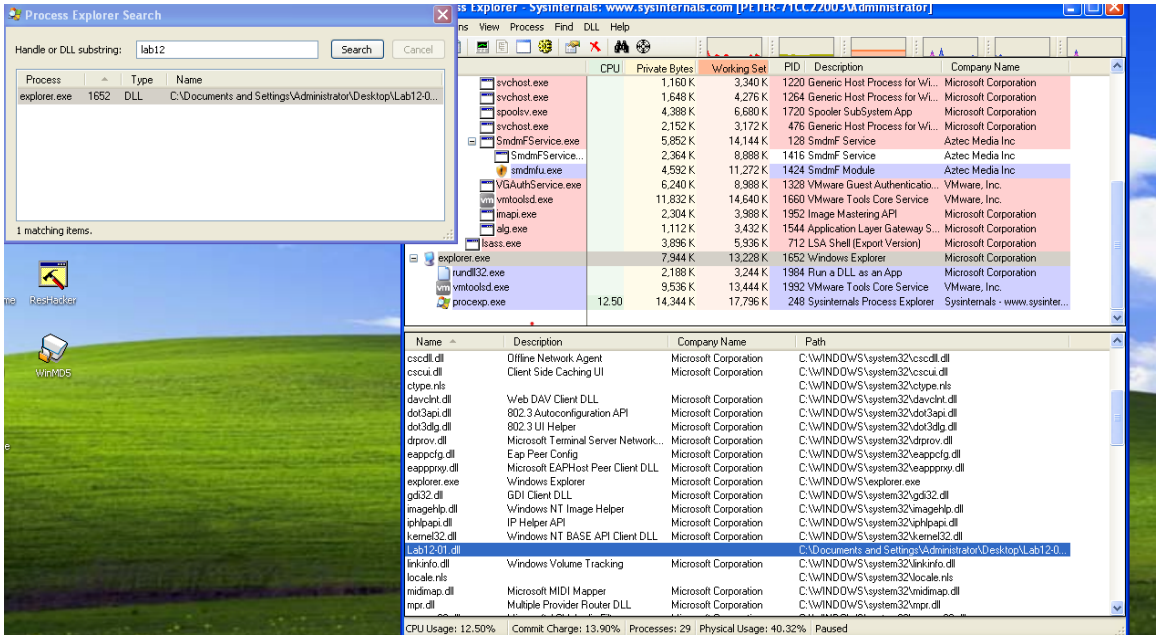
```
.data:0040602B                 db    0
.data:0040602C                 db    0
.data:0040602D                 db    0
.data:0040602E                 db    0
.data:0040602F                 db    0
.data:00406030 ; char aExplorer_exe[]
.data:00406030 aExplorer_exe   db 'explorer.exe',0
.data:0040603D                 align 10h
.data:00406040 dword_406040    dd 6B6E753Ch
.data:00406044 dword_406044    dd 6E776F6Eh
.data:00406048 word_406048     dw 3Eh
.data:0040604A                 align 4
.data:0040604C ; char aLoadlibrarya[]
.data:0040604C aLoadlibrarya   db 'LoadLibraryA',0
.data:00406059                 align 4
.data:0040605C ; char ModuleName[]
.data:0040605C ModuleName      db 'kernel32.dll',0
.data:00406069                 align 4
.data:0040606C ; char aLab1201_dll[]
.data:0040606C aLab1201_dll    db 'Lab12-01.dll',0
.data:00406079                 align 4
.data:0040607C ; char String2[]
.data:0040607C String2:
.data:0040607C                 unicode 0, <\>,0
.data:00406080 ; char aEnumprocesses[]
.data:00406080 aEnumprocesses  db 'EnumProcesses',0
.data:0040608E                 align 10h
.data:00406090 ; char aGetmodulebasen[]
.data:00406090 aGetmodulebasen db 'GetModuleBaseName
.data:004060A3                 align 4
.data:004060A4 ; char LibFileName[]
.data:004060A4 LibFileName     db 'psapi.dll',0
.data:004060A4
```

The process being injected within opening the application is explorer.exe which is given the second screenshot above. One can also anaylze the exe in IDAPro, when anayzing you can retrieve the libraries, and programs used within the malware by looking at the strings of the executable. There are many more programs, libraries, and functions being called such as kernel32.dll, EnumProcesses, psapi.dll, etc.

iii. How can you make the malware stop the pop-ups?

One can stop the malware by first right-clicking the explorer.exe process and killing the process in ProcessExplorer. Once killing the process malware stops the pop-ups. An alternative can also be by manually restarting the application as well and rebooting the system.

iv. How does this malware operate?



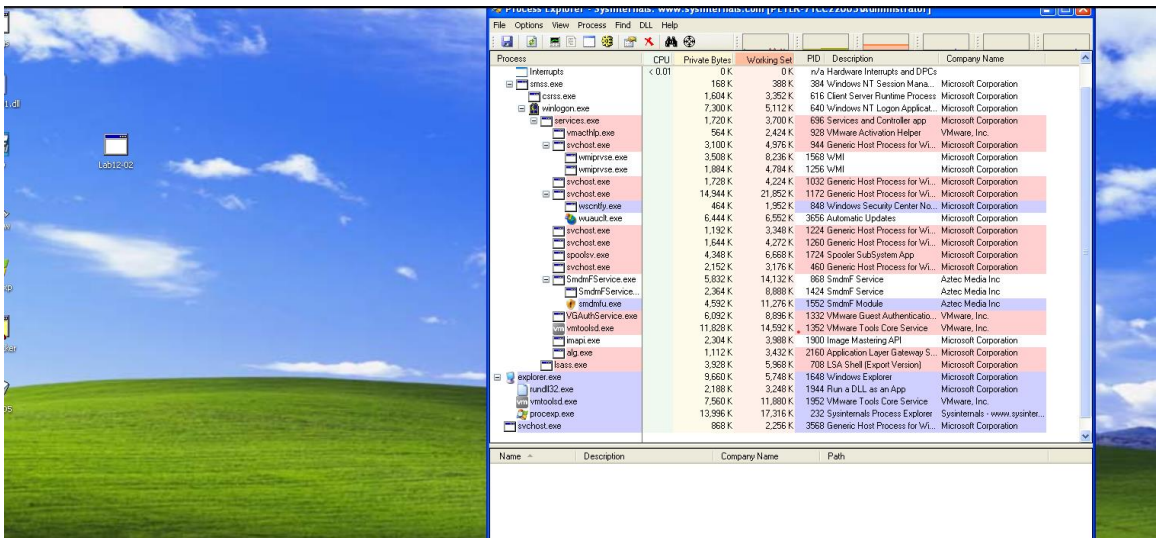
In order to launch Lab12-01.dll, the malware executes dll injection on explorer.exe. The message boxes are controlled by the dll executable. Every minute, a message box with a counter indicating the number of minutes that have passed appears on the screen.

LAB 12-2

Analyze the malware found in the file Lab12-02.exe

Questions

i. What is the purpose of this program?



This malware's objective is to launch a keylogger while impersonating an svchost.exe process utilizing process substitution, beginning a different

program. One is able to analyze this by launching the application in the virtual machine to not affect host. Once completed, the svchost.exe can be viewed while analyzing ProcessExplorer.

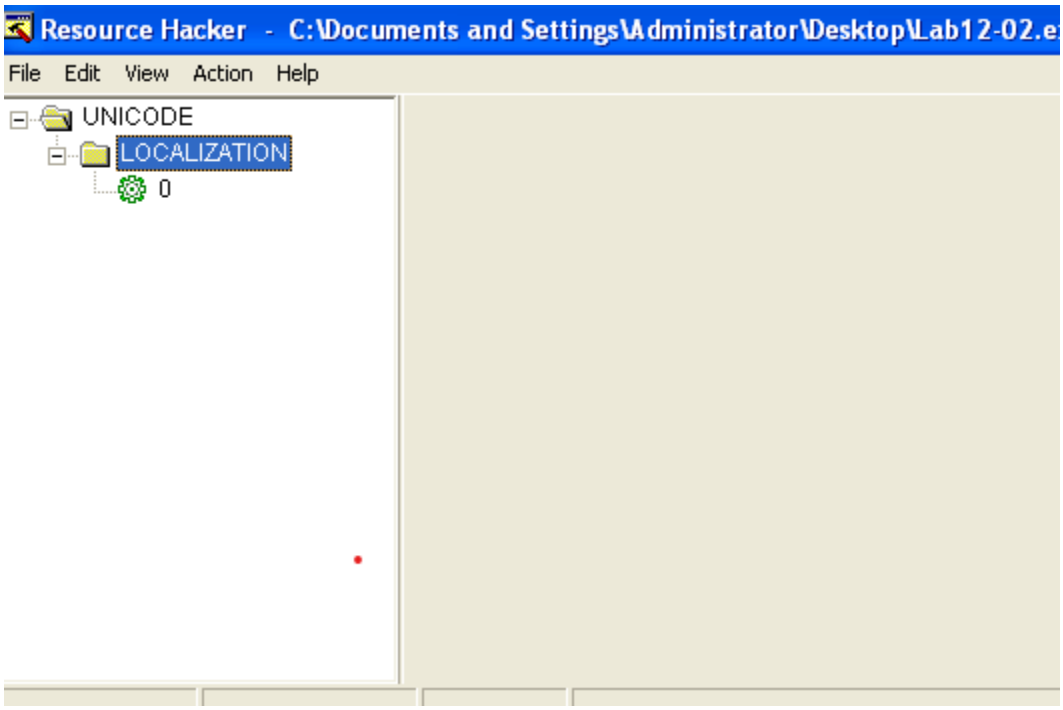
ii. How does the launcher program hide execution?

```
mov     [ebp+lpBuffer], 0
push    0                       ; lpModuleName
call    ds:GetModuleHandleA
mov     [ebp+hModule], eax
push    400h                    ; uSize
lea     eax, [ebp+ApplicationName]
push    eax                     ; lpBuffer
push    offset aSvchost_exe ; "\\svchost.exe"
call    sub_40149D
add     esp, 0Ch
mov     ecx, [ebp+hModule]
push    ecx                     ; hModule
call    sub_40132C
add     esp, 4
mov     [ebp+lpBuffer], eax
cmp     [ebp+lpBuffer], 0
jz      short loc_401573
```

The launcher replaces svchost.exe's process to conceal execution.

When a malware author wishes to disguise their program as a legitimate process. Without running the risk of using process injection to crash the target process, they utilize process replacement.

iii. Where is the malicious payload stored?



The malicious payload is contained in the localization resource. This resource is a file containing executable code. One can explore this data using the ResourceHacker tool.

iv. How is the malicious payload protected?

The malicious payload placed in the resource section of the application is encoded using XOR. This decoding procedure is located at sub 40132C. The location of the XOR byte is 0x0040141B.

v. How are strings protected?

The malicious payload placed in the resource section of the application is encoded using XOR. This decoding procedure is located at sub 40132C. The location of the XOR byte is 0x0040141B.

LAB 12-3

Analyze the malware extracted during the analysis of Lab 12-2, or use the file Lab12-03.exe.

Questions

i. What is the purpose of this malicious payload?

```
mov     eax, [ebp+lpContext]
mov     ecx, [eax+0A4h]
add     ecx, 8
push    ecx             ; lpBaseAddress
mov     edx, [ebp+hProcess]
push    edx             ; hProcess
call    ds:ReadProcessMemory
push    offset ProcName ; "NtUnmapViewOfSection"
push    offset ModuleName ; "ntdll.dll"
call    ds:GetModuleHandleA
push    eax             ; hModule
call    ds:GetProcAddress
mov     [ebp+var_64], eax
cmp     [ebp+var_64], 0
jnz     short loc_4011FE
xor     eax, eax
```

Looking further down after the screenshot given, the program calls many parameters. These parameters are label as idHook,lpfn, hMod, dwThreadId which are very common in keylogger functions. Therefore, the malware has a purpose of installing a keylogger.

ii. How does the malicious payload inject itself?

Through an application-defined hook procedure, malicious payload is injected. Through these injections the program is able to steal keystrokes.

iii. What filesystem residue does this program create?

The filesystem creates a file entitled practicalmalwareanalysis.log that includes the keystrokes logged throughout its run.