

Forum:[Hack Forums](#)

Thread: [Hooking winapi + hooking in general](#)

Io0sr - July 11, 2019, 9:15 am

Hi so I am pretty decent at coding in c++, but I never did any hooking. So I just wanted to ask if anyone could recommend some good articles. ?

Sai Hung - July 11, 2019, 1:58 pm

You better good with Libminhook , i have used it once (libminhook.x86.lib) good library to hook 32 bit process, cannot say for x64
if you good with C++, how good are you with Reverse Engineering?

Io0sr - July 11, 2019, 2:31 pm

(Jul 11, 2019 01:58 PM)Sai Hung Wrote: [\[CITE\]](#) You better good with Libminhook , i have used it once (libminhook.x86.lib) good library to hook 32 bit process, cannot say for x64
if you good with C++, how good are you with Reverse Engineering?
So I never reverse engineered anything really... Btw I would like to learn to code the whole thing myself so pref. no libraries besides default ones

YouthInAsia - July 12, 2019, 4:17 am

A hook is easy you don't need an engine
patch the first five+ bytes with a jmp + offset to an address it's common sense

Sai Hung - July 12, 2019, 3:00 pm

(Jul 12, 2019 04:17 AM)YouthInAsia Wrote: [\[CITE\]](#) A hook is easy you don't need an engine
patch the first five+ bytes with a jmp + offset to an address it's common sense
Initially i never knew that, i have to be very honest. hence i started using Detour libraries. even until now, i dont have a good tutor whos gonna show me things in RE. I learnt everything on here , i even misfire most times.

If i knew how to count byte instructions manually without the use of detour libs , i think i am fine.
Can I pm you for that?

YouthInAsia - July 12, 2019, 3:14 pm

(Jul 12, 2019 03:00 PM)Sai Hung Wrote: [\[CITE\]](#) Initially i never knew that, i have to be very honest.
hence i started using Detour libraries. even until now, i dont have a good tutor whos gonna show me things in RE. I learnt everything on here , i even misfire most times.

If i knew how to count byte instructions manually without the use of detour libs , i think i am fine.
Can I pm you for that?

Yup if you just screen shot the instruction you want to hook I can tell you how many bytes to patch
but but any disassembler will show you
by screen shot I mean screenshot the disassembly of the instruction

Io0sr - July 13, 2019, 7:03 am

Ok so I am currently trying to Hook NtQuerySystemInformation. I managed to write a function that should be able to replace it, and a dll injector that can inject 64bit and 32bit dlls with no issue or crash. My main issue right now is: How do I replace the real NtQuerySysteminformation function with mine? I looked at a lot of github code (tried most) and I never am able to properly hook anything

with it (taskmanager just crashes).

int main() - July 13, 2019, 7:11 am

(Jul 13, 2019 07:03 AM)Invasi0nZ Wrote: [\[CITE\]](#) Ok so I am currently trying to Hook NtQuerySystemInformation. I managed to write a function that should be able to replace it, and a dll injector that can inject 64bit and 32bit dlls with no issue or crash. My main issue right now is: How do I replace the real NtQuerySystemInformation function with mine? I looked at a lot of github code (tried most) and I never am able to properly hook anything with it (taskmanager just crashes). you don't make a hook to replace an entire function... especially not ntquerysysteminformation.

lo0sr - July 13, 2019, 7:24 am

(Jul 13, 2019 07:11 AM)int main() Wrote: [\[CITE\]](#) you don't make a hook to replace an entire function... especially not ntquerysysteminformation.Code #define _CRT_SECURE_NO_WARNINGS

```
/* Includes */
#include <Windows.h>
#include <TIHelp32.h>
#include <winternl.h>

#define STATUS_SUCCESS ((NTSTATUS)0x00000000L)

#pragma comment(lib, "ntdll.lib")
```

```
typedef struct _SYSTEM_PROCESSES {
    ULONG          NextEntryDelta;
    ULONG          ThreadCount;
    ULONG          Reserved1[6];
    LARGE_INTEGER  CreateTime;
    LARGE_INTEGER  UserTime;
    LARGE_INTEGER  KernelTime;
    UNICODE_STRING ProcessName;
    KPRIORITY      BasePriority;
    ULONG          ProcessId;
    ULONG          InheritedFromProcessId;
    ULONG          HandleCount;
    ULONG          Reserved2[2];
} SYSTEM_PROCESSES, *PSYSTEM_PROCESSES;
```

```
NTSTATUS NTAPI NtHookedQuerySystemInformation(SYSTEM_INFORMATION_CLASS
SystemInformationClass, PSYSTEM_PROCESSES SystemInformation, ULONG
SystemInformationLength, PULONG ReturnLength)
{
```

```
    NTSTATUS ntResult;
```

```
    PSYSTEM_PROCESSES pSystemProcesses;
    PSYSTEM_PROCESSES pNextSystemProcess;
```

```
    WCHAR wTargetProcess[] = L"notepad.exe";
```

```
    ntResult = NtQuerySystemInformation(SystemInformationClass, SystemInformation,
SystemInformationLength, ReturnLength);
```

```

switch (SystemInformationClass)
{
case SystemProcessInformation:
    pSystemProcesses = SystemInformation;
    pNextSystemProcess = (PSYSTEM_PROCESSES)((LPBYTE)pSystemProcesses +
pSystemProcesses->NextEntryDelta);

    while (pNextSystemProcess->NextEntryDelta != NULL)
    {
        if (lstrcmpW((&pNextSystemProcess->ProcessName)->Buffer, wTargetProcess) == NULL)
        {
            pSystemProcesses->NextEntryDelta += pNextSystemProcess->NextEntryDelta;
        }

        pSystemProcesses = pNextSystemProcess;
        pNextSystemProcess = (PSYSTEM_PROCESSES)((LPBYTE)pSystemProcesses +
pSystemProcesses->NextEntryDelta);
    }

    break;
}

return ntResult;
}

```

This is the function that I wrote

Sai Hung - July 13, 2019, 12:01 pm

You did not show us your hook function .

Hooking is not just as you think it is. after replacing the jmp bytes of instruction, you would still need a function to be redirected to your own function after which, it goes back to the original function.

Have a good look at this

```

Code int WINAPI DetourMessageBoxW(HWND hWnd, LPCWSTR lpText, LPCWSTR lpCaption,
UINT uType)
{
    return fpMessageBoxW(hWnd, L"Hooked!", lpCaption, uType);
}

```

This is a hook (Detoured Function) which should be called after you have replaced the 5 byte jmp instruction now when you want to call it You do something like this

```

Code MH_CreateHookApiEx(L"user32", "MessageBoxW", &DetourMessageBoxW,
&fpMessageBoxW) != MH_OK)
I am using Libminhook here

```

You can read more here

[MinHook - The Minimalistic x86/x64 API Hooking Library - CodeProject](#)

doxerhacker - July 13, 2019, 8:11 pm

Why do people use libraries for trivial inline hooks? It baffles my mind. Do you guys not understand

assembly?

Sai Hung - July 14, 2019, 4:50 am

(Jul 13, 2019 08:11 PM)doxerhacker Wrote: [\[CITE\]](#) Why do people use libraries for trivial inline hooks? It baffles my mind. Do you guys not understand assembly?

Not every one does sir. Rather than make fun of people, start a thread to show how to do RE and calculate bytes .

More popularity and respect awaits you, than asking this question you just asked. Its a forum, You too can contribute :D

doxerhacker - July 14, 2019, 1:23 pm

(Jul 14, 2019 04:50 AM)Sai Hung Wrote: [\[CITE\]](#) Not every one does sir. Rather than make fun of people, start a thread to show how to do RE and calculate bytes .

More popularity and respect awaits you, than asking this question you just asked. Its a forum, You too can contribute :D

Calculating instruction lengths can be done in three ways:

- Using a length disassembler engine
- Writing your own length disassembler engine
- Hardcode function prologues (or any segment of the procedure for that matter) (assuming they are the same across different library versions)

The former is trivial but could be used as a signature based scan for AV solutions.

Sai Hung - July 14, 2019, 3:23 pm

(Jul 14, 2019 01:23 PM)doxerhacker Wrote: [\[CITE\]](#) Calculating instruction lengths can be done in three ways:

- Using a length disassembler engine
- Writing your own length disassembler engine
- Hardcode function prologues (or any segment of the procedure for that matter) (assuming they are the same across different library versions)

The former is trivial but could be used as a signature based scan for AV solutions.

No time to code my own LDE at this Moment , i wish i had someone who can give a guide on that. i dont know anything about Intel XED, sorry. like i said, most things i know on here i had to read alone :D

I used HDE (Hacker Disassembly Engine) as my own LDE sometime when i tried to make an x86 Detour Engine so i could try Heavens gate much later on. gave me some issues and hence i practically resolved to using the Disassemblers like IDA and the likes to calculate and then hard code the byte of instruction One problem... I dont have a guide to school me thru RE fully.