GAME HACKING

Speacher : Jasey

2018-01-15

CONTENT

How to Hack Game

AssaultCube
https://assault.cubers.net/

Cheat Engine
http://www.cheatengine.org/

Hack Theory

Api Hook
hook/APIHOOK/apihookdemo/apihookdemo/

Remote Injector
hook/apihookDemo2/apihookdemo/Injector/

https://github.com/Jasey/hook.git

GAME HACKING

1. Find out the player's health address. The are some tips to find that

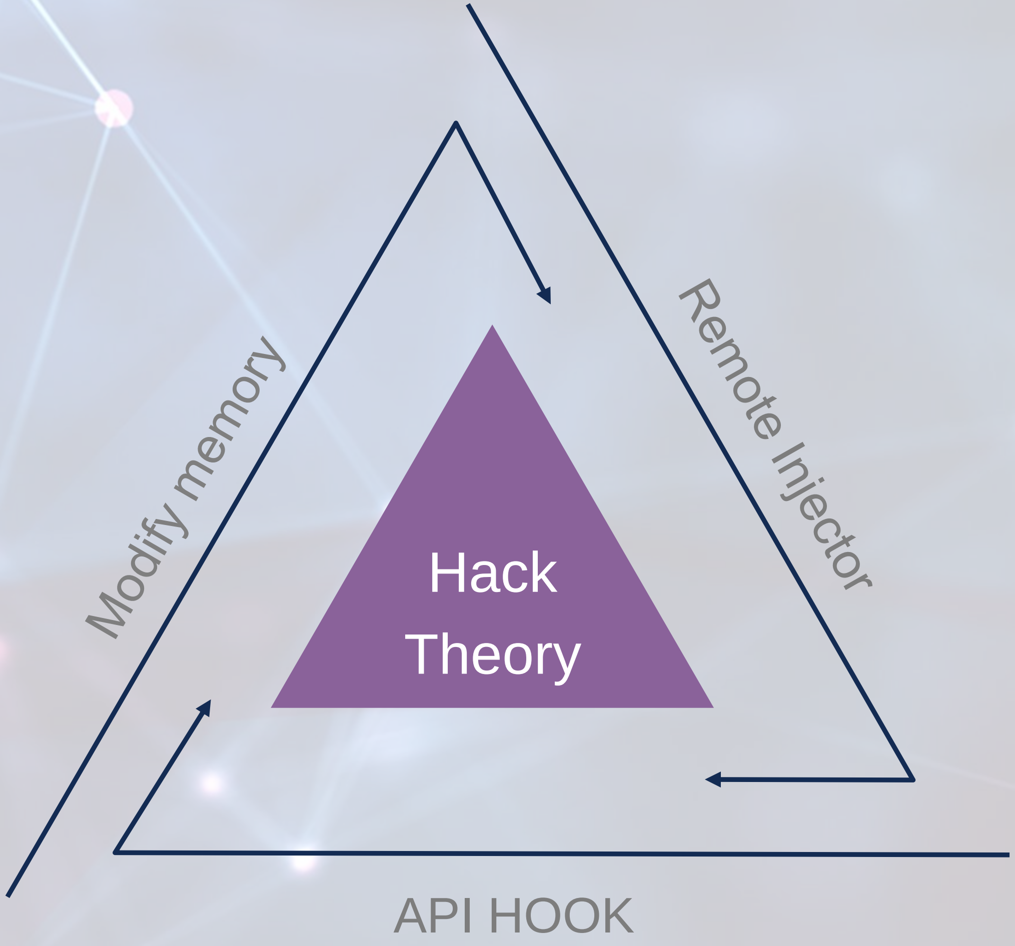
2. Find out which assemble instruction decrease player health value when player be attacked

3. Use injector technology to modify the dissemble code and make the player's health keeping fully

4. Make sure your assemble code is correct. Attention that only when the player's health will not be decreased when attacked

Showing time ...

GAME HACKING



Hack Theory

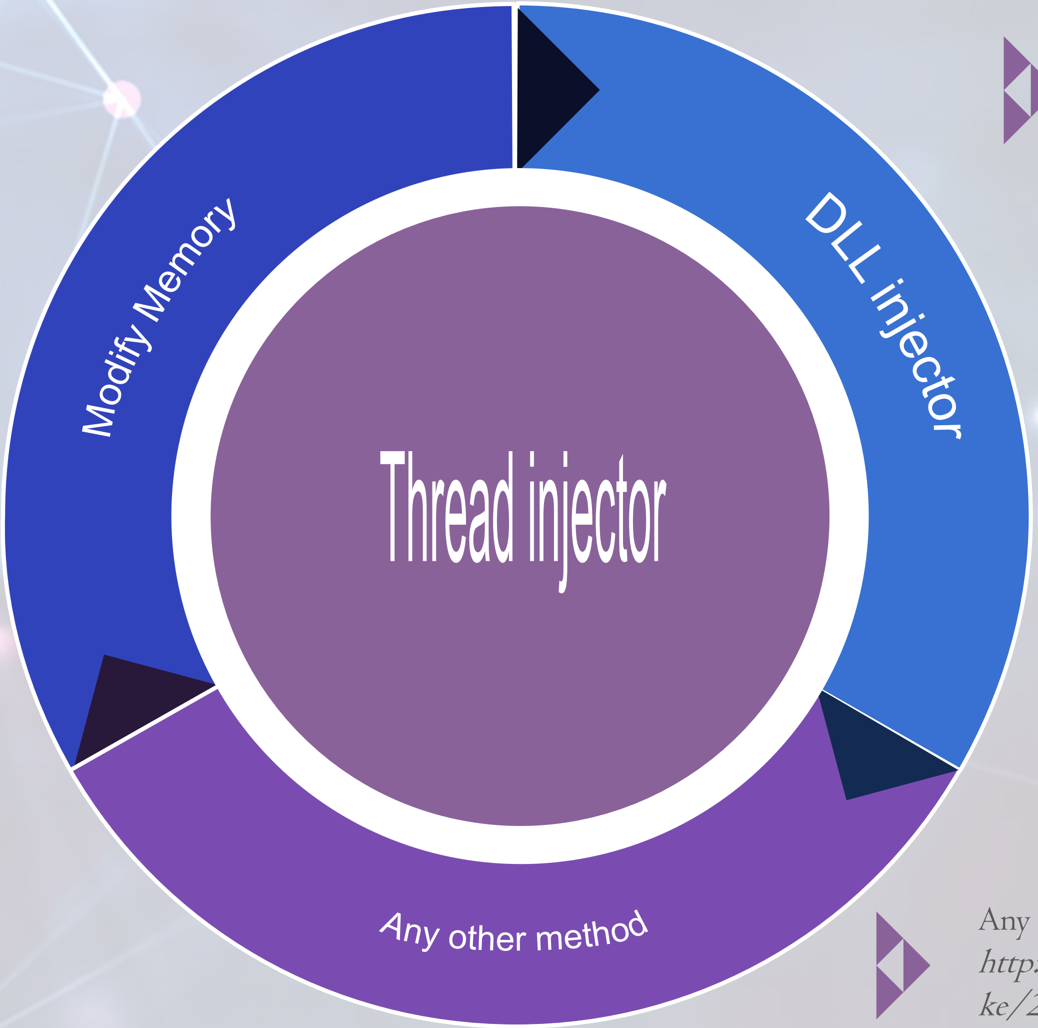
Modify memory

Remote Injector

API HOOK

GAME HACKING

Create remote thread to load the virus DLL
The virus DLL will find out the hook function API address and make it jump to your function



Get remote process handle with process id
Locate the hook address in remote process
Modify the remote process address's value

Any other technology:
http://www.360zhijia.com/360anquanke/274186.html

GAME HACKING

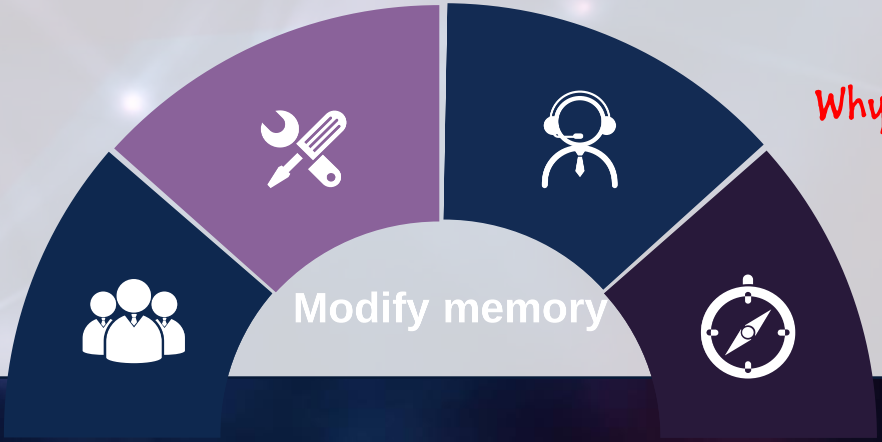
```
hProcHandle = OpenProcess( PROCESS_ALL_ACCESS, FALSE, dwProcId );
WriteProcessMemory( hProcHandle, (BYTE*)addressToWrite, &value, sizeof(value), NULL);
```

Get process handle with process id

Write any thing you want to the remote process address you have found

See a demo:
D:\myfile\git-hub\hook\APIHOOK\modifyApiHookDemoMemory

comment

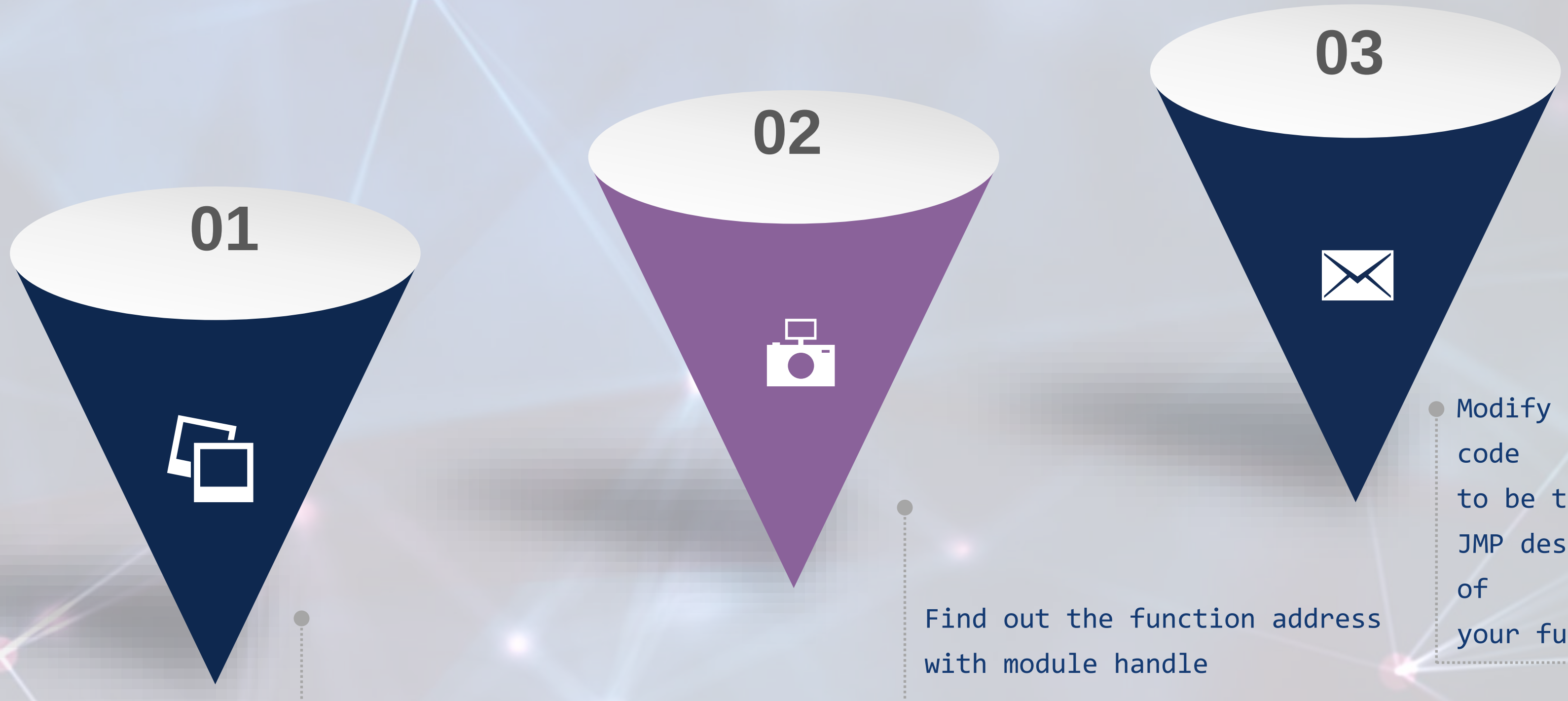


Modify memory

Why not modify the memory directly?

GAME HACKING

API HOOK



01

Find out the API module and get the module handle

02

Find out the function address with module handle

03

Modify the function assemble code
to be the JMP assemble code
JMP destination is the address of
your function

D:\myfile\git-hub\hook\APIHOOK\apihookdemo

GAME HACKING

DLL Injector

PROMOTION PRIVILEGE

Modify local process privilege

OPEN REMOTE PORCESS

Open remote process and make sure local process have some privilege access remote process

PROCESS_CREATE_THREAD

PROCESS_VM_OPERATION

PROCESS_VM_WRITE

PROCESS_ALL_ACCESS

ALLOCATE MEMORY

Allocate VM to store DLL address

COPY DLL PATH

Copy DLL address to remote VM you have allocated

GET API ADDR

Calculate LoadLibraryA start address

Remember the address you calculated is the local process address, but it is the same as the remote process

CREATE REMOTE THREAD

Create remote thread to execute the LoadLibraryA function in remote porcess

D:\myfile\git-hub\hook\apihookDemo2\apihookdemo\Injector

API HOOK

1. Assume the function name to be hooked as fhooked and your own function named fhook

2. Get the module handle of fhooked

3. Get the VM start address of hooked

4. Replace the begin of the hooked function assemble instruction as "jmp XXX" (total 5 byets)

5. Calculate the jmp destination:

   XXX = fhook – fhooked-5

'When CPU calculate jmp instruction's destination, PC value is the next assemble instruction's address, not jmp instruction address'

GAME HACKING

EXAMPLE

```
fhooked:
01FF0000 : ?? ?? ?? ??
01FF0004 : ?? ?? ?? ??
01FF0008 : ?? ?? ?? ??
...

fhook:
01FF00A0 : ?? ?? ?? ??
01FF00A4 : ?? ?? ?? ??
01FF00A8 : ?? ?? ?? ??
...

relpalce the first 5 bytes fhooked as jmp(E9 xx xx xx xx),the content of fhooked:
fhooked:
01FF0000 : E9 xx xx xx
01FF0004 : xx ?? ?? ??
01FF0008 : ?? ?? ?? ??
...

satisfy :  01FF00A0 = PC + XXX
                 PC = 01FF0000 + 5
then the addr : XXX = 01FF00A0 - 01FF0000 - 5
```

GAME HACKING

AOB Injection

The content of allocation:

```
03420000 - 50                - push eax
03420001 - 51                - push ecx
03420002 - A1 749B5000       - mov eax,[ac_client.exe+109B74] { [005287C0] }
03420007 - 05 F8000000       - add eax,000000F8 { 248 }
0342000C - 8B CB             - mov ecx,ebx
0342000E - 83 C1 04          - add ecx,04 { 4 }
03420011 - 39 C8             - cmp eax,ecx
03420013 - 59                - pop ecx
03420014 - 58                - pop eax
03420015 - 0F85 05000000     - jne 03420020
0342001B - BF 00000000       - mov edi,00000000 { 0 }
03420020 - 29 7B 04          - sub [ebx+04],edi
03420023 - 8B C7             - mov eax,edi
03420025 - E9 FA9C00FD       - jmp ac_client.exe+29D24
```

Before injection:

```
ac_client.exe+29D1D - 2B F8            - sub edi,eax
ac_client.exe+29D1F - 29 7B 04         - sub [ebx+04],edi
ac_client.exe+29D22 - 8B C7            - mov eax,edi
ac_client.exe+29D24 - 5F               - pop edi
ac_client.exe+29D25 - 5E               - pop esi
ac_client.exe+29D26 - 8B E5            - mov esp,ebp
```

After injection:

```
ac_client.exe+29D1D - 2B F8            - sub edi,eax
INJECT              - E9 DC62FF02      - jmp 03420000
ac_client.exe+29D24 - 5F               - pop edi
ac_client.exe+29D25 - 5E               - pop esi
ac_client.exe+29D26 - 8B E5            - mov esp,ebp
```

2018

THANK YOU