全国信息安全竞赛-easyGo writeup

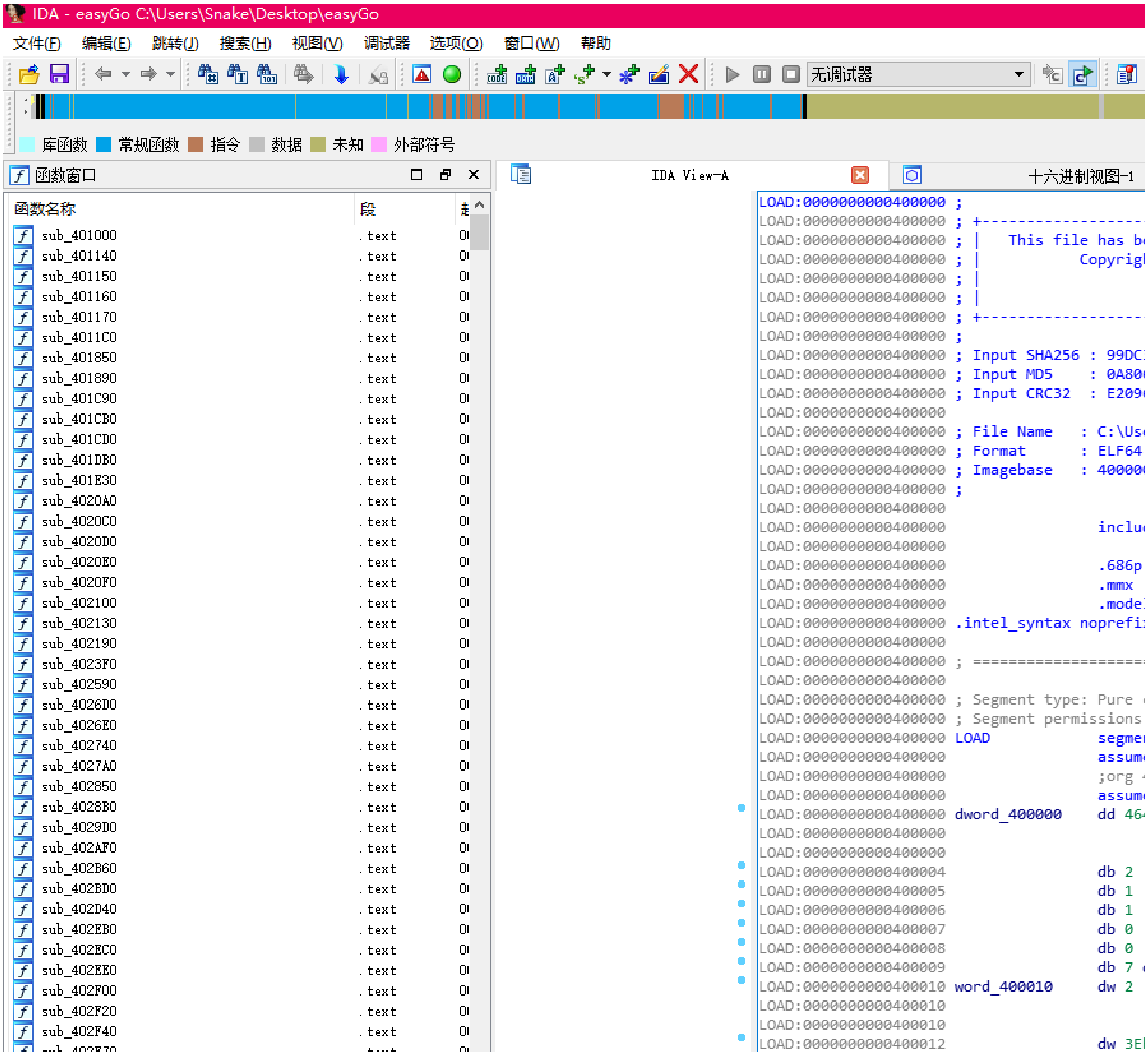
原创

bufsnake 于 2019-04-23 16:11:59 发布 1292 收藏 4
分类专栏： CTF 文章标签： writeup 十二届全国信息安全竞赛

CTF 专栏收录该内容
3 篇文章 0 订阅
订阅专栏
所需工具：

> ida pro，IDAGolangHelper，easyGo程序

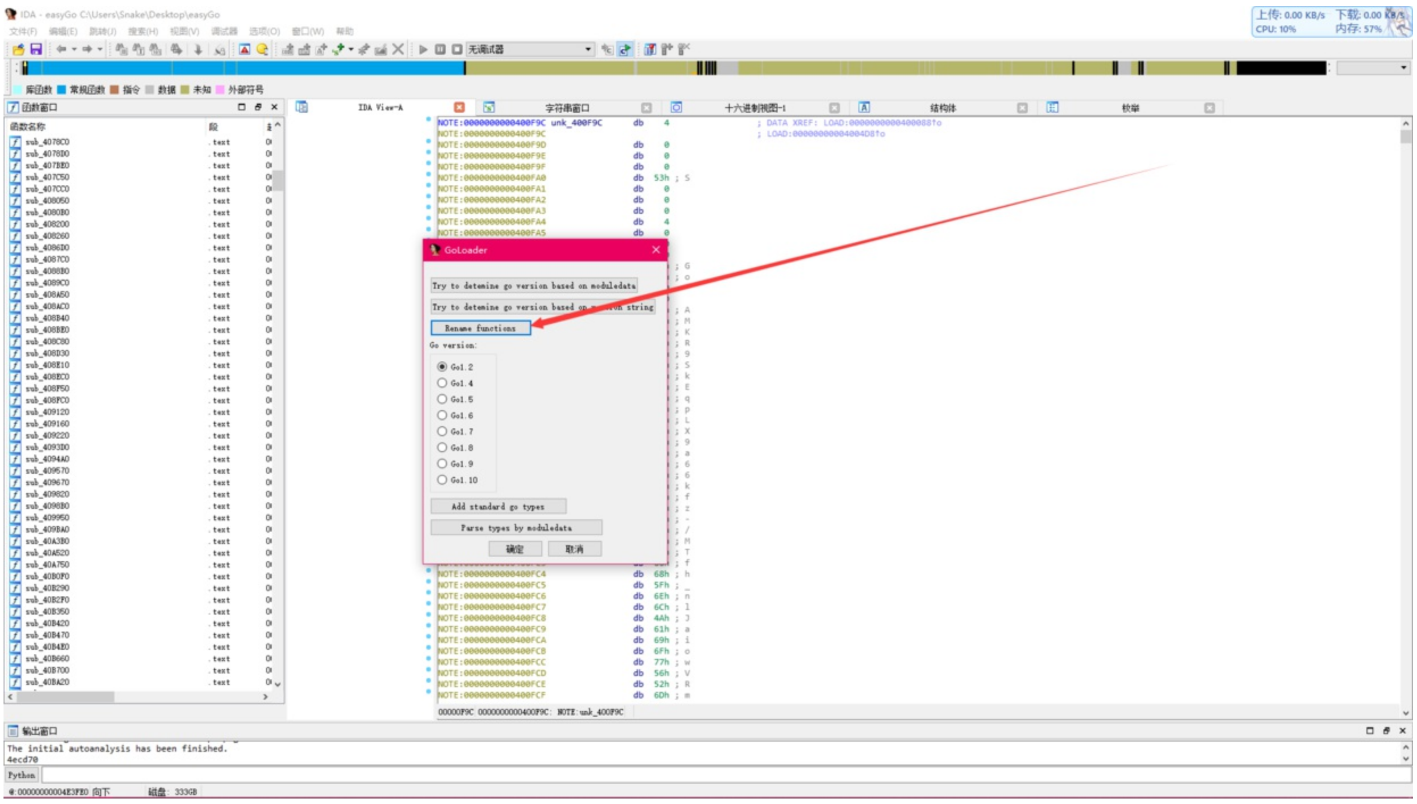
将程序拖入对应位数的IDA中观察函数窗口，可以很容易的看出来，程序进行了无符号处理，Google翻阅资料，查看无符号golang逆向技巧，找到IDAGolangHelper这一脚本(项目地址在文章最下方)


IDA - easyGo C:\Users\Snake\Desktop\easyGo
文件(F) 编辑(E) 跳转(J) 搜索(H) 视图(V) 调试器 选项(O) 窗口(W) 帮助
无调试器
库函数 常规函数 指令 数据 未知 外部符号
函数窗口
IDA View-A
十六进制视图-1

函数名称	段	
sub_401000	.text	0
sub_401140	.text	0
sub_401150	.text	0
sub_401160	.text	0
sub_401170	.text	0
sub_4011C0	.text	0
sub_401850	.text	0
sub_401890	.text	0
sub_401C90	.text	0
sub_401CB0	.text	0
sub_401CD0	.text	0
sub_401DB0	.text	0
sub_401E30	.text	0
sub_4020A0	.text	0
sub_4020C0	.text	0
sub_4020D0	.text	0
sub_4020E0	.text	0
sub_4020F0	.text	0
sub_402100	.text	0
sub_402130	.text	0
sub_402190	.text	0
sub_4023F0	.text	0
sub_402590	.text	0
sub_4026D0	.text	0
sub_4026E0	.text	0
sub_402740	.text	0
sub_4027A0	.text	0
sub_402850	.text	0
sub_4028B0	.text	0
sub_4029D0	.text	0
sub_402AF0	.text	0
sub_402B60	.text	0
sub_402BD0	.text	0
sub_402D40	.text	0
sub_402EB0	.text	0
sub_402EC0	.text	0
sub_402EE0	.text	0
sub_402F00	.text	0
sub_402F20	.text	0
sub_402F40	.text	0

```
LOAD:0000000000400000 ;
LOAD:0000000000400000 ; +------------------
LOAD:0000000000400000 ; |   This file has b
LOAD:0000000000400000 ; |          Copyrig
LOAD:0000000000400000 ; |
LOAD:0000000000400000 ; |
LOAD:0000000000400000 ; +------------------
LOAD:0000000000400000 ;
LOAD:0000000000400000 ; Input SHA256 : 99DC
LOAD:0000000000400000 ; Input MD5    : 0A80
LOAD:0000000000400000 ; Input CRC32  : E209
LOAD:0000000000400000
LOAD:0000000000400000 ; File Name   : C:\Us
LOAD:0000000000400000 ; Format      : ELF64
LOAD:0000000000400000 ; Imagebase   : 40000
LOAD:0000000000400000 ;
LOAD:0000000000400000
LOAD:0000000000400000                 inclu
LOAD:0000000000400000
LOAD:0000000000400000                 .686p
LOAD:0000000000400000                 .mmx
LOAD:0000000000400000                 .mode
LOAD:0000000000400000 .intel_syntax noprefi
LOAD:0000000000400000
LOAD:0000000000400000 ; ==================
LOAD:0000000000400000
LOAD:0000000000400000 ; Segment type: Pure
LOAD:0000000000400000 ; Segment permissions
LOAD:0000000000400000 LOAD            segme
LOAD:0000000000400000                 assum
LOAD:0000000000400000                 ;org
LOAD:0000000000400000                 assum
LOAD:0000000000400000 dword_400000    dd 46
LOAD:0000000000400000
LOAD:0000000000400000
LOAD:0000000000400004                 db 2
LOAD:0000000000400005                 db 1
LOAD:0000000000400006                 db 1
LOAD:0000000000400007                 db 0
LOAD:0000000000400008                 db 0
LOAD:0000000000400009                 db 7
LOAD:0000000000400010 word_400010     dw 2
LOAD:0000000000400010
LOAD:0000000000400010
LOAD:0000000000400012                 dw 3E
```

接下来，我们打开脚本(文件-脚本文件)找到IDAGolangHelper脚本所在的文件，打开，选中rename function 后选择go版本，默认是Go1.2，然后点确认，发现函数窗口变成了我们能够理解的函数名


IDA - easyGo C:\Users\Snake\Desktop\easyGo
文件(F) 编辑(E) 跳转(J) 搜索(H) 视图(V) 调试器 选项(O) 窗口(W) 帮助
无调试器
上传: 0.00 KB/s 下载: 0.00 KB/s
CPU: 10% 内存: 57%
库函数 常规函数 指令 数据 未知 外部符号
函数窗口
IDA View-A
字符串窗口
十六进制视图-1
结构体
枚举
输出窗口

```
函数名称        段
sub_4078C0     .text
sub_4078D0     .text
sub_407BE0     .text
sub_407C50     .text
sub_407CC0     .text
sub_408050     .text
sub_4080B0     .text
sub_408200     .text
sub_408260     .text
sub_4086D0     .text
sub_4087C0     .text
sub_4088B0     .text
sub_4089C0     .text
sub_408A50     .text
sub_408AC0     .text
sub_408B40     .text
sub_408BE0     .text
sub_408C80     .text
sub_408D30     .text
sub_408E10     .text
sub_408EC0     .text
sub_408F50     .text
sub_408FC0     .text
sub_409120     .text
sub_409160     .text
sub_409220     .text
sub_4093D0     .text
sub_4094A0     .text
sub_409570     .text
sub_409670     .text
sub_409820     .text
sub_4098B0     .text
sub_409950     .text
sub_409BA0     .text
sub_40A3B0     .text
sub_40A520     .text
sub_40A750     .text
sub_40B0F0     .text
sub_40B290     .text
sub_40B2F0     .text
sub_40B350     .text
sub_40B420     .text
sub_40B470     .text
sub_40B4E0     .text
sub_40B660     .text
sub_40B700     .text
sub_40BA20     .text
```

```
NOTE:000000000400F9C unk_400F9C      db    4                ; DATA XREF: LOAD:00000000004000088↑o
NOTE:000000000400F9C                                        ; LOAD:00000000004004D8↑o
NOTE:000000000400F9D                 db    0
NOTE:000000000400F9E                 db    0
NOTE:000000000400F9F                 db    0
NOTE:000000000400FA0                 db  53h ; S
NOTE:000000000400FA1                 db    0
NOTE:000000000400FA2                 db    0
NOTE:000000000400FA3                 db    0
NOTE:000000000400FA4                 db    4
NOTE:000000000400FA5                 db    0
                                             ; G
                                             ; o

                                             ; A
                                             ; M
                                             ; K
                                             ; R
                                             ; 9
                                             ; S
                                             ; k
                                             ; E
                                             ; q
                                             ; p
                                             ; L
                                             ; X
                                             ; 9
                                             ; a
                                             ; 6
                                             ; 6
                                             ; k
                                             ; f
                                             ; z
                                             ; -
                                             ; /
                                             ; M
                                             ; T
                                             ; f
NOTE:000000000400FC4                 db  68h ; h
NOTE:000000000400FC5                 db  5Fh ; _
NOTE:000000000400FC6                 db  6Eh ; n
NOTE:000000000400FC7                 db  6Ch ; l
NOTE:000000000400FC8                 db  4Ah ; J
NOTE:000000000400FC9                 db  61h ; a
NOTE:000000000400FCA                 db  69h ; i
NOTE:000000000400FCB                 db  6Fh ; o
NOTE:000000000400FCC                 db  77h ; w
NOTE:000000000400FCD                 db  56h ; V
NOTE:000000000400FCE                 db  52h ; R
NOTE:000000000400FCF                 db  6Dh ; m
00000F9C 000000000400F9C: NOTE:unk_400F9C
```

GoLoader

Try to detemine go version based on moduledata

Try to detemine go version based on version string

Rename functions

Go version:

[✓ Go1.2] Go1.4 Go1.5 Go1.6 Go1.7 Go1.8 Go1.9 Go1.10

Add standard go types

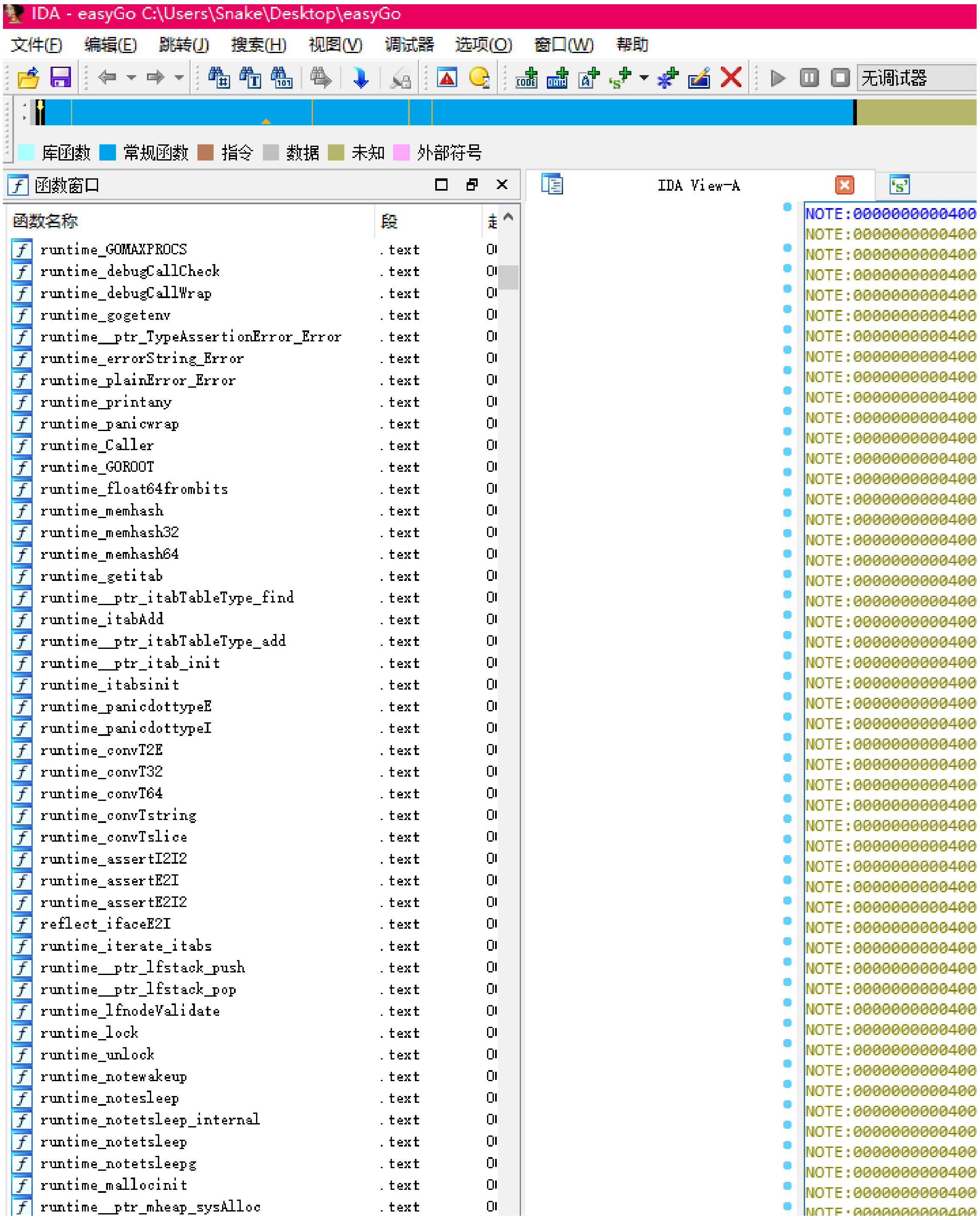
Parse types by moduledata

确定 取消

```
The initial autoanalysis has been finished.
4ecd70
```

Python

@: 00000000004E3FE0 向下   磁盘: 333GB


IDA - easyGo C:\Users\Snake\Desktop\easyGo

文件(F) 编辑(E) 跳转(J) 搜索(H) 视图(V) 调试器 选项(O) 窗口(W) 帮助

无调试器

库函数 常规函数 指令 数据 未知 外部符号

函数窗口

函数名称	段	起
runtime_GOMAXPROCS	.text	0
runtime_debugCallCheck	.text	0
runtime_debugCallWrap	.text	0
runtime_gogetenv	.text	0
runtime__ptr_TypeAssertionError_Error	.text	0
runtime_errorString_Error	.text	0
runtime_plainError_Error	.text	0
runtime_printany	.text	0
runtime_panicwrap	.text	0
runtime_Caller	.text	0
runtime_GOROOT	.text	0
runtime_float64frombits	.text	0
runtime_memhash	.text	0
runtime_memhash32	.text	0
runtime_memhash64	.text	0
runtime_getitab	.text	0
runtime__ptr_itabTableType_find	.text	0
runtime_itabAdd	.text	0
runtime__ptr_itabTableType_add	.text	0
runtime__ptr_itab_init	.text	0
runtime_itabsinit	.text	0
runtime_panicdottypeE	.text	0
runtime_panicdottypeI	.text	0
runtime_convT2E	.text	0
runtime_convT32	.text	0
runtime_convT64	.text	0
runtime_convTstring	.text	0
runtime_convTslice	.text	0
runtime_assertI2I2	.text	0
runtime_assertE2I	.text	0
runtime_assertE2I2	.text	0
reflect_ifaceE2I	.text	0
runtime_iterate_itabs	.text	0
runtime__ptr_lfstack_push	.text	0
runtime__ptr_lfstack_pop	.text	0
runtime_lfnodeValidate	.text	0
runtime_lock	.text	0
runtime_unlock	.text	0
runtime_notewakeup	.text	0
runtime_notesleep	.text	0
runtime_notetsleep_internal	.text	0
runtime_notetsleep	.text	0
runtime_notetsleepg	.text	0
runtime_mallocinit	.text	0
runtime__ptr_mheap_sysAlloc	.text	0

IDA View-A

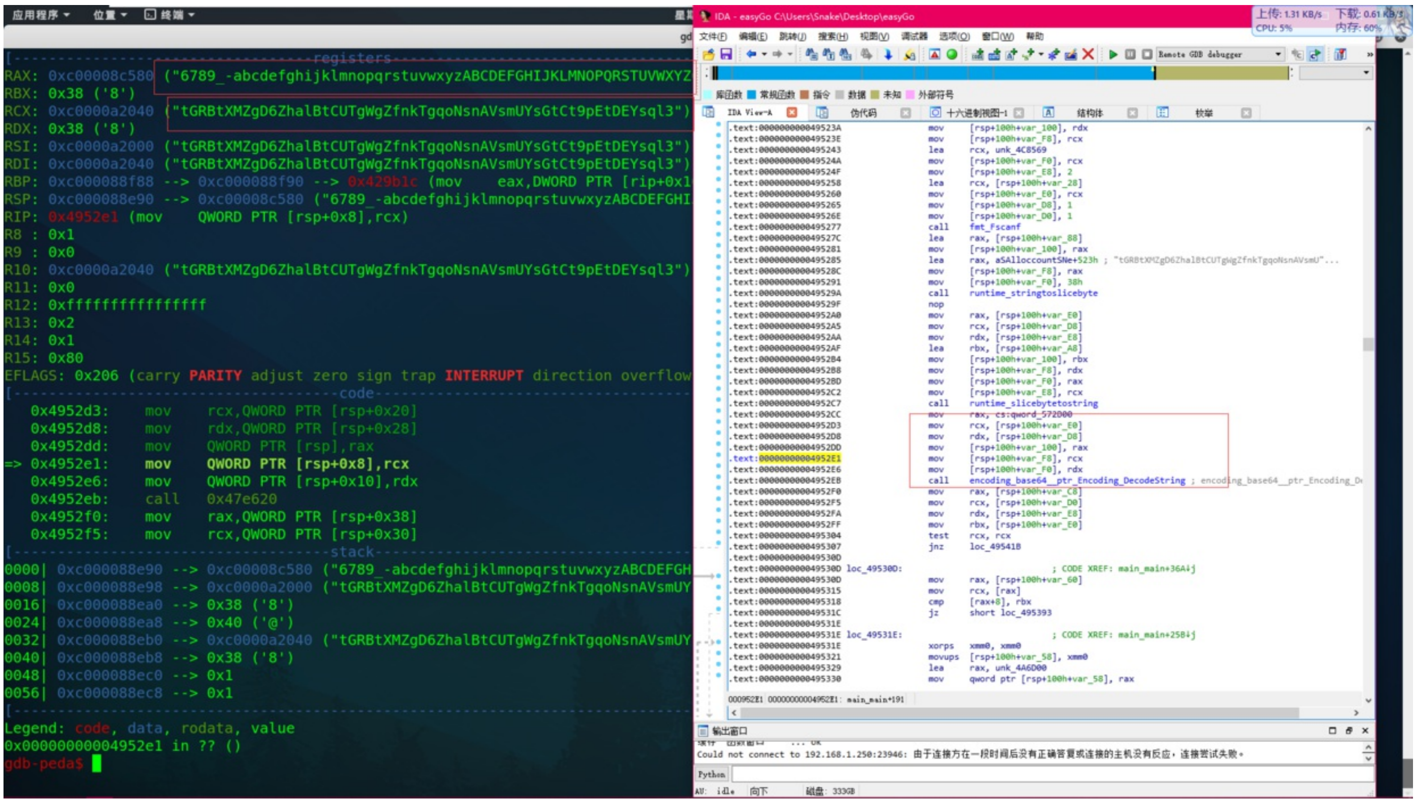
```
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
NOTE:0000000000400
```

接下来，找到main_main函数，F5得到伪代码，分析伪代码逻辑得到其使用的算法

```
  __int64 v17; // [rsp+8h] [rbp-F8h]
  __int64 v18; // [rsp+10h] [rbp-F0h]
  char v19; // [rsp+58h] [rbp-A8h]
  char v20; // [rsp+80h] [rbp-80h]
  __int64 v21; // [rsp+98h] [rbp-68h]
  __int64 v22; // [rsp+A0h] [rbp-60h]
  __int128 v23; // [rsp+A8h] [rbp-58h]
  __int128 v24; // [rsp+B8h] [rbp-48h]
  __int128 v25; // [rsp+C8h] [rbp-38h]
  __int128 v26; // [rsp+D8h] [rbp-28h]
  __int128 v27; // [rsp+E8h] [rbp-18h]

  if ( &v20 <= *(__readfsqword(0xFFFFFFF8) + 16) )
    runtime_morestack_noctxt(a1, a2);
  runtime_newobject(a1, a2);
  v22 = v17;
  *&v27 = &unk_4A6D00;
  *(&v27 + 1) = &off_4E1130;
  fmt_Fprintln(a1, a2, &v27, &unk_4A6D00, v2);  // Please input you flag like flag{123} to judge:
  *&v26 = &unk_4A3E80;
  *(&v26 + 1) = v22;
  fmt_Fscanf(a1, a2, &off_4E2880);              // 输入
  runtime_stringtoslicebyte(a1, a2, v3, v4, v5, v6);
  *&v7 = &v19;
  *(&v7 + 1) = 2LL;
  runtime_slicebytetostring(a1, a2, 2LL, 1, v8, v7);
  encoding_base64__ptr_Encoding_DecodeString(a1, a2, 1LL, &v26, v9, v10, qword_572B00, &v26, 1uLL);// 魔改的base64解密
  v21 = 1LL;
  MEMORY[0x19](a1);
  runtime_convTstring(a1, a2, v14);
  *&v25 = &unk_4A6D00;
  *(&v25 + 1) = v18;
  fmt_Fprintln(a1, a2, &off_4E28A0, &unk_4A6D00, v15);
  if ( *(v22 + 8) == &v26 )
  {
    runtime_memequal(a1, a2, v21, *v22);
    *&v24 = &unk_4A6D00;
    *(&v24 + 1) = &off_4E1140;
    result = fmt_Fprintln(a1, a2, v12, &off_4E28A0, v13);
  }
  else
  {
    *&v23 = &unk_4A6D00;
    *(&v23 + 1) = &off_4E1150;
    result = fmt_Fprintln(a1, a2, v21, &off_4E28A0, v16);
  }
```

gdb动态调试一波，看base加密是给谁解密，是我们输入的值，亦或是程序自带的值，以及其自定义的编码表(base系列加密的原理，这里就不深究了)，动态调试知道到了需要解密的字符串以及编码表



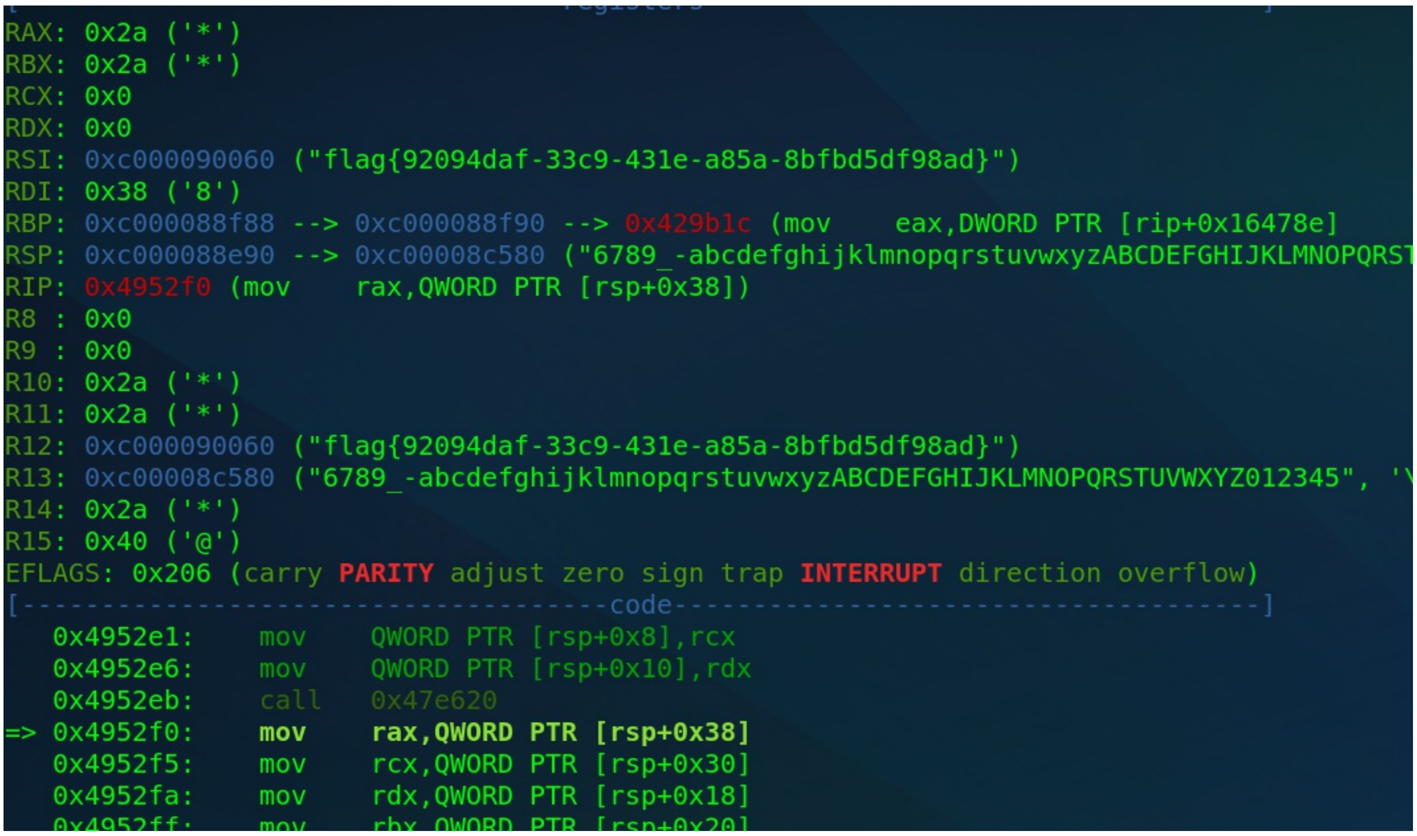
应用程序 位置 终端
IDA - easyGo C:\Users\Snake\Desktop\easyGo
文件(F) 编辑(E) 跳转(J) 搜索(H) 视图(V) 调试器 选项(O) 窗口(W) 帮助
Remote GDB debugger
库函数 常规函数 指令 数据 未知 外部符号
IDA View-A 伪代码 十六进制视图-1 结构体 枚举
上传: 1.31 KB/s 下载: 0.61 KB/s CPU: 5% 内存: 60%
输出窗口

```
[----------------------------------registers-----------------------------------
RAX: 0xc00008c580 ("6789_-abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
RBX: 0x38 ('8')
RCX: 0xc0000a2040 ("tGRBtXMZgD6ZhalBtCUTgWgZfnkTgqoNsnAVsmUYsGtCt9pEtDEYsql3")
RDX: 0x38 ('8')
RSI: 0xc0000a2000 ("tGRBtXMZgD6ZhalBtCUTgWgZfnkTgqoNsnAVsmUYsGtCt9pEtDEYsql3")
RDI: 0xc0000a2040 ("tGRBtXMZgD6ZhalBtCUTgWgZfnkTgqoNsnAVsmUYsGtCt9pEtDEYsql3")
RBP: 0xc000088f88 --> 0xc000088f90 --> 0x429b1c (mov    eax,DWORD PTR [rip+0x1
RSP: 0xc000088e90 --> 0xc00008c580 ("6789_-abcdefghijklmnopqrstuvwxyzABCDEFGHI
RIP: 0x4952e1 (mov    QWORD PTR [rsp+0x8],rcx)
R8 : 0x1
R9 : 0x0
R10: 0xc0000a2040 ("tGRBtXMZgD6ZhalBtCUTgWgZfnkTgqoNsnAVsmUYsGtCt9pEtDEYsql3")
R11: 0x0
R12: 0xffffffffffffffff
R13: 0x2
R14: 0x1
R15: 0x80
EFLAGS: 0x206 (carry PARITY adjust zero sign trap INTERRUPT direction overflow
[-------------------------------------code-------------------------------------
   0x4952d3:    mov    rcx,QWORD PTR [rsp+0x20]
   0x4952d8:    mov    rdx,QWORD PTR [rsp+0x28]
   0x4952dd:    mov    QWORD PTR [rsp],rax
=> 0x4952e1:    mov    QWORD PTR [rsp+0x8],rcx
   0x4952e6:    mov    QWORD PTR [rsp+0x10],rdx
   0x4952eb:    call   0x47e620
   0x4952f0:    mov    rax,QWORD PTR [rsp+0x38]
   0x4952f5:    mov    rcx,QWORD PTR [rsp+0x30]
[------------------------------------stack-------------------------------------
0000| 0xc000088e90 --> 0xc00008c580 ("6789_-abcdefghijklmnopqrstuvwxyzABCDEFGH
0008| 0xc000088e98 --> 0xc0000a2000 ("tGRBtXMZgD6ZhalBtCUTgWgZfnkTgqoNsnAVsmUY
0016| 0xc000088ea0 --> 0x38 ('8')
0024| 0xc000088ea8 --> 0x40 ('@')
0032| 0xc000088eb0 --> 0xc0000a2040 ("tGRBtXMZgD6ZhalBtCUTgWgZfnkTgqoNsnAVsmUY
0040| 0xc000088eb8 --> 0x38 ('8')
0048| 0xc000088ec0 --> 0x1
0056| 0xc000088ec8 --> 0x1
[------------------------------------------------------------------------------
Legend: code, data, rodata, value
0x00000000004952e1 in ?? ()
gdb-peda$
```

```
.text:000000000049523A                 mov     [rsp+100h+var_100], rdx
.text:000000000049523E                 mov     [rsp+100h+var_F8], rcx
.text:0000000000495243                 lea     rcx, unk_4C8569
.text:000000000049524A                 mov     [rsp+100h+var_F0], rcx
.text:000000000049524F                 mov     [rsp+100h+var_E8], 2
.text:0000000000495258                 lea     rcx, [rsp+100h+var_28]
.text:0000000000495260                 mov     [rsp+100h+var_E0], rcx
.text:0000000000495265                 mov     [rsp+100h+var_D8], 1
.text:000000000049526E                 mov     [rsp+100h+var_D0], 1
.text:0000000000495277                 call    fmt_Fscanf
.text:000000000049527C                 lea     rax, [rsp+100h+var_88]
.text:0000000000495281                 mov     [rsp+100h+var_100], rax
.text:0000000000495285                 lea     rax, aSAlloccountSNe+523h ; "tGRBtXMZgD6ZhalBtCUTgWgZfnkTgqoNsnAVsmU"...
.text:000000000049528C                 mov     [rsp+100h+var_F8], rax
.text:0000000000495291                 mov     [rsp+100h+var_F0], 38h
.text:000000000049529A                 call    runtime_stringtoslicebyte
.text:000000000049529F                 nop
.text:00000000004952A0                 mov     rax, [rsp+100h+var_E0]
.text:00000000004952A5                 mov     rcx, [rsp+100h+var_D8]
.text:00000000004952AA                 mov     rdx, [rsp+100h+var_E8]
.text:00000000004952AF                 lea     rbx, [rsp+100h+var_A8]
.text:00000000004952B4                 mov     [rsp+100h+var_100], rbx
.text:00000000004952B8                 mov     [rsp+100h+var_F8], rdx
.text:00000000004952BD                 mov     [rsp+100h+var_F0], rax
.text:00000000004952C2                 mov     [rsp+100h+var_E8], rcx
.text:00000000004952C7                 call    runtime_slicebytetostring
.text:00000000004952CC                 mov     rax, cs:qword_572B00
.text:00000000004952D3                 mov     rcx, [rsp+100h+var_E0]
.text:00000000004952D8                 mov     rdx, [rsp+100h+var_D8]
.text:00000000004952DD                 mov     [rsp+100h+var_100], rax
.text:00000000004952E1                 mov     [rsp+100h+var_F8], rcx
.text:00000000004952E6                 mov     [rsp+100h+var_F0], rdx
.text:00000000004952EB                 call    encoding_base64__ptr_Encoding_DecodeString ; encoding_base64__ptr_Encoding_D
.text:00000000004952F0                 mov     rax, [rsp+100h+var_C8]
.text:00000000004952F5                 mov     rcx, [rsp+100h+var_D0]
.text:00000000004952FA                 mov     rdx, [rsp+100h+var_E8]
.text:00000000004952FF                 mov     rbx, [rsp+100h+var_E0]
.text:0000000000495304                 test    rcx, rcx
.text:0000000000495307                 jnz     loc_495418
.text:000000000049530D
.text:000000000049530D loc_49530D:                             ; CODE XREF: main_main+36A↓j
.text:000000000049530D                 mov     rax, [rsp+100h+var_60]
.text:0000000000495315                 mov     rcx, [rax]
.text:0000000000495318                 cmp     [rax+8], rbx
.text:000000000049531C                 jz      short loc_495393
.text:000000000049531E
.text:000000000049531E loc_49531E:                             ; CODE XREF: main_main+25B↓j
.text:000000000049531E                 xorps   xmm0, xmm0
.text:0000000000495321                 movups  [rsp+100h+var_58], xmm0
.text:0000000000495329                 lea     rax, unk_4A6D00
.text:0000000000495330                 mov     qword ptr [rsp+100h+var_58], rax
000952E1 00000000004952E1: main_main+191
```

```
Could not connect to 192.168.1.250:23946: 由于连接方在一段时间后没有正确答复或连接的主机没有反应，连接尝试失败。
Python
AU:  idle   向下     磁盘: 333GB
```

继续运行程序，到0x4952eb后即可发现flag



```
RAX: 0x2a ('*')
RBX: 0x2a ('*')
RCX: 0x0
RDX: 0x0
RSI: 0xc000090060 ("flag{92094daf-33c9-431e-a85a-8bfbd5df98ad}")
RDI: 0x38 ('8')
RBP: 0xc000088f88 --> 0xc000088f90 --> 0x429b1c (mov    eax,DWORD PTR [rip+0x16478e]
RSP: 0xc000088e90 --> 0xc00008c580 ("6789_-abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRST
RIP: 0x4952f0 (mov    rax,QWORD PTR [rsp+0x38])
R8 : 0x0
R9 : 0x0
R10: 0x2a ('*')
R11: 0x2a ('*')
R12: 0xc000090060 ("flag{92094daf-33c9-431e-a85a-8bfbd5df98ad}")
R13: 0xc00008c580 ("6789_-abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ012345", '
R14: 0x2a ('*')
R15: 0x40 ('@')
EFLAGS: 0x206 (carry PARITY adjust zero sign trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x4952e1:    mov    QWORD PTR [rsp+0x8],rcx
   0x4952e6:    mov    QWORD PTR [rsp+0x10],rdx
   0x4952eb:    call   0x47e620
=> 0x4952f0:    mov    rax,QWORD PTR [rsp+0x38]
   0x4952f5:    mov    rcx,QWORD PTR [rsp+0x30]
   0x4952fa:    mov    rdx,QWORD PTR [rsp+0x18]
   0x4952ff:    mov    rbx,QWORD PTR [rsp+0x20]
```

用py脚本自定义base64加密，解得flag

```
import string

import base64

flag = 'tGRBtXMZgD6ZhalBtCUTgWgZfnkTgqoNsnAVsmUYsGtCt9pEtDEYsql3'

std_table = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

my_table = "6789_-abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ012345"

flag = flag.translate(string.maketrans(my_table,std_table))

print base64.b64decode(flag)
```

IDAGolangHelper项目地址
https://github.com/sibears/IDAGolangHelper

参考链接

无符号Golang程序逆向方法解析 https://www.anquanke.com/post/id/170332

golang base64加密与解密 https://studygolang.com/articles/6926

MIPS架构的CTF逆向题--SUCTFbabyre题目
writeup https://blog.csdn.net/xiangshangbashaonian/article/details/83146678