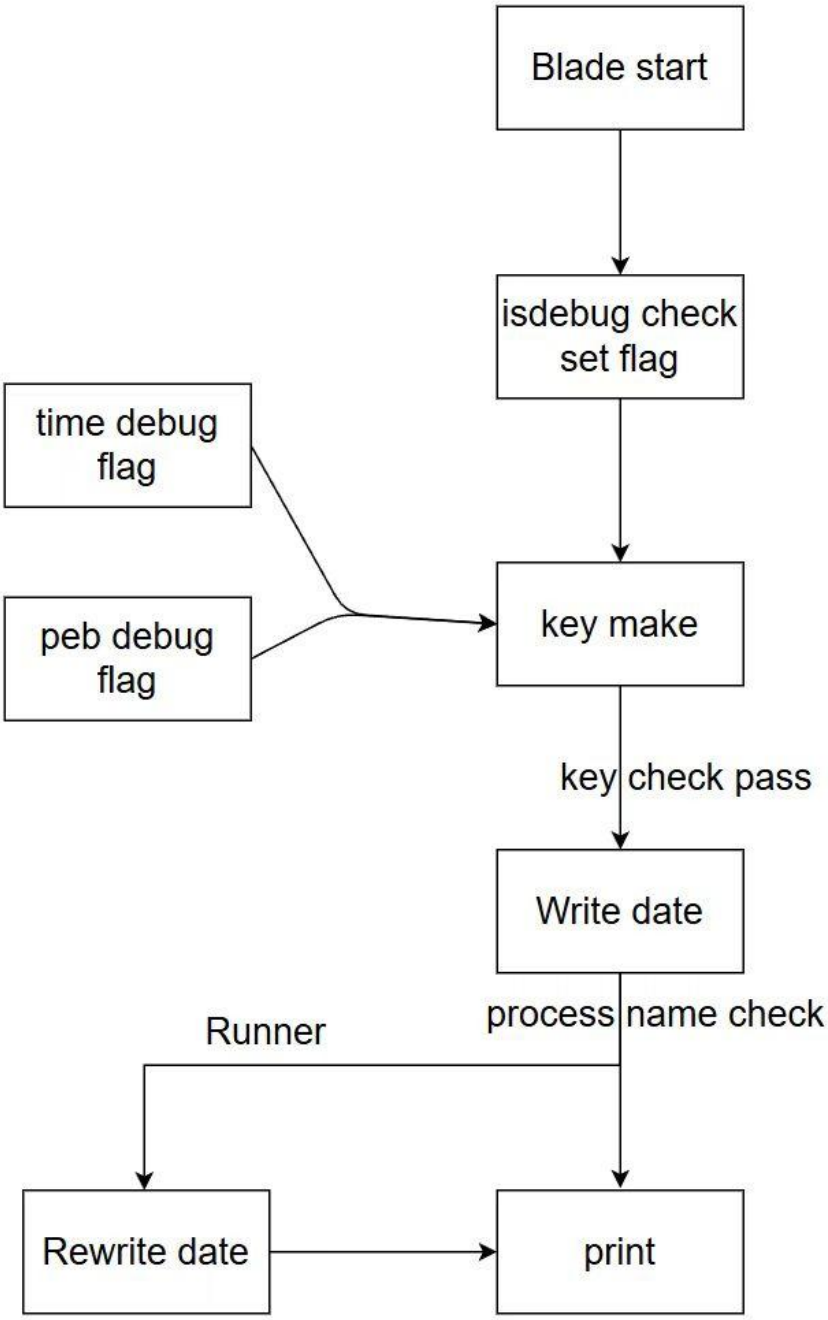
Blade WP

Cgo 在程序中拥有三种调用约定，即 go 寄存器调用，go 栈调用，x64fastcall，c 调用 go、go 调 c 都通过大量系统函数，使用函数指针的方式进行调用。Go 拥有大量系统库函数，建议制作符号文件将 go 的部分基础库函数解析出来再作分析。注：使用 go_parser 目前无法解开 garble 混淆，不过可以通过修改 pclinetable header 的魔数和修改脚本解开部分混淆。

选手需要熟悉 cgo 整体特性和流程，并且将整体程序逻辑分析清楚即可解题。
程序整体逻辑较为简单：



根据字符串明文交叉引用定位 main 函数:

```
.rdata:00007FF6E7EECBFE unk_7FF6E7EECBFE db  52h ; R
.rdata:00007FF6E7EECBFF db  61h ; a
.rdata:00007FF6E7EECC00 db  69h ; i
.rdata:00007FF6E7EECC01 db  6Eh ; n
.rdata:00007FF6E7EECC02 db  73h ; s
.rdata:00007FF6E7EECC03 db  3Ah ; :
.rdata:00007FF6E7EECC04 db  0Ah
```

```
v17 = v11;
v33 = sub_7FF6E7E99360(v5, v4, v6, v7, *(_QWORD *)&v11[7], v12);
v25 = v11;
v27 = sub_7FF6E7E99C40();
v18 = v11;
v29 = sub_7FF6E7E9A3C0();
v20 = v11;
v32 = sub_7FF6E7E9A580();
v24 = v11;
sub_7FF6E7E9AA00();
v8 = sub_7FF6E7D81D62();
v35 = v23;
v34 = v31;
v37 = v19;
v36 = v28;
v39 = v17;
v38 = v26;
v41 = v25;
v40 = v33;
v43 = v18;
v42 = v27;
v45 = v20;
v44 = v29;
v47 = v24;
v46 = v32;
v49 = v11;
v48 = v8;
v30 = sub_7FF6E7E91E60(8i64);
v21 = 8i64;
v22 = v9;
sub_7FF6E7E92160();
if ( (BYTE5(qword_7FF6E80008C0) ^ 2) == 1 )
  sub_7FF6E7E93440();
sub_7FF6E7E92280(v22);
sub_7FF6E7E92480();
sub_7FF6E7E92600();
return sub_7FF6E7E927A0();
```

过 IsDebuggerPresent，修改该标志为 2：

```
  sub_7FF6E7E92160();
  if ( (unk_7FF6E80008C5 ^ 2) == 1 )
    sub_7FF6E7E93440();
```

```
.bss:00007FF6E80008C4 db  4Dh ; M
.bss:00007FF6E80008C5 unk_7FF6E80008C5 db    2                ; DAT
.bss:00007FF6E80008C5                                          ; sub
```

在 main 函数中的密钥制作处，根据分析将反调试的 c6 和 c7 的值改为相等即可，使 v7=0x4f，此处 c1 为 win10 和 win11 的 NtCreateThread 函数的系统调用号，都为 4e。
得到密钥：

```
  v48 = v8;
  v30 = sub_7FF6E7E91E60(8i64);
  v21 = 8i64;
  v22 = v9;
  sub_7FF6E7E92160();
  if ( (BYTE5(byte_7FF6E80008C0) ^ 2) == 1 )
    sub_7FF6E7E93440();
  sub_7FF6E7E92280(v22);
  sub_7FF6E7E92480();
  sub_7FF6E7E92600();
  return sub_7FF6E7E927A0();
```

```
__int64 __fastcall sub_7FF6E7E92600(__int64 a1, __int64 a2, __int64 a3, __int64 a4)
{
  __int64 v4; // rax
  unsigned __int64 v5; // rbx
  __int64 v6; // r14
  __int64 v7; // rdx
  __int64 i; // rsi
  char v9; // al
  int v10; // edx
  int v11; // r8d
  int v12; // r9d
  int v13; // edx
  int v14; // r8d
  int v15; // r9d
  _QWORD v17[2]; // [rsp+38h] [rbp-20h] BYREF
  _QWORD v18[2]; // [rsp+48h] [rbp-10h] BYREF
  void *retaddr; // [rsp+60h] [rbp+8h] BYREF
  __int64 key; // [rsp+68h] [rbp+10h]

  if ( (unsigned __int64)&retaddr <= *(_QWORD *)(v6 + 16) )
    sub_7FF6E7D7F7C0(a1, a2, a3, a4);
  v7 = unk_7FF6E80008C6 + unk_7FF6E80008C1 + unk_7FF6E80008C5 - (unsigned int)unk_7FF6E80008C7 - 1;
  for ( i = 0i64; i < 8; ++i )
  {
    if ( v5 <= i )
      sub_7FF6E7D81AC0(v5, v7);
    *(_BYTE *)(v4 + i) ^= v7;
  }
  key = v4;
  sub_7FF6E7E8FE00(a1, v7, a3, a4);
  sub_7FF6E7E96900();
  if ( v5 == 16 )
    v9 = sub_7FF6E7D23520(16i64);
  else
    v9 = 0;
  if ( v9 )
  {
    sub_7FF6E7E96A00();
    v18[0] = &unk_7FF6E7EC47E0;
    v18[1] = sub_7FF6E7D76940();
```

密钥:

```
1182CF db     0
1182D0 unk_C0001182D0 db 0BBh
1182D1 db   7Dh ; }
1182D2 db   83h
1182D3 db   2Dh ; -
1182D4 db 0ABh
1182D5 db   45h ; E
1182D6 db   93h
1182D7 db   95h
1182D8 db     0
1182D9 db     0
1182DA db     0
```

进入 main 函数最后:

```
v42 = v27;
v45 = v20;
v44 = v29;
v47 = v24;
v46 = v32;
v49 = v11;
v48 = v8;
v30 = sub_7FF6E7E91E60(8i64);
v21 = 8i64;
v22 = v9;
sub_7FF6E7E92160();
if ( (BYTE5(byte_7FF6E80008C0) ^ 2) == 1 )
  sub_7FF6E7E93440();
sub_7FF6E7E92280(v22);
sub_7FF6E7E92480();
sub_7FF6E7E92600();
return sub_7FF6E7E927A0();
}
```

根据分析可知 8D0 为 pipe_write，最终结果通过 8C8 pipe_read 读取并打印:

```
sub_7FF6E7E964C0();
sub_7FF6E7E96760();
v49 = v1;
v23 = sub_7FF6E7D76940();
*(_QWORD *)&v49 = &unk_7FF6E7EC47E0;
*((_QWORD *)&v49 + 1) = v23;
sub_7FF6E7E73560((unsigned int)&v49, v24, v25, v26);
sub_7FF6E7E93440();
sub_7FF6E7E91820();
qword_7FF6E80008C8 = *v37;
qword_7FF6E80008D0 = v37[1];
sub_7FF6E7E92D60();
sub_7FF6E7D2C5A0();
sub_7FF6E7E8E280();
sub_7FF6E7E96880();
v48 = sub_7FF6E7E92CE0();
v46 = v27;
```

分析可知，需要使用当前的程序名的 md5 作对比，程序名很简单，此处也可以使用哈希爆破得出程序名:

```
sub_14016FE00(v9, v8, v10, v11);
result = sub_140175AA0();
if ( v14 == 0x10 )
{
  result = sub_140003520(16i64);              // filename md5 check
  if ( (_BYTE)result )
  {
    sub_1401760E0();
    sub_140171460();
    return sub_140171780();
  }
}
return result;
```

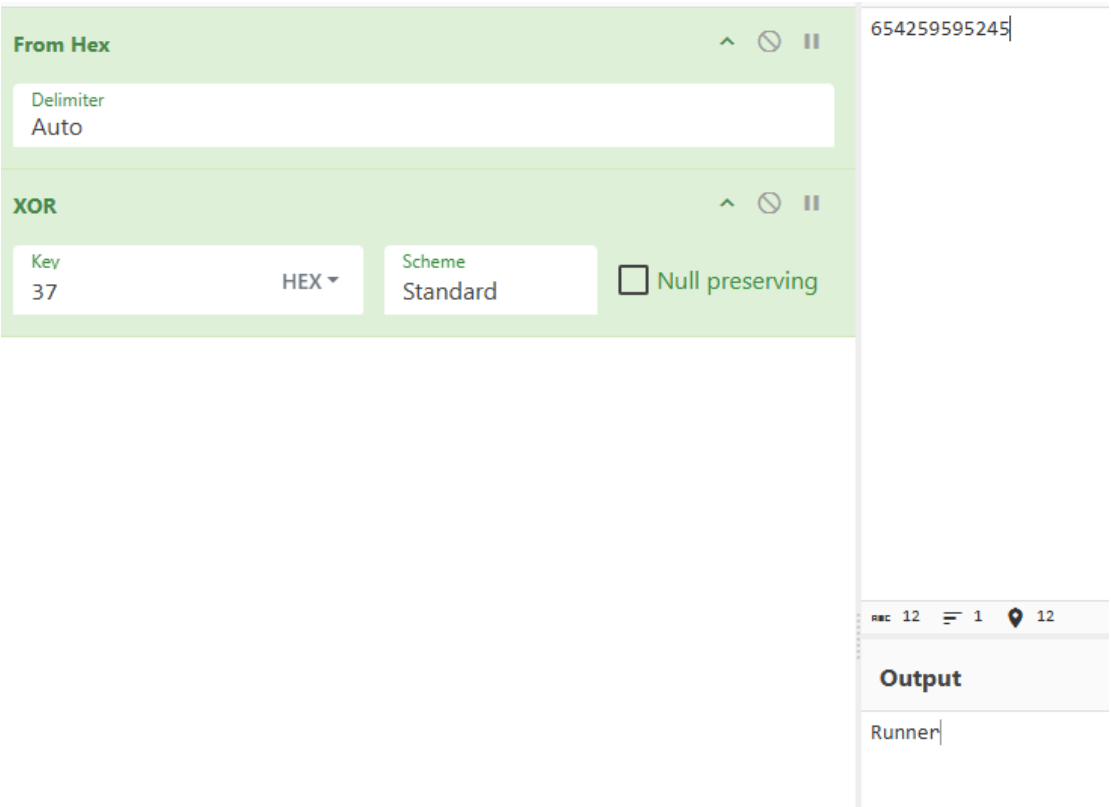
利用程序名的异或结果指定跳转：

```
  v6 = *a4 ^ 0x37;
  v7 = a4[1] ^ 0x37;
  v8 = a4[2] ^ 0x37;
  v9 = a4[3] ^ 0x37;
  v10 = a4[4] ^ 0x37;
  v11 = a4[5] ^ 0x37;
  while ( *(unsigned __int8 *)i == v6
       && *((unsigned __int8 *)i + 1) == v7
       && *((unsigned __int8 *)i + 2) == v8
       && *((unsigned __int8 *)i + 3) == v9
       && *((unsigned __int8 *)i + 4) == v10 )
  {
    if ( *((unsigned __int8 *)i + 5) == v11 )
      goto LABEL_21;
    if ( v6 == *v4 )
      goto LABEL_11;
.ABEL_3:
    i = (__int64 (__golang *)())((char *)i + 1);
    --v4;
  }
  if ( v6 != *v4 )
    goto LABEL_3;
.ABEL_11:
  if ( v4[1] != v7 || v4[2] != v8 || v4[3] != v9 || v4[4] != v10 || v4[5] != v11 )
    goto LABEL_3;
.ABEL_21:
  for ( i = (__int64 (__golang *)())v4;
        *(_BYTE *)i != 85 || *((_BYTE *)i + 1) != 87 || *((_BYTE *)i + 2) != 86 || *((_BYTE *)i + 3) != 83;
        i = (__int64 (__golang *)())((char *)i - 1) )
  {
    ;
  }
  i();
```

数据一定是通过 pipe_write 写入管道的，因此可以交叉引用 pipe_write，从 cgo 调用中，向上层寻找，可找到需要跳转的目的地：

```
  v0 = off_140196700;
  v5[0] = 0x455259594265i64;
  v1 = v5;
  do
  {
    v2 = *(_BYTE *)v1;
    v1 = (__int64 *)((char *)v1 + 1);
    v2 ^= 0x37u;
    *((_BYTE *)v1 - 1) = v2;
    sub_14017B5C0("%c", (unsigned int)v2);
    v3 = (FILE *)((__int64 (__fastcall *)(__int64))v0)(1i64);
    fflush(v3);
  }
  while ( v1 != (__int64 *)((char *)v5 + 6) );
  return sub_14017AFF0();
```

使用 0x37 异或可得程序名：



将程序名改为 Runner.exe，运行并输入密钥：

```
Rains:
0xF4 0x32 0xCC 0x62 0xE4 0x0A 0xDC 0xDA
There is a man sitting in the rain...
"Do Androids Dream of Electric Sheep? I am a replicant from the past, I forgot something."
"Rains hide something, read it for me."
eg:1a2a3a4a5a6a7a8a
bb7d832dab459395
Runner
I am Roy Batty.I am from 2019.
I've seen things you people wouldn't believe.
Attack ships on fire off shoulder of Orion.
I watched C-beams glitter in the dark near the Tannhauser Gate.
All those moments will be lost in time,
like tears in rain.
Time to die
flag{`` `` `` `` ``tear`` `` `` `` ``i`` `` `` `` ``n`` `` `` `` ``rain`` `` `` `` ``}
```

得到 flag{`` `` `` `` ``tear`` `` `` `` ``i`` `` `` `` ``n`` `` `` `` ``rain`` `` `` `` ``}

熟悉这段 Roy 对 Runner 的绝美雨中泪诀别或许可以直接猜到需要的程序名是什么（- :