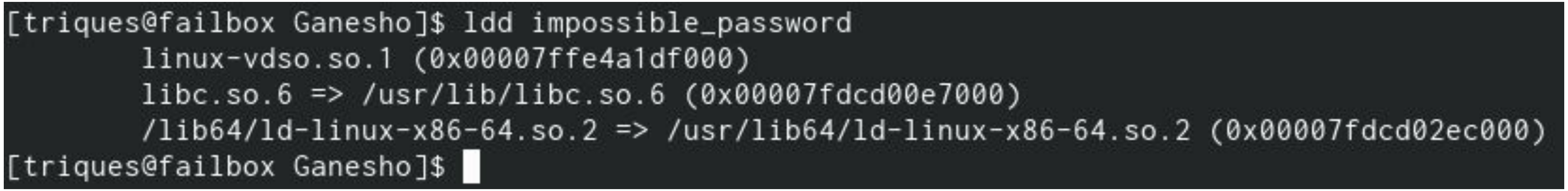
LD_PRELOAD

LDD

- Printa os shared objects (.so) que serão carregados pelo LD



```
[triques@failbox Ganesho]$ ldd impossible_password
        linux-vdso.so.1 (0x00007ffe4a1df000)
        libc.so.6 => /usr/lib/libc.so.6 (0x00007fdcd00e7000)
        /lib64/ld-linux-x86-64.so.2 => /usr/lib64/ld-linux-x86-64.so.2 (0x00007fdcd02ec000)
[triques@failbox Ganesho]$ 
```

Shared Objects

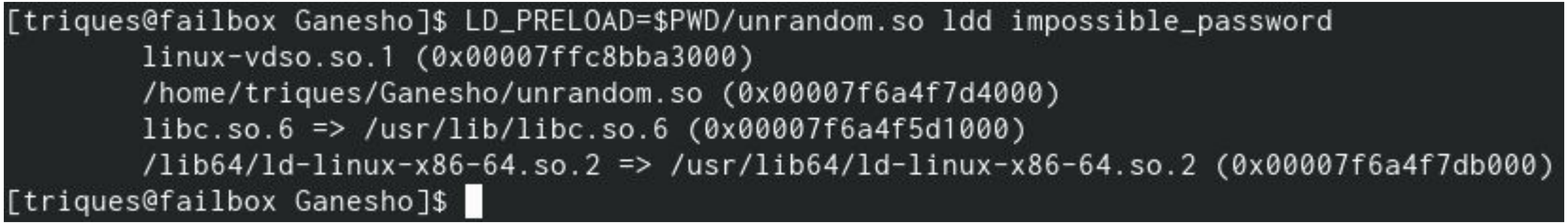
- Bibliotecas
- Reuso de funções
    - libc - scanf, printf ...
    - Diminui tamanho do binário

LD

- dynamic linker/loader
- Procura e carrega .so, prepara o programa e roda

LD_PRELOAD

- Variável de ambiente
- “Força” carregamento de um shared object primeiro

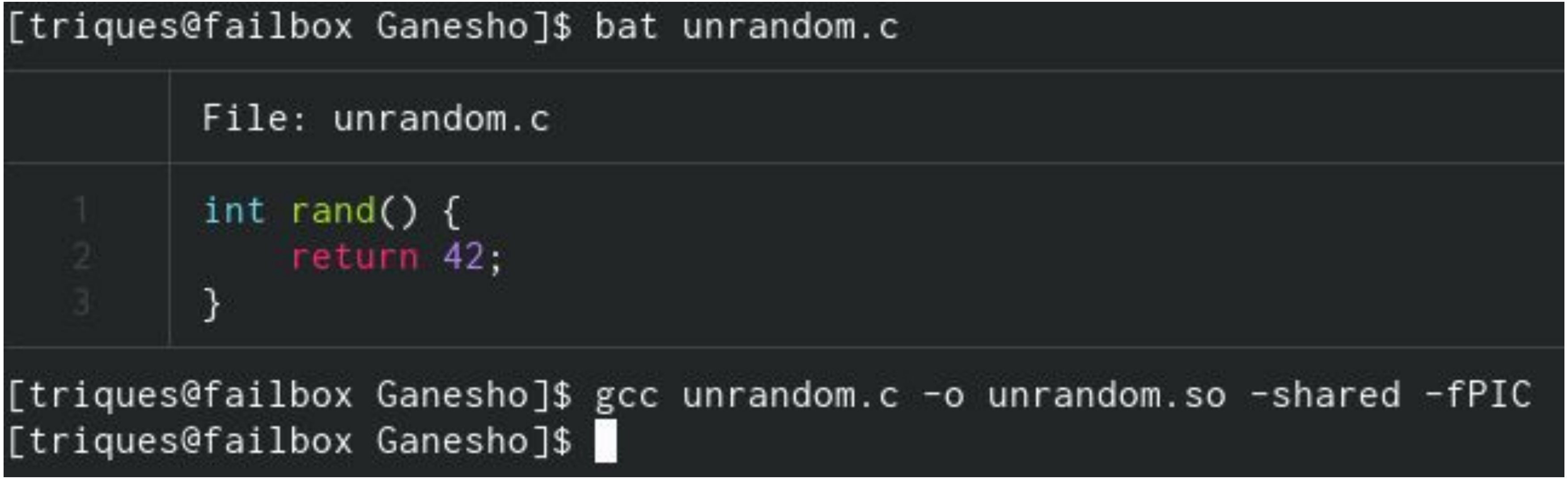


```
[triques@failbox Ganesho]$ LD_PRELOAD=$PWD/unrandom.so ldd impossible_password
	linux-vdso.so.1 (0x00007ffc8bba3000)
	/home/triques/Ganesho/unrandom.so (0x00007f6a4f7d4000)
	libc.so.6 => /usr/lib/libc.so.6 (0x00007f6a4f5d1000)
	/lib64/ld-linux-x86-64.so.2 => /usr/lib64/ld-linux-x86-64.so.2 (0x00007f6a4f7db000)
[triques@failbox Ganesho]$ 
```

Parte divertida

Idéia

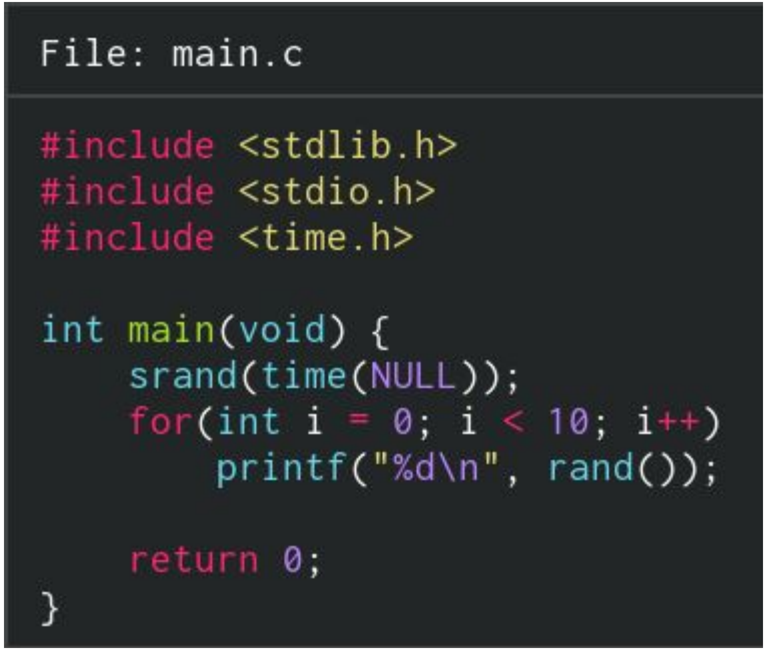
- Recriar uma função já existente



```
[triques@failbox Ganesho]$ bat unrandom.c

       File: unrandom.c

       int rand() {
           return 42;
       }

[triques@failbox Ganesho]$ gcc unrandom.c -o unrandom.so -shared -fPIC
[triques@failbox Ganesho]$ 
```

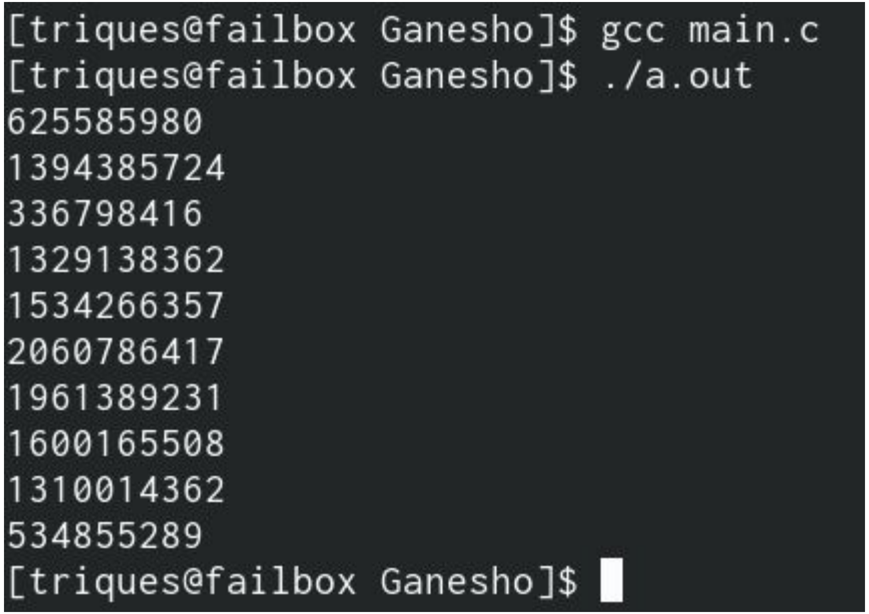
Idéia

- Gerar .so e fazer preload
- Exportar x Setar LD_PRELOAD
    - válido para sessão x única vez

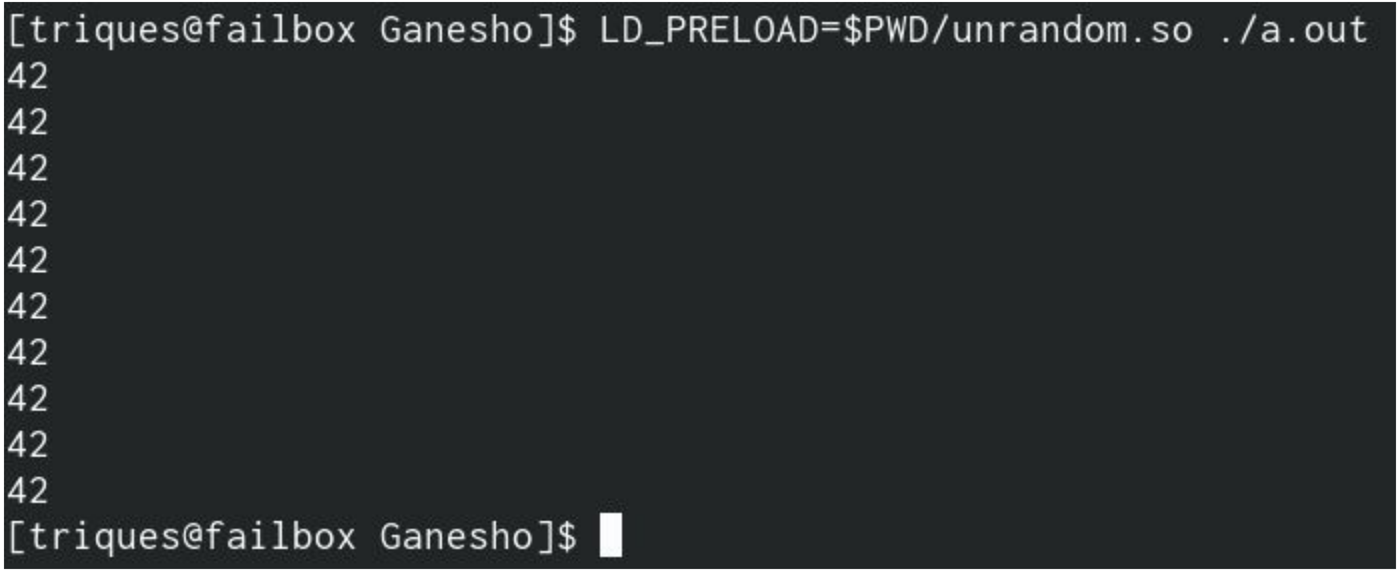


```
File: main.c
```

```c
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

int main(void) {
    srand(time(NULL));
    for(int i = 0; i < 10; i++)
        printf("%d\n", rand());

    return 0;
}
```

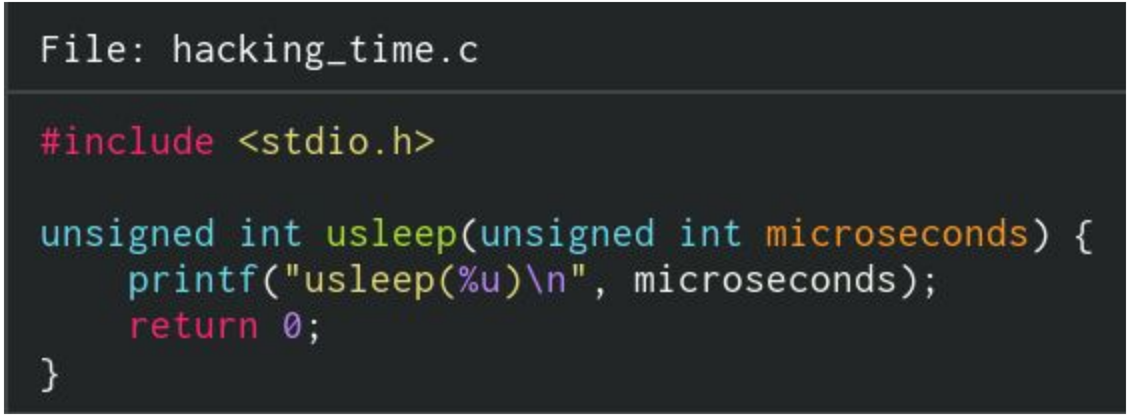


```
[triques@failbox Ganesho]$ gcc main.c
[triques@failbox Ganesho]$ ./a.out
625585980
1394385724
336798416
1329138362
1534266357
2060786417
1961389231
1600165508
1310014362
534855289
[triques@failbox Ganesho]$ 
```



```
[triques@failbox Ganesho]$ LD_PRELOAD=$PWD/unrandom.so ./a.out
42
42
42
42
42
42
42
42
42
42
[triques@failbox Ganesho]$ 
```

CTF da CrytoRave

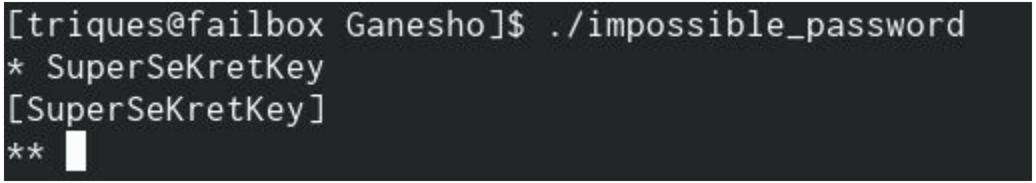


File: hacking_time.c

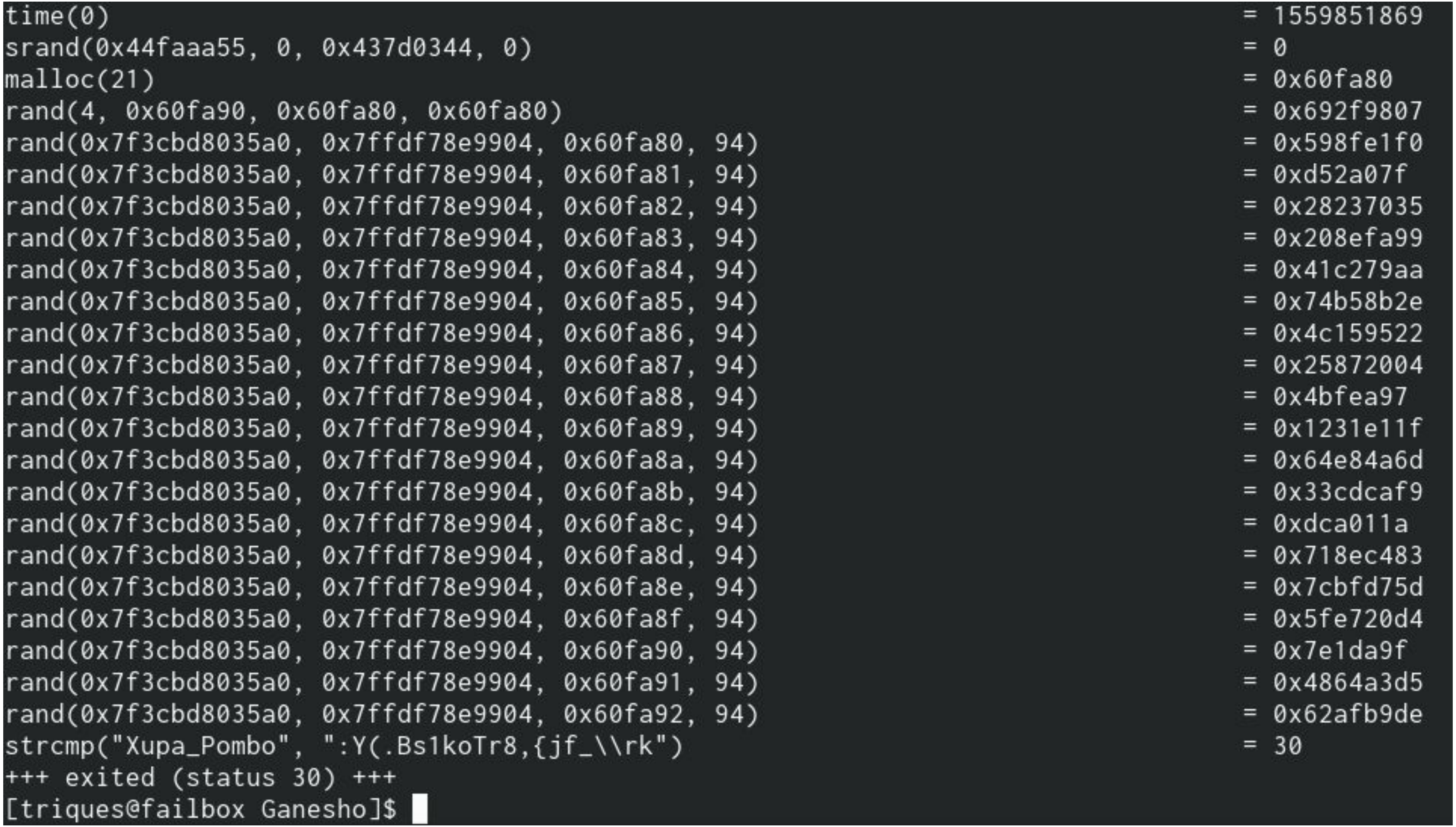
```c
#include <stdio.h>

unsigned int usleep(unsigned int microseconds) {
    printf("usleep(%u)\n", microseconds);
    return 0;
}
```

Chall do HackTheBox

Impossible Password



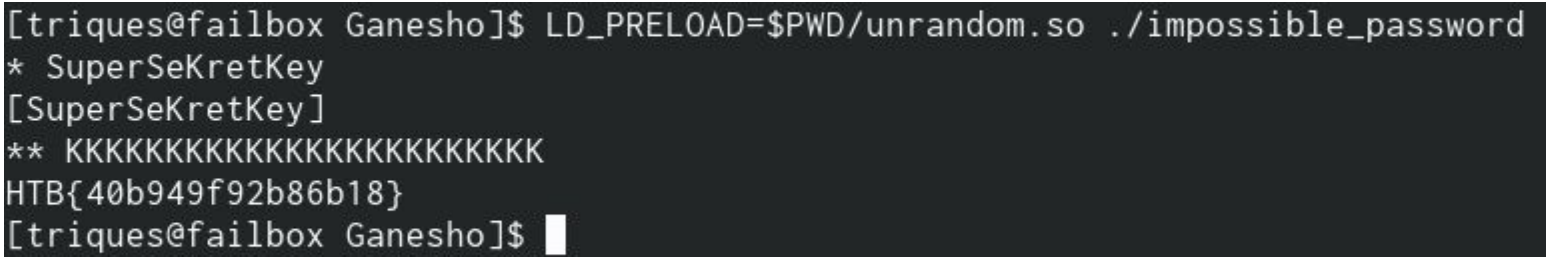
```
[triques@failbox Ganesho]$ ./impossible_password
* SuperSeKretKey
[SuperSeKretKey]
**
```



```
time(0)                                                                           = 1559851869
srand(0x44faaa55, 0, 0x437d0344, 0)                                               = 0
malloc(21)                                                                        = 0x60fa80
rand(4, 0x60fa90, 0x60fa80, 0x60fa80)                                             = 0x692f9807
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa80, 94)                                = 0x598fe1f0
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa81, 94)                                = 0xd52a07f
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa82, 94)                                = 0x28237035
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa83, 94)                                = 0x208efa99
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa84, 94)                                = 0x41c279aa
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa85, 94)                                = 0x74b58b2e
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa86, 94)                                = 0x4c159522
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa87, 94)                                = 0x25872004
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa88, 94)                                = 0x4bfea97
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa89, 94)                                = 0x1231e11f
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa8a, 94)                                = 0x64e84a6d
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa8b, 94)                                = 0x33cdcaf9
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa8c, 94)                                = 0xdca011a
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa8d, 94)                                = 0x718ec483
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa8e, 94)                                = 0x7cbfd75d
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa8f, 94)                                = 0x5fe720d4
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa90, 94)                                = 0x7e1da9f
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa91, 94)                                = 0x4864a3d5
rand(0x7f3cbd8035a0, 0x7ffdf78e9904, 0x60fa92, 94)                                = 0x62afb9de
strcmp("Xupa_Pombo", ":Y(.Bs1koTr8,{jf_\\rk")                                     = 30
+++ exited (status 30) +++
[triques@failbox Ganesho]$ 
```

Chall do HackTheBox com LD_PRELOAD

fixando rand



```
[triques@failbox Ganesho]$ LD_PRELOAD=$PWD/unrandom.so ./impossible_password
* SuperSeKretKey
[SuperSeKretKey]
** KKKKKKKKKKKKKKKKKKKKKKKK
HTB{40b949f92b86b18}
[triques@failbox Ganesho]$ 
```

Pwn Adventure 3

- Série de vídeos do LiveOverflow

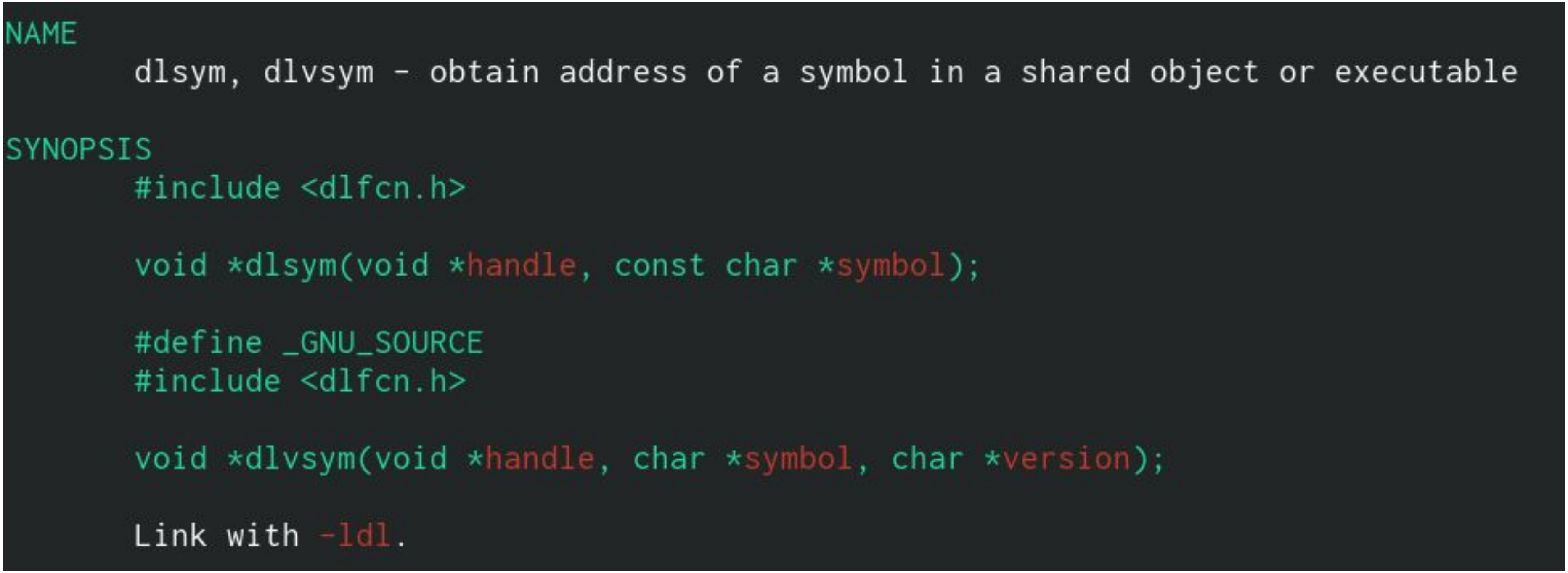
Problema

- Impossível chamar função “original”

Solução

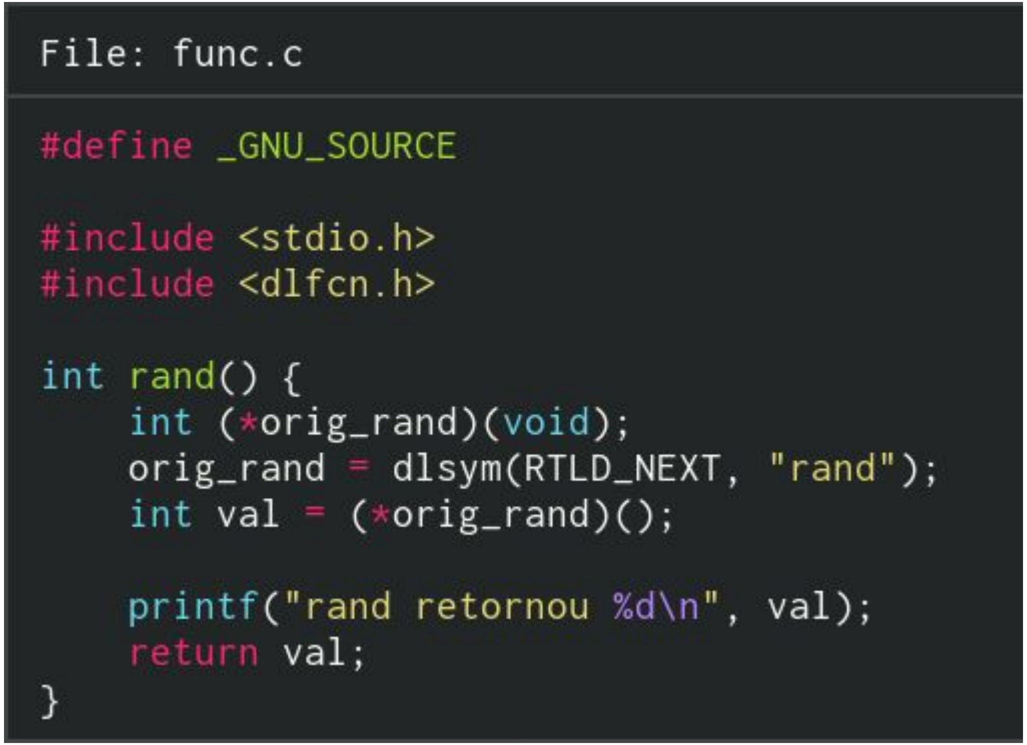
- dlsym
  - obtém endereço da função original

man dlsym



```
NAME
       dlsym, dlvsym - obtain address of a symbol in a shared object or executable

SYNOPSIS
       #include <dlfcn.h>

       void *dlsym(void *handle, const char *symbol);

       #define _GNU_SOURCE
       #include <dlfcn.h>

       void *dlvsym(void *handle, char *symbol, char *version);

       Link with -ldl.
```

Solução

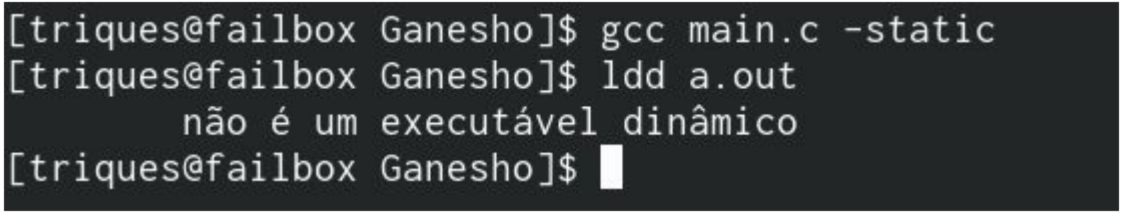


File: func.c

```
#define _GNU_SOURCE

#include <stdio.h>
#include <dlfcn.h>

int rand() {
    int (*orig_rand)(void);
    orig_rand = dlsym(RTLD_NEXT, "rand");
    int val = (*orig_rand)();

    printf("rand retornou %d\n", val);
    return val;
}
```

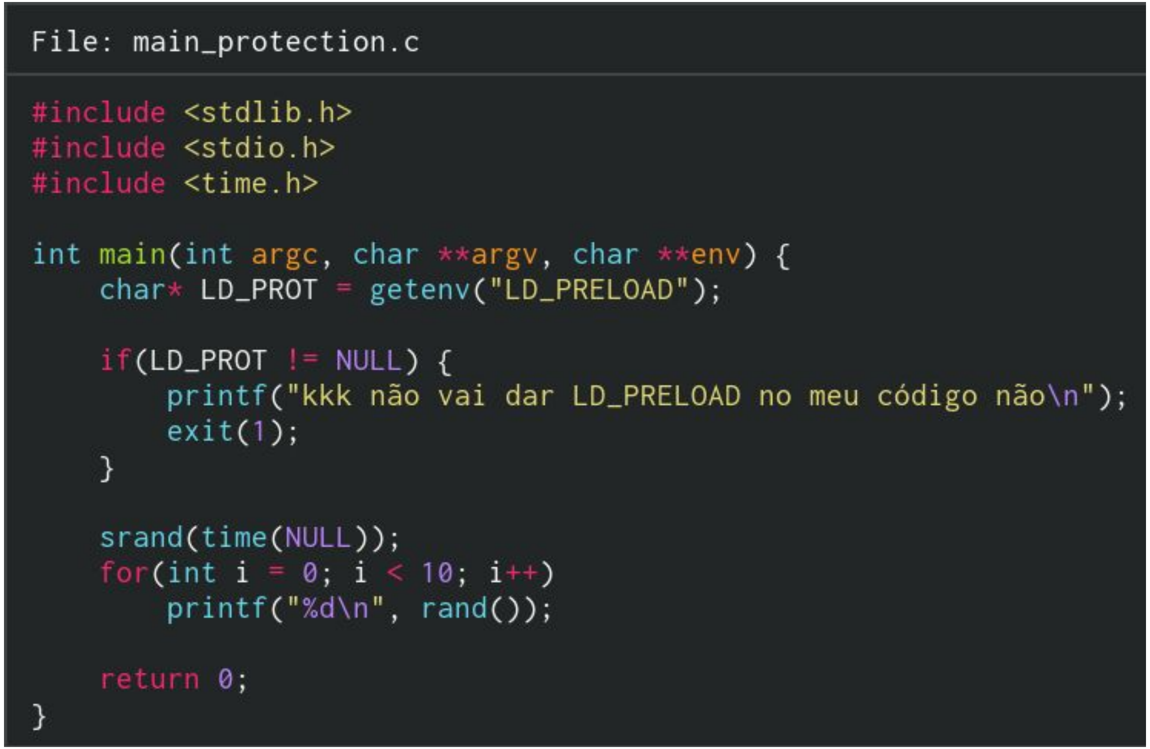
Proteção

- Linkagem estática



```
[triques@failbox Ganesho]$ gcc main.c -static
[triques@failbox Ganesho]$ ldd a.out
        não é um executável dinâmico
[triques@failbox Ganesho]$ 
```

- Checar variável LD_PRELOAD



File: main_protection.c

```c
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

int main(int argc, char **argv, char **env) {
    char* LD_PROT = getenv("LD_PRELOAD");

    if(LD_PROT != NULL) {
        printf("kkk não vai dar LD_PRELOAD no meu código não\n");
        exit(1);
    }

    srand(time(NULL));
    for(int i = 0; i < 10; i++)
        printf("%d\n", rand());

    return 0;
}
```