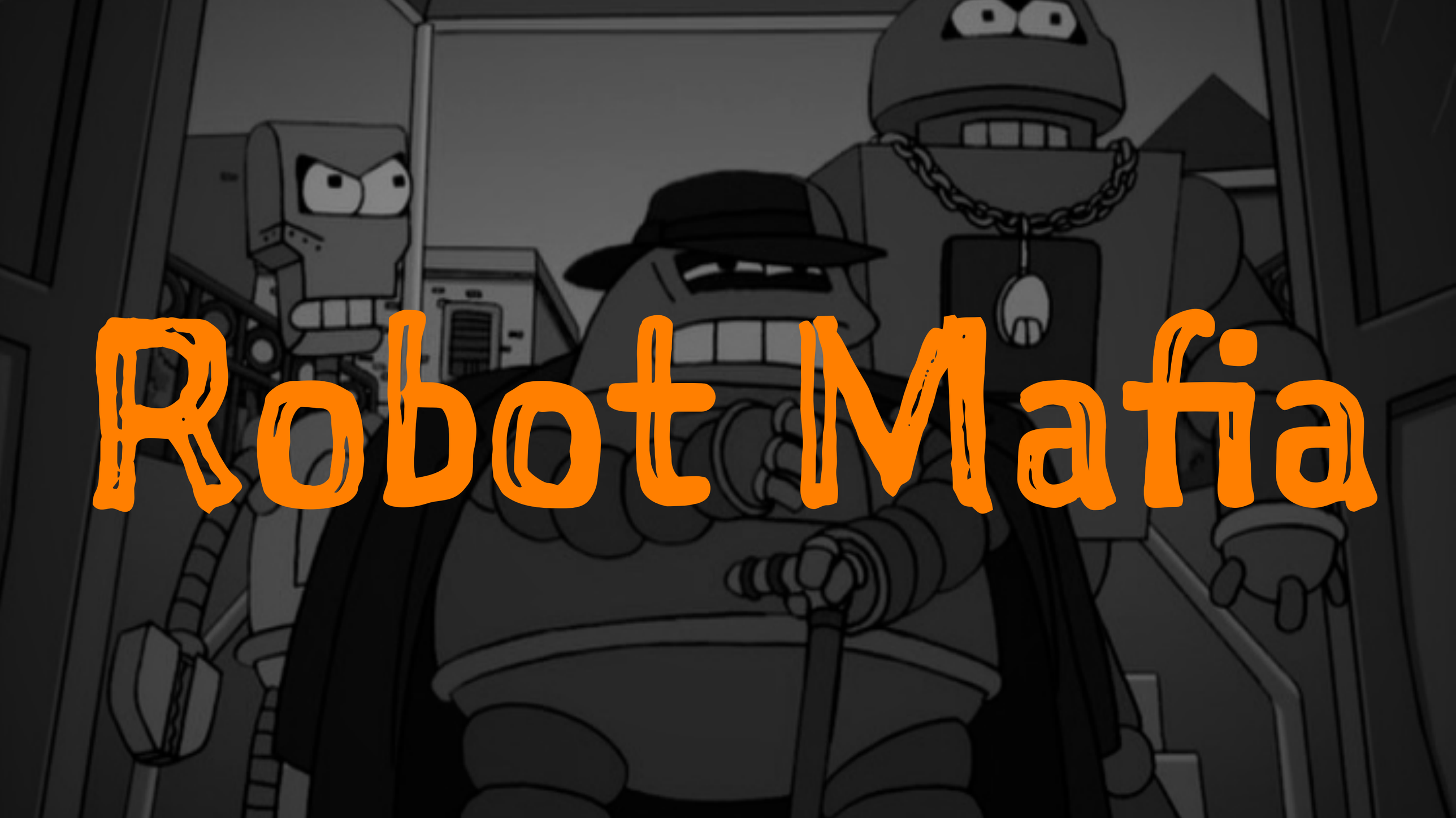

Make PWN'ng Great Again


PWN
THINGS


Who?


Robot Mafia

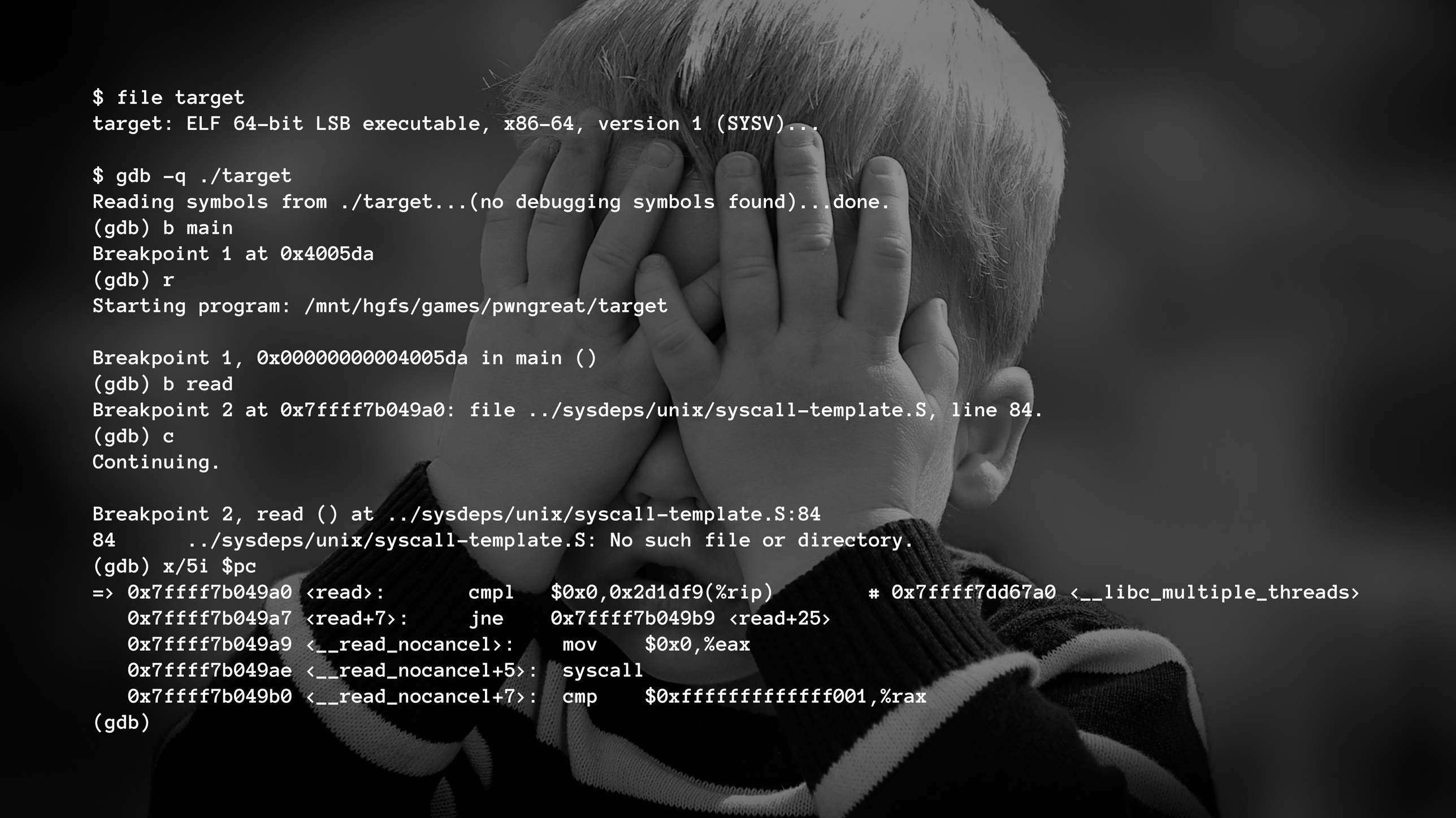
wat?

pwndbg /poʊndbæg/

https://github.com/pwndbg/pwndbg

PWNTOOLS

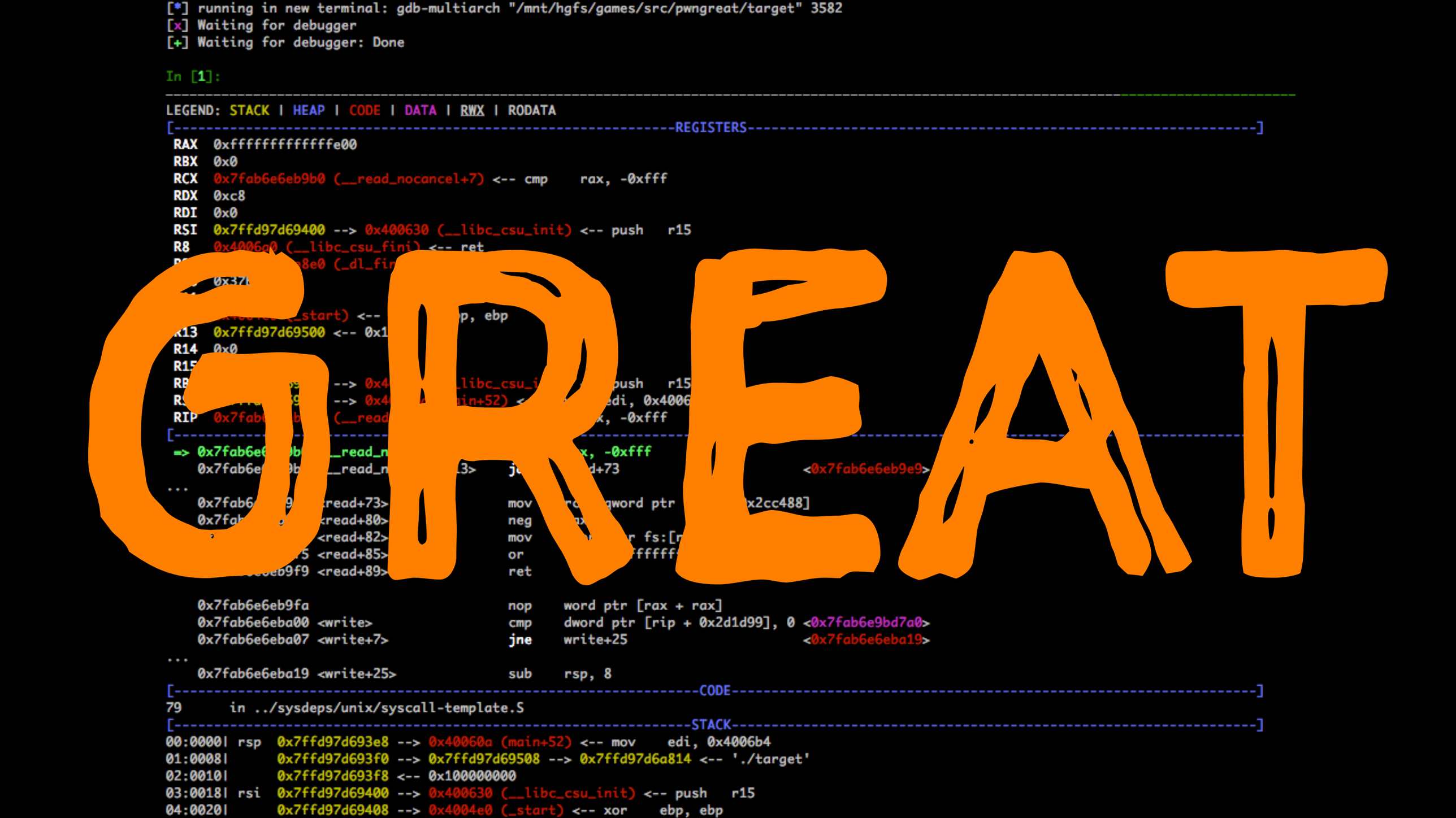
http://www.github.com/Gallopsled/pwntools



```
$ file target
target: ELF 64-bit LSB executable, x86-64, version 1 (SYSV)...

$ gdb -q ./target
Reading symbols from ./target...(no debugging symbols found)...done.
(gdb) b main
Breakpoint 1 at 0x4005da
(gdb) r
Starting program: /mnt/hgfs/games/pwngreat/target

Breakpoint 1, 0x00000000004005da in main ()
(gdb) b read
Breakpoint 2 at 0x7ffff7b049a0: file ../sysdeps/unix/syscall-template.S, line 84.
(gdb) c
Continuing.

Breakpoint 2, read () at ../sysdeps/unix/syscall-template.S:84
84      ../sysdeps/unix/syscall-template.S: No such file or directory.
(gdb) x/5i $pc
=> 0x7ffff7b049a0 <read>:       cmpl   $0x0,0x2d1df9(%rip)        # 0x7ffff7dd67a0 <__libc_multiple_threads>
   0x7ffff7b049a7 <read+7>:     jne    0x7ffff7b049b9 <read+25>
   0x7ffff7b049a9 <__read_nocancel>:    mov    $0x0,%eax
   0x7ffff7b049ae <__read_nocancel+5>:  syscall
   0x7ffff7b049b0 <__read_nocancel+7>:  cmp    $0xfffffffffffff001,%rax
(gdb)
```

```
[*] '/mnt/hgfs/games/src/pwngreat/target'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE
[x] Starting program './target'
[+] Starting program './target': Done
[*] running in new terminal: gdb-multiarch "/mnt/hgfs/games/src/pwngreat/target" 3582
[x] Waiting for debugger
[+] Waiting for debugger: Done

In [1]:
```

```
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
[-------------------------------------------------------------------REGISTERS-------------------------------------------------------------------]
 RAX  0xfffffffffffffe00
 RBX  0x0
 RCX  0x7fab6e6eb9b0 (__read_nocancel+7) <-- cmp    rax, -0xfff
 RDX  0xc8
 RDI  0x0
 RSI  0x7ffd97d69400 --> 0x400630 (__libc_csu_init) <-- push   r15
 R8   0x4006a0 (__libc_csu_fini) <-- ret
 R9   0x7fab6e9ce8e0 (_dl_fini) <-- push   rbp
 R10  0x37b
 R11  0x246
 R12  0x4004e0 (_start) <-- xor    ebp, ebp
 R13  0x7ffd97d69500 <-- 0x1
 R14  0x0
 R15  0x0
 RBP  0x7ffd97d69420 --> 0x400630 (__libc_csu_init) <-- push   r15
 RSP  0x7ffd97d693e8 --> 0x40060a (main+52) <-- mov    edi, 0x4006b4
 RIP  0x7fab6e6eb9b0 (__read_nocancel+7) <-- cmp    rax, -0xfff
[---------------------------------------------------------------------CODE--------------------------------------------------------------------]
 => 0x7fab6e6eb9b0 <__read_nocancel+7>     cmp    rax, -0xfff
    0x7fab6e6eb9b6 <__read_nocancel+13>    jae    read+73                     <0x7fab6e6eb9e9>
...
    0x7fab6e6eb9e9 <read+73>               mov    rcx, qword ptr [rip + 0x2cc488]
    0x7fab6e6eb9f0 <read+80>               neg    eax
    0x7fab6e6eb9f2 <read+82>               mov    dword ptr fs:[rcx], eax
    0x7fab6e6eb9f5 <read+85>               or     rax, 0xffffffffffffffff
    0x7fab6e6eb9f9 <read+89>               ret

    0x7fab6e6eb9fa                         nop    word ptr [rax + rax]
    0x7fab6e6eba00 <write>                 cmp    dword ptr [rip + 0x2d1d99], 0 <0x7fab6e9bd7a0>
    0x7fab6e6eba07 <write+7>               jne    write+25                    <0x7fab6e6eba19>
...
    0x7fab6e6eba19 <write+25>              sub    rsp, 8
[---------------------------------------------------------------------CODE--------------------------------------------------------------------]
79      in ../sysdeps/unix/syscall-template.S
[-------------------------------------------------------------------STACK-------------------------------------------------------------------]
00:0000| rsp  0x7ffd97d693e8 --> 0x40060a (main+52) <-- mov    edi, 0x4006b4
01:0008|      0x7ffd97d693f0 --> 0x7ffd97d69508 --> 0x7ffd97d6a814 <-- './target'
02:0010|      0x7ffd97d693f8 <-- 0x100000000
03:0018| rsi  0x7ffd97d69400 --> 0x400630 (__libc_csu_init) <-- push   r15
04:0020|      0x7ffd97d69408 --> 0x4004e0 (_start) <-- xor    ebp, ebp
05:0028|      0x7ffd97d69410 --> 0x7ffd97d69500 <-- 0x1
06:0030|      0x7ffd97d69418 <-- 0xdebc7841cec4bb00
07:0038| rbp  0x7ffd97d69420 --> 0x400630 (__libc_csu_init) <-- push   r15
[-----------------------------------------------------------------BACKTRACE-----------------------------------------------------------------]
>  f 0     7fab6e6eb9b0 __read_nocancel+7
   f 1          40060a main+52
   f 2     7fab6e615830 __libc_start_main+240
pwndbg>
```


[*] running in new terminal: gdb-multiarch "/mnt/hgfs/games/src/pwngreat/target" 3582
[x] Waiting for debugger
[+] Waiting for debugger: Done
In [1]:
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
REGISTERS
RAX 0xfffffffffffffe00
RBX 0x0
RCX 0x7fab6e6eb9b0 (__read_nocancel+7) <-- cmp rax, -0xfff
RDX 0xc8
RDI 0x0
RSI 0x7ffd97d69400 --> 0x400630 (__libc_csu_init) <-- push r15
R8 0x4006a0 (__libc_csu_fini) <-- ret
R13 0x7ffd97d69500 <-- 0x1
R14 0x0
R15
RIP 0x7fab
0x7fab6e6eb9fa nop word ptr [rax + rax]
0x7fab6e6eba00 <write> cmp dword ptr [rip + 0x2d1d99], 0 <0x7fab6e9bd7a0>
0x7fab6e6eba07 <write+7> jne write+25 <0x7fab6e6eba19>
0x7fab6e6eba19 <write+25> sub rsp, 8
CODE
79 in ../sysdeps/unix/syscall-template.S
STACK
00:0000| rsp 0x7ffd97d693e8 --> 0x40060a (main+52) <-- mov edi, 0x4006b4
01:0008| 0x7ffd97d693f0 --> 0x7ffd97d69508 --> 0x7ffd97d6a814 <-- './target'
02:0010| 0x7ffd97d693f8 <-- 0x100000000
03:0018| rsi 0x7ffd97d69400 --> 0x400630 (__libc_csu_init) <-- push r15
04:0020| 0x7ffd97d69408 --> 0x4004e0 (_start) <-- xor ebp, ebp
GREAT

```
[*] '/mnt/hgfs/games/src/pwngreat/target'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE
[x] Starting program './target'
[+] Starting program './target': Done
[*] running in new terminal: gdb-multiarch "/mnt/hgfs/games/src/pwngreat/target" 3582
[x] Waiting for debugger
[+] Waiting for debugger: Done

In [1]:
```

```
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
[-------------------------------------------------------------------REGISTERS-------------------------------------------------------------------]
 RAX  0xfffffffffffffe00
 RBX  0x0
 RCX  0x7fab6e6eb9b0 (__read_nocancel+7) <-- cmp    rax, -0xfff
 RDX  0xc8
 RDI  0x0
 RSI  0x7ffd97d69400 --> 0x400630 (__libc_csu_init) <-- push   r15
 R8   0x4006a0 (__libc_csu_fini) <-- ret
 R9   0x7fab6e9ce8e0 (_dl_fini) <-- push   rbp
 R10  0x37b
 R11  0x246
 R12  0x4004e0 (_start) <-- xor    ebp, ebp
 R13  0x7ffd97d69500 <-- 0x1
 R14  0x0
 R15  0x0
 RBP  0x7ffd97d69420 --> 0x400630 (__libc_csu_init) <-- push   r15
 RSP  0x7ffd97d693e8 --> 0x40060a (main+52) <-- mov    edi, 0x4006b4
 RIP  0x7fab6e6eb9b0 (__read_nocancel+7) <-- cmp    rax, -0xfff
[---------------------------------------------------------------------CODE--------------------------------------------------------------------]
 => 0x7fab6e6eb9b0 <__read_nocancel+7>     cmp    rax, -0xfff
    0x7fab6e6eb9b6 <__read_nocancel+13>    jae    read+73                     <0x7fab6e6eb9e9>
...
    0x7fab6e6eb9e9 <read+73>               mov    rcx, qword ptr [rip + 0x2cc488]
    0x7fab6e6eb9f0 <read+80>               neg    eax
    0x7fab6e6eb9f2 <read+82>               mov    dword ptr fs:[rcx], eax
    0x7fab6e6eb9f5 <read+85>               or     rax, 0xffffffffffffffff
    0x7fab6e6eb9f9 <read+89>               ret

    0x7fab6e6eb9fa                         nop    word ptr [rax + rax]
    0x7fab6e6eba00 <write>                 cmp    dword ptr [rip + 0x2d1d99], 0 <0x7fab6e9bd7a0>
    0x7fab6e6eba07 <write+7>               jne    write+25                    <0x7fab6e6eba19>
...
    0x7fab6e6eba19 <write+25>              sub    rsp, 8
[---------------------------------------------------------------------CODE--------------------------------------------------------------------]
79      in ../sysdeps/unix/syscall-template.S
[-------------------------------------------------------------------STACK-------------------------------------------------------------------]
00:0000| rsp  0x7ffd97d693e8 --> 0x40060a (main+52) <-- mov    edi, 0x4006b4
01:0008|      0x7ffd97d693f0 --> 0x7ffd97d69508 --> 0x7ffd97d6a814 <-- './target'
02:0010|      0x7ffd97d693f8 <-- 0x100000000
03:0018| rsi  0x7ffd97d69400 --> 0x400630 (__libc_csu_init) <-- push   r15
04:0020|      0x7ffd97d69408 --> 0x4004e0 (_start) <-- xor    ebp, ebp
05:0028|      0x7ffd97d69410 --> 0x7ffd97d69500 <-- 0x1
06:0030|      0x7ffd97d69418 <-- 0xdebc7841cec4bb00
07:0038| rbp  0x7ffd97d69420 --> 0x400630 (__libc_csu_init) <-- push   r15
[-----------------------------------------------------------------BACKTRACE-----------------------------------------------------------------]
>  f 0     7fab6e6eb9b0 __read_nocancel+7
   f 1          40060a main+52
   f 2     7fab6e615830 __libc_start_main+240
pwndbg>
```


/games/src/pwngreat$ ipython -i sploit.py
nes/src/pwngreat/target'
64-64-little
tial RELRO
ary found
enabled
PIE
ram 'target'
ram 'target': Done
termina gdb multia "/hgfs/ga es/sr /wngr
ebugg
ebugg Done
shell

```
evan@xub:/mnt/hgfs/games/src/pwngreat$ ipython -i sploit.py
[*] '/mnt/hgfs/games/src/pwngreat/target'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE
[x] Starting program './target'
[+] Starting program './target': Done
[*] running in new terminal: gdb-multiarch "/mnt/hgfs/games/src/pwngreat/target" 70937
[x] Waiting for debugger
[+] Waiting for debugger: Done

In [1]:
```


f31f5d4f9b0 (__read_nocancel+7) <-- cmp rax, -0x
8
ffefbcf59e0 --> 0x400630 (__libc_csu_init) <-- push
006a0 (__libc_csu_fini) <-- ret
7b
46
004e0 (_start) <-- xor ebp, ebp
ffefbcf5ae0 <-- 0x1
registers

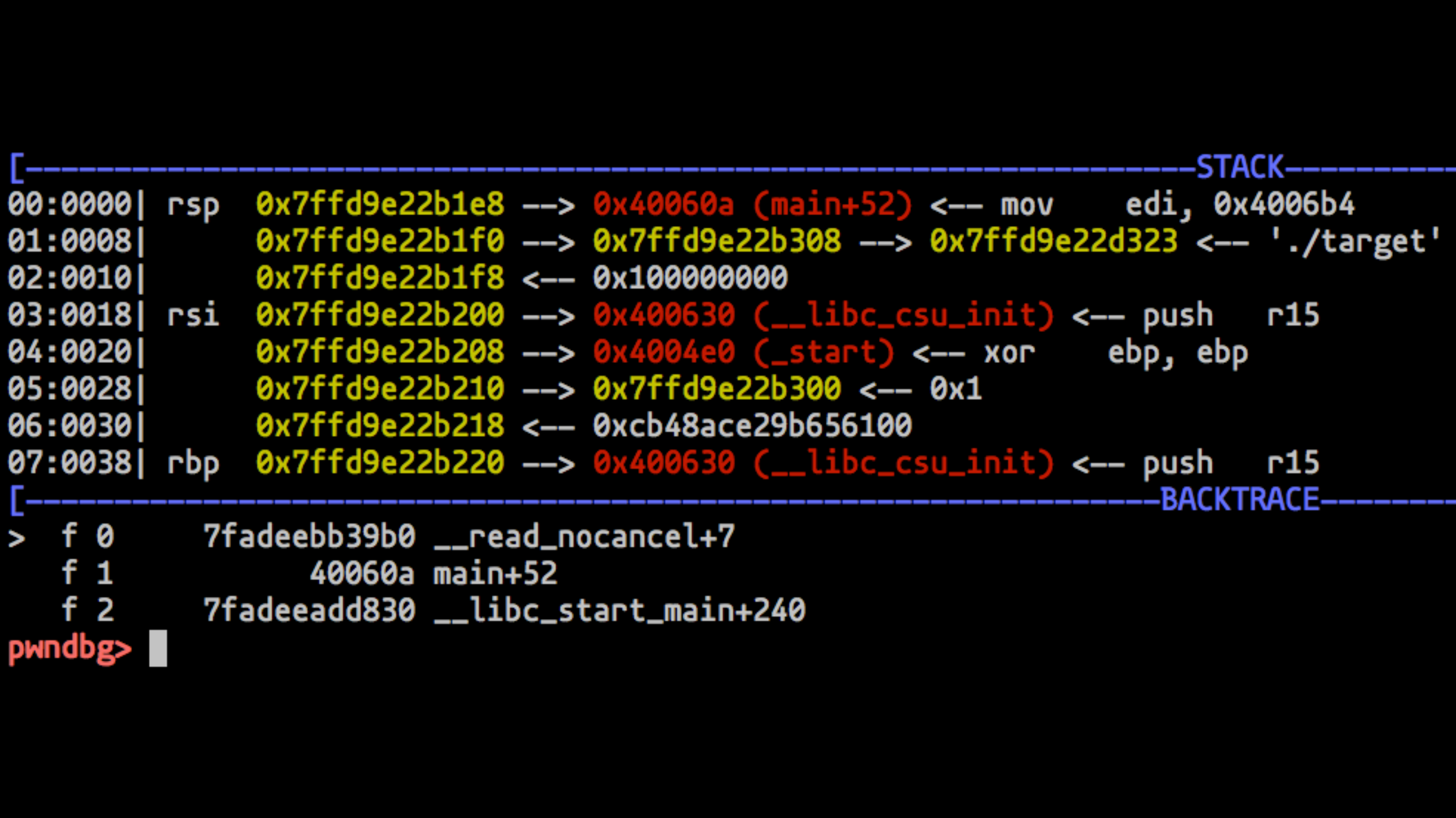
```
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
[-------------------------------------------------------------------
 RAX  0xfffffffffffffe00
 RBX  0x0
 RCX  0x7f31f5d4f9b0 (__read_nocancel+7) <-- cmp    rax, -0xfff
 RDX  0xc8
 RDI  0x0
 RSI  0x7ffefbcf59e0 --> 0x400630 (__libc_csu_init) <-- push   r15
 R8   0x4006a0 (__libc_csu_fini) <-- ret
 R9   0x7f31f60328e0 (_dl_fini) <-- push   rbp
 R10  0x37b
 R11  0x246
 R12  0x4004e0 (_start) <-- xor    ebp, ebp
 R13  0x7ffefbcf5ae0 <-- 0x1
 R14  0x0
 R15  0x0
 RBP  0x7ffefbcf5a00 --> 0x400630 (__libc_csu_init) <-- push   r15
 RSP  0x7ffefbcf59c8 --> 0x40060a (main+52) <-- mov    edi, 0x4006b4
 RIP  0x7f31f5d4f9b0 (__read_nocancel+7) <-- cmp    rax, -0xfff
```


ead_nocancel+7> cmp rax, -0xfff
ead_nocancel+13> jae read+73
d+73> mov rcx, q d ptr [rip + 0x
d+80> neg eax
d+82> mov dword t fs:[rcx] eax
d+85 a f ffff ff
d+8 r
m rd p [ra rax]
te> o rd [r 0x2d1d9
te+7 w
te+25> sub rsp, 8
code

```
[----------------------------------------------------------------------------------
 => 0x7f31f5d4f9b0 <__read_nocancel+7>     cmp    rax, -0xfff
    0x7f31f5d4f9b6 <__read_nocancel+13>    jae    read+73                    <0x7f31f5d4f9e9>
...
    0x7f31f5d4f9e9 <read+73>               mov    rcx, qword ptr [rip + 0x2cc488]
    0x7f31f5d4f9f0 <read+80>               neg    eax
    0x7f31f5d4f9f2 <read+82>               mov    dword ptr fs:[rcx], eax
    0x7f31f5d4f9f5 <read+85>               or     rax, 0xffffffffffffffff
    0x7f31f5d4f9f9 <read+89>               ret

    0x7f31f5d4f9fa                         nop    word ptr [rax + rax]
    0x7f31f5d4fa00 <write>                 cmp    dword ptr [rip + 0x2d1d99], 0 <0x7f31f60217a0>
    0x7f31f5d4fa07 <write+7>               jne    write+25                   <0x7f31f5d4fa19>
...
    0x7f31f5d4fa19 <write+25>              sub    rsp, 8
[----------------------------------------------------------------------------------
```

```
[-------------------------------------------------------CODE-------------------------------------------------------]
   0x8049404 <menu+629>                          lea    esp, [ebp - 0xc]
   0x8049407 <menu+632>                          pop    ebx
   0x8049408 <menu+633>                          pop    esi
   0x8049409 <menu+634>                          pop    edi
   0x804940a <menu+635>                          pop    ebp
 => 0x804940b <menu+636>                         ret                                <0x806f510; __lll_unlock_wake_private+32>
...
   0x806f510 <__lll_unlock_wake_private+32>      pop    edx
   0x806f511 <__lll_unlock_wake_private+33>      pop    ecx
   0x806f512 <__lll_unlock_wake_private+34>      pop    ebx
   0x806f513 <__lll_unlock_wake_private+35>      ret
...
   0x80bb926 <_Unwind_GetDataRelBase+6>         pop    eax
[-------------------------------------------------------STACK------------------------------------------------------]
```


stack


[---STACK---------
00:0000| rsp 0x7ffd9e22b1e8 --> 0x40060a (main+52) <-- mov   edi, 0x4006b4
01:0008| 0x7ffd9e22b1f0 --> 0x7ffd9e22b308 --> 0x7ffd9e22d323 <-- './target'
02:0010| 0x7ffd9e22b1f8 <-- 0x100000000
03:0018| rsi 0x7ffd9e22b200 --> 0x400630 (__libc_csu_init) <-- push   r15
04:0020| 0x7ffd9e22b208 --> 0x4004e0 (_start) <-- xor   ebp, ebp
05:0028| 0x7ffd9e22b210 --> 0x7ffd9e22b300 <-- 0x1
06:0030| 0x7ffd9e22b218 <-- 0xcb48ace29b656100
07:0038| rbp 0x7ffd9e22b220 --> 0x400630 (__libc_csu_init) <-- push   r15
[---BACKTRACE-------
> f 0   7fadeebb39b0 __read_nocancel+7
   f 1   40060a main+52
   f 2   7fadeeadd830 __libc_start_main+240
pwndbg>

Comes with LOTS of commands

pwndbg command to list of them all


0x602000 rw-p 1000 1000 /mnt/hgfs/games/
0x7f31f5e19000 r-xp 1c0000 0 /lib/x86_64-linu
0x7f31f6018000 ---p 1ff000 1c0000 /lib/x86_64-linu
0x7f31f601c000 r--p 4000 1bf000 /lib/x86_64-linu
0x7f31f601e000 rw-p 2000 1c3000 /lib/x86_64-linu
0x7f31f6022000 rw-p 4000 0
0x7f31f624a000 rw-p 1000 0
0x7ffefbcf7000 rw-p 21000 0 [stac
0x7ffefbd6f000 r--p 2000 0 [vvar]
0x7ffefbd71000 r-xp 2000 0 [vdso]
vmmap

```
pwndbg> vmmap
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
          0x400000           0x401000 r-xp     1000 0      /mnt/hgfs/games/src/pwngreat/target
          0x600000           0x601000 r--p     1000 0      /mnt/hgfs/games/src/pwngreat/target
          0x601000           0x602000 rw-p     1000 1000   /mnt/hgfs/games/src/pwngreat/target
    0x7f31f5c59000     0x7f31f5e19000 r-xp   1c0000 0      /lib/x86_64-linux-gnu/libc-2.23.so
    0x7f31f5e19000     0x7f31f6018000 ---p   1ff000 1c0000 /lib/x86_64-linux-gnu/libc-2.23.so
    0x7f31f6018000     0x7f31f601c000 r--p     4000 1bf000 /lib/x86_64-linux-gnu/libc-2.23.so
    0x7f31f601c000     0x7f31f601e000 rw-p     2000 1c3000 /lib/x86_64-linux-gnu/libc-2.23.so
    0x7f31f601e000     0x7f31f6022000 rw-p     4000 0
    0x7f31f6022000     0x7f31f6048000 r-xp    26000 0      /lib/x86_64-linux-gnu/ld-2.23.so
    0x7f31f6227000     0x7f31f622a000 rw-p     3000 0
    0x7f31f6245000     0x7f31f6247000 rw-p     2000 0
    0x7f31f6247000     0x7f31f6248000 r--p     1000 25000  /lib/x86_64-linux-gnu/ld-2.23.so
    0x7f31f6248000     0x7f31f6249000 rw-p     1000 26000  /lib/x86_64-linux-gnu/ld-2.23.so
    0x7f31f6249000     0x7f31f624a000 rw-p     1000 0
    0x7ffefbcd6000     0x7ffefbcf7000 rw-p    21000 0      [stack]
    0x7ffefbd6d000     0x7ffefbd6f000 r--p     2000 0      [vvar]
    0x7ffefbd6f000     0x7ffefbd71000 r-xp     2000 0      [vdso]
0xffffffffff600000 0xffffffffff601000 r-xp     1000 0      [vsyscall]
pwndbg> vmmap 0x7f31f5de558b
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
    0x7f31f5c59000     0x7f31f5e19000 r-xp   1c0000 0      /lib/x86_64-linux-gnu/libc-2.23.so
```


0x68732f6e69622f /* '/bin/sh' */
bytes of target memory at
search

```
pwndbg> search /bin/sh
libc-2.23.so    0x7f31f5de558b 0x68732f6e69622f /* '/bin/sh' */
warning: Unable to access 16000 bytes of target memory at 0x7f31f5e20d06, halting search.
pwndbg> hexdump 0x7f31f5de558b
+0000 0x7f31f5de558b  2f 62 69 6e  2f 73 68 00  65 78 69 74  20 30 00 63  |/bin|/sh.|exit| 0.c|
+0010 0x7f31f5de559b  61 6e 6f 6e  69 63 61 6c  69 7a 65 2e  63 00 4d 53  |anon|ical|ize.|c.MS|
+0020 0x7f31f5de55ab  47 56 45 52  42 00 53 45  56 5f 4c 45  56 45 4c 00  |GVER|B.SE|V_LE|VEL.|
+0030 0x7f31f5de55bb  54 4f 20 46  49 58 3a 20  00 20 20 00  25 73 25 73  |TO F|IX: |.  .|%s%s|
+0040 0x7f31f5de55cb
```

```
pwndbg> search -h
usage: search [-h] [-t {byte,short,dword,qword,pointer,string,bytes}] [-1] [-2] [-4] [-8] [-p] [-x] [-s] [-e] [-w]
              value [mapping]

Search memory for byte sequences, strings, pointers, and integer values

positional arguments:
  value                 Value to search for
  mapping               Mapping to search [e.g. libc]

optional arguments:
  -h, --help            show this help message and exit
  -t {byte,short,dword,qword,pointer,string,bytes}, --type {byte,short,dword,qword,pointer,string,bytes}
                        Size of search target (default: bytes)
  -1, --byte            Search for a 1-byte integer
  -2, --word            Search for a 2-byte integer
  -4, --dword           Search for a 4-byte integer
  -8, --qword           Search for an 8-byte integer
  -p, --pointer         Search for a pointer-width integer
  -x, --hex             Target is a hex-encoded (for bytes/strings) (default: False)
  -s, --string          Target is a raw string (default: False)
  -e, --executable      Search executable segments only (default: False)
  -w, --writable        Search writable segments only (default: False)
```

what was in sploit.py ???

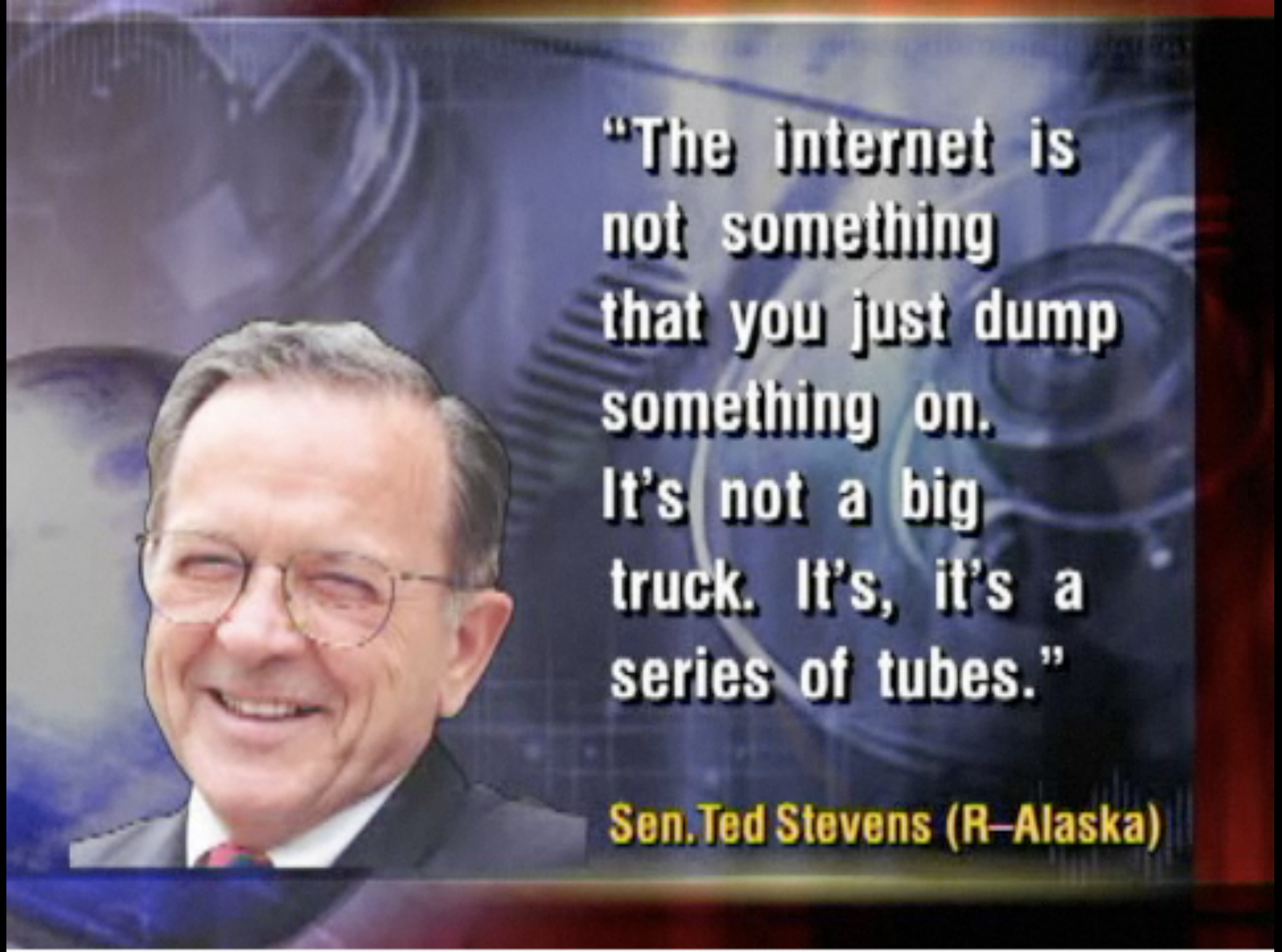
```
$ cat sploit.py
1 from pwn import *

2 elf = ELF('./target')
3 r = process('./target')
4 gdb.attach(r)
```


mov QWORD PTR [rbp-0x30],rsi
ELF

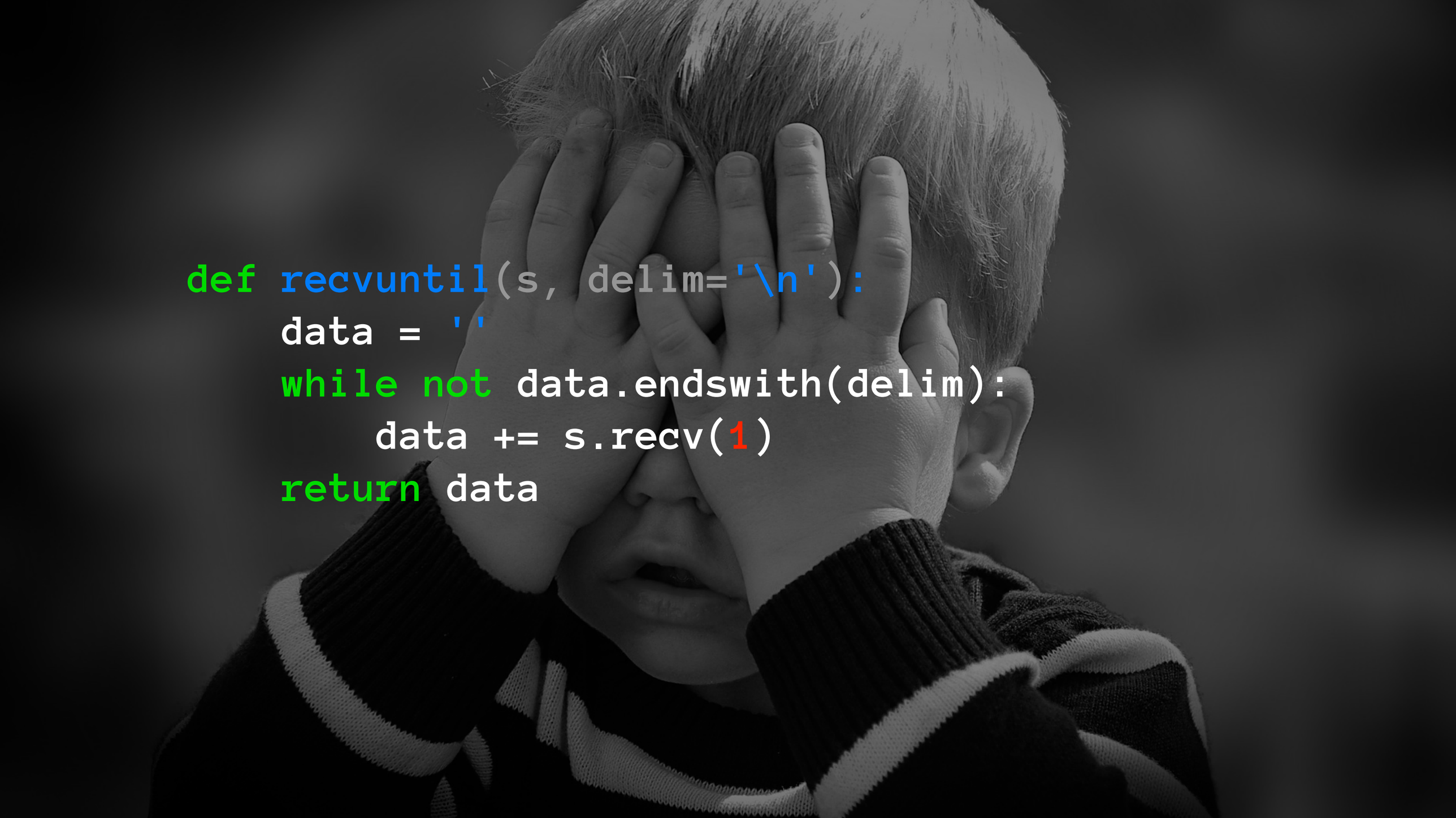
```
In [9]: print elf.disasm(elf.symbols['main'], 15)
  4005d6:       55                      push   rbp
  4005d7:       48 89 e5                mov    rbp,rsp
  4005da:       48 83 ec 30             sub    rsp,0x30
  4005de:       89 7d dc                mov    DWORD PTR [rbp-0x24],edi
  4005e1:       48 89 75 d0             mov    QWORD PTR [rbp-0x30],rsi

In [10]: elf.disasm?
Signature: elf.disasm(address, n_bytes)
Docstring:
Returns a string of disassembled instructions at
the specified virtual memory address
File:      ~/.virtualenvs/pwn/local/lib/python2.7/site-packages/pwnlib/elf/elf.py
Type:      instancemethod

In [11]: elf.
```

```
elf.address           elf.asm           elf.canary          elf.dynamic_by_tag
elf.address_offsets   elf.bits          elf.checksec        elf.dynamic_string
elf.arch              elf.bss           elf.data            elf.e_ident_raw
elf.asan              elf.buildid       elf.disasm          elf.elfclass
elf.aslr              elf.bytes         elf.dwarf           elf.elftype
```

ython2.7/site-packages/pwnlib/tubes/tube.py


"The internet is not something that you just dump something on. It's not a big truck. It's, it's a series of tubes."
Sen. Ted Stevens (R–Alaska)

```python
r = process('./target')  # spawn a process and wrap stdin/stdout/stderr

r = remote('host', 4141)             # connect to the remote host
r = remote('host', 4141, typ='udp') # connect on udp

l = listen(4141)                     # listen for incoming connections
r = l.wait_for_connection()          # accept

# connect to remote host
s = ssh(host='host', user='user', password='pass')
r = s.process('./target') # spawn a process
```



```
def recvuntil(s, delim='\n'):
    data = ''
    while not data.endswith(delim):
        data += s.recv(1)
    return data
```

```
In [19]: r.recvuntil?
Signature: r.recvuntil(delims, drop=False, timeout=pwnlib.timeout.Timeout.default)
Docstring:
recvuntil(delims, timeout = default) -> str

Receive data until one of `delims` is encountered.

If the request is not satisfied before ``timeout`` seconds pass,
all data is buffered and an empty string (``''``) is returned.

arguments:
    delims(str,tuple): String of delimiters characters, or list of delimiter strings.
    drop(bool): Drop the ending.  If ``True`` it is removed from the end of the return value.
```

```
In [4]: r.recvline?
Signature: r.recvline(keepends=True, timeout=pwnlib.timeout.Timeout.default)
Docstring:
recvline(keepends = True) -> str

Receive a single line from the tube.

A "line" is any sequence of bytes terminated by the byte sequence
set in :attr:`newline`, which defaults to ``'\n'``.

If the request is not satisfied before ``timeout`` seconds pass,
all data is buffered and an empty string (``''``) is returned.

Arguments:
    keepends(bool): Keep the line ending (``True``).
    timeout(int): Timeout
```

```
In [11]: r.sendline('hello')

In [12]: r.sendline?
Signature: r.sendline(line='')
Docstring:
sendline(data)

Shorthand for ``t.send(data + t.newline)``.

Examples:

    >>> def p(x): print repr(x)
    >>> t = tube()
    >>> t.send_raw = p
    >>> t.sendline('hello')
    'hello\n'
    >>> t.newline = '\r\n'
    >>> t.sendline('hello')
    'hello\r\n'
File:      ~/.virtualenvs/pwn/local/lib/python2.7/site-packages/pwnlib/tubes/tube.py
Type:      instancemethod

In [13]: r.
         r.addHandler          r.can_recv            r.communicate         r.connected_directions r.critical
         r.alarm               r.can_recv_raw        r.connect_both        r.connected_raw        r.cwd
         r.argv                r.clean               r.connect_input       r.corefile             r.debug
         r.aslr                r.clean_and_log       r.connect_output      r.countdown            r.default
         r.buffer              r.close               r.connected           r.countdown_active     r.display
```

```
In [16]: r.interactive?
Signature: r.interactive(prompt='\x1b[1m\x1b[31m$\x1b[m ')
Docstring:
interactive(prompt = pwnlib.term.text.bold_red('$') + ' ')

Does simultaneous reading and writing to the tube. In principle this just
connects the tube to standard in and standard out, but in practice this
is much more usable, since we are using :mod:`pwnlib.term` to print a
floating prompt.

Thus it only works in while in :data:`pwnlib.term.term_mode`.
File:      ~/.virtualenvs/pwn/local/lib/python2.7/site-packages/pwnlib/tubes/tube.py
Type:      instancemethod
```


rm: <pwnlib.tubes.ssh.ssh object at 0x7f2efd7ea450>
~/.virtualenvs/pwn/local/lib/python2.7/site-packages/pwnlib/tube
s(self, attr)
SSH

```
In [11]: s = ssh(host='localhost', user='evan')
[x] Connecting to localhost on port 22
[+] Connecting to localhost on port 22: Done

In [12]: s.pwd()
[x] Opening new channel: 'pwd'
[+] Opening new channel: 'pwd': Done
[x] Receiving all data
[x] Receiving all data: 0B
[x] Receiving all data: 11B
[+] Receiving all data: Done (11B)
[*] Closed SSH channel with localhost
Out[12]: '/home/evan'

In [13]: s.uname('-a')
[x] Opening new channel: 'uname -a'
[+] Opening new channel: 'uname -a': Done
[x] Receiving all data
[x] Receiving all data: 0B
[x] Receiving all data: 101B
[+] Receiving all data: Done (101B)
[*] Closed SSH channel with localhost
Out[13]: 'Linux xub 4.4.0-38-generic #57-Ubuntu SMP Tue Sep 6 15:42:33 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux'
```

```
In [22]: s.download_data('/tmp/download')
[x] Downloading '/tmp/download'
[x] Downloading '/tmp/download': 12B/12B
[+] Downloading '/tmp/download': Done
Out[22]: 'hello there\n'

In [23]: s.upload_data('hello back', '/tmp/upload')
```

```
In [32]: r = s.remote('www.google.com', 80)
[x] Connecting to www.google.com:80 via SSH to localhost
[+] Connecting to www.google.com:80 via SSH to localhost: Done

In [33]: r.sendline('GET / HTTP/1.1\r\n\r')

In [34]: r.recv(20)
Out[34]: 'HTTP/1.1 200 OK\r\nDat'
```

```
In [35]: a = s.listen(4141)
[x] Waiting on port 4141 via SSH to localhost

In [36]: a.recv()
[+] Waiting on port 4141 via SSH to localhost: Got connection from 127.0.0.1:58934
Out[36]: 'hello\n'
```

```
In [37]: r = s.process('/mnt/hgfs/games/src/pwngreat/target')
[x] Opening new channel: execve('/mnt/hgfs/games/src/pwngreat/target', ['/mnt/hgfs/games/src/pwngreat/target'], os.environ)
[+] Opening new channel: execve('/mnt/hgfs/games/src/pwngreat/target', ['/mnt/hgfs/games/src/pwngreat/target'], os.environ): Don
e

In [38]: r.
```

```
r.addHandler          r.clean              r.connect_input       r.countdown          r.default
r.argv                r.clean_and_log      r.connect_output      r.countdown_active   r.env
r.buffer              r.close              r.connected           r.critical           r.error
r.can_recv            r.closed             r.connected_directions r.cwd               r.exception
r.can_recv_raw        r.connect_both       r.connected_raw       r.debug              r.executable
```

```
$ cat sploit.py
1 from pwn import *

2 elf = ELF('./target')
3 #r = process('./target')
4 #gdb.attach(r)
5 r = remote('HOST', 1234)
```


2.7/site-packages/pwnlib/shellcraft/__init_
ellcode
y operating system
shellcraft

```
In [12]: shellcraft?
Type:        module
String form: <module 'pwnlib.shellcraft' from '/home/evan/.virtualenvs/pwn/local/lib/python2.7/site-packages/pwnlib/shellcraft/__init__.pyc
'>
File:        ~/.virtualenvs/pwn/local/lib/python2.7/site-packages/pwnlib/shellcraft/__init__.py
Docstring:
The shellcode module.

This module contains functions for generating shellcode.

It is organized first by architecture and then by operating system.

In [13]: print shellcraft.arm.mov('r0', 10)
    mov  r0, #0xa


In [14]: print shellcraft.mips.mov('$r0', 10)
    li $t9, ~0xa
    not $r0, $t9
```

```
In [25]: print shellcraft.cat('flag.txt')
    /* push 'flag.txt\x00' */
    push 1
    dec byte ptr [esp]
    push 0x7478742e
    push 0x67616c66
    /* call open('esp', 0, 'O_RDONLY') */
    push (SYS_open) /* 5 */
    pop eax
    mov ebx, esp
    xor ecx, ecx
    cdq /* edx=0 */
    int 0x80
    /* call sendfile(1, 'eax', 0, 0x7fffffff) */
    mov ecx, eax
    xor eax, eax
    mov al, 0xbb
    push 1
    pop ebx
    push 0x7fffffff
    pop esi
    cdq /* edx=0 */
    int 0x80
```

```
In [26]: context.os='linux'

In [27]: context.arch='arm'

In [28]: print shellcraft.cat('flag.txt')
    /* push 'flag.txt\x00AAA' */
    movw r7, #0x41414100 & 0xffff
    movt r7, #0x41414100 >> 16
    push {r7}
    movw r7, #0x7478742e & 0xffff
    movt r7, #0x7478742e >> 16
    push {r7}
    movw r7, #0x67616c66 & 0xffff
    movt r7, #0x67616c66 >> 16
    push {r7}
    /* call open('sp', 0, 0) */
    mov  r0, sp
    eor  r1, r1 /* 0 (#0) */
    eor  r2, r2 /* 0 (#0) */
    mov  r7, #(SYS_open) /* 5 */
    svc  0
    /* call sendfile(1, 'r0', 0, 2147483647) */
    mov  r1, r0
    mov  r0, #1
    eor  r2, r2 /* 0 (#0) */
    mvn  r3, #(0x7fffffff ^ (-1))
    mov  r7, #(SYS_sendfile) /* 0xbb */
    svc  0
```

```
In [30]: asm(shellcraft.cat('flag.txt'))
Out[30]: '\x00q\x04\xe3AqD\xe3\x04p-\xe5.t\x07\xe3xtG\xe3\x04p-\xe5f|\x06\xe3awF\xe3\x04p-\xe5\r\x00\xa0\x
e1\x01\x10!\xe0\x02 "\xe0\x05p\xa0\xe3\x00\x00\x00\xef\x00\x10\xa0\xe1\x01\x00\xa0\xe3\x02 "\xe0\x021\xe0\
xe3\xbbp\xa0\xe3\x00\x00\x00\xef'
```

```
In [32]: asm('mov r0, #(SYS_open)')
Out[32]: '\x05\x00\xa0\xe3'

In [33]: asm('mov r0, #(SYS_read)')
Out[33]: '\x03\x00\xa0\xe3'

In [34]: asm('mov r0, #(SYS_dup2)')
Out[34]: '?\x00\xa0\xe3'
```

```
In [7]: disasm('\x05\x00\xa0\xe3')
Out[7]: '   0:   e3a00005        mov     r0, #5'
```


[x] Starting local process '/usr/bin/qemu-arm-static'
[+] Starting local process '/usr/bin/qemu-arm-static': Done
[*] running in new terminal: gdb-multiarch -q -x "/tmp/pwnWDbAdI.gdb"
Out[5]: <pwnlib.tubes.process.process at 0x7fdb0111afd0>
LEGEND: STACK | HEAP | CODE | DATA | RWX | ROD
REGISTERS
R0 0x0
R1 0xf6fff653 <-- '/tmp/pwn-asm-sk...'
R2 0x0
R3 0x0
R4 0x0
R5 0x0
R6 0x0
R7 0x0
R8 0x0
R9 0x0
R10 0x10000000
R11 0x0
R12 0x0
SP 0xf6fff530
PC 0x10000000
#5 /* 0xe3a00005 */
CODE
=> 0x10000000 mov r0, #5
0x10000004 andeq r1, r0, r1, asr #24
gdb

```
In [5]: gdb.debug_assembly('mov r0, #(SYS_open)')
[x] Starting local process '/usr/bin/qemu-arm-static'
[+] Starting local process '/usr/bin/qemu-arm-static': Done
[*] running in new terminal: gdb-multiarch -q  -x "/tmp/pwnWDbAdI.gdb"
Out[5]: <pwnlib.tubes.process.process at 0x7fdb0111afd0>

LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
[----------------------------------------------------------------REGISTERS---------]
 R0   0x0
 R1   0xf6fff653 <-- '/tmp/pwn-asm-sk...'
 R2   0x0
 R3   0x0
 R4   0x0
 R5   0x0
 R6   0x0
 R7   0x0
 R8   0x0
 R9   0x0
 R10  0x10000000 <-- mov    r0, #5 /* 0xe3a00005 */
 R11  0x0
 R12  0x0
 SP   0xf6fff530 <-- 0x1
 PC   0x10000000 <-- mov    r0, #5 /* 0xe3a00005 */
[-----------------------------------------------------------------CODE-----------]
 => 0x10000000    mov    r0, #5
    0x10000004    andeq  r1, r0, r1, asr #24
```


THE
OTHER
GUYS

```
In [16]: ROP?
Init signature: ROP(self, elfs, base=None, **kwargs)
Docstring:
Class which simplifies the generation of ROP-chains.

Example:

.. code-block:: python

   elf = ELF('ropasaurusrex')
   rop = ROP(elf)
   rop.read(0, elf.bss(0x80))
   rop.dump()
   # ['0x0000:        0x80482fc (read)',
   #  '0x0004:       0xdeadbeef',
   #  '0x0008:              0x0',
   #  '0x000c:        0x80496a8']
   str(rop)
   # '\xfc\x82\x04\x08\xef\xbe\xad\xde\x00\x00\x00\x00\xa8\x96\x04\x08'
```

```
In [38]: print hexdump('\x01\x02\x031234567890abcdef')
00000000  01 02 03 31  32 33 34 35  36 37 38 39  30 61 62 63  |···1|2345|6789|0abc|
00000010  64 65 66                                           |def|
00000013
```

```
In [41]: p32(1)
Out[41]: '\x01\x00\x00\x00'

In [42]: u32('\x01\x00\x00\x00')
Out[42]: 1

In [43]: p64(1)
Out[43]: '\x01\x00\x00\x00\x00\x00\x00\x00'

In [44]: u64('\x01\x00\x00\x00\x00\x00\x00\x00')
Out[44]: 1
```

```
In [5]: context.endianness = 'big'

In [6]: p32(1)
Out[6]: '\x00\x00\x00\x01'

In [7]: u32('\x00\x00\x00\x01')
Out[7]: 1

In [8]: context.endianness = 'little'

In [9]: u32('\x00\x00\x00\x01')
Out[9]: 16777216

In [10]: u32('\x00\x00\x00\x01', endianness='big')
Out[10]: 1
```

```
In [7]: cyclic(100)
Out[7]: 'aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaayaaa'

In [8]: cyclic_find('aaag')
Out[8]: 21
```

```
In [4]: errno.errorcode?
Type:        dict
String form: {1: 'EPERM', 2: 'ENOENT', 3: 'ESRCH', 4: 'EINTR', 5: 'EIO', 6: 'ENXIO', 7: 'E2BIG', 8: 'ENOEXEC', <...> , 117: 'EUCLEAN', 118: 'ENOTNAM',
119: 'ENAVAIL', 120: 'EISNAM', 121: 'EREMOTEIO', 122: 'EDQUOT'}
Length:      120
Docstring:
dict() -> new empty dictionary
dict(mapping) -> new dictionary initialized from a mapping object's
    (key, value) pairs
dict(iterable) -> new dictionary initialized as if via:
    d = {}
    for k, v in iterable:
        d[k] = v
dict(**kwargs) -> new dictionary initialized with the name=value pairs
    in the keyword argument list.  For example:  dict(one=1, two=2)

In [5]:

In [5]: errno.errorcode[7]
Out[5]: 'E2BIG'

In [6]: errno.E2BIG
Out[6]: 7
```

```
In [2]: log.info('inform')
[*] inform

In [3]: log.warn('warning')
[!] warning

In [4]: log.error('OH NOES!')
[ERROR] OH NOES!
---------------------------------------------------------------------------
PwnlibException                           Traceback (most recent call last)
<ipython-input-4-cf3185232889> in <module>()
----> 1 log.error('OH NOES!')

/home/evan/.virtualenvs/pwn/local/lib/python2.7/site-packages/pwnlib/log.pyc in error(self, message, *args, **kwargs)
    412         """
    413         self._log(logging.ERROR, message, args, kwargs, 'error')
--> 414         raise PwnlibException(message % args)
    415
    416     def exception(self, message, *args, **kwargs):

PwnlibException: OH NOES!

In [5]: context.log_level = 'error'

In [6]: log.info('nope')

In [7]:
```


$ install

install pwntools

```
sudo apt-get update
sudo apt-get install -y \
    binutils \
    binutils-arm-linux-gnueabihf \ # whatever arch you need
    git \
    libffi-dev \
    libss-dev \
    python-pip \
sudo pip install --upgrade git+https://github.com/Gallopsled/pwntools.git
```

install pwndbg

```
git clone https://github.com/pwndbg/pwndbg
cd pwndbg
./setup.sh
```

install MISC

```
sudo apt-get install tmux
sudo apt-get install qemu-static
sudo pip install --upgrade ipython
```

MOAR

https://github.com/pwndbg/pwndbg/blob/master/FEATURES.md

https://docs.pwntools.com/en/stable/

https://github.com/Gallopsled/pwntools-write-ups

@syndrowm