GOT/PLT

Presented by Justin Zhu

What is the Global Offset Table (GOT)?

```
It's a table of offsets!



Offsets to what?
Dynamically linked libraries.



Like functions in libc.
```

~~What's PLT?~~

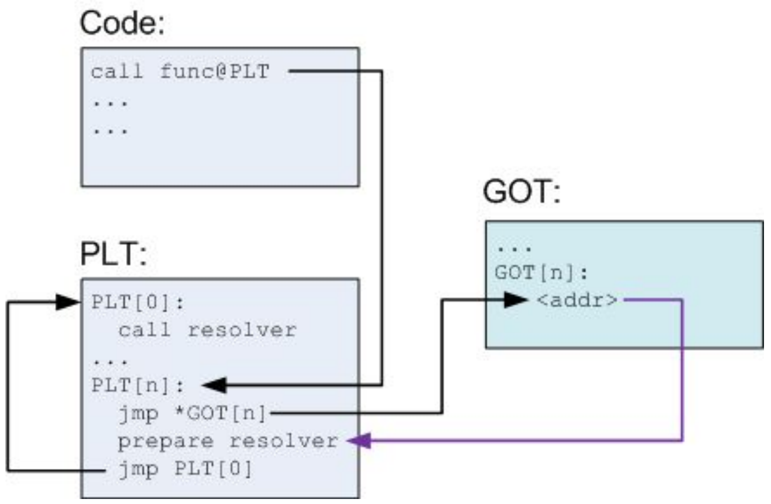
What's the Procedure Linkage Table?

It's like the interface the actual program uses.

It's what uses the GOT to give the program the function it wants.

Everything makes more sense with examples.

How does GOT/PLT Work?



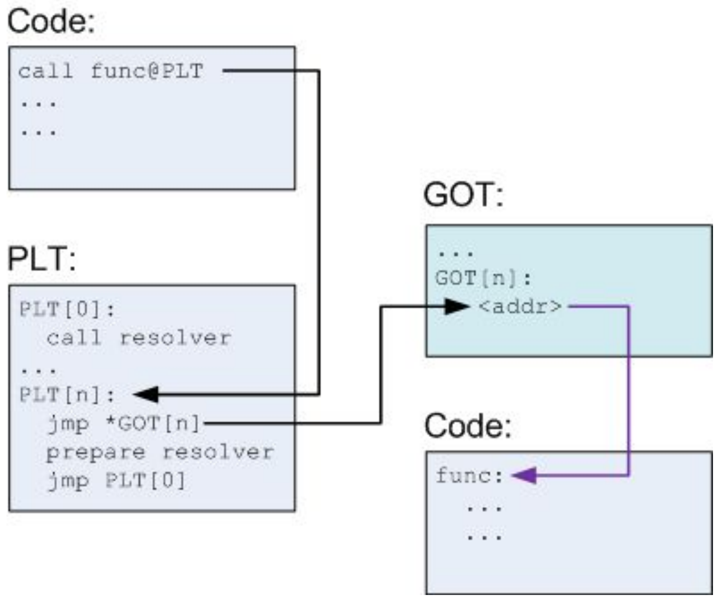
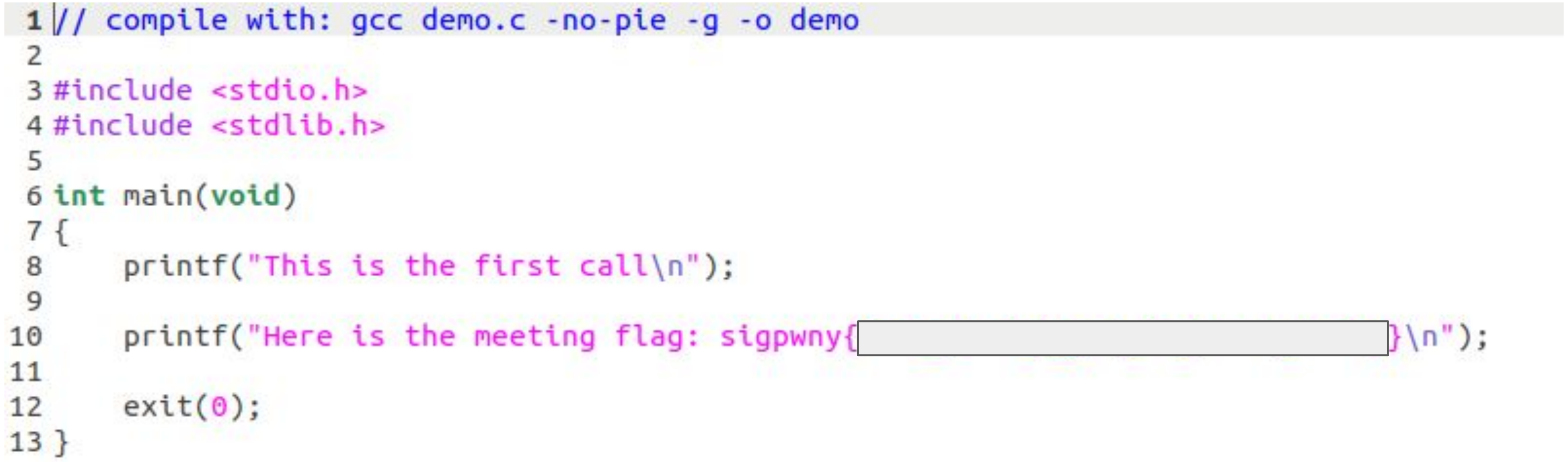


Diagram Credits:

https://nuc13us.wordpress.com/2015/12/25/hack-using-global-offset-table/

How does GOT/PLT Work? (demo)

demo.c

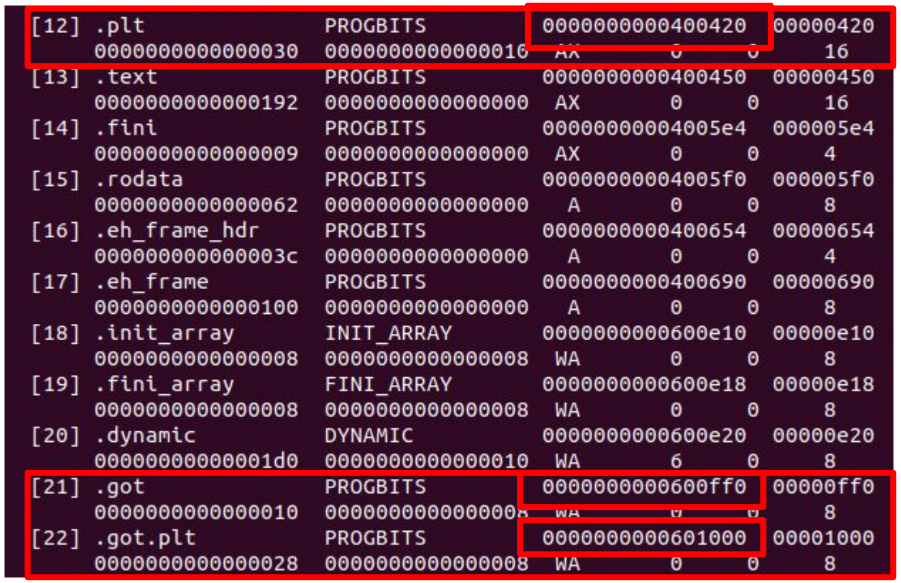


```c
// compile with: gcc demo.c -no-pie -g -o demo

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

int main(void)
{
    printf("This is the first call\n");

    printf("Here is the meeting flag: sigpwny{[redacted]}\n");

    exit(0);
}
```

How does GOT/PLT Work? (demo) (cont'd)

```
.plt - 0x00400420
.got - 0x00600ff0
.got.plt - 00x601000
```

```
readelf -S demo
```

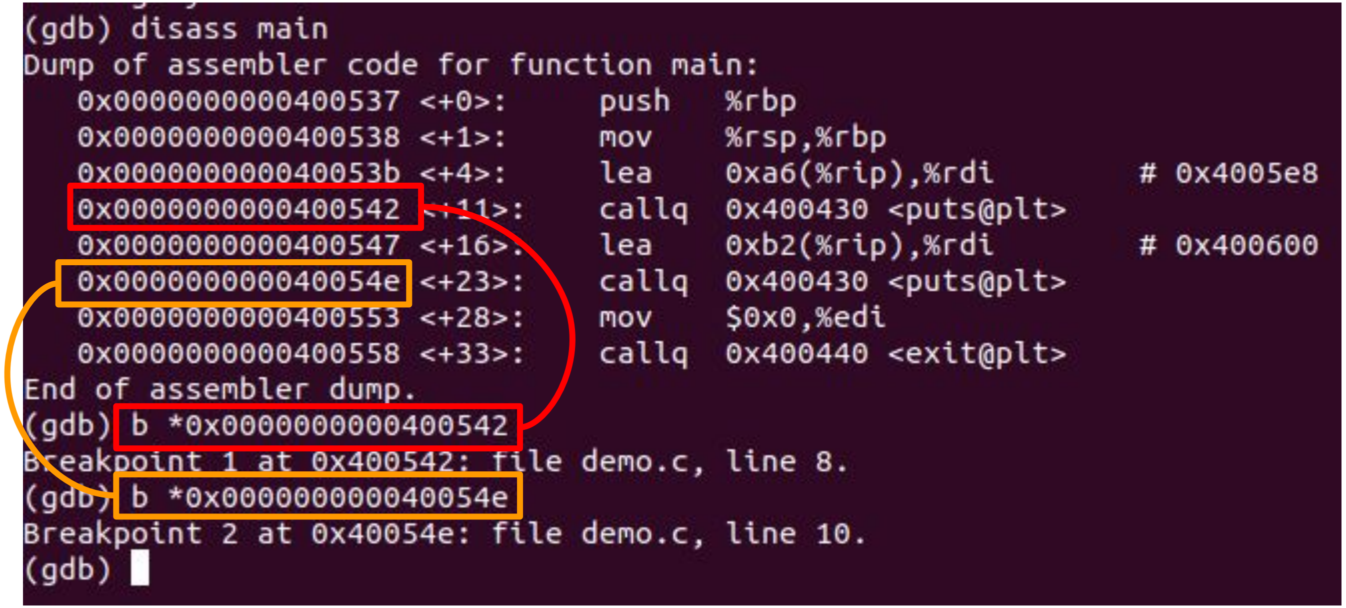


```
[12] .plt              PROGBITS          0000000000400420  00000420
     0000000000000030  0000000000000010  AX       0     0     16
[13] .text             PROGBITS          0000000000400450  00000450
     0000000000000192  0000000000000000  AX       0     0     16
[14] .fini             PROGBITS          00000000004005e4  000005e4
     0000000000000009  0000000000000000  AX       0     0     4
[15] .rodata           PROGBITS          00000000004005f0  000005f0
     0000000000000062  0000000000000000   A       0     0     8
[16] .eh_frame_hdr     PROGBITS          0000000000400654  00000654
     000000000000003c  0000000000000000   A       0     0     4
[17] .eh_frame         PROGBITS          0000000000400690  00000690
     0000000000000100  0000000000000000   A       0     0     8
[18] .init_array       INIT_ARRAY        0000000000600e10  00000e10
     0000000000000008  0000000000000008  WA       0     0     8
[19] .fini_array       FINI_ARRAY        0000000000600e18  00000e18
     0000000000000008  0000000000000008  WA       0     0     8
[20] .dynamic          DYNAMIC           0000000000600e20  00000e20
     00000000000001d0  0000000000000010  WA       6     0     8
[21] .got              PROGBITS          0000000000600ff0  00000ff0
     0000000000000010  0000000000000008  WA       0     0     8
[22] .got.plt          PROGBITS          0000000000601000  00001000
     0000000000000028  0000000000000008  WA       0     0     8
```

How does GOT/PLT Work? (demo) (cont'd)

```
.plt - 0x00400420
.got - 0x00600ff0
.got.plt - 00x601000
```

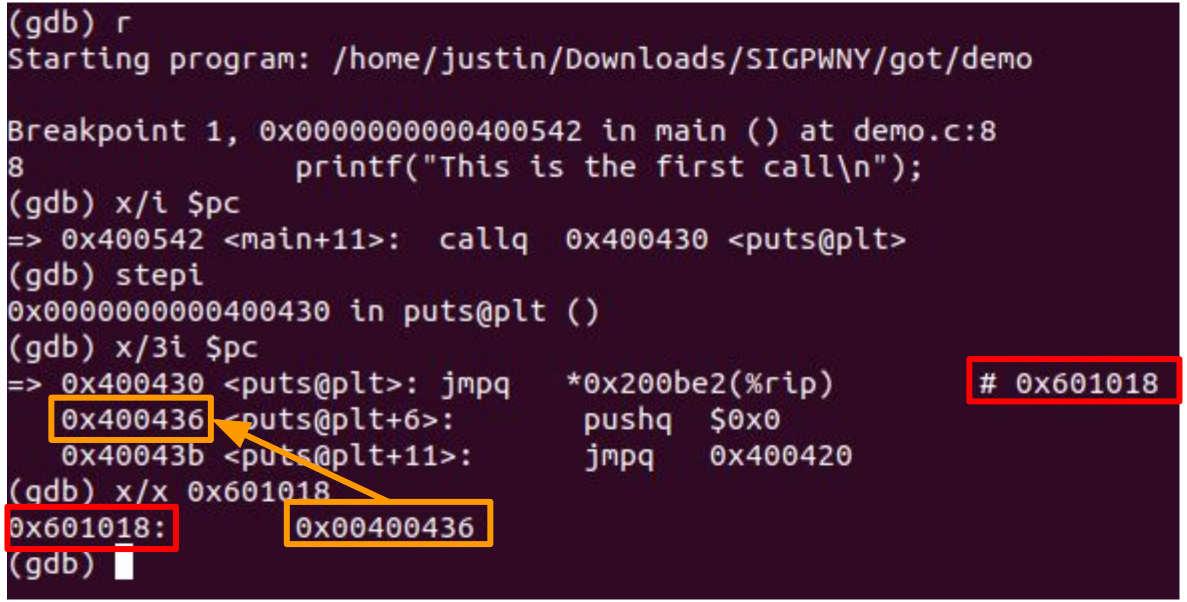
gdb demo



```
(gdb) disass main
Dump of assembler code for function main:
   0x0000000000400537 <+0>:     push   %rbp
   0x0000000000400538 <+1>:     mov    %rsp,%rbp
   0x000000000040053b <+4>:     lea    0xa6(%rip),%rdi        # 0x4005e8
   0x0000000000400542 <+11>:    callq  0x400430 <puts@plt>
   0x0000000000400547 <+16>:    lea    0xb2(%rip),%rdi        # 0x400600
   0x000000000040054e <+23>:    callq  0x400430 <puts@plt>
   0x0000000000400553 <+28>:    mov    $0x0,%edi
   0x0000000000400558 <+33>:    callq  0x400440 <exit@plt>
End of assembler dump.
(gdb) b *0x0000000000400542
Breakpoint 1 at 0x400542: file demo.c, line 8.
(gdb) b *0x000000000040054e
Breakpoint 2 at 0x40054e: file demo.c, line 10.
(gdb) 
```

How does GOT/PLT Work? (demo) (cont'd)

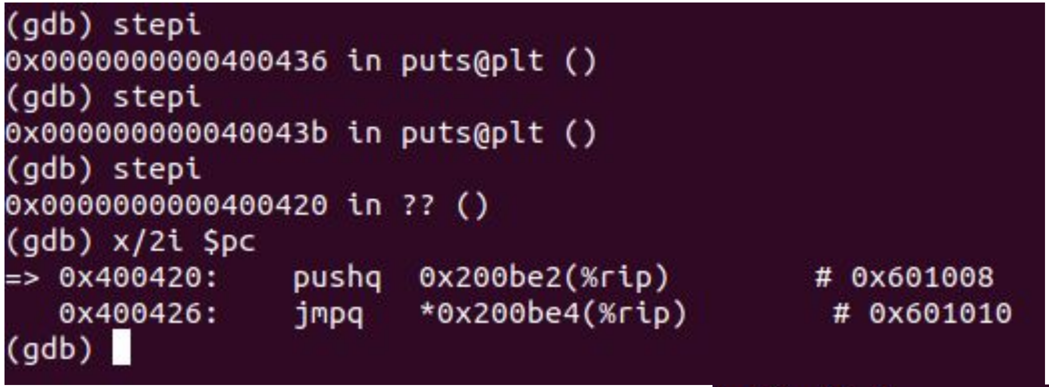
```
.plt - 0x00400420
.got - 0x00600ff0
.got.plt - 00x601000
```



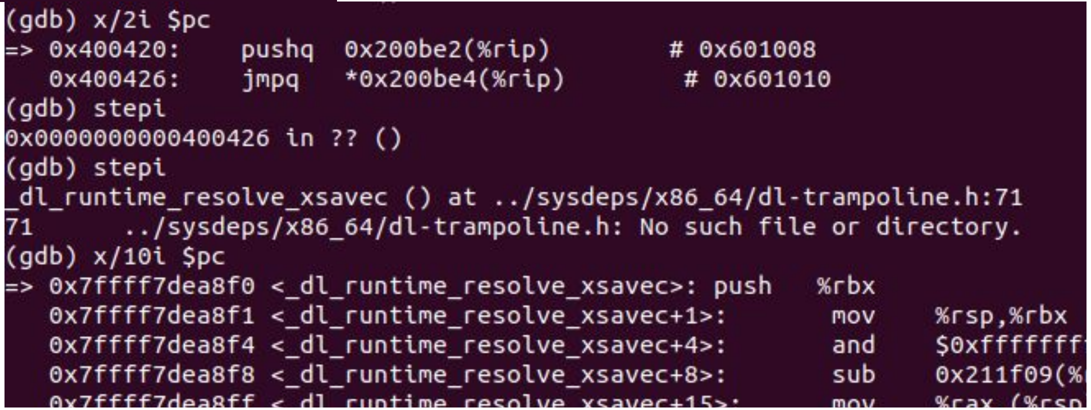
```
(gdb) r
Starting program: /home/justin/Downloads/SIGPWNY/got/demo

Breakpoint 1, 0x0000000000400542 in main () at demo.c:8
8               printf("This is the first call\n");
(gdb) x/i $pc
=> 0x400542 <main+11>:  callq  0x400430 <puts@plt>
(gdb) stepi
0x0000000000400430 in puts@plt ()
(gdb) x/3i $pc
=> 0x400430 <puts@plt>: jmpq   *0x200be2(%rip)        # 0x601018
   0x400436 <puts@plt+6>:       pushq  $0x0
   0x40043b <puts@plt+11>:      jmpq   0x400420
(gdb) x/x 0x601018
0x601018:       0x00400436
(gdb) 
```

How does GOT/PLT Work? (demo) (cont'd)

```
.plt - 0x00400420
.got - 0x00600ff0
.got.plt - 00x601000
```



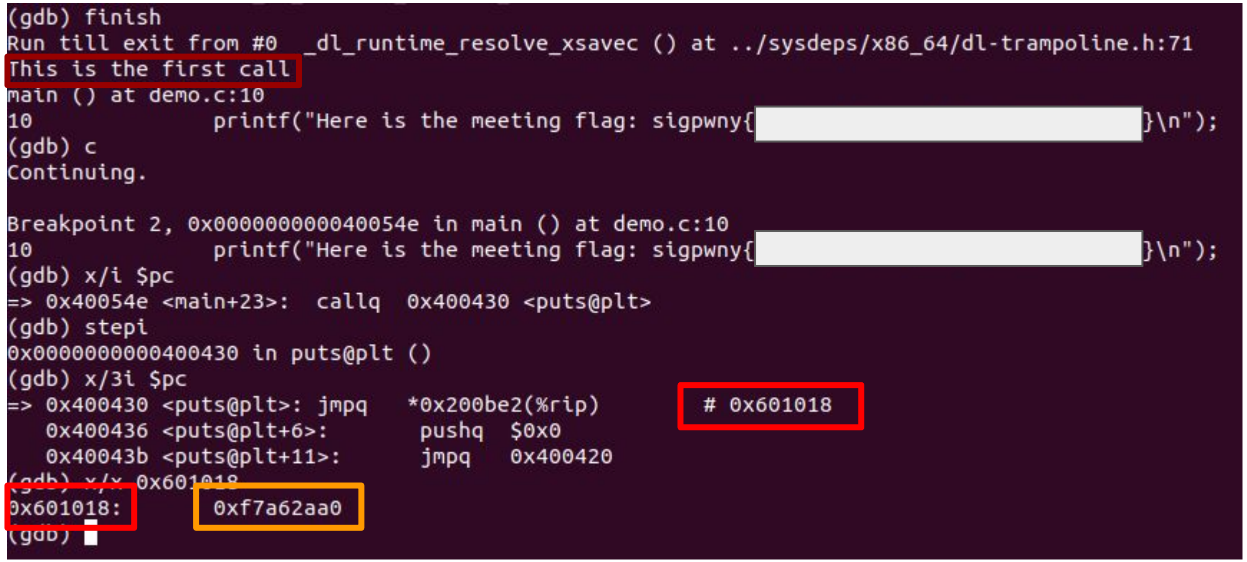
```
(gdb) stepi
0x0000000000400436 in puts@plt ()
(gdb) stepi
0x000000000040043b in puts@plt ()
(gdb) stepi
0x0000000000400420 in ?? ()
(gdb) x/2i $pc
=> 0x400420:    pushq  0x200be2(%rip)        # 0x601008
   0x400426:    jmpq   *0x200be4(%rip)        # 0x601010
(gdb)
```



```
(gdb) x/2i $pc
=> 0x400420:    pushq  0x200be2(%rip)        # 0x601008
   0x400426:    jmpq   *0x200be4(%rip)        # 0x601010
(gdb) stepi
0x0000000000400426 in ?? ()
(gdb) stepi
_dl_runtime_resolve_xsavec () at ../sysdeps/x86_64/dl-trampoline.h:71
71      ../sysdeps/x86_64/dl-trampoline.h: No such file or directory.
(gdb) x/10i $pc
=> 0x7ffff7dea8f0 <_dl_runtime_resolve_xsavec>: push   %rbx
   0x7ffff7dea8f1 <_dl_runtime_resolve_xsavec+1>:      mov    %rsp,%rbx
   0x7ffff7dea8f4 <_dl_runtime_resolve_xsavec+4>:      and    $0xfffffff
   0x7ffff7dea8f8 <_dl_runtime_resolve_xsavec+8>:      sub    0x211f09(%
   0x7ffff7dea8ff <_dl_runtime_resolve_xsavec+15>:     mov    %rax,(%rsp
```

How does GOT/PLT Work? (demo) (cont'd)

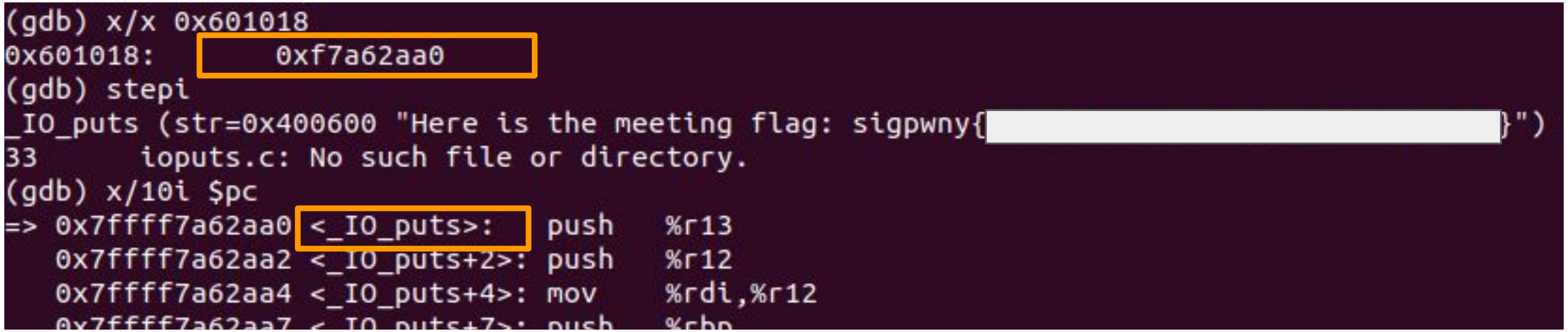
```
.plt - 0x00400420
.got - 0x00600ff0
.got.plt - 00x601000
```



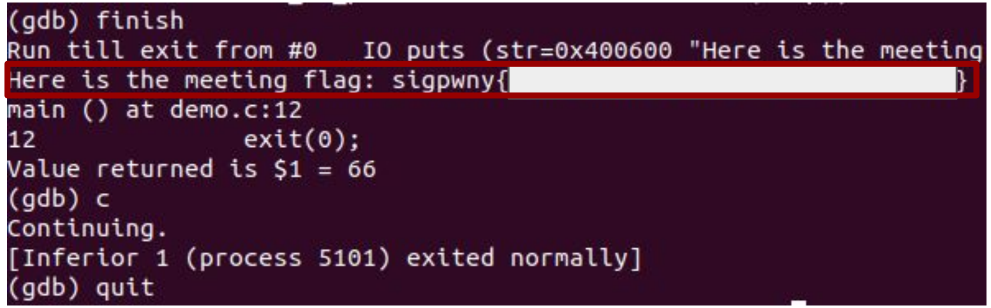
```
(gdb) finish
Run till exit from #0  _dl_runtime_resolve_xsavec () at ../sysdeps/x86_64/dl-trampoline.h:71
This is the first call
main () at demo.c:10
10                printf("Here is the meeting flag: sigpwny{[redacted]}\n");
(gdb) c
Continuing.

Breakpoint 2, 0x000000000040054e in main () at demo.c:10
10                printf("Here is the meeting flag: sigpwny{[redacted]}\n");
(gdb) x/i $pc
=> 0x40054e <main+23>:  callq  0x400430 <puts@plt>
(gdb) stepi
0x0000000000400430 in puts@plt ()
(gdb) x/3i $pc
=> 0x400430 <puts@plt>: jmpq   *0x200be2(%rip)        # 0x601018
   0x400436 <puts@plt+6>:       pushq  $0x0
   0x40043b <puts@plt+11>:      jmpq   0x400420
(gdb) x/x 0x601018
0x601018:       0xf7a62aa0
(gdb) 
```

How does GOT/PLT Work? (demo) (cont'd)

```
.plt - 0x00400420
.got - 0x00600ff0
.got.plt - 00x601000
```



```
(gdb) x/x 0x601018
0x601018:       0xf7a62aa0
(gdb) stepi
_IO_puts (str=0x400600 "Here is the meeting flag: sigpwny{[redacted]}")
33      ioputs.c: No such file or directory.
(gdb) x/10i $pc
=> 0x7ffff7a62aa0 <_IO_puts>:   push   %r13
   0x7ffff7a62aa2 <_IO_puts+2>: push   %r12
   0x7ffff7a62aa4 <_IO_puts+4>: mov    %rdi,%r12
   0x7ffff7a62aa7 <_IO_puts+7>: push   %rbp
```



```
(gdb) finish
Run till exit from #0  _IO_puts (str=0x400600 "Here is the meeting
Here is the meeting flag: sigpwny{[redacted]}
main () at demo.c:12
12              exit(0);
Value returned is $1 = 66
(gdb) c
Continuing.
[Inferior 1 (process 5101) exited normally]
(gdb) quit
```

Your Mission

`Overwrite entries in the GOT to call the function you want to call.`

Mitigations

RELRO - Relocation Read-Only

ASLR - Address Space Layout Randomization

PIE - Position Independent Execution

Please
Leave (but actually stay for help and questions),
Then

Go
Online
To_SIGPwny_CTF_And_Solve_Challenges

Presentation

Now

GOT Overwrite 2

Walkthrough

(hey Justin, open up your terminal)

Format String Vulns

```
%s - print random string

%x - print hex word

%n - write number of printed chars

[num]$ - use the [num]-th parameter

%[num][format specifier] - use for padding


Arbitrary write: Put [addr] onto stack and printf("%[value]n")
```