

HGAME Week1 WriteUp

WEB

这是我第一次打 ctf，真的很有趣 hhhh，辛苦各位出题的学长们啦。

Cosmos 的博客

你好。欢迎你来到我的博客。

大茄子让我把 flag 藏在我的这个博客里。但我前前后后改了很多遍，还是觉得不满意。不过有大茄子告诉我的**版本管理工具**以及 GitHub，我改起来也挺方便的。

1. Cosmos 的博客

打开题目的页面是这样，提取关键信息版本管理工具和 github，推知题目应该是.git 文件泄露，看源码没啥特别的，上百度搜索找到.git 文件泄露的利用方法。发现需要工具 githack 工具

运行

得到.git 文件夹，里面应该包含了一些非常重要的信息。我一开始理解错了，走了弯路，以为是要用 HEAD 文件里的哈希值查看文件差异，结果弄了半天发现在服务器上只提交过一次。再回到开头页面上给的信息“GitHub”，想到可能是一个提交在 github 上面的项目，于是在文件

```
lazyboy@ubuntu:~$ cd '/home/lazyboy/Desktop/web/GitHack-master'
lazyboy@ubuntu:~/Desktop/web/GitHack-master$ python '/home/lazyboy/Desktop/web/GitHack-master/GitHack.py' http://cosmos.hgame.n3ko.co/
[*] Check Depends
[*] Check depends end
[*] Set Paths
[*] Target Url: http://cosmos.hgame.n3ko.co/.git/
[*] Initialize Target
[*] Try to Clone straightly
[*] Clone done
正克隆到 '/home/lazyboy/Desktop/web/GitHack-master/dist/cosmos.hgame.n3ko.co'...
remote: 404 Not Found
fatal: repository 'http://cosmos.hgame.n3ko.co/.git/' not found
[-] Clone Error
[*] Try to Clone with Directory Listing
[*] http://cosmos.hgame.n3ko.co/.git/ is not support Directory Listing
[-] [Skip][First Try] Target is not support Directory Listing
[*] Try to clone with Cache
[*] Initialize Git
[*] Cache files
[*] packed-refs
[*] config
[*] HEAD
[*] COMMIT_EDITMSG
[*] FETCH_HEAD
[*] /refs/heads/master
[*] index
[*] logs/HEAD
[*] refs/heads/master
[*] logs/refs/heads/master
[*] Fetch Commit Objects
[*] objects/05/1894e2ed400a7195008f7022a241e68f5a1335
[*] objects/00/6aca924eaaf556120ca6728099ed0233c10a679
[*] objects/1e/b0928962cc43gd9e647f32f68018a13e56a908
[*] objects/1f/c55fea295446d597542447165f9c57b1c54bf
[*] objects/d4/c5b9733cae27fa7a6cb2e6f24608db7fe4571
[*] objects/ed/3905e0ec91d4ed7d8aa14412df7eb038745f
[*] objects/b6/699e80ede013112e3210bd47655438fa57541f
[*] Fetch Commit Objects End
[*] refs/stash
[*] Valid Repository
[*] Valid Repository Success
[*] Clone Success. Dist File : /home/lazyboy/Desktop/web/GitHack-master/dist/cosmos.hgame.n3ko.co
lazyboy@ubuntu:~/Desktop/web/GitHack-master$
```

夹里找项目的网址。首先在文件夹查看选项勾上查看隐藏文件，就可以在文件夹里看到.git 文件夹，翻找后再 config 文件里找到目标 url 地址

访问后查看 commits 发现一个叫 new file 的文件打开后再根据说明 base64 解码得到 flag

```
[remote "origin"]
  url = https://github.com/FeYcYodhrPDJSru/8LTUKCL83VLhXbc
```

2. 接头霸王

看题目，“头”这个字很关键，不仅是指我老婆“换头大师”，还很有可能是指要改请求头。打开查看源码，没啥特别的。

- 1) 网页上显示 You need to come from <https://vidar.club/>.

由于只知道有 header 这个东西其他，继续百度，查看各个字段的意义。然后用 BurpSuite 抓包工具进行改头。首先改 Referer 字段（Referer 字段用于向服务器发送请求时告诉服务器我是从哪个页面链接过来的）加入 Referer: https://vidar.club/之后就类似这样，见招拆招。

- 2) 网页显示 You need to visit it locally.

在 header 里面加入 X-forwarded-for:localhost (X-Forwarded-For (XFF 头) 是用来识别通过 HTTP 代理或负载均衡方式连接到 web 服务器的客户端最原始的 IP 地址的 http 字段。常用此字段实现伪造 ip 地址的目的)

- 3) 网页显示 You need to use Cosmos Brower to visit.

修改 User-Agent 来伪造访问用的浏览器，把 User-Agent 字段值改为查到的值 YAS-COSMOS/1.0 UP.Browser/6.1.0.7.3 (GUI) MMP/1.0。

- 4) 网页显示 Your should use POST method :)

该第一个字段名请求方式为 POST

- 5) 网页显示 The flag will be updated after 2077, please wait for it patiently. 查看服务的回应和我们的请求只有一个字段与事件有关 If-Modified-Since，百度它，明白是询问服务器在某个时间后文件是否有更改，有的话则更新画面内容。F12 后可以看到服务器返回的 Last-Modified 是 Fri, 01 Jan 2077 00:00:00 GMT 那么我们就直接在请求的 header 里面加上 If-Unmodified-Since: Fri, 01 Jan 2077 00:00:00 GMT，直接要求服务器更新页面 flag 就出来啦 hhhh

3. Code World

这一题开幕雷击，一打开就 403，但是我们根本不慌（滑稽）hhh。查看源码找到提示，说页面用了 302 重定向，也就是说我们要想办法防止页面重定向。想到 linux 的命令 curl 可以防止重定向。输入命令后发现呗 405 拒绝访问，百度得知是请求方式不对，搜索 curl 命令如何用 POST 方式访问发现 url 后面加-X POST 即可照着输入。得到

```
azyboy@ubuntu:~/Desktop/web/GitHack-master$ curl http://codeworld.hgame.day-day
work/ -X POST
<center><h1>人机验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的相加,
参数为a<br><br>现在,需要让结果为10</center>
```

人机验证，一开始看不太懂，大胆去问了学长，就是要用 url 直接传导参数值，不会，查。得知直接后面接?a=blabla 这里要两个数加起来为 10 那么就可以是?a=1%2B9 (+号不是直接简单地输入，url 需要转义)。

输入，得到 flag。

4. 🎉尼泰玟

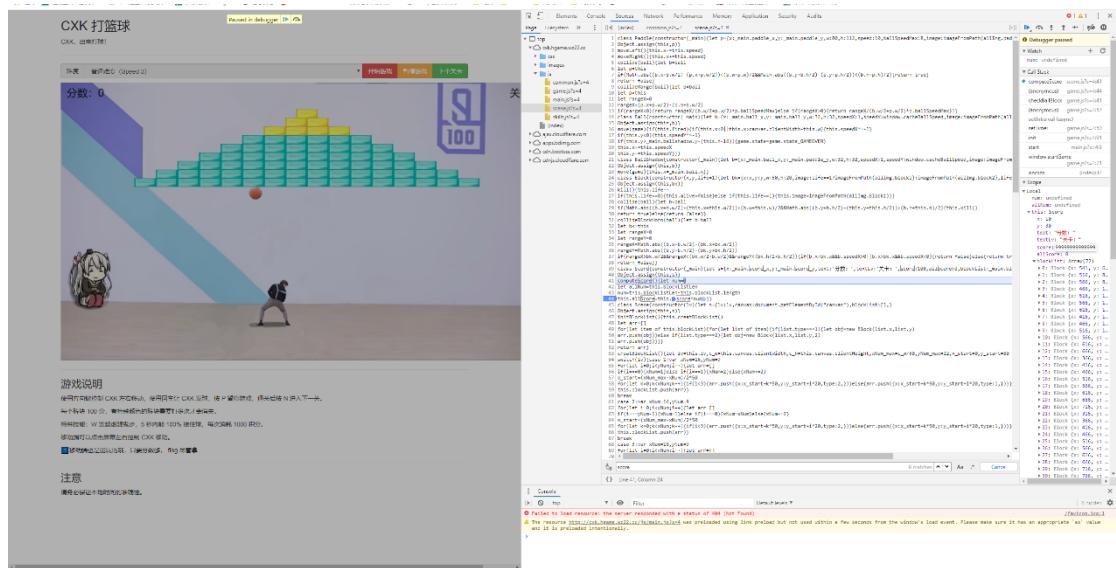
Xswl，学长游戏做的真好，这题根据提示，只要积分到了 30000 就可以拿到 flag。当然是可以慢慢地玩到 30000 分咯，但是我这种三好青年怎么会浪费时间玩游戏（滑稽）。直接 f12 阅读代码，直奔 js 文件夹，找到与分数计算有关的文件，就是它。



Ctrl+f 找 score，搜出来这么一句

```
this.allScore=this.score*num})
```

可以猜想 this.allScore 是总分，num 是砖块数，this.score 是单个砖块分数，选择改单个砖块分数。在这一句设断点，修改变量值。



分数直接爆表 hhhh，得到 flag。

MISC

1. 欢迎参加 HGame !

L10tIC4uL0tIC4tLi4gLS4tLiAtLS0tLSAtLSAuIC4uLS0uLSAtIC0tLSAuLi0tLi0gLi4tLS0gLS0tLS0gLi4tLS0gLS0tLS0gLi4tLS4tIC4uLi4gLS0uIC4tIC0tIC4uLi0t

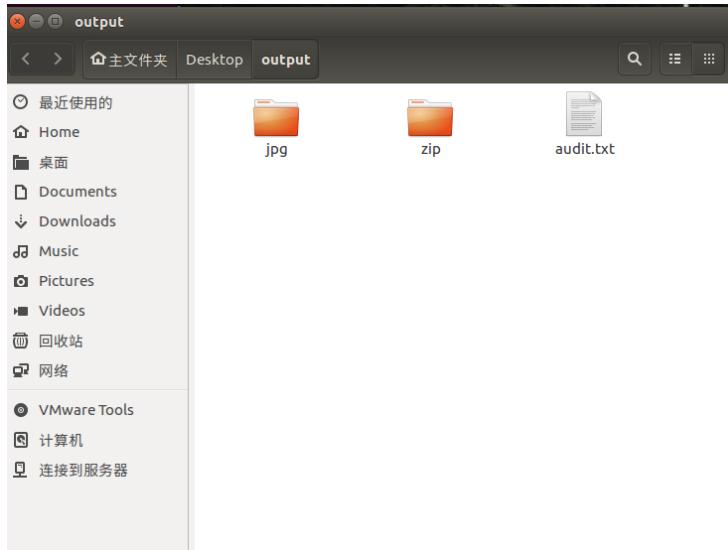
看不出来，乖乖百度，正好贴吧有个帖子，有类似的字符串是 base64，解码得到一串摩斯码



解码，得到 flag。

2. 壁纸

下载得到一张图片，感觉文件大小稍微有点大，决定用 foremost 分离一下，拖进虚拟机。果然，得到一个压缩包



压缩包有密码，但是暂时没有发现提示，于是猜想可能有压缩包注释，拖回物理机，找到 Password is picture ID. 字样

这里贡献我珍藏多年的网站 <https://saucenao.com/index.php>。找 p 站的图非常方便，导入，得到 id。



打开 flag.txt 得到一堆/u 啥啥啥的，感觉应该应该是 unicode 码查了一下再把几种 Unicode 的编码形式都试了一下，发现是 ascii 码，得到 flag。

3. 克苏鲁神话

打开文件发现一个 Bacon.txt 和一个带密码的压缩包，打开 txt 看到密文和提示，立马想到了培根密码，小写字母为 a，大写字母为 b 解得

Your message: (Swap A and B)
aabababaaaaaaabbbaabbbbabaaaaaabbaabbabaababaaaabbbaabbabbbaaaa
This is your encoded or decoded text:

FLAGHIDDENINDOC

只知道 flag 藏在 doc 文件中，试了一下这句话并不是解压密码，百度 zip 解密的几种套路（花了好久 QWQ）看到压缩包里也有 Bacon.txt 且大小一致，得知是同一个文件，采用明文攻击，得到没有密码的 zip 文件，打开 doc，查看选项，显示隐藏文字，ok，得到 flag。

4. 签到题 ProPlus

得到一个 txt 和一个带密码的 zip, txt 写的很清楚栅栏三凯撒五，当然凯撒也有可能是负五，然后前面的解出来应该是一句句子，后面的解出来是密码。

txt 内容：

```
Rdjxfwxjfimkn z,ts wntzi xtjrwmxsfjt jm ywt rtntwhf f y h jnsxf qjFjf jnb rg fiyykwtbsnkm  
tm xa jsdwqjfmkjy wlviHtzqsGsffywjjyynf yssm xfjypnyihjn.
```

JRFVJYFZVRUAGMAI

* Three fences first, Five Caesar next. English sentence first, zip password next.

网上的脚本都不好用，所以自己写了两个一个用的 c，一个用的 py

 caesar.c

 fences.py

解出来句子内容是 Many years later as he faced the firing squad, Colonel Aureliano Buendia was to remember that distant afternoon when his father took him to discover ice. (百年孤独，正好看过，虽然与解题无关 hhh) 然后发现很恐怖的一件事是密码那一段是没有办法被 3 整除的那就只能是 655 的排列然后横着和竖着都试一下，试出解压密码。打开压缩包里面是 txt，全是 ook，查了一下，是 ook 语言，解密，

```
data:text;base32,NFLEET2S04YEWR3HN5AUCQKBJZJVK2CFKVTUCQKBKFIU  
CQKBIVCUGQKZIFAUCRCPINCW6S2BIFAU6V2VNRCVCVSSGRXE6MTBKM3DIRKOO  
53UIMZPGB3DOV3ZIMJWOOCHMNCCTQ33LMJFXEQKPHBQSW3CBKNLC6MRTIFAUIK  
ZVMW4WIQKBIRVW0Q2F1F3UCY2NIFIUCK2ZIFTUCOCBIZCECSKBKDUCSKBMZG  
UCUKBJ5AUI2DHIFAUQN2ZJU2GKL3WNIXWINBQM5CTANJZOZH2YLYLJQW6L2K  
MIYXGM3YJMYFIVRWK52G25LNMFYGMYKZF5GFUMKRHF3TMY2WLBQXC4DNOVLVO  
4KQPFLTSYSOHBJKX1RJRMVWHEWTSOBWXCWBBSNVIHSMTEKVIGGT30IZLDE4LRLJ  
ZGY3DRNI4GY5SXPJTEK4SSJZMHAYJSME3FU4LMHFYGUODUNZLEIM2EOB4FMZD  
ROFWWCNK2MFXS6STCFGFTG6CLGBKFMNSXORWXK3LBOBTGCWJPJRNDCUJJZ043G  
GVSYMFYXA3LVK5LXCUDZK44WE TRYKN2EKMLFNRZFU4TQNYYVQMTNKB4TEZCWJ
```

开头有 base32 提示，用 base32 解码

 opcSyQozZhN9fk3I

 BJRU5ErkJgg==

得到的密码结尾有两个等号，基本确定为 base64，解码

 PNG



开头有 PNG 字样猜想可能是 png 图片，找到一个脚本将 base64 的码导入 png 文件中，打开，得到一个二维码，扫一下得到 flag。

5. 每日推荐

拿到一个 pcapng 文件，在此之前我都不知道这是啥，见都没见过 QAQ。查询得知要用 wireshark 打开，于是下载工具，打开

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::3016:7e4c:4f7... ff02::1:ff:0:2222	ICMPv6	86	Neighbor Solicitation for fe80::250:56ff:fe0:2222 from 00:50:56:c0:00:08	
2	1.000021	fe80::3016:7e4c:4f7... ff02::1:ff:0:2222	ICMPv6	86	Neighbor Solicitation for fe80::250:56ff:fe0:2222 from 00:50:56:c0:00:08	
3	2.099279	192.168.146.132	114.114.114.114	DNS	75	Standard query 0x95c4 A www.gstatic.com
4	2.121295	192.168.146.132	114.114.114.114	DNS	89	Standard query 0xa935 A clientservices.googleapis.com
5	2.121296	192.168.146.132	114.114.114.114	DNS	79	Standard query 0xd3c A clients2.google.com
6	2.155248	114.114.114.114	192.168.146.132	DNS	139	Standard query response 0x95c4 A www.gstatic.com A 203.208.50.56 A 203.208.50.55 A 203.208.50.55
7	2.175517	114.114.114.114	192.168.146.132	DNS	119	Standard query response 0x8d3c A clients2.google.com CNAME clients1.google.com A 174.36.229.126
8	2.179724	114.114.114.114	192.168.146.132	DNS	153	Standard query response 0xa935 A clientservices.googleapis.com A 203.208.39.239 A 203.208.39.239
9	2.181660	192.168.146.132	203.208.39.239	TCP	66	50106 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
10	2.182117	192.168.146.132	172.217.160.78	TCP	66	50107 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
11	2.182718	192.168.146.132	203.208.50.56	TCP	66	50108 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
12	2.183144	192.168.146.132	203.208.50.56	TCP	66	50109 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
13	2.184516	192.168.146.132	114.114.114.114	DNS	79	Standard query 0xfe16 A accounts.google.com
14	2.185034	192.168.146.132	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
15	2.195609	192.168.146.132	114.114.114.114	DNS	74	Standard query 0x80ad A www.google.com
16	2.219584	203.208.50.56	192.168.146.132	TCP	66	443 → 50108 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
17	2.219650	192.168.146.132	203.208.50.56	TCP	54	50108 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
18	2.220088	203.208.39.239	192.168.146.132	TCP	66	443 → 50108 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
19	2.220117	192.168.146.132	203.208.39.239	TCP	54	50106 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
20	2.220693	192.168.146.132	203.208.50.56	TLSv1.3	571	Client Hello
21	2.220992	203.208.50.56	192.168.146.132	TCP	66	443 → 50108 [ACK] Seq=1 Ack=518 Win=64240 Len=0
22	2.221198	192.168.146.132	203.208.39.239	TLSv1.3	571	Client Hello
23	2.221507	203.208.39.239	192.168.146.132	TCP	66	443 → 50106 [ACK] Seq=1 Ack=518 Win=64240 Len=0
24	2.221647	203.208.50.56	192.168.146.132	TCP	66	443 → 50109 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
25	2.222163	192.168.146.132	203.208.50.56	TCP	54	50109 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
26	2.222460	192.168.146.132	203.208.50.56	TLSv1.3	571	Client Hello
27	2.222703	203.208.50.56	192.168.146.132	TCP	66	443 → 50109 [ACK] Seq=1 Ack=518 Win=64240 Len=0
28	2.370854	192.168.146.132	172.217.160.78	TCP	66	50110 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
29	2.463571	114.114.114.114	192.168.146.132	DNS	95	Standard query response 0xfe16 A accounts.google.com A 216.58.200.237
30	2.463577	114.114.114.114	192.168.146.132	dns	99	Standard query response 0x80ad A www.google.com A 174.36.229.126

```
> Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{2E17B068-F3FB-4B25-A802-400A70D39FE6}, id 0
> Ethernet II, Src: VMWare_00:00:08 (00:50:56:c0:00:08), Dst: IPv6mcast_ff:c0:22:22 (33:33:ff:c0:22:22)
> Internet Protocol Version 6, Src: fe80::3016:7e4c:4f70:2ff6, Dst: ff02::1:ff:0:2222
> Internet Control Message Protocol v6
```

0000	33	33	ff	c0	22	22	00	50	56	c0	00	08	86	dd	60	00	33	"", P V, -.
0010	00	00	00	20	3a	ff	fe	80	00	00	00	00	00	00	30	16	...	;
0020	7e	4c	70	2f	f6	ff	02	00	00	00	00	00	00	00	00	00	-LOp/.....	
0030	00	01	ff	c0	22	22	87	00	5a	a7	00	00	00	fe	80	...	"", Z, -.	
0040	00	00	00	00	00	00	02	50	56	ff	fe	c0	22	22	01	01	P V, -."	
0050	00	50	56	c0	00	00	08	PV, -.	

一窍不通，不知道是个啥，于是现学。很久很久之后，知道这种题目叫流量分析，总之就是要在这些流中找到有用的信息。一般来说 POST 请求里的东西价值比较大，筛选 http，

3048	28.449637	192.168.146.132	192.168.146.1	HTTP	790	POST /wp-admin/async-upload.php	HTTP/1.1	(application/x-www-form-urlencoded)
7258	56.542139	192.168.146.132	192.168.146.1	HTTP	1094	POST /wp-admin/admin-ajax.php	HTTP/1.1	(application/x-www-form-urlencoded)
1996	21.009896	192.168.146.132	192.168.146.1	HTTP	1091	POST /wp-admin/admin-ajax.php	HTTP/1.1	(application/x-www-form-urlencoded)
7393	70.622019	192.168.146.132	192.168.146.1	HTTP	526	POST /wp-admin/admin-ajax.php	HTTP/1.1	(application/x-www-form-urlencoded)

POST 有这么几个，都翻看一下得知这两个 IP 之间应该是传输了一个 song.zip 的文件（还有 E99 NB！滑稽）然后根据题目的描述，这道题大概率和歌有关，所以把目标定为找到这个 zip 文件

- ✓ Member Key: title
String value: E99 nb!
- ✓ Key: title
- ✓ Member Key: content
String value [truncated]: <!-- wp:file {"id":12,"href":"http://192.168.146.1:8008/wp-content/uploads/2020/01/song.zip"}
Key: content

用了 wireshark 的自动提取文件功能，然而并没有帮我提取出来，这时我发现了之前遗漏的一个点

790 POST /wp-admin/async-upload.php HTTP/1.1

这里有个关键词 upload! 在里面翻找，找到一个 8m 左右的文件，大胆猜测就是它，我们要找的 song.zip。打开查看 16 进制，开头是 PK，这波稳了 hhhh

[Response in frame: 3067]

File Data: 8290400 bytes

保存下来，改后缀为 zip，果然！根据注释说密码为六位数字，这明摆着让我们暴力破解，于是爆破得出 zip 的密码，歌我很喜欢，前几天也正好出现在我的日推里 hhhh。上网查了一下音频文件隐写的几种主要方式，听了一下感觉没啥异常，基本把隐写方式定在频谱图和 MP3Stego 隐写两者。先用记事本打开搜了一下 pass 没有搜到东西，感觉可能不是 MP3Stego。看了一下频谱图，找到 flag。

Crypto

1. InfantRSA

Rsa 加密，就是算，算出来的数值用 int. to_bytes 函数再转化得到 flag。这道题莫名其妙的卡了我好久，直接用 python 算给的值都是错的，佛了 QAQ。幸好学会了用 gmpy2 库，收获很大 hhhh。

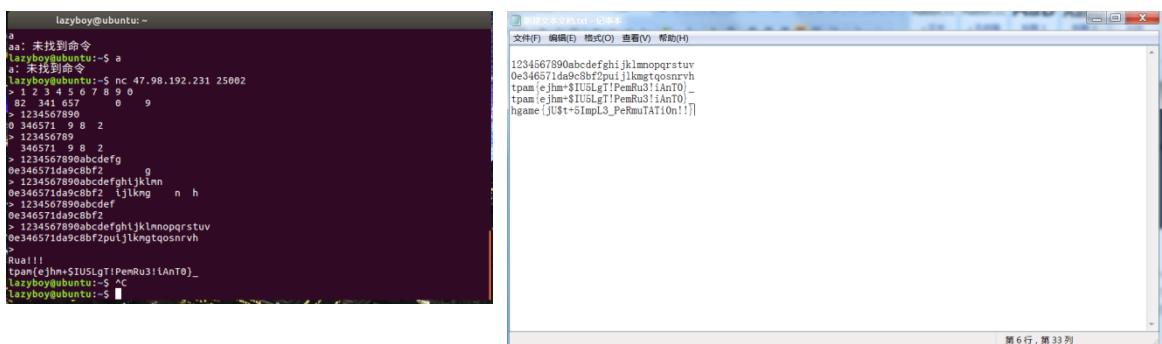
2. Affine

查看代码，发现整个加密过程都写出来了，然后就思考计算了一下，总的来说就是根据加密程序反着运算，写了个 c 程序来解密，运行，得到 flag

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char table[100] = "zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZCVBNM";
6     int mod, count = 0, n, i, ii, k = 1, flag = 0;
7     mod = strlen(table);
8     while(pass[count] != '\0'){
9         flag = 0;
10        for(n = 0; n < 100; n++){
11            if(pass[count] == table[n]){
12                ii = n;
13                while(1){
14                    if((48+mod*k+ii)%13 == 0)
15                        break;
16                    else
17                        k++;
18                }
19                i = (48+mod*k+ii)/13;
20                //printf("\ni = %d ii = %d k = %d\n", i, ii, k);
21                printf("%c", table[i]);
22                k = 1;
23                flag = 1;
24            }
25        }
26        if(flag == 0)
27            printf("\n");
28        count++;
29    }
30    return 0;
31 }
```

4. Reorder

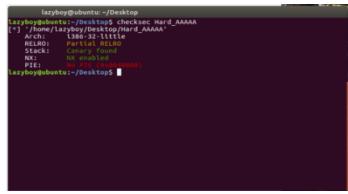
根据题名，应该是一种重排序。不知道到底具体怎么样，先用虚拟机 nc 命令连上去看看。试了一回发现是每次连接都不一样的规律，然后输入一定次数后会给出 Rua！！！加一串字符，一看就是 reorder 过的 flag 啊（滑稽）。那就很方便了，既然这样我们只需要输入和 flag 同样长度的字符串，一一对应之后就可以把 flag 重现啦



Pwn

1. Hard_AAAAA

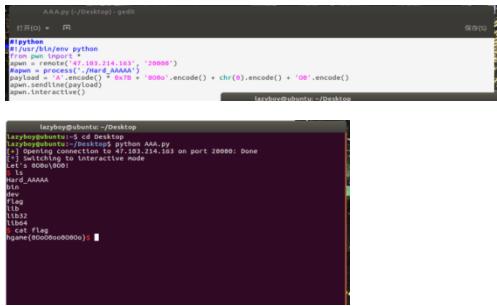
这道题由于我回了老家，换了我爸的电脑，ida 还没有装，所以没有办法很好地重现写题过程，也莫得 ida 的截图，希望哥哥们见谅。过程大概是这样的。拿到文件后，先 checksec



```
lazyboy@ubuntu:~/Desktop$ ./Hard_d_AAAAA
[+] /home/lazyboy/Desktop/Hard_d_AAAAA
Arch: little-endian
LITTLE_ENDIAN
FILE: Hard_d_AAAAA
PIE: No PIE (Execution)
lazyboy@ubuntu:~/Desktop$
```

发现是有 nx 保护的，那就不能无脑 aaa 了 hhhh，果然是 hard 一点的。

拖进 ida，找到 main 函数，f5，我记得是可以看到一个叫 backdoor 的函数，打开一看是可以看到这是一个可以拿到 shell 的函数，这道题显然是拿到 shell 就成功了。然后找到这个函数是在一个 if 语句中那么我们只要让条件实现就 ok 啦。具体没有 ida，我也忘了，但是脚本我带了过来，以下是截图



```
AAA.py (-/Desktop): ./Hard_d_AAAAA
[+] /home/lazyboy/Desktop/Hard_d_AAAAA
Arch: little-endian
FILE: Hard_d_AAAAA
PIE: No PIE (Execution)
lazyboy@ubuntu:~/Desktop$ ./AAA.py
[*] Starting interaction
[*] Opening connection to 47.103.214.103 on port 20000: Done
[*] Let's rock!(000)
[*] Hard_d_AAAAA
[*] dev
[*] fd0
[*] l1d
[*] l1s3t
[*] l1b54
[*] l1c_f1ag
[*] flagname(0000000000000000)
```

得到 flag。在写这题之前我是从来没接触过类似的软件或是这样的任务，我认识到了 pwn 的魅力，它也许是最接近我脑海中黑客技能的一个方向，但是我同时也发现它是真的难，希望自己能够把 pwn 玩的明白一些 QwQ（其实根本不是玩，是快乐地受虐）。我想写 pwn 啊啊啊啊！！