

xctf攻防世界 MISC高手进阶区 D1f

原创

[18947943](#) 于 2022-01-14 17:43:17 发布 3613 收藏

分类专栏: [攻防世界misc之路](#) 文章标签: [misc](#)

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/18947943/article/details/122498767>

版权



[攻防世界misc之路](#) 专栏收录该内容

68 篇文章 2 订阅

订阅专栏

1. 进入环境, 下载附件

是一张美图:

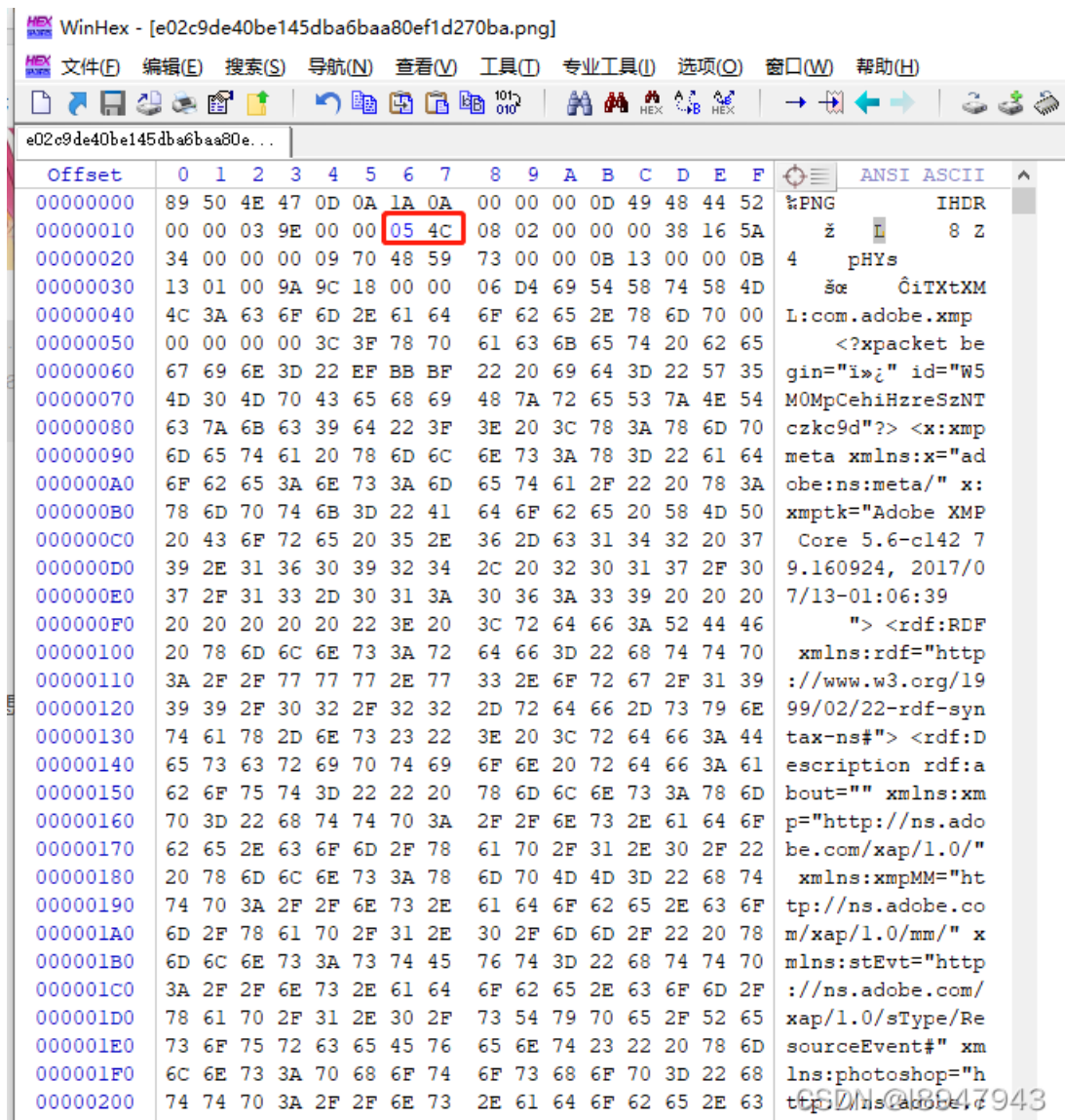


没有其他提示信息, 接着走。

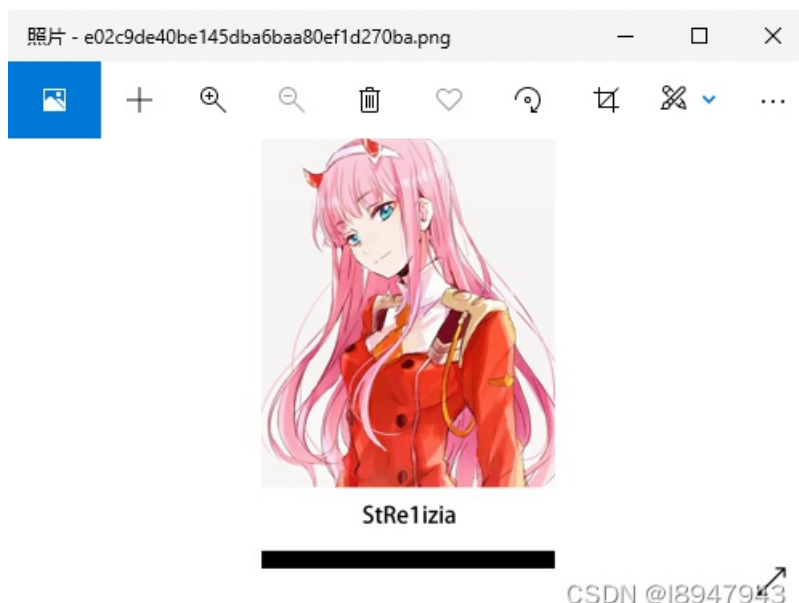
2. 问题分析

1. StegSolve探索

使用StegSolve打开图片，发现图片巨长。联想到之前做的一个题，使用winhex修改图片高度



发现了隐藏的信息:



图片文件分离

我们将图片塞入kali中，使用binwalk查看一下，是否有隐含文件，如图：



```

└─$ binwalk -e e02c9de40be145dba6baa80ef1d270ba.png
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         PNG image, 926 x 1100, 8-bit/color RGB, non-interlaced
1822        0x71E      Zlib compressed data, default compression
989714      0xF1A12    RAR archive data version 4.x, first volume type: M
AIN_HEAD
CSDN @I8947943

```

有个rar文件，我们使用foremost分离一下，如图：

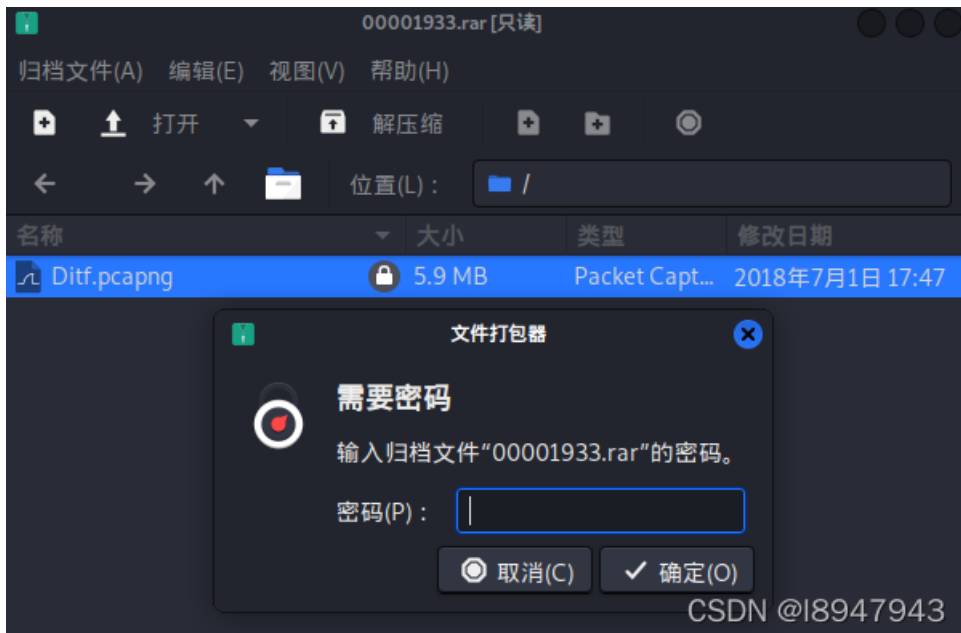
```

(zhangfa@kali)-[~/下载]
└─$ foremost e02c9de40be145dba6baa80ef1d270ba.png
Processing: e02c9de40be145dba6baa80ef1d270ba.png
|*|
(zhangfa@kali)-[~/下载]
└─$

```

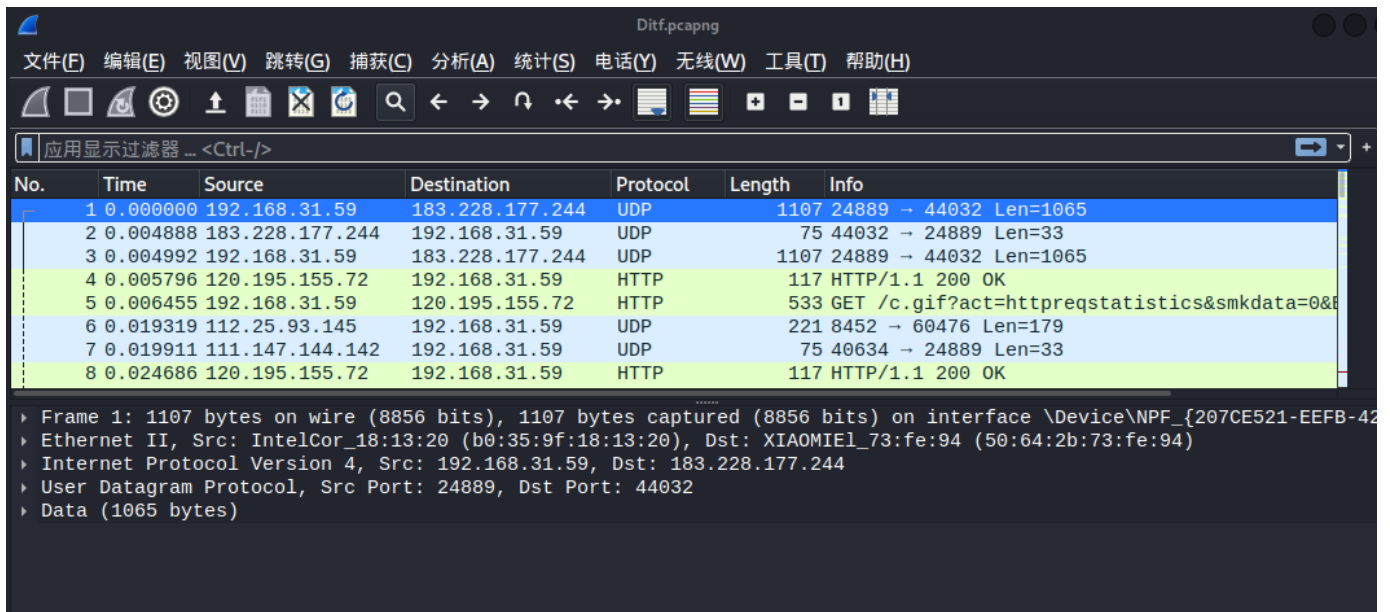
解密压缩包

解压过程发现需要密码，联想到刚才图片的字符StRe1izia，尝试一下：



好家伙，还真解压成功了，是个pcag流量包文件，果断塞入wireshark，看看猫腻

分析数据流



使用分组字节流搜索关键词flag和ctf，都没有对应内容！无思路了，查看wp，发现是要找png相关的关键词。。。如图：

The image shows a Wireshark interface with a packet capture of a TCP stream. The selected packet (No. 6983) is an HTTP GET request for /kiss.png. The details pane shows the request method, URI, and version.

No.	Time	Source	Destination	Protocol	Length	Info
6974	20.305...	192.168.31.59	123.206.131.120	HTTP	432	GET / HTTP/1.1
6978	20.317...	123.206.131.120	192.168.31.59	TCP	54	80 → 33307 [ACK] Seq=1 Ack=379 Win=30336 Len=0
6979	20.318...	123.206.131.120	192.168.31.59	HTTP	567	HTTP/1.1 200 OK (text/html)
6983	20.324...	192.168.31.59	123.206.131.120	HTTP	398	GET /kiss.png HTTP/1.1
6993	20.338...	123.206.131.120	192.168.31.59	TCP	1458	80 → 33307 [ACK] Seq=514 Ack=723 Win=31360 Len=0
6994	20.338...	123.206.131.120	192.168.31.59	TCP	1458	80 → 33307 [ACK] Seq=1918 Ack=723 Win=31360 Len=0
6995	20.338...	123.206.131.120	192.168.31.59	TCP	1458	80 → 33307 [ACK] Seq=3322 Ack=723 Win=31360 Len=0
6996	20.339...	192.168.31.59	123.206.131.120	TCP	54	33307 → 80 [ACK] Seq=723 Ack=3986 Win=66816 Len=0

Details pane for packet 6983:

- [SEQ/ACK analysis]
- [Timestamps]
- TCP payload (344 bytes)
- Hypertext Transfer Protocol
 - GET /kiss.png HTTP/1.1\r\n
 - [Expert Info (Chat/Sequence): GET /kiss.png HTTP/1.1\r\n]
 - Request Method: GET
 - Request URI: /kiss.png
 - Request Version: HTTP/1.1

追踪其http流，查看具体内容：

The image shows the HTML content of the response. The `` tag is highlighted, and its src attribute value is shown as a base64-encoded string: `ZmxhZ3tPel180bmRfSGlyMF9sb3ZzX0ZvcjN2ZXJ9`.

```

Content-Type: text/html

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  </head>
  <body>
    
    ZmxhZ3tPel180bmRfSGlyMF9sb3ZzX0ZvcjN2ZXJ9
  </body>
</html>

```

发现一串奇怪的东东（其实我也不知道为啥大佬们就知道这个是结果，o(∩_∩)o），使用base64解码，得到最终结果：

The image shows a browser's developer tools interface. The URL bar contains the flag: `flag{Oz_4nd_Hir0_lov3_For3ver}`.

答案为： `flag{Oz_4nd_Hir0_lov3_For3ver}`