西湖论剑预选赛Misc第二题Write-UP

原创

ObjectNF 于 2019-06-07 11:45:31 发布

分类专栏： WriteUp 网络安全 文章标签： 西湖论剑 Misc WriteUp

本文链接：https://blog.csdn.net/weixin_44911246/article/details/91126701

近期铺天盖地宣传的“西湖论剑”网络安全技能赛预选已经结束了。在这里随便糊一篇文章（也是我第一次写Write-Up文章），就聊聊杂项最先放出的那个第二题的解法。

首先拿到题，解压，发现里面有“题目描述”，先看描述，是这么写的：

> 我们截获了一些IP数据报，发现报文头中的TTL值特别可疑，怀疑是通信方嵌入了数据到TTL，我们将这些TTL值提取了出来，你能看出什么端倪吗？

然后看给出的另一个文件，`ttl.txt`，里面的内容是这样的：

不难发现TTL值只有 63，127，191，255 四种，都是2的某次幂-1的值。于是将这四个数都转换成二进制，得到 111111 、

1111111、10111111、11111111 四个二进制数。从后面两个数字可以观察到二进制数的开头两位似乎有关系。又因为TTL值为一个8位整数，进行合理猜想，不妨将不足8位的二进制数开头补0，变为8位后再取开头两位。即：00111111、01111111、10111111、11111111 提取开头两位为：00、01、10、11，恰好为全排列，可以用于数据的存储。

这样每组两个比特，四组就可以组成一个字节。博客园

上也有文章提到了这种数据隐藏方式（点击这里）。

随后编写脚本，将这些数据进行处理并将得到的二进制数据转为十六进制明文。Python脚本代码如下：（假设输入文件为`ttl.txt`，输出文件为`ttl_res.txt`）

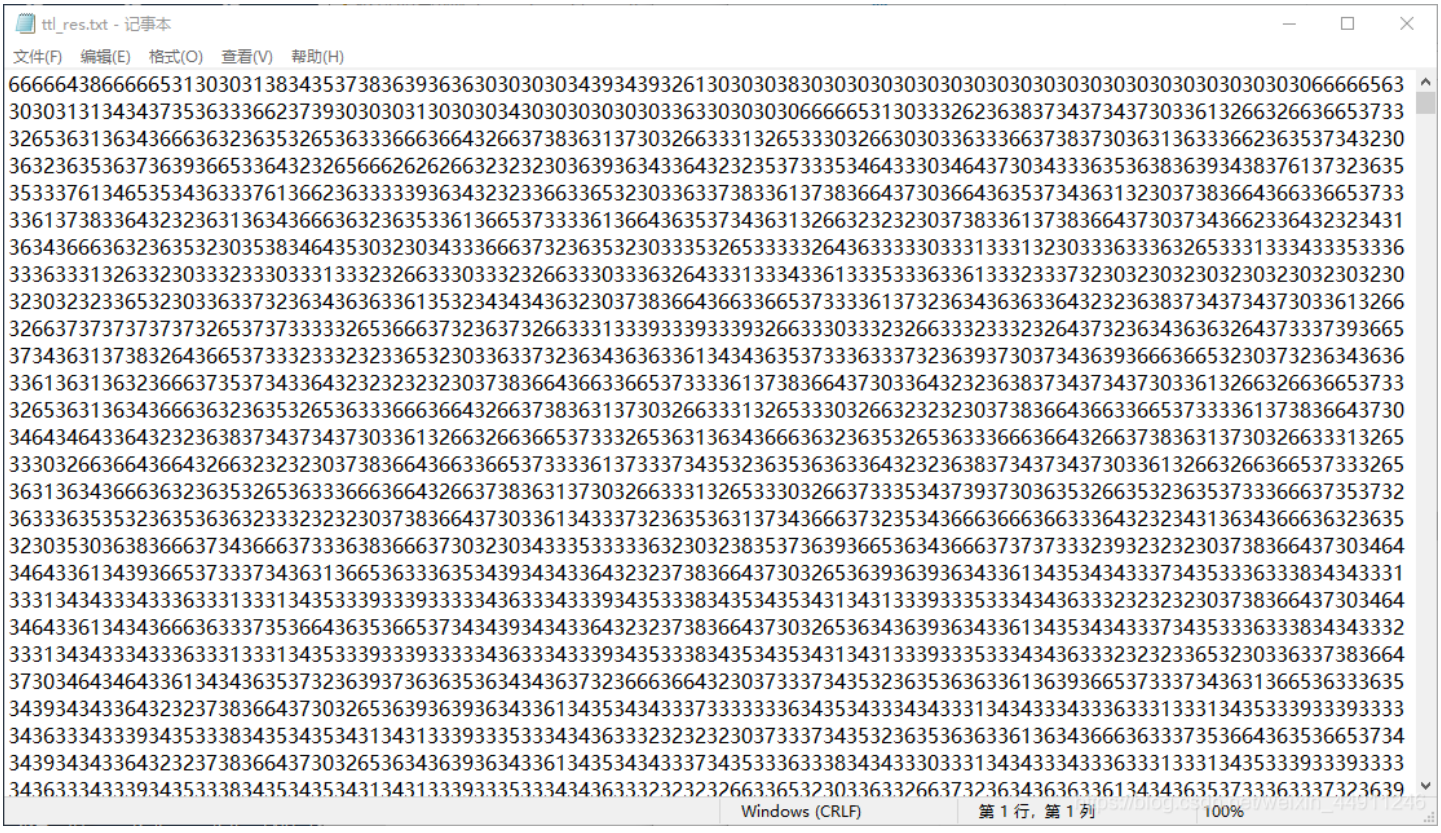
```
infile = open("ttl.txt", "r")
outfile = open("ttl_res.txt", "w")

ascii_data = ""
num = 0

for i in range(295376):
    tmp = infile.readline();
    tmp = tmp[4:len(tmp)-1]
    bin_data = bin(int(tmp))[2:]
    bin_len = len(bin_data)
    if bin_len != 8:
        for j in range(8-bin_len):
            bin_data = "0" + bin_data
    bin_val = bin_data[:2]
    ascii_data += bin_val
    num += 1
    if num == 4:
        num = 0
        msg_hex = hex(int(ascii_data, 2))[2:]
        if len(msg_hex) < 2:
            msg_hex = "0" + msg_hex
        outfile.write(msg_hex)
        ascii_data = ""

infile.close()
outfile.close()
```

得到的文件是这个样子的：



随后打开WinHex，新建文件，将记事本中的内容全选、复制，粘贴到WinHex中，粘贴方式选择“ASCII Hex”，得到如下结果：

00000000	66	66	64	38	66	66	65	31	30	30	31	38	34	35	37	38	ffd8ffe100184578
00000016	36	39	36	36	30	30	30	30	34	39	34	39	32	61	30	30	6966000049492a00
00000032	30	38	30	30	30	30	30	30	30	30	30	30	30	30	30	30	0800000000000000
00000048	30	30	30	30	30	30	30	30	66	66	65	63	30	30	31	31	00000000ffec0011
00000064	34	34	37	35	36	33	36	62	37	39	30	30	30	31	30	30	4475636b79000100
00000080	30	34	30	30	30	30	30	30	33	63	30	30	30	30	66	66	040000003c0000ff
00000096	65	31	30	33	32	62	36	38	37	34	37	34	37	30	33	61	e1032b687474703a
00000112	32	66	32	66	36	65	37	33	32	65	36	31	36	34	36	66	2f2f6e732e61646f
00000128	36	32	36	35	32	65	36	33	36	66	36	64	32	66	37	38	62652e636f6d2f78
00000144	36	31	37	30	32	66	33	31	32	65	33	30	32	66	30	30	61702f312e302f00
00000160	33	63	33	66	37	38	37	30	36	31	36	33	36	62	36	35	3c3f787061636b65
00000176	37	34	32	30	36	32	36	35	36	37	36	39	36	65	33	64	7420626567696e3d
00000192	32	32	65	66	62	62	62	66	32	32	32	30	36	39	36	34	22efbbbf22206964
00000208	33	64	32	32	35	37	33	35	34	64	33	30	34	64	37	30	3d2257354d304d70
00000224	34	33	36	35	36	38	36	39	34	38	37	61	37	32	36	35	43656869487a7265
00000240	35	33	37	61	34	65	35	34	36	33	37	61	36	62	36	33	537a4e54637a6b63
00000256	33	39	36	34	32	32	33	66	33	65	32	30	33	63	37	38	3964223f3e203c78
00000272	33	61	37	38	36	64	37	30	36	64	36	35	37	34	36	31	3a786d706d657461
00000288	32	30	37	38	36	64	36	63	36	65	37	33	33	61	37	38	20786d6c6e733a78
00000304	33	64	32	32	36	31	36	34	36	66	36	32	36	35	33	61	3d2261646f62653a
00000320	36	65	37	33	33	61	36	64	36	35	37	34	36	31	32	66	6e733a6d6574612f
00000336	32	32	32	30	37	38	33	61	37	38	36	64	37	30	37	34	2220783a786d7074
00000352	36	62	33	64	32	32	34	31	36	34	36	66	36	32	36	35	6b3d2241646f6265
00000368	32	30	35	38	34	64	35	30	32	30	34	33	36	66	37	32	20584d5020436f72
00000384	36	35	32	30	33	35	32	65	33	33	32	64	36	33	33	30	6520352e332d6330
00000400	33	31	33	31	32	30	33	36	33	36	32	65	33	31	33	34	31312036362e3134
00000416	33	35	33	36	33	36	33	31	32	63	32	30	33	32	33	30	353636312c203230
00000432	33	31	33	32	32	66	33	30	33	32	32	66	33	30	33	36	31322f30322f3036

D:\编程与比赛\CTF题目\西湖论剑\

文件大小: 132 KB
135,667 字节

缺省编辑模式
状态: 原始的
撤消级数: 0
反向撤消: 暂无信息

创建时间: 2019/04/07 11:45:02
最后写入时间: 2019/04/07 11:44:03
属性: A
图标: 0

模式: 十六进制
偏移地址: decimal
每页字节数: 45x16=720
当前窗口: 1
窗口总数: 1

剪贴板: 无数据
暂存文件夹: 124 GB 空余
.DMINI~1\AppData\Local\Temp

另存为txt文件，怀疑其仍为16进制数据。在WinHex中再次新建文件，重复上述操作，得到：

decode_2.jpg

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000	FF	D8	FF	E1	00	18	45	78	69	66	00	00	49	49	2A	00	ÿØÿá Exif II*
00000016	08	00	00	00	00	00	00	00	00	00	00	00	FF	EC	00	11	ÿì
00000032	44	75	63	6B	79	00	01	00	04	00	00	00	3C	00	00	FF	Ducky < ÿ
00000048	E1	03	2B	68	74	74	70	3A	2F	2F	6E	73	2E	61	64	6F	á +http://ns.ado
00000064	62	65	2E	63	6F	6D	2F	78	61	70	2F	31	2E	30	2F	00	be.com/xap/1.0/
00000080	3C	3F	78	70	61	63	6B	65	74	20	62	65	67	69	6E	3D	<?xpacket begin=
00000096	22	EF	BB	BF	22	20	69	64	3D	22	57	35	4D	30	4D	70	"ï»¿" id="W5M0Mp
00000112	43	65	68	69	48	7A	72	65	53	7A	4E	54	63	7A	6B	63	CehiHzreSzNTczkc
00000128	39	64	22	3F	3E	20	3C	78	3A	78	6D	70	6D	65	74	61	9d"?> <x:xmpmeta
00000144	20	78	6D	6C	6E	73	3A	78	3D	22	61	64	6F	62	65	3A	xmlns:x="adobe:
00000160	6E	73	3A	6D	65	74	61	2F	22	20	78	3A	78	6D	70	74	ns:meta/" x:xmpt
00000176	6B	3D	22	41	64	6F	62	65	20	58	4D	50	20	43	6F	72	k="Adobe XMP Cor
00000192	65	20	35	2E	33	2D	63	30	31	31	20	36	36	2E	31	34	e 5.3-c011 66.14
00000208	35	36	36	31	2C	20	32	30	31	32	2F	30	32	2F	30	36	5661, 2012/02/06
00000224	2D	31	34	3A	35	36	3A	32	37	20	20	20	20	20	20	20	-14:56:27
00000240	20	22	3E	20	3C	72	64	66	3A	52	44	46	20	78	6D	6C	"> <rdf:RDF xml
00000256	6E	73	3A	72	64	66	3D	22	68	74	74	70	3A	2F	2F	77	ns:rdf="http://w
00000272	77	77	2E	77	33	2E	6F	72	67	2F	31	39	39	39	2F	30	ww.w3.org/1999/0
00000288	32	2F	32	32	2D	72	64	66	2D	73	79	6E	74	61	78	2D	2/22-rdf-syntax-
00000304	6E	73	23	22	3E	20	3C	72	64	66	3A	44	65	73	63	72	ns#"> <rdf:Descr
00000320	69	70	74	69	6F	6E	20	72	64	66	3A	61	62	6F	75	74	iption rdf:about
00000336	3D	22	22	20	78	6D	6C	6E	73	3A	78	6D	70	3D	22	68	="" xmlns:xmp="h
00000352	74	74	70	3A	2F	2F	6E	73	2E	61	64	6F	62	65	2E	63	ttp://ns.adobe.c
00000368	6F	6D	2F	78	61	70	2F	31	2E	30	2F	22	20	78	6D	6C	om/xap/1.0/" xml
00000384	6E	73	3A	78	6D	70	4D	4D	3D	22	68	74	74	70	3A	2F	ns:xmpMM="http:/
00000400	2F	6E	73	2E	61	64	6F	62	65	2E	63	6F	6D	2F	78	61	/ns.adobe.com/xa
00000416	70	2F	31	2E	30	2F	6D	6D	2F	22	20	78	6D	6C	6E	73	p/1.0/mm/" xmlns
00000432	3A	73	74	52	65	66	3D	22	68	74	74	70	3A	2F	2F	6E	:stRef="http://n
00000448	73	2E	61	64	6F	62	65	2E	63	6F	6D	2F	78	61	70	2F	s.adobe.com/xap/
00000464	31	2E	30	2F	73	54	79	70	65	2F	52	65	73	6F	75	72	1.0/sType/Resour
00000480	63	65	52	65	66	23	22	20	78	6D	70	3A	43	72	65	61	ceRef#" xmp:Crea

decode_2.jpg
D:\编程与比赛\CTF题目\西湖论剑\

文件大小: 36.1 KB
36,922 字节

缺省编辑模式
状态: 原始的
撤消级数: 0
反向撤消: 暂无信息

创建时间: 2019/04/07 11:47:19
最后写入时间: 2019/04/07 11:47:19
属性: A
图标: 0

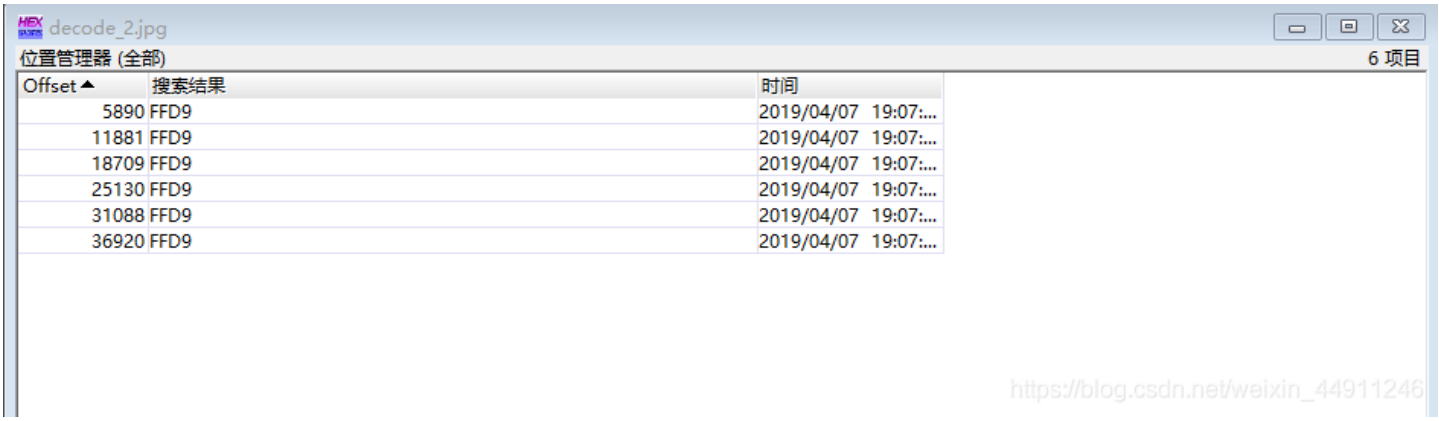
模式: 十六进制
偏移地址: decimal
每页字节数: 45x16=720
当前窗口: 1
窗口总数: 1

剪贴板: 无数据
暂存文件夹: 122 GB 空余
.DMINI~1\AppData\Local\Temp

`FF D8 FF E1`是JPEG文件头部标志，因此保存为jpg。能从缩略图发现这是一个二维码，但是不完整。

decode_2.jpg

怀疑该文件中含有多张JPEG图片。搜索JPEG文件尾标志`FF D9`，结果如下：



将6个JPEG文件分别提取保存为新的文件，得：

分别是同一个二维码的不同部分。利用Photoshop拼接得到完整二维码：

二维码识别，扫描结果为：

```
key:AutomaticKey cipher:fftu{2028mb39927wn1f96o6e12z03j58002p}
```

这并非flag原文而是密文。前面的key字段有两个含义：加密方式为**AutoKey Chiper**且加密密钥为`AutomaticKey`。找到在线解密网站进行解密（注意：这个网站只会将字母解密，因为符号和数字在密码加密前后不变，因此网站自动忽略了对它们的处理），用解密后的字母替换解密之前的字母得到真正的flag：

```
flag{2028ab39927df1d96e6a12b03e58002e}
```

原创。发于：http://www.zhouweitong.site/2019/04/07/016-xihuctf-misc2-writeup/