CTFHUB技能树-Misc-流量分析

原创

老大的豆豆酱 于 2020-08-17 18:37:57 发布 2135 收藏 6

分类专栏： CTF 文章标签： 安全

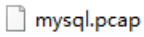
本文链接：https://blog.csdn.net/weixin_43486981/article/details/108058698

文章目录

数据库类流量

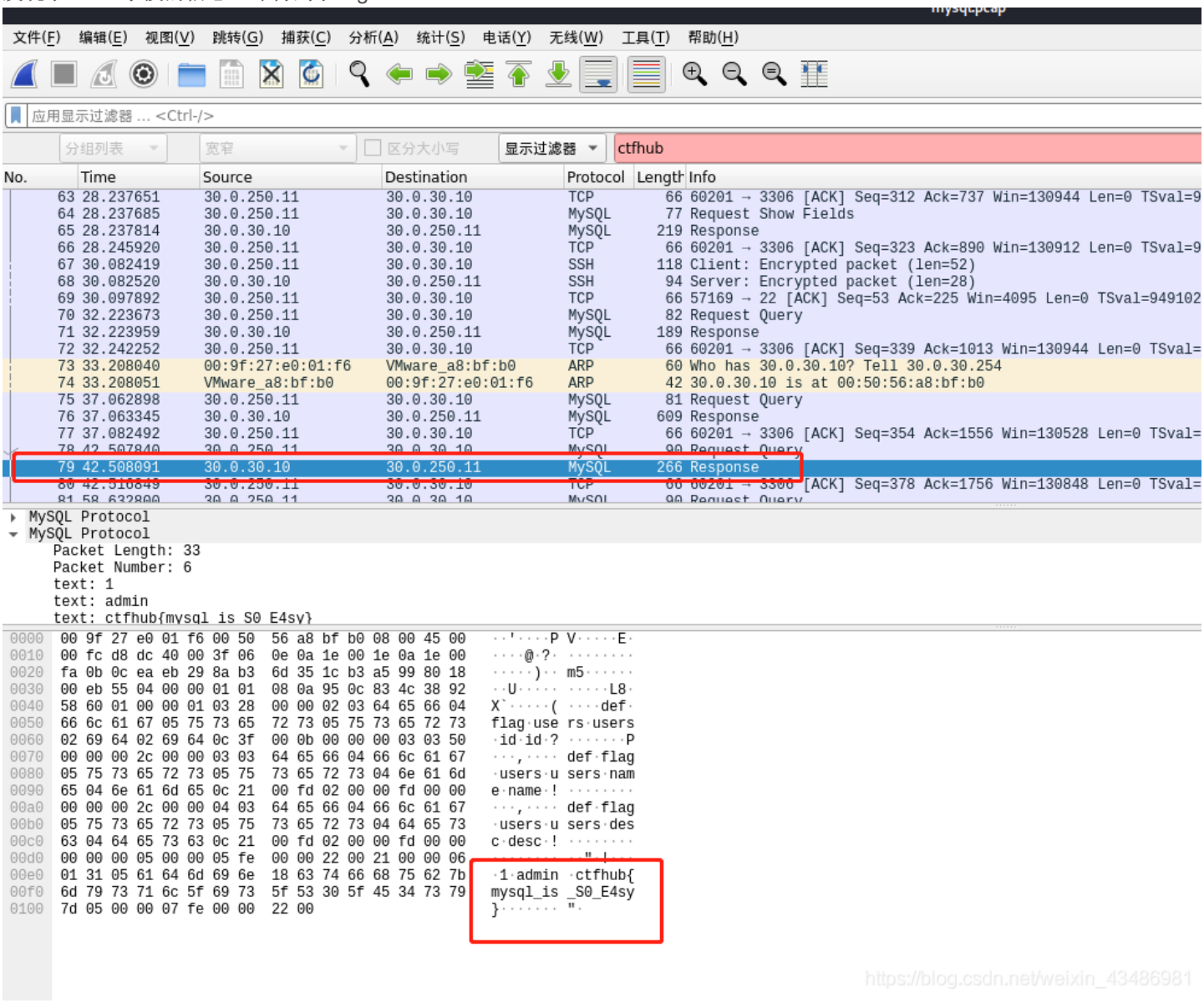
MySQL流量

下载题目附件，是一个`.pcap`文件



使用wireshark工具分析流量包

发现了ctfhub字段的信息，即得到了flag



（小白一个，一个包一个包找的，并不知道怎么过滤关键包……）

Redis流量

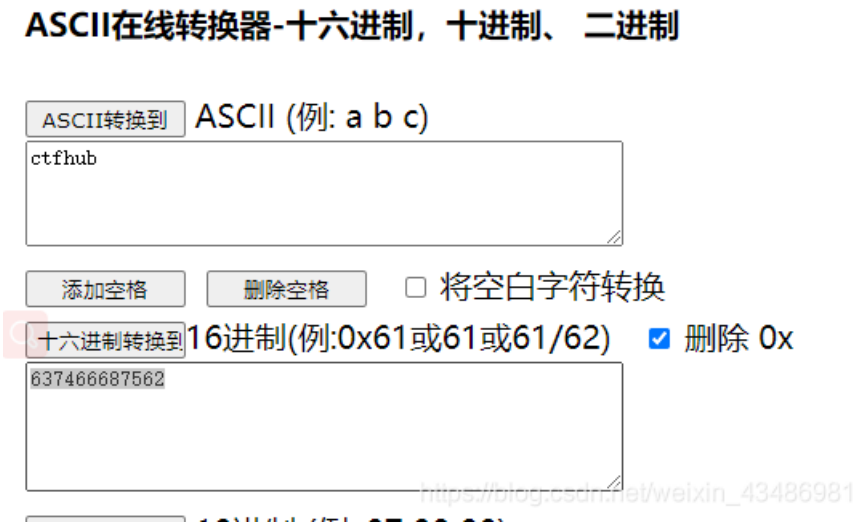
Redis:

- nosql数据库，**非关系型数据库**
- 支持5大数据类型 (字符串String，列表list、字典hash，集合set，zset)
  - 与之相似的有memcache，但memcache只支持string类型
- 单进程单线程，好处在于不用考虑并发

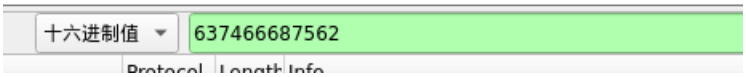
下载题目附件，打开是一个.pcap文件

使用**wireshark**工具分析流量包。

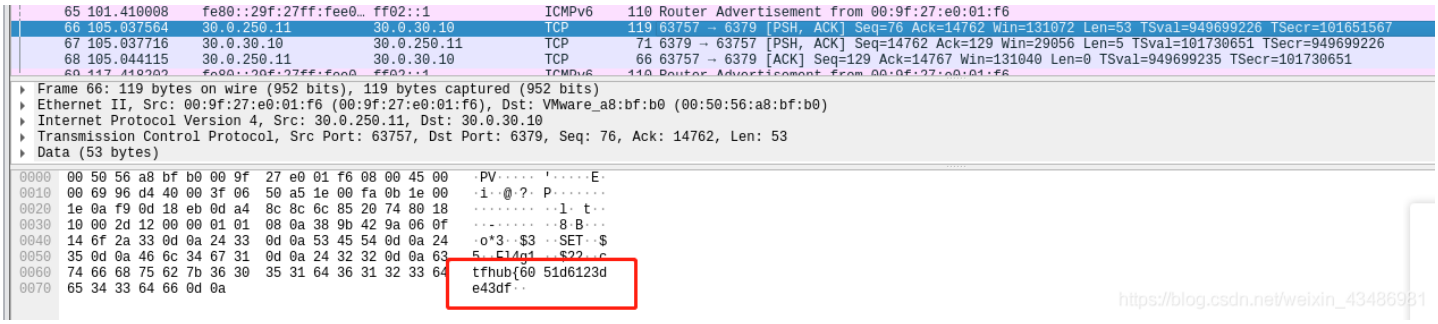
使用在线工具，**将ctfhub转换为十六进制数值**



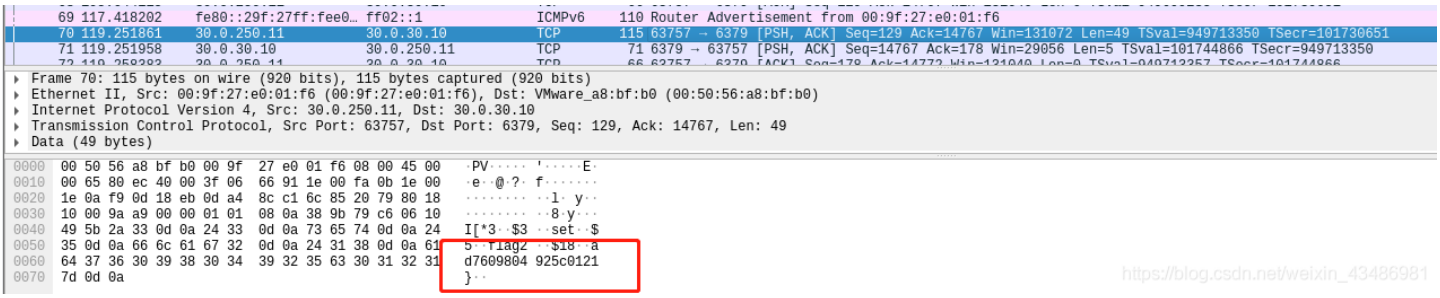
在wireshark中搜索该十六进制关键字



发现了包含ctfhub字段信息的数据包，第66个包发现只有部分的flag信息，并不完整。

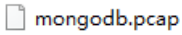


在66号包后面继续寻找，发现70号数据包出现了另一半flag信息

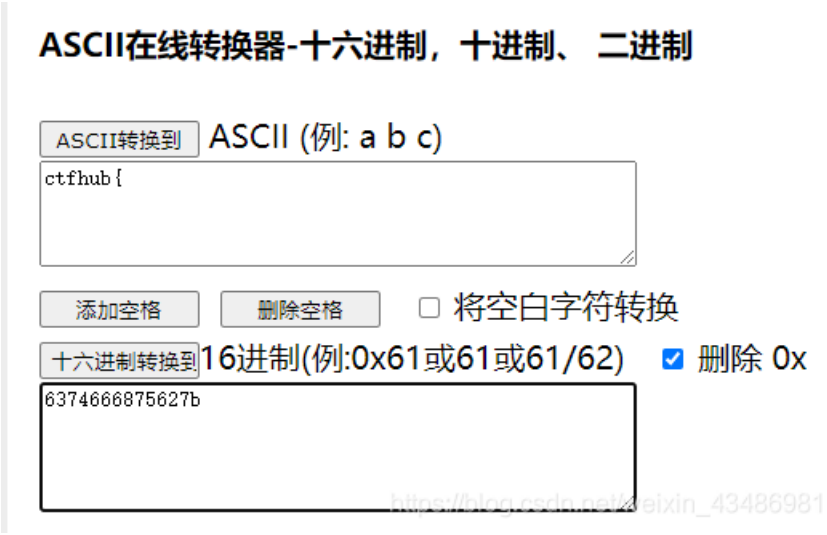


MongoDB流量

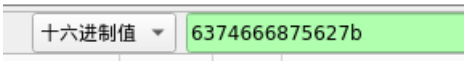
下载题目附件，打开是一个.pcap文件



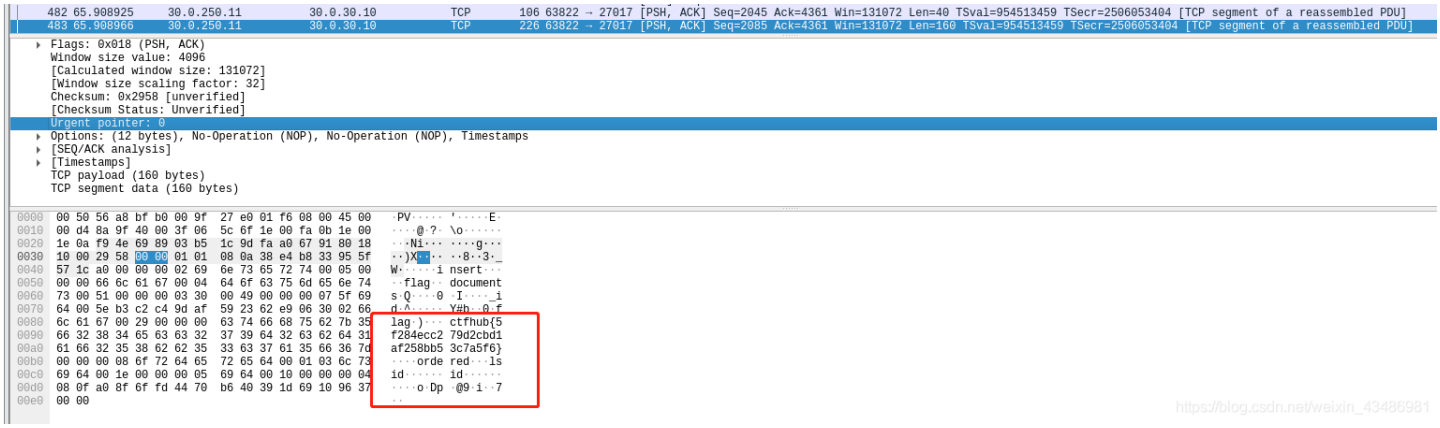
使用在线转换工具将ctfhub{转换为十六进制表示



在wireshark中搜索该十六进制



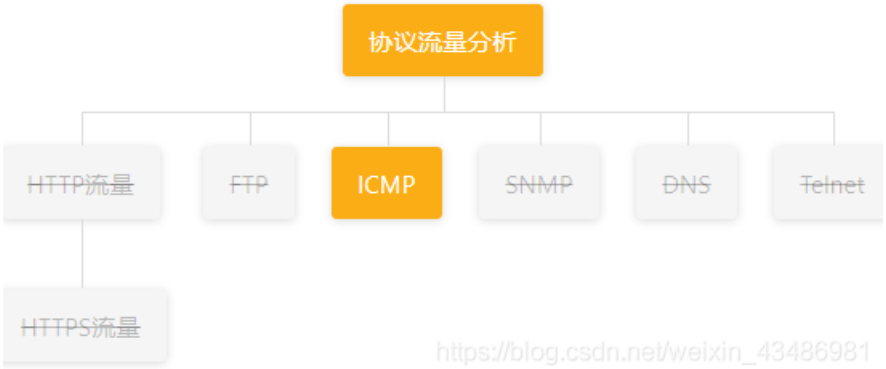
即可查找到包含关键字ctfhub{的数据包

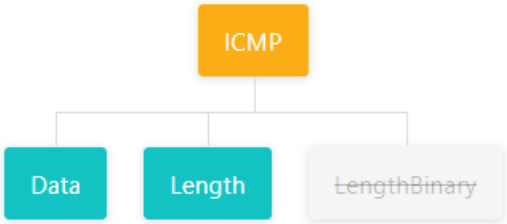


邮件流量

这部分并没有发布环境

协议流量分析

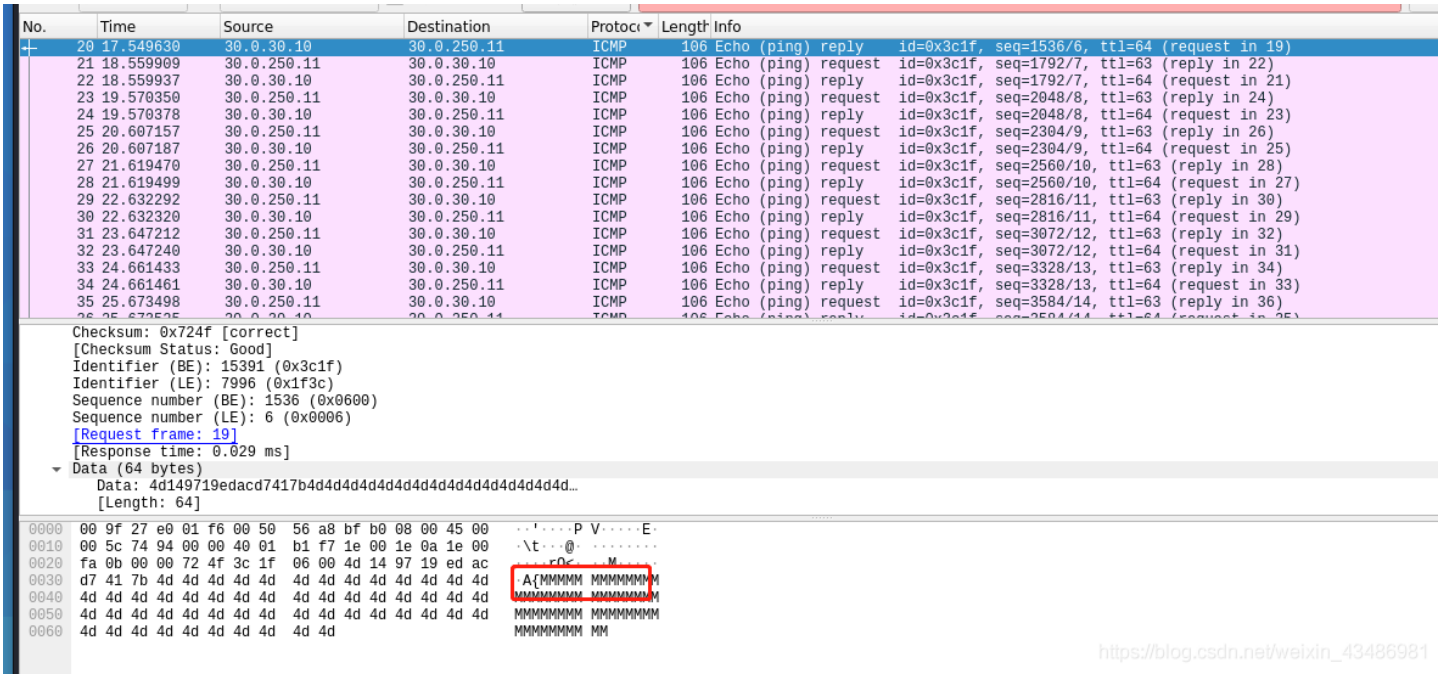




ICMP-Data

题目附件下载打开是一个.pcap包

使用wireshark打开，过滤ICMP包。观察数据包。

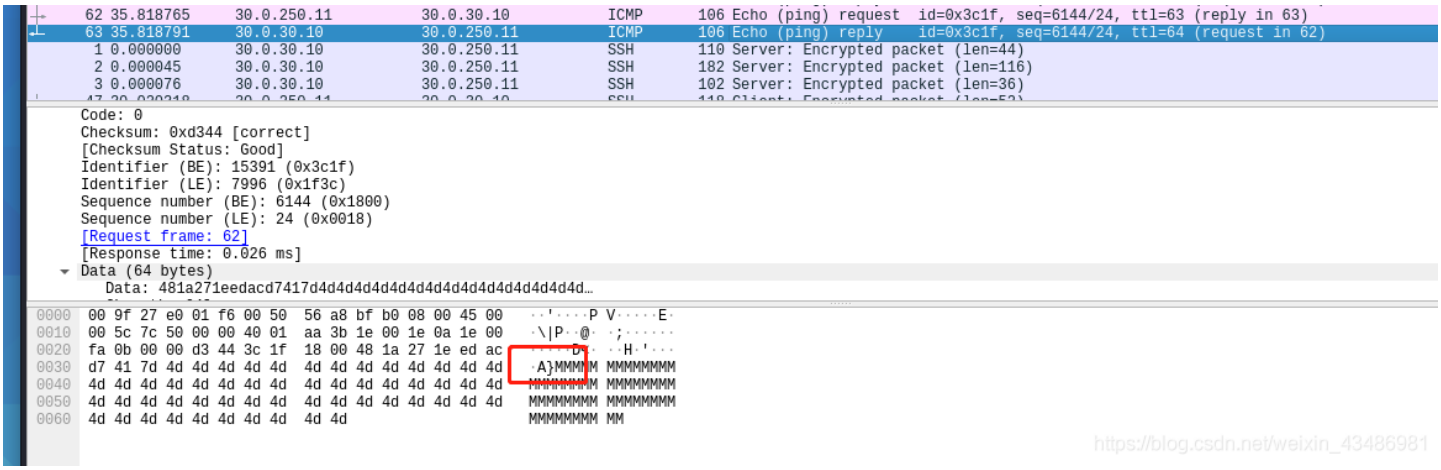


No.	Time	Source	Destination	Protoc	Length	Info
20	17.549630	30.0.30.10	30.0.250.11	ICMP	106	Echo (ping) reply id=0x3c1f, seq=1536/6, ttl=64 (request in 19)
21	18.559909	30.0.250.11	30.0.30.10	ICMP	106	Echo (ping) request id=0x3c1f, seq=1792/7, ttl=63 (reply in 22)
22	18.559937	30.0.30.10	30.0.250.11	ICMP	106	Echo (ping) reply id=0x3c1f, seq=1792/7, ttl=64 (request in 21)
23	19.570350	30.0.250.11	30.0.30.10	ICMP	106	Echo (ping) request id=0x3c1f, seq=2048/8, ttl=63 (reply in 24)
24	19.570378	30.0.30.10	30.0.250.11	ICMP	106	Echo (ping) reply id=0x3c1f, seq=2048/8, ttl=64 (request in 23)
25	20.607157	30.0.250.11	30.0.30.10	ICMP	106	Echo (ping) request id=0x3c1f, seq=2304/9, ttl=63 (reply in 26)
26	20.607187	30.0.30.10	30.0.250.11	ICMP	106	Echo (ping) reply id=0x3c1f, seq=2304/9, ttl=64 (request in 25)
27	21.619470	30.0.250.11	30.0.30.10	ICMP	106	Echo (ping) request id=0x3c1f, seq=2560/10, ttl=63 (reply in 28)
28	21.619499	30.0.30.10	30.0.250.11	ICMP	106	Echo (ping) reply id=0x3c1f, seq=2560/10, ttl=64 (request in 27)
29	22.632292	30.0.250.11	30.0.30.10	ICMP	106	Echo (ping) request id=0x3c1f, seq=2816/11, ttl=63 (reply in 30)
30	22.632320	30.0.30.10	30.0.250.11	ICMP	106	Echo (ping) reply id=0x3c1f, seq=2816/11, ttl=64 (request in 29)
31	23.647212	30.0.250.11	30.0.30.10	ICMP	106	Echo (ping) request id=0x3c1f, seq=3072/12, ttl=63 (reply in 32)
32	23.647240	30.0.30.10	30.0.250.11	ICMP	106	Echo (ping) reply id=0x3c1f, seq=3072/12, ttl=64 (request in 31)
33	24.661433	30.0.250.11	30.0.30.10	ICMP	106	Echo (ping) request id=0x3c1f, seq=3328/13, ttl=63 (reply in 34)
34	24.661461	30.0.30.10	30.0.250.11	ICMP	106	Echo (ping) reply id=0x3c1f, seq=3328/13, ttl=64 (request in 33)
35	25.673498	30.0.250.11	30.0.30.10	ICMP	106	Echo (ping) request id=0x3c1f, seq=3584/14, ttl=63 (reply in 36)

```
Checksum: 0x724f [correct]
[Checksum Status: Good]
Identifier (BE): 15391 (0x3c1f)
Identifier (LE): 7996 (0x1f3c)
Sequence number (BE): 1536 (0x0600)
Sequence number (LE): 6 (0x0006)
[Request frame: 19]
[Response time: 0.029 ms]
Data (64 bytes)
   Data: 4d149719edacd7417b4d4d4d4d4d4d4d4d4d4d4d4d4d4d4d…
   [Length: 64]
```

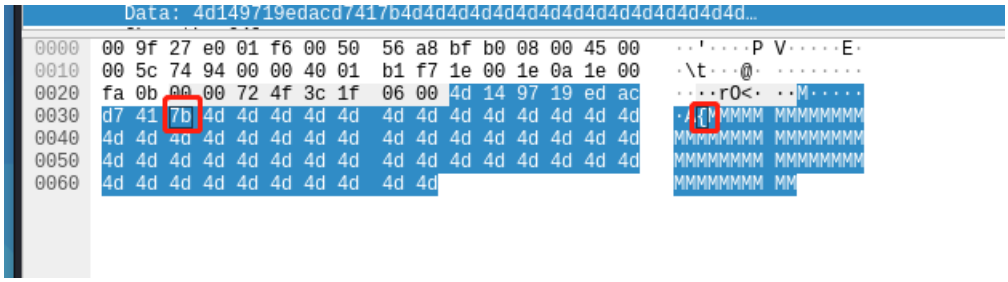
```
0000  00 9f 27 e0 01 f6 00 50  56 a8 bf b0 08 00 45 00   ··'····P V·····E·
0010  00 5c 74 94 00 00 40 01  b1 f7 1e 00 1e 0a 1e 00   ·\t···@· ········
0020  fa 0b 00 00 72 4f 3c 1f  06 00 4d 14 97 19 ed ac   ····rO<· ··M·····
0030  d7 41 7b 4d 4d 4d 4d 4d  4d 4d 4d 4d 4d 4d 4d 4d   ·A{MMMMM MMMMMMMM
0040  4d 4d 4d 4d 4d 4d 4d 4d  4d 4d 4d 4d 4d 4d 4d 4d   MMMMMMMM MMMMMMMM
0050  4d 4d 4d 4d 4d 4d 4d 4d  4d 4d 4d 4d 4d 4d 4d 4d   MMMMMMMM MMMMMMMM
0060  4d 4d 4d 4d 4d 4d 4d 4d  4d 4d                     MMMMMMMM MM
```

最后一个ICMP包中有}字段，

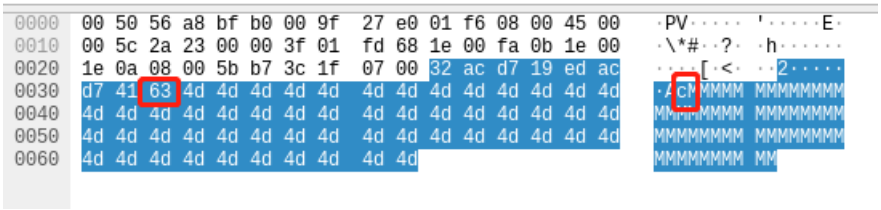


猜测{}与flag有关。猜测可能是”{“到”}“之间**相对应的位置的字母**可能就是flag。

相对应位置的意思是如图，{位置的值为flag信息



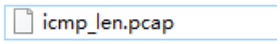
如下图的c



（注意ICMP包中有请求包和响应包，所以flag信息只需取同一个类型的包的信息就可以了）

ctfhub{c87eb99796406ac0b}

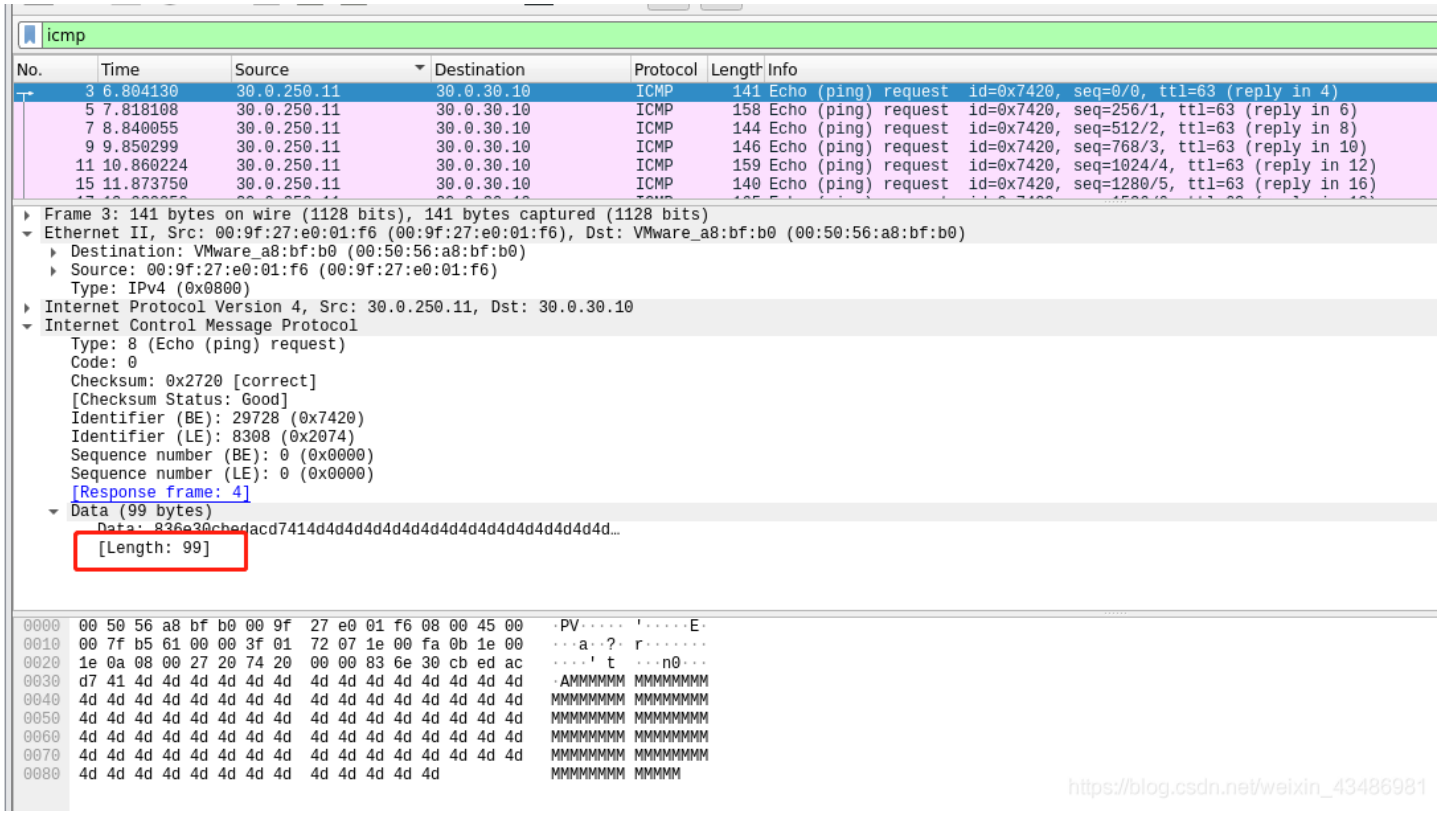
ICMP-Length



使用wireshark分析数据包

过滤ICMP包，分析……

根据题目提示，需要使用Length字段



将所有ICMP数据包的Length字段取出来，转换为ASCII码值，即可得到flag

99116102104117981239799985453571024850511125

ctfhub{acb659f023}

原始流量