

Buuctf 被偷走的文件

原创

Dexret 于 2021-11-22 19:55:31 发布 1083 收藏

分类专栏: [Buuctf Misc](#) 文章标签: [安全](#) [加密解密](#) [wireshark](#)

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/Dexret/article/details/121476617>

版权



[Buuctf Misc](#) 专栏收录该内容

47 篇文章 0 订阅

订阅专栏

下载该文件, 发现该文件为一个流量包文件

通过wireshark打开该文件

Wireshark interface showing a network capture. The packet list pane displays a list of packets, with packet 10 highlighted. The packet details pane shows the structure of the selected packet: Ethernet II, Internet Protocol, User Datagram Protocol, and Data (2 bytes). The packet bytes pane shows the raw data in hexadecimal and ASCII.

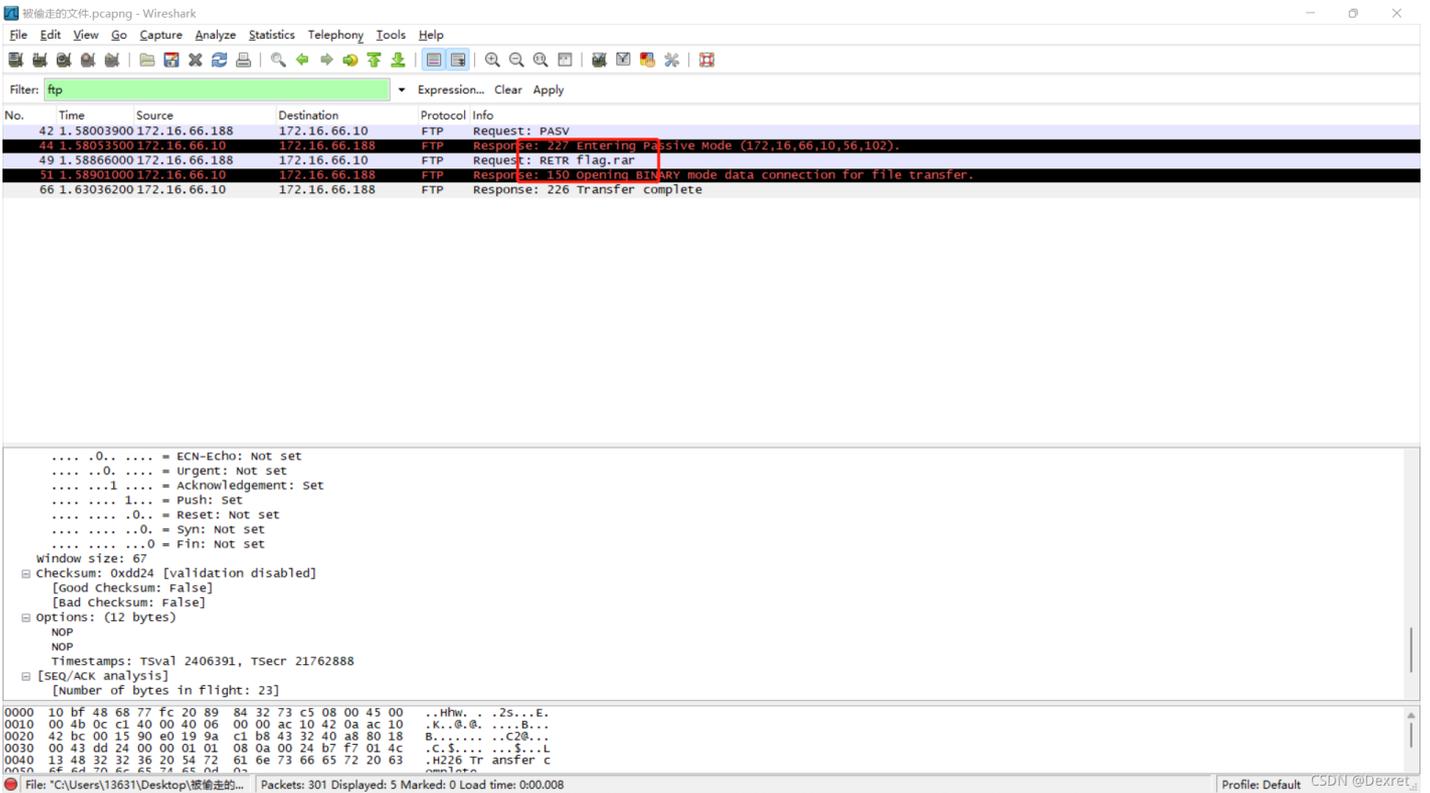
No.	Time	Source	Destination	Protocol	Info
1	0.00000000	172.16.66.10	218.81.111.242	UDP	Source port: 18961 Destination port: 20467
2	0.00013500	172.16.66.10	116.22.166.15	UDP	Source port: 18961 Destination port: hydap
3	0.00021300	172.16.66.10	62.111.134.76	eDonkey Kademlia	UDP: KADEMLIA2_BOOTSTRAP_REQ
4	0.00030500	172.16.66.10	125.40.138.211	eDonkey Kademlia	UDP: KADEMLIA2_BOOTSTRAP_REQ
5	0.00037700	172.16.66.10	24.732.106.25	UDP	Source port: 18961 Destination port: 15321
6	0.00046500	172.16.66.10	220.211.195.238	UDP	Source port: 18961 Destination port: 6267
7	0.00055300	172.16.66.10	222.125.241.60	UDP	Source port: 18961 Destination port: 9758
8	0.00062400	172.16.66.10	59.175.114.90	eDonkey Kademlia	UDP: KADEMLIA2_BOOTSTRAP_REQ
9	0.00457400	172.16.66.10	115.200.232.1	ICMP	Echo (ping) request (id=0x0100, seq(be/le)=23285/62810, ttl=64)
10	0.00742100	115.200.232.1	172.16.66.10	ICMP	Echo (ping) reply (id=0x0100, seq(be/le)=23285/62810, ttl=250)
11	0.07776700	172.16.66.10	114.112.93.51	TCP	14437 > http [SYN] seq=0 win=8192 Len=0 MSS=1460 WS=2 SACK_PERM=1
12	0.59350000	172.16.66.10	192.168.31.1	DNS	Standard query PTR 10.66.16.172.in-addr.arpa
13	0.59909300	172.16.66.10	192.168.31.1	DNS	Standard query PTR 242.111.81.218.in-addr.arpa
14	0.60466800	172.16.66.10	192.168.31.1	DNS	Standard query PTR 15.166.22.116.in-addr.arpa
15	0.61020000	172.16.66.10	192.168.31.1	DNS	Standard query PTR 76.134.111.62.in-addr.arpa
16	0.61572700	172.16.66.10	192.168.31.1	DNS	Standard query PTR 211.138.40.125.in-addr.arpa
17	0.62128300	172.16.66.10	192.168.31.1	DNS	Standard query PTR 235.106.232.24.in-addr.arpa
18	0.62681300	172.16.66.10	192.168.31.1	DNS	Standard query PTR 238.195.211.220.in-addr.arpa
19	0.63234100	172.16.66.10	192.168.31.1	DNS	Standard query PTR 60.241.125.222.in-addr.arpa
20	0.63791600	172.16.66.10	192.168.31.1	DNS	Standard query PTR 90.114.175.59.in-addr.arpa
21	0.64345000	172.16.66.10	192.168.31.1	DNS	Standard query PTR 1.232.200.115.in-addr.arpa

Frame 1: 44 bytes on wire (352 bits), 44 bytes captured (352 bits)
Ethernet II, Src: 20:89:84:32:73:c5 (20:89:84:32:73:c5), Dst: 0c:da:41:9e:cc:85 (0c:da:41:9e:cc:85)
Internet Protocol, Src: 172.16.66.10 (172.16.66.10), Dst: 218.81.111.242 (218.81.111.242)
User Datagram Protocol, Src Port: 18961 (18961), Dst Port: 20467 (20467)
Data (2 bytes)

```
0000 0c da 41 9e cc 85 20 89 84 32 73 c5 08 00 45 00  ..A... ..2s...E.  
0010 00 1e 0c 91 00 00 40 11 00 00 ac 10 42 0a da 51  ....@. ....B..Q  
0020 6f f2 4a 11 4f f3 00 0a 38 7a e4 01          o..O... 8z..
```

通过题意得知该题需要我们分析被盗走的文件

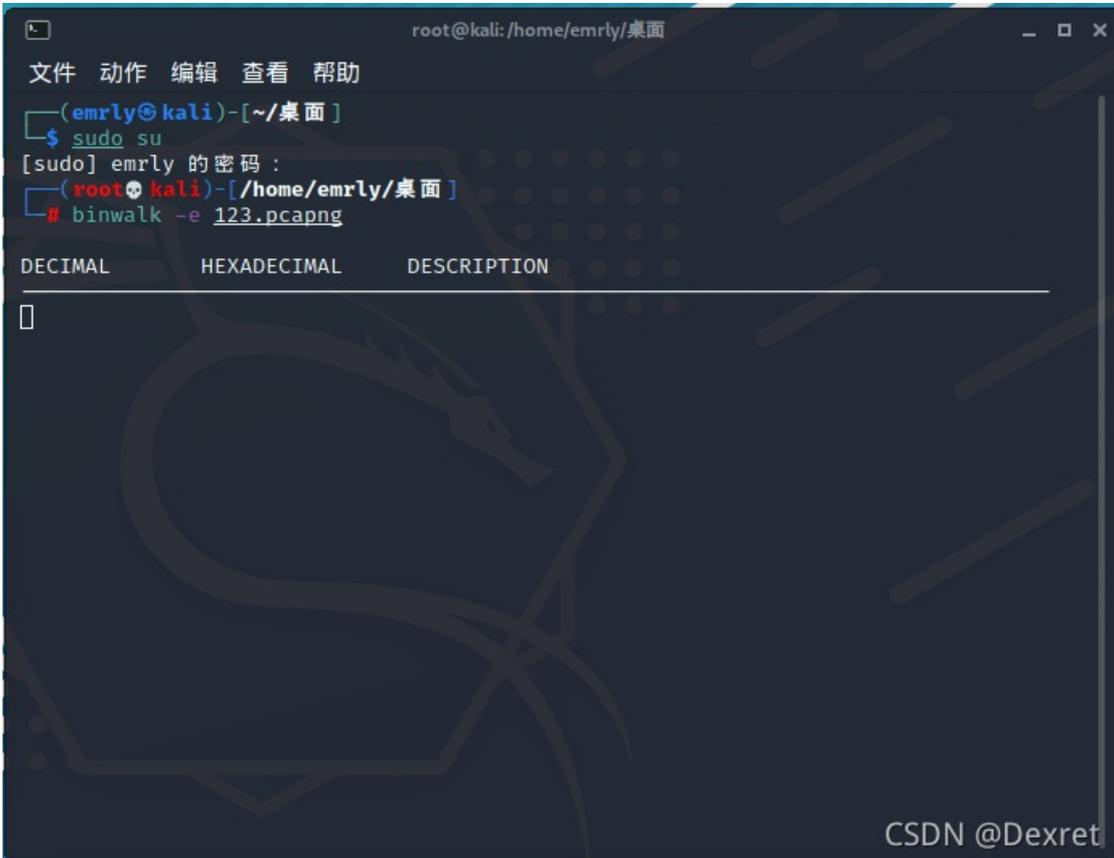
在该数据包中看到有ftp传输, 过滤ftp数据包



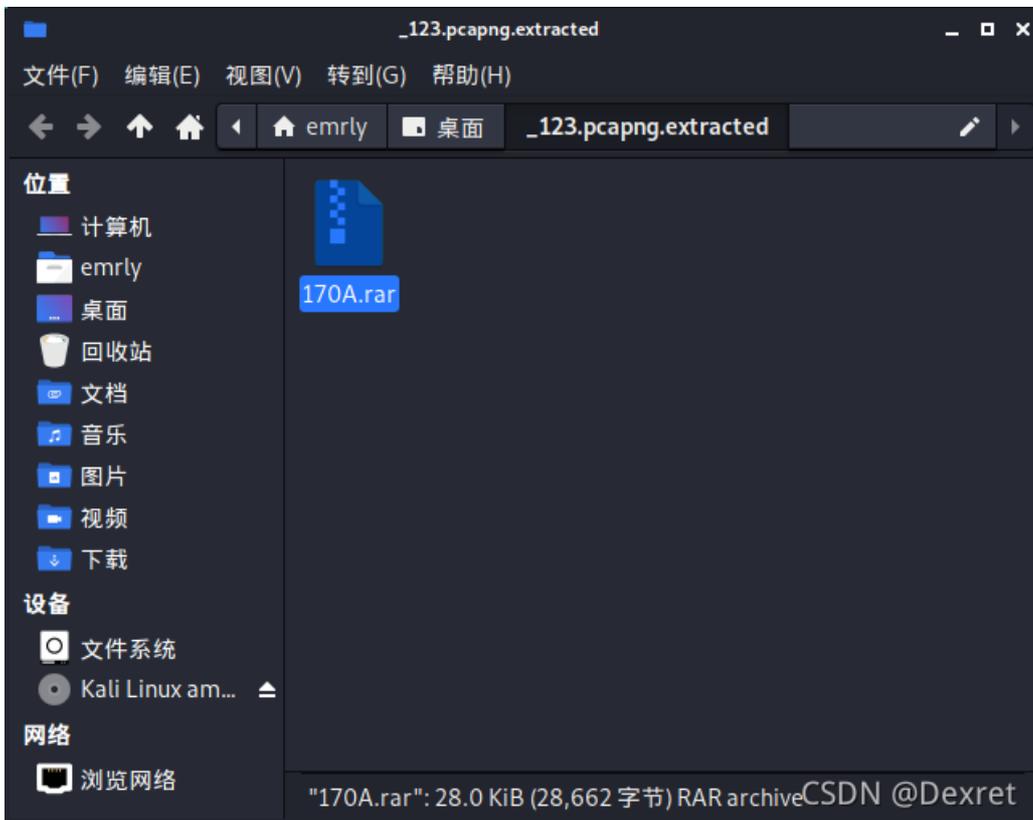
可以看到ftp传输有个flag.rar文件

通过kail中的binwalk对该文件进行分离(这里改了下名称，容易分离出来)

```
binwalk -e 该文件名称
```



分离得到一个压缩包文件



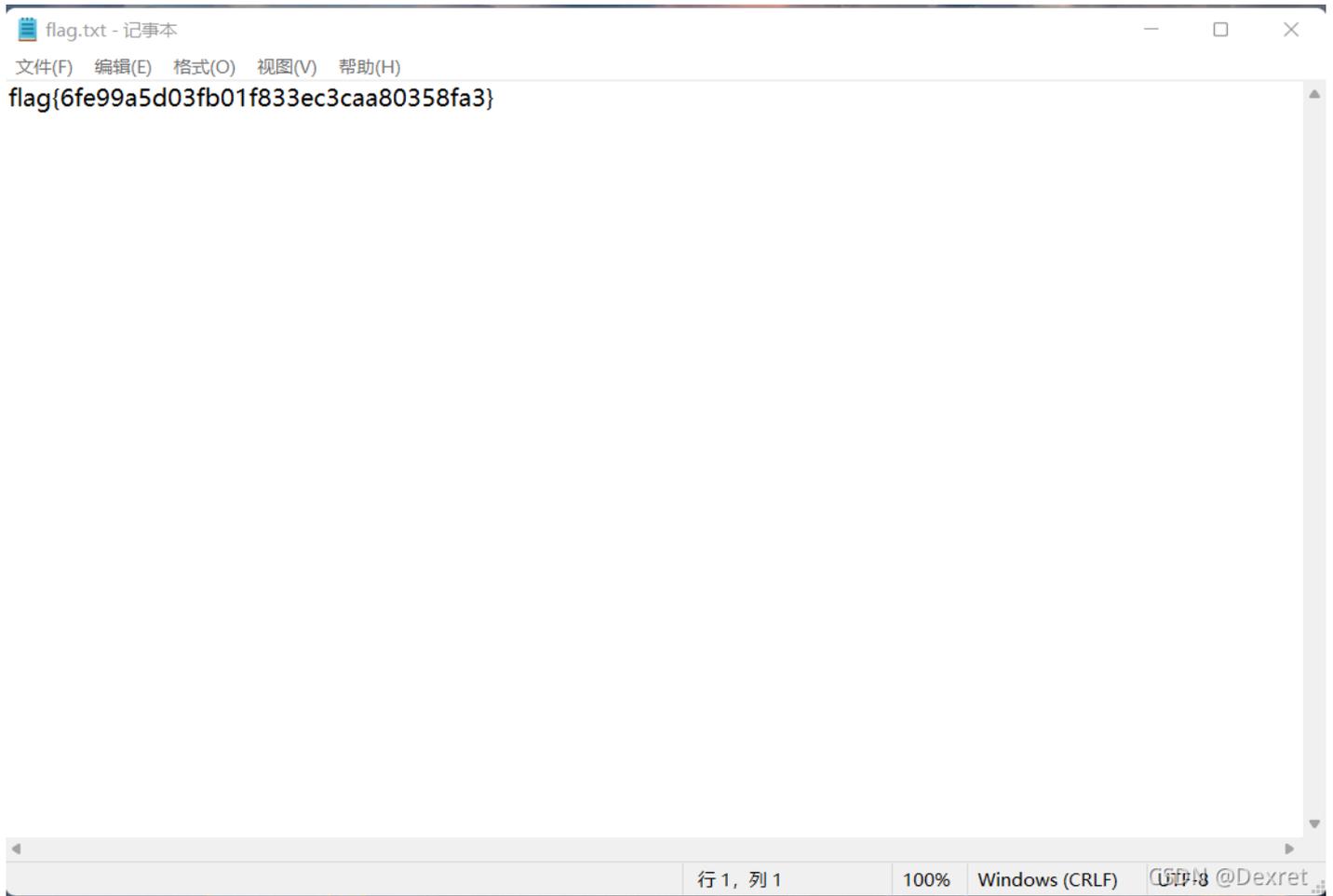
打开压缩包文件发现该压缩包文件有密码

利用ARCHPR对该文件进行密码爆破



最终得到该压缩包密码为: 5790

通过该密码成功解压压缩包, 发现里面有个flag.txt文件, 将其打开



成功获取到该题的flag:

```
flag{6fe99a5d03fb01f833ec3caa80358fa3}
```