一、前言

capinfos是Wireshark默认配套安装的命令行工具之一，从其命名来看也能顾名思义，主要用于显示抓包文件的信息，如文件格式、数据包数量、时间范围（首尾包）、数据包类型等。

使用场景大致为以下几种：

- 检查抓包文件的基本信息：前面说过，用于查看抓包文件的格式、数据包数量、时间范围、数据包类型等基本信息，便于了解抓包文件的内容和特征；
- 检查抓包文件的完整性：检查抓包文件是否完整，是否存在数据丢失或损坏的情况；
- 检查抓包文件的时间范围：查看抓包文件中数据包的时间范围，以便于了解抓包文件中数据包的时间分布情况，利于快速判断抓包文件时间范围是否已经覆盖故障出现时间；
- 检查抓包文件的数据包类型：查看抓包文件中数据包的类型，了解抓包文件中数据包的协议分布情况；
- 检查抓包文件的过滤器：检查抓包文件中是否存在过滤器，了解抓包文件中数据包的过滤情况。

本文将详细介绍capinfos的用法案例。

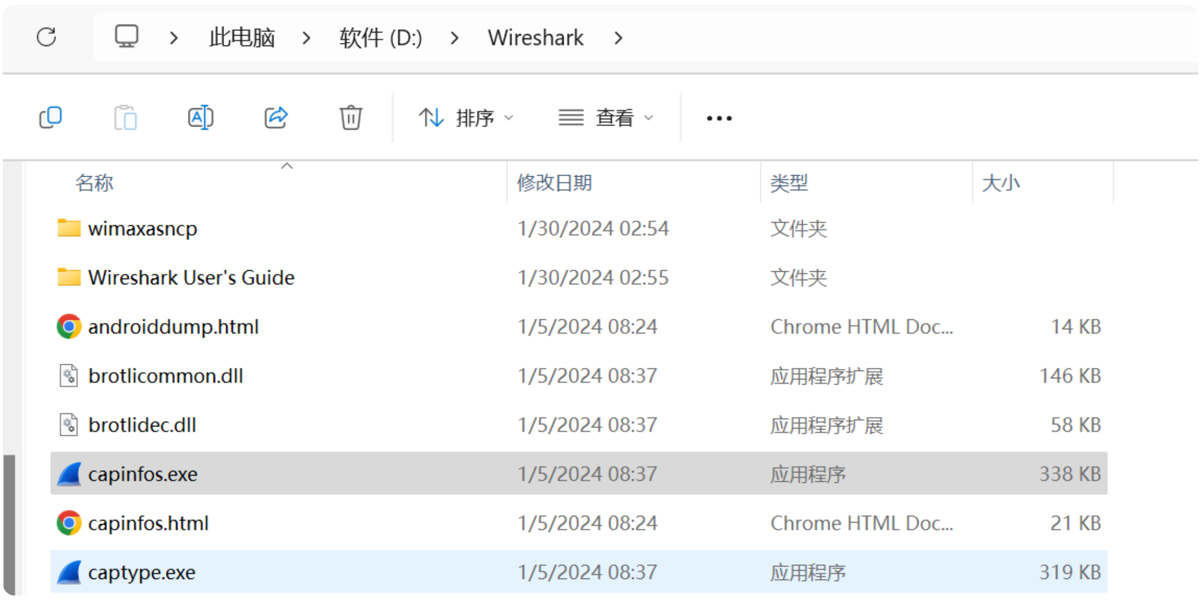
二、安装

Linux

发行版	**安装命令**
Archlinux	pacman -Sy wireshark-cli
CentOS/Redhat	yum install -y wireshark
Debian/Ubuntu	apt install -y wireshark
Gentoo	emerge --ask wireshark

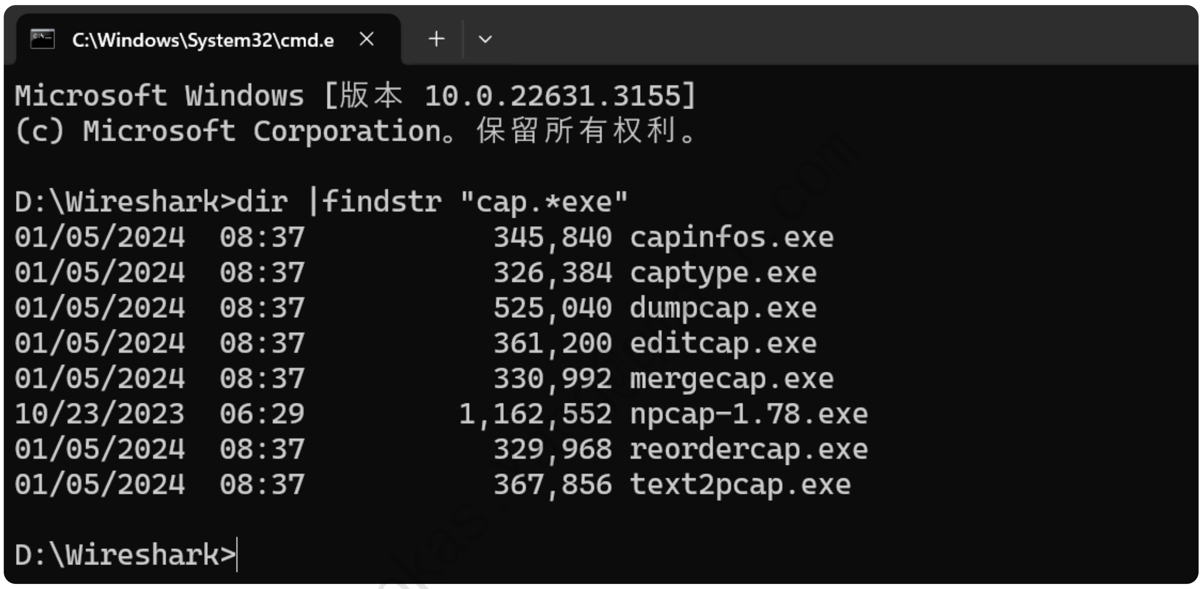
Windows

安装wireshark后，**capinfos**默认在wireshark安装路径：


此电脑 > 软件 (D:) > Wireshark >

名称	修改日期	类型	大小
wimaxasncp	1/30/2024 02:54	文件夹	
Wireshark User's Guide	1/30/2024 02:55	文件夹	
androiddump.html	1/5/2024 08:24	Chrome HTML Doc...	14 KB
brotlicommon.dll	1/5/2024 08:37	应用程序扩展	146 KB
brotlidec.dll	1/5/2024 08:37	应用程序扩展	58 KB
capinfos.exe	1/5/2024 08:37	应用程序	338 KB
capinfos.html	1/5/2024 08:24	Chrome HTML Doc...	21 KB
captype.exe	1/5/2024 08:37	应用程序	319 KB

其它配套命令也都在这个路径下：



```
Microsoft Windows [版本 10.0.22631.3155]
(c) Microsoft Corporation。保留所有权利。

D:\Wireshark>dir |findstr "cap.*exe"
01/05/2024  08:37           345,840 capinfos.exe
01/05/2024  08:37           326,384 captype.exe
01/05/2024  08:37           525,040 dumpcap.exe
01/05/2024  08:37           361,200 editcap.exe
01/05/2024  08:37           330,992 mergecap.exe
10/23/2023  06:29         1,162,552 npcap-1.78.exe
01/05/2024  08:37           329,968 reordercap.exe
01/05/2024  08:37           367,856 text2pcap.exe

D:\Wireshark>
```

添加路径到环境变量还是直接在路径下使用，可自行选择。

MacOS

前提：需要安装homebrew

使用homebrew安装wireshark，默认也会将**capinfos**安装上去：

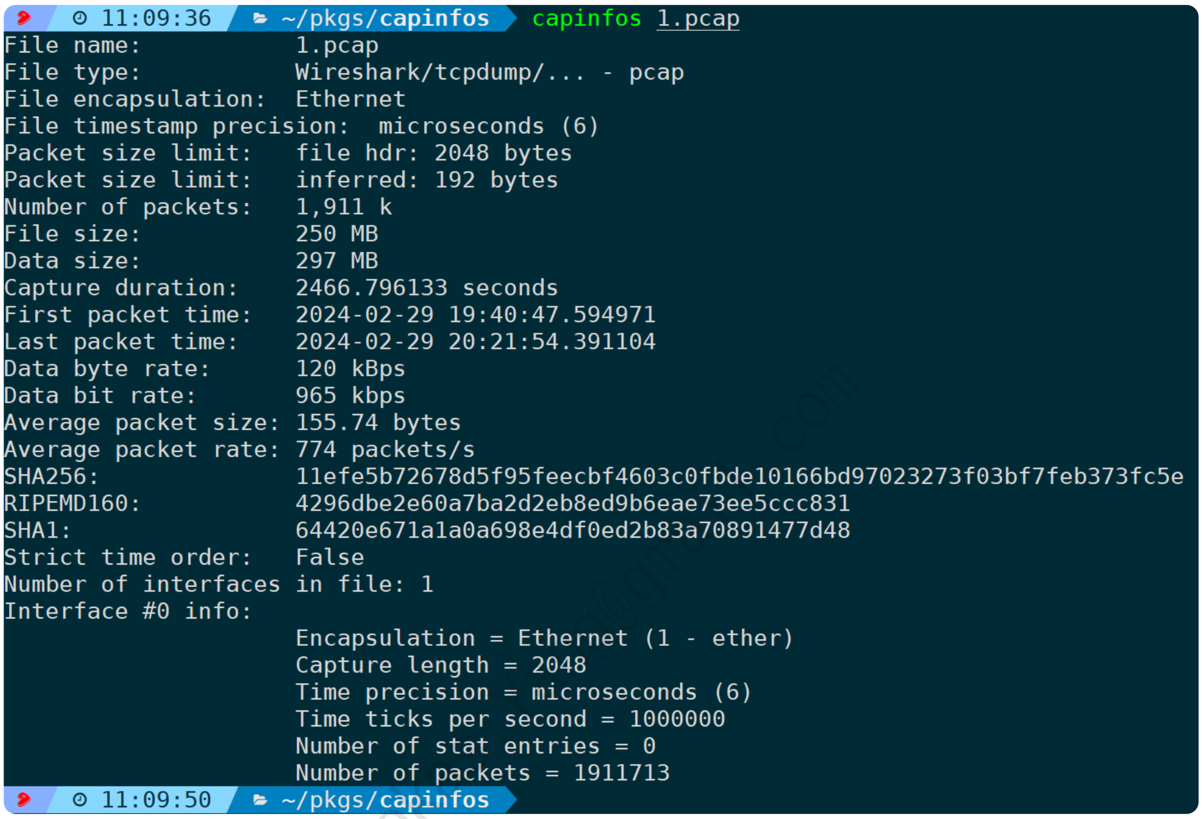
```
brew install wireshark
brew install wireshark-chmodbpf
```

三、用法案例分析

0.输出所有信息

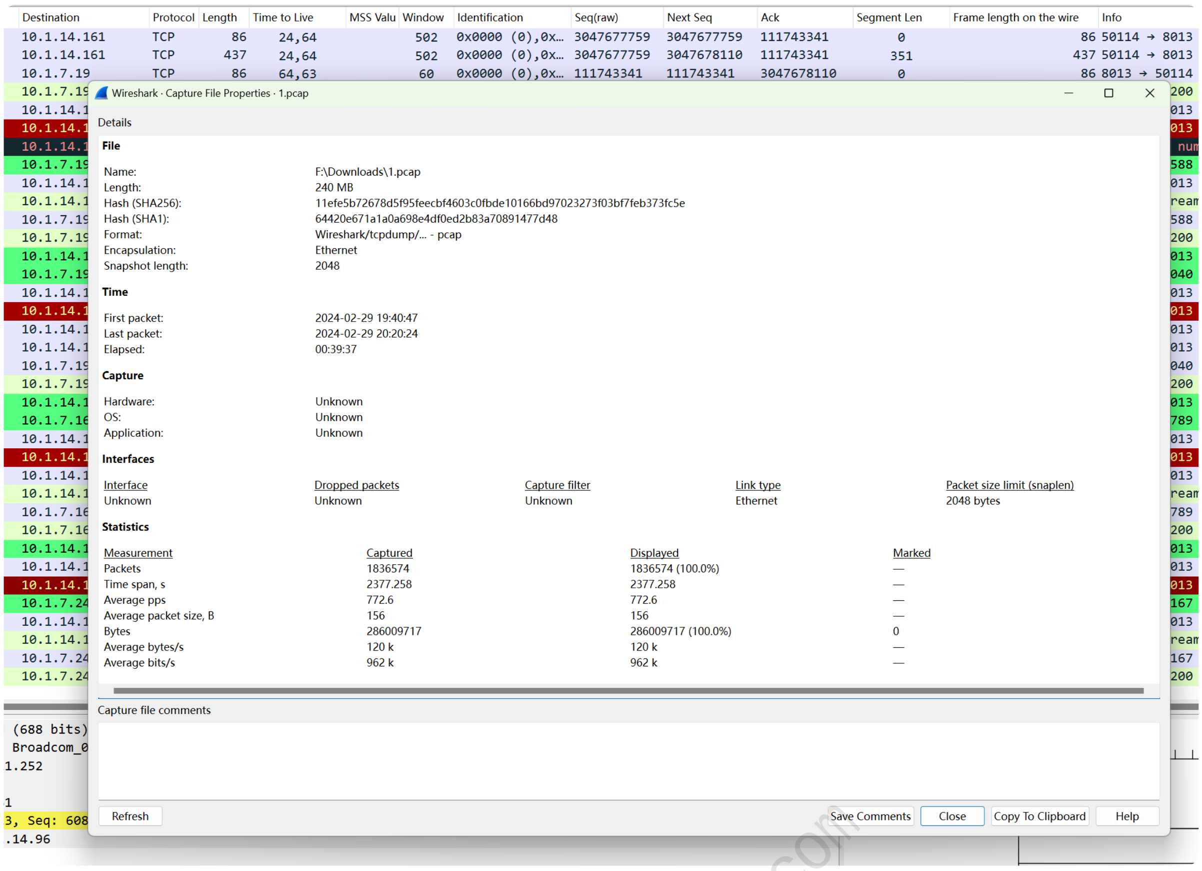
不接任何参数的情况下默认会引用-A参数，输出所有信息字段。

```
capinfos <文件名>
```



```
 ⊙ 11:09:36  ~/pkgs/capinfos  capinfos 1.pcap
File name:           1.pcap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 2048 bytes
Packet size limit:   inferred: 192 bytes
Number of packets:   1,911 k
File size:           250 MB
Data size:           297 MB
Capture duration:    2466.796133 seconds
First packet time:   2024-02-29 19:40:47.594971
Last packet time:    2024-02-29 20:21:54.391104
Data byte rate:      120 kBps
Data bit rate:       965 kbps
Average packet size: 155.74 bytes
Average packet rate: 774 packets/s
SHA256:              11efe5b72678d5f95feecbf4603c0fbde10166bd97023273f03bf7feb373fc5e
RIPEMD160:           4296dbe2e60a7ba2d2eb8ed9b6eae73ee5ccc831
SHA1:                64420e671a1a0a698e4df0ed2b83a70891477d48
Strict time order:   False
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 2048
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 1911713
 ⊙ 11:09:50  ~/pkgs/capinfos
```

这些信息在Wireshark的**统计（Statistics）** --> **捕获文件属性（Capture File Properties）**也有同样的输出：



```
Destination   Protocol Length Time to Live MSS Valu Window Identification  Seq(raw)   Next Seq   Ack        Segment Len Frame length on the wire Info
10.1.14.161   TCP          86      24,64               502 0x0000 (0),0x… 3047677759 3047677759 111743341            0                      86 50114 → 8013
10.1.14.161   TCP         437      24,64               502 0x0000 (0),0x… 3047677759 3047678110 111743341          351                     437 50114 → 8013
10.1.7.19     TCP          86      64,63                60 0x0000 (0),0x… 111743341  111743341  3047678110           0                      86 8013 → 50114
```

Wireshark · Capture File Properties · 1.pcap

Details

File

Name:	F:\Downloads\1.pcap
Length:	240 MB
Hash (SHA256):	11efe5b72678d5f95feecbf4603c0fbde10166bd97023273f03bf7feb373fc5e
Hash (SHA1):	64420e671a1a0a698e4df0ed2b83a70891477d48
Format:	Wireshark/tcpdump/... - pcap
Encapsulation:	Ethernet
Snapshot length:	2048

Time

First packet:	2024-02-29 19:40:47
Last packet:	2024-02-29 20:20:24
Elapsed:	00:39:37

Capture

Hardware:	Unknown
OS:	Unknown
Application:	Unknown

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Unknown	Unknown	Unknown	Ethernet	2048 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	1836574	1836574 (100.0%)	—
Time span, s	2377.258	2377.258	—
Average pps	772.6	772.6	—
Average packet size, B	156	156	—
Bytes	286009717	286009717 (100.0%)	0
Average bytes/s	120 k	120 k	—
Average bits/s	962 k	962 k	—

Capture file comments

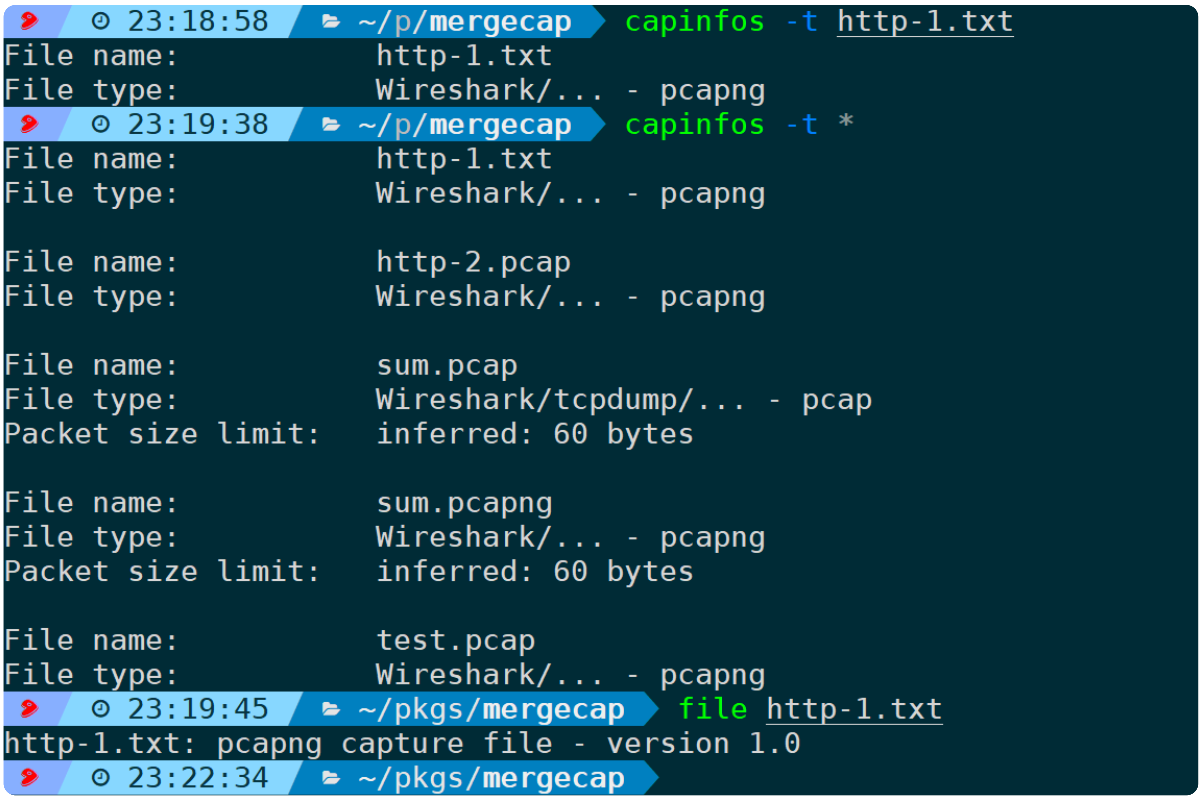
每个字段代表什么含义实际已经写的很清晰了，没有精准过滤的需求其实已经满足你的需求了，需要更精细化控制和更多拓展用法，则继续阅读下文。

1.通用选项

1）显示文件类型（-t）

-t显示抓包文件的格式类型，文件后缀不一定和实际保存时的文件格式类型完全一致，后缀是可以通过修改文件名后缀来任意进行修改的，-t参数则分析实际的文件注入格式，而不是通过分析文件后缀：

```
capinfos -t <文件名>
```



```
 23:18:58  ~/p/mergecap  capinfos -t http-1.txt
File name:           http-1.txt
File type:           Wireshark/... - pcapng
 23:19:38  ~/p/mergecap  capinfos -t *
File name:           http-1.txt
File type:           Wireshark/... - pcapng

File name:           http-2.pcap
File type:           Wireshark/... - pcapng

File name:           sum.pcap
File type:           Wireshark/tcpdump/... - pcap
Packet size limit:   inferred: 60 bytes

File name:           sum.pcapng
File type:           Wireshark/... - pcapng
Packet size limit:   inferred: 60 bytes

File name:           test.pcap
File type:           Wireshark/... - pcapng
 23:19:45  ~/pkgs/mergecap  file http-1.txt
http-1.txt: pcapng capture file - version 1.0
 23:22:34  ~/pkgs/mergecap
```

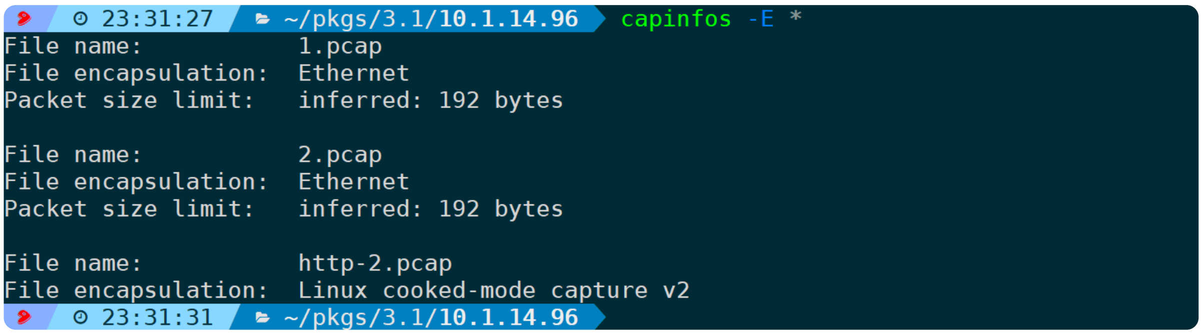
比如上面这个示例，文件http-1.txt以txt结尾的后缀，实际文件格式为pcapng，file命令也能查看文件存储使用的格式；同时，使用通配符*则匹配当前目录下的所有文件，其中sum.pcap、sum.pcapng两个文件多出了一行：**Packet size limit: inferred: 60bytes**，这一行是包文件中数据帧的推断长度（inferred），这两个文件实际是通过**mergecap -s 60**来截断后合并保存的。

2）显示数据链路层协议封装类型（-E）

此参数将显示数据链路层使用的封装协议，通常情况下都是以太网（Ethernet），也可能会出现Linux cooked-mode capture，至于Linux cooked-mode capture是什么，可以参考笔者写的**这篇文章**。简单来讲，它是虚拟协议，在Linux抓包时指定抓包设备为所有时（-i any）可能会出现的情况。

比如下面的案例：

```
capinfos -E <文件名>
```



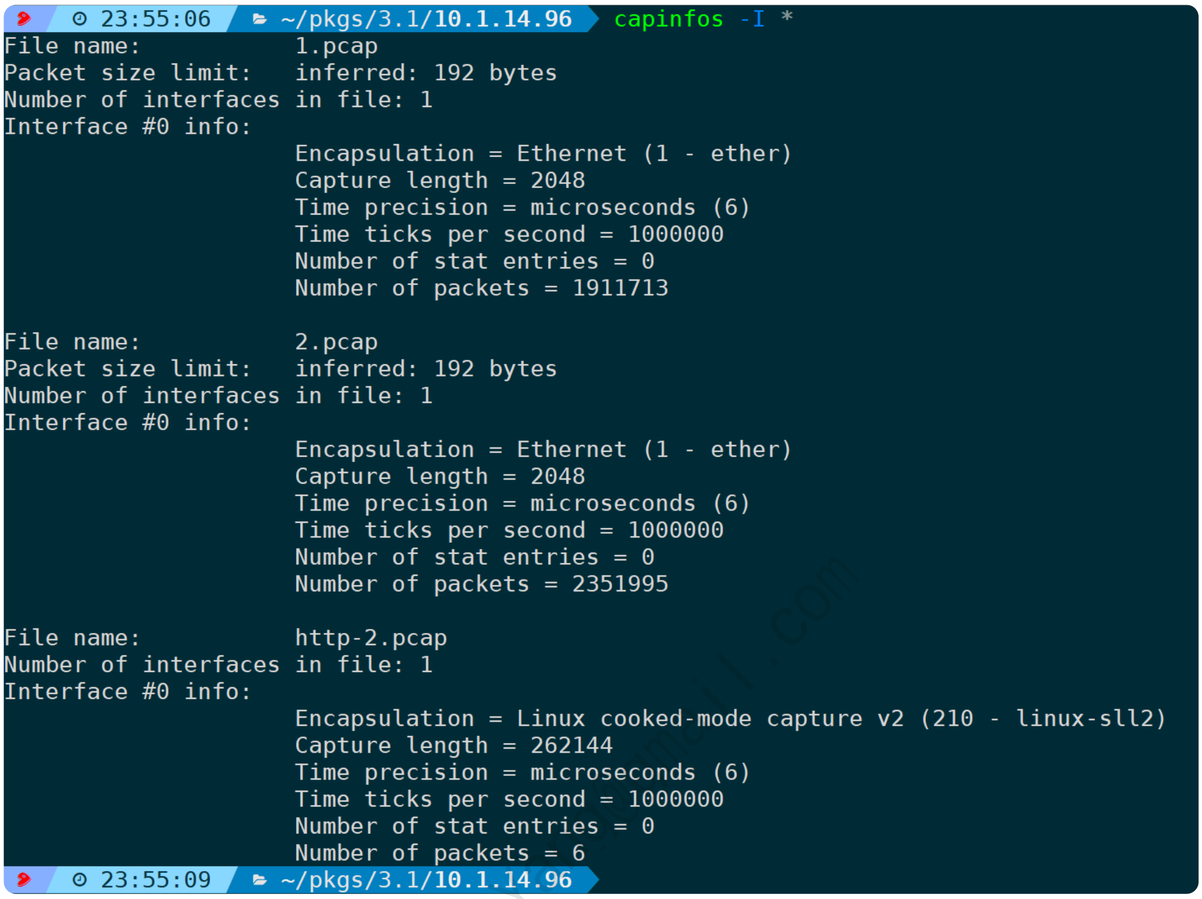
```
 23:31:27  ~/pkgs/3.1/10.1.14.96  capinfos -E *
File name:             1.pcap
File encapsulation:    Ethernet
Packet size limit:     inferred: 192 bytes

File name:             2.pcap
File encapsulation:    Ethernet
Packet size limit:     inferred: 192 bytes

File name:             http-2.pcap
File encapsulation:    Linux cooked-mode capture v2
 23:31:31  ~/pkgs/3.1/10.1.14.96
```

1.pcap、2.pcap的链路层协议均为以太网，且包文件中数据帧的推断长度（inferred）大小为192字节，http-2.pcap的链路层协议为Linux cooked-mode capture，因为这个包是通过`tcpdump -i any`来捕获保存的。

3）显示包文件接口信息、链路层协议（-I）

-I选项可以帮助了解抓包文件中的数据包来源，譬如网络接口、链路层协议等：

```
capinfos -I <文件名>
```



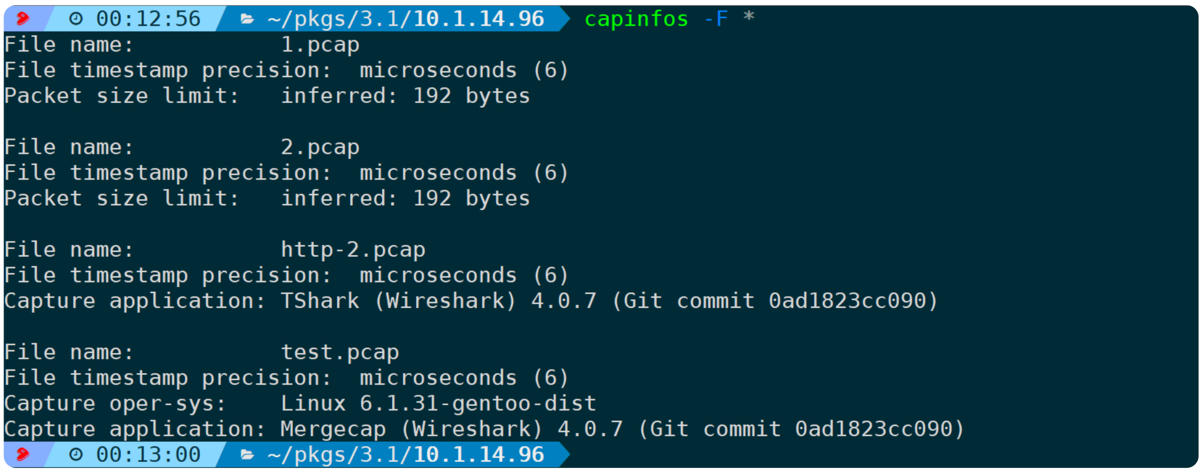
```
 ⊙ 23:55:06  ~/pkgs/3.1/10.1.14.96  capinfos -I *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 2048
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 1911713

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 2048
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 2351995

File name:           http-2.pcap
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Linux cooked-mode capture v2 (210 - linux-sll2)
                     Capture length = 262144
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 6
 ⊙ 23:55:09  ~/pkgs/3.1/10.1.14.96
```

同时还显示了总包量、时间精度、捕获长度等详细信息。

4）显示包文件的附加信息（-F）

这个选项会尽可能显示能识别到的抓包文件的额外信息，比如时间精度、包文件中每个数据帧的推断长度（inferred）、抓包时使用的抓包程序版本、使用的操作系统：

```
capinfos -F <文件名>
```

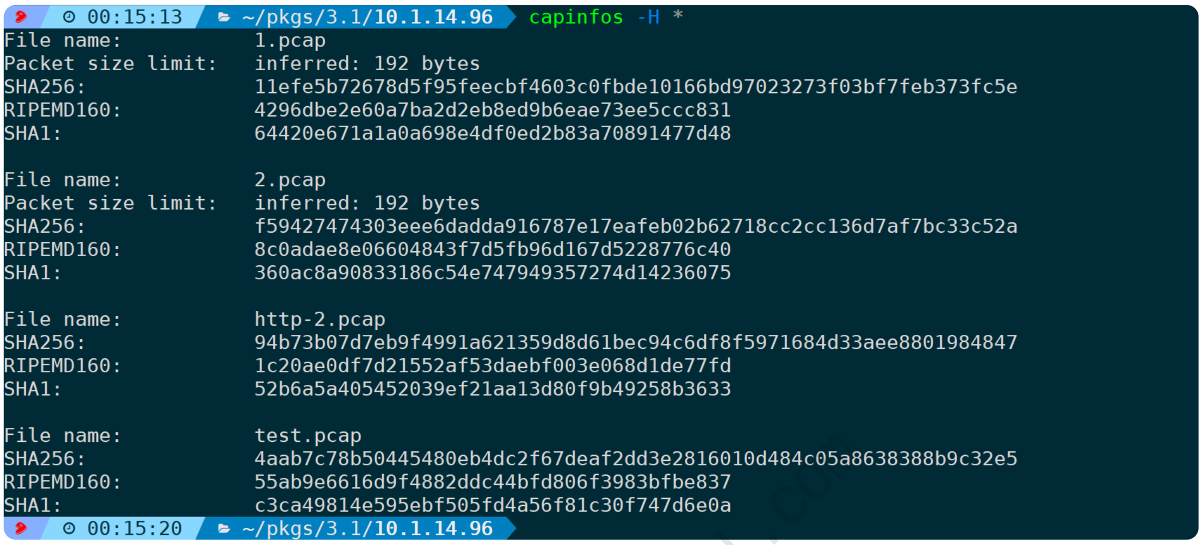


```
 ⊙ 00:12:56  ~/pkgs/3.1/10.1.14.96  capinfos -F *
File name:           1.pcap
File timestamp precision:  microseconds (6)
Packet size limit:   inferred: 192 bytes

File name:           2.pcap
File timestamp precision:  microseconds (6)
Packet size limit:   inferred: 192 bytes

File name:           http-2.pcap
File timestamp precision:  microseconds (6)
Capture application: TShark (Wireshark) 4.0.7 (Git commit 0ad1823cc090)

File name:           test.pcap
File timestamp precision:  microseconds (6)
Capture oper-sys:    Linux 6.1.31-gentoo-dist
Capture application: Mergecap (Wireshark) 4.0.7 (Git commit 0ad1823cc090)
 ⊙ 00:13:00  ~/pkgs/3.1/10.1.14.96
```

5）显示文件的SHA256、RIPEMD160和SHA1散列（-H）

这个参数相当于把sha256sum、sha1sum、ripemd160等用来计算文件hash值的工具合并输出了，有利于校验文件一致性，避免抓包文件被篡改的情况：

```
capinfos -H <包文件>
```



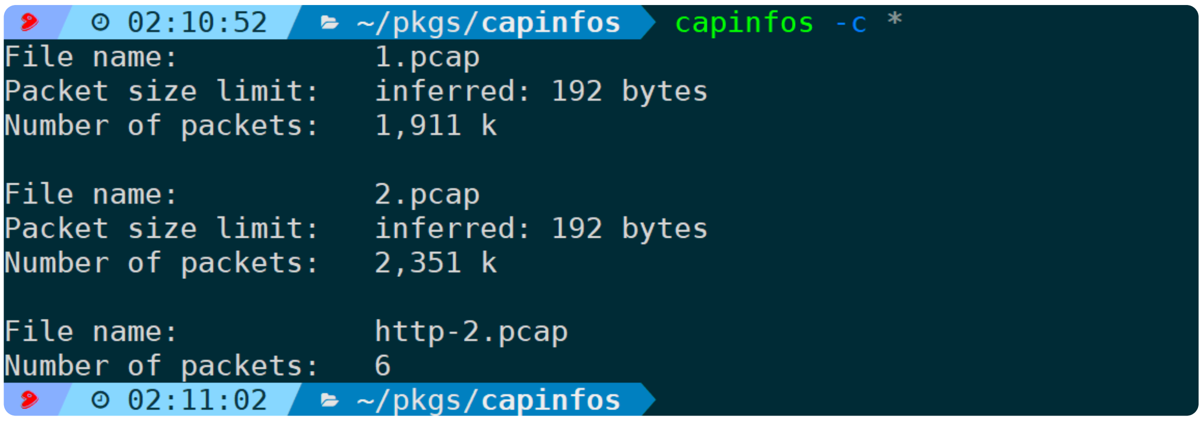
```
 00:15:13  ~/pkgs/3.1/10.1.14.96  capinfos -H *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
SHA256:              11efe5b72678d5f95feecbf4603c0fbde10166bd97023273f03bf7feb373fc5e
RIPEMD160:           4296dbe2e60a7ba2d2eb8ed9b6eae73ee5ccc831
SHA1:                64420e671a1a0a698e4df0ed2b83a70891477d48

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
SHA256:              f59427474303eee6dadda916787e17eafeb02b62718cc2cc136d7af7bc33c52a
RIPEMD160:           8c0adae8e06604843f7d5fb96d167d5228776c40
SHA1:                360ac8a90833186c54e747949357274d14236075

File name:           http-2.pcap
SHA256:              94b73b07d7eb9f4991a621359d8d61bec94c6df8f5971684d33aee8801984847
RIPEMD160:           1c20ae0df7d21552af53daebf003e068d1de77fd
SHA1:                52b6a5a405452039ef21aa13d80f9b49258b3633

File name:           test.pcap
SHA256:              4aab7c78b50445480eb4dc2f67deaf2dd3e2816010d484c05a8638388b9c32e5
RIPEMD160:           55ab9e6616d9f4882ddc44bfd806f3983bfbe837
SHA1:                c3ca49814e595ebf505fd4a56f81c30f747d6e0a
 00:15:20  ~/pkgs/3.1/10.1.14.96
```

2.文件大小选项

1）显示包量（-c）

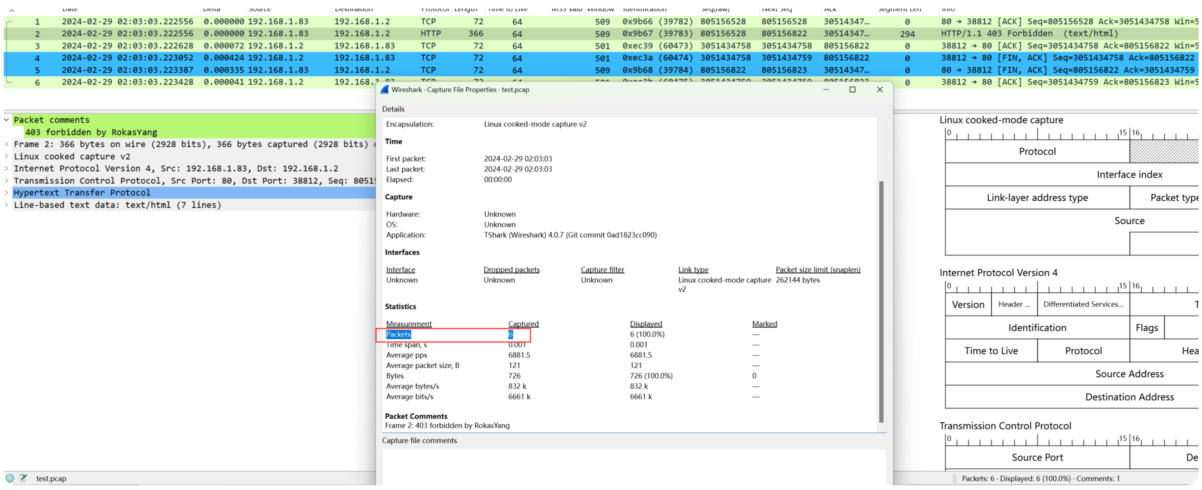
此选项用于打印包文件里的帧数量：

```
capinfos -c <文件名>
```



```
 02:10:52  ~/pkgs/capinfos  capinfos -c *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Number of packets:   1,911 k

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Number of packets:   2,351 k

File name:           http-2.pcap
Number of packets:   6
 02:11:02  ~/pkgs/capinfos
```

对应在wireshark页面的 **统计（Statistics）** --> **捕获文件属性（Capture File Properties）** ，也有这部分信息：



```
1  2024-02-29 02:03:03.222556 0.000000 192.168.1.83 192.168.1.2  TCP   72  64  509 0x9b66 (39782) 805156528  805156528  30514347…  0   80 → 38812 [ACK] Seq=805156528 Ack=3051434758 Win=5
2  2024-02-29 02:03:03.222556 0.000000 192.168.1.83 192.168.1.2  HTTP 366  64  509 0x9b67 (39783) 805156528  805156822  30514347… 294  HTTP/1.1 403 Forbidden  (text/html)
3  2024-02-29 02:03:03.222628 0.000072 192.168.1.2  192.168.1.83 TCP   72  64  501 0xec39 (60473) 3051434758 3051434758 805156822  0   38812 → 80 [ACK] Seq=3051434758 Ack=805156822 Win=5
4  2024-02-29 02:03:03.223052 0.000424 192.168.1.2  192.168.1.83 TCP   72  64  501 0xec3a (60474) 3051434758 3051434759 805156822  0   38812 → 80 [FIN, ACK] Seq=3051434758 Ack=805156822
5  2024-02-29 02:03:03.223387 0.000335 192.168.1.83 192.168.1.2  TCP   72  64  509 0x9b68 (39784) 805156822  805156823  30514347…  0   80 → 38812 [FIN, ACK] Seq=805156822 Ack=3051434759
6  2024-02-29 02:03:03.223428 0.000041 192.168.1.2                                                                                0   38812 → 80 [ACK] Seq=3051434759 Ack=805156823 Win=5
```

```
Packet comments
    403 forbidden by RokasYang
Frame 2: 366 bytes on wire (2928 bits), 366 bytes captured (2928 bits)
Linux cooked capture v2
Internet Protocol Version 4, Src: 192.168.1.83, Dst: 192.168.1.2
Transmission Control Protocol, Src Port: 80, Dst Port: 38812, Seq: 8051
Hypertext Transfer Protocol
Line-based text data: text/html (7 lines)
```

Wireshark · Capture File Properties · test.pcap

Details

Encapsulation: Linux cooked-mode capture v2

Time

First packet: 2024-02-29 02:03:03
Last packet: 2024-02-29 02:03:03
Elapsed: 00:00:00

Capture

Hardware: Unknown
OS: Unknown
Application: TShark (Wireshark) 4.0.7 (Git commit 0ad1823cc090)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Unknown	Unknown	Unknown	Linux cooked-mode capture v2	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	6	6 (100.0%)	—
Time span, s	0.001	0.001	—
Average pps	6881.5	6881.5	—
Average packet size, B	121	121	—
Bytes	726	726 (100.0%)	0
Average bytes/s	832 k	832 k	—
Average bits/s	6661 k	6661 k	—

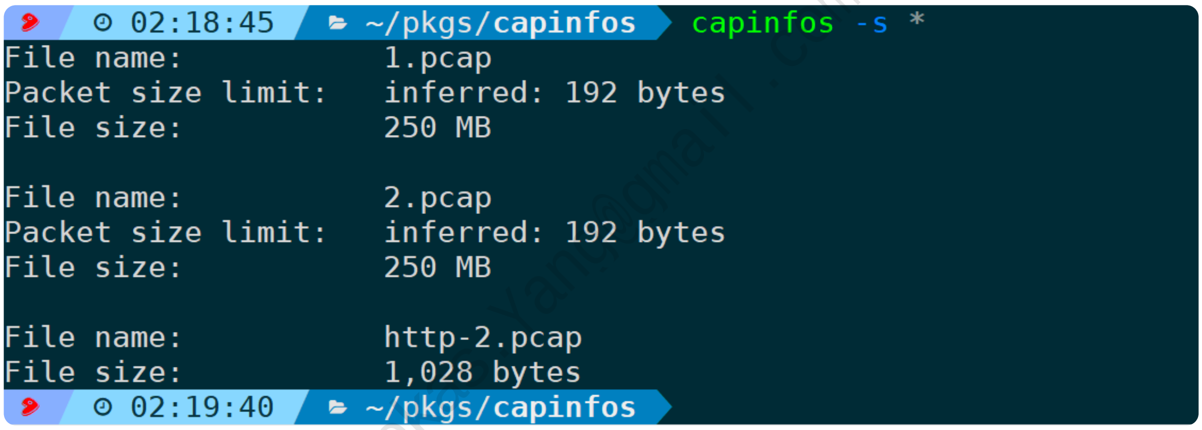
Packet Comments
Frame 2: 403 forbidden by RokasYang

Capture file comments

2）显示捕获文件的大小（-s）

以字节为单位，统计包文件大小：

```
capinfos -s <文件名>
```



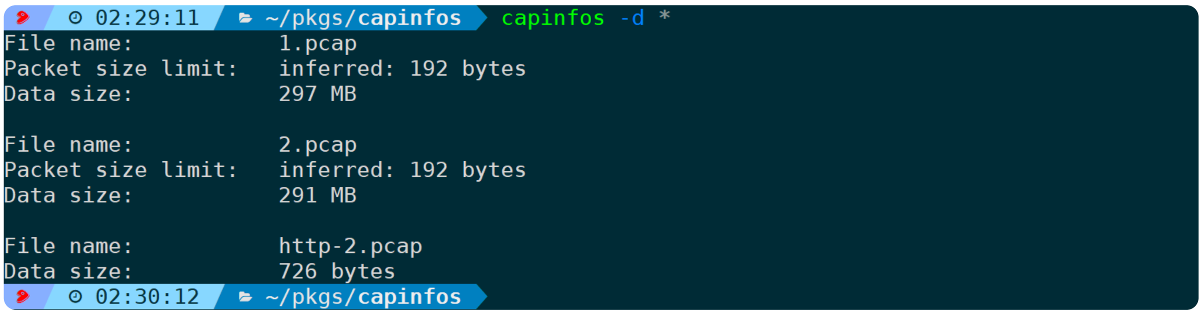
```
 02:18:45  ~/pkgs/capinfos  capinfos -s *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
File size:           250 MB

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
File size:           250 MB

File name:           http-2.pcap
File size:           1,028 bytes
 02:19:40  ~/pkgs/capinfos
```

如图，**File size**即为文件大小字段，如果文件过大会自动进行单位转换。

3）显示所有数据包的总长度（-d）

统计包文件中所有包的Length总大小：

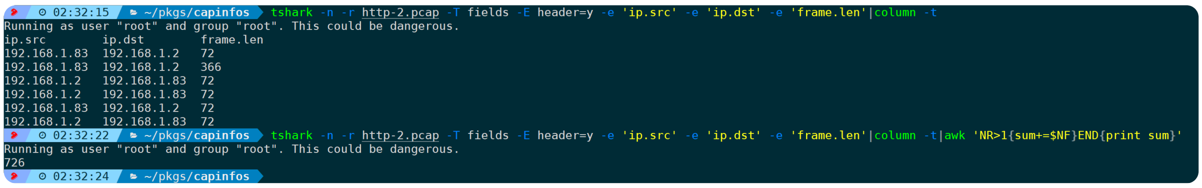
```
capinfos -d <文件名>
```



```
 02:29:11  ~/pkgs/capinfos  capinfos -d *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Data size:           297 MB

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Data size:           291 MB

File name:           http-2.pcap
Data size:           726 bytes
 02:30:12  ~/pkgs/capinfos
```

以http-2.pcap为例，统计的大小为726字节，我们通过tshark把每个包的**frame.len**字段值输出出来，并且用awk做一个累加，刚好为726字节：

```
tshark -n -r <文件名> -T fields -E header=y -e 'ip.src' -e 'ip.dst' -e
'frame.len'|column -t|awk 'NR>1{sum+=$NF}END{print sum}'
```

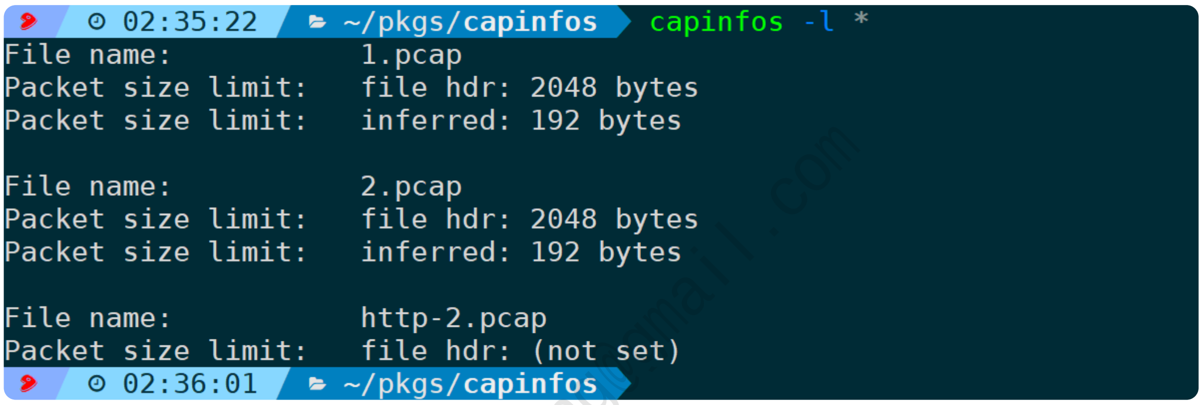


```
 02:32:15  ~/pkgs/capinfos  tshark -n -r http-2.pcap -T fields -E header=y -e 'ip.src' -e 'ip.dst' -e 'frame.len'|column -t
Running as user "root" and group "root". This could be dangerous.
ip.src        ip.dst        frame.len
192.168.1.83  192.168.1.2   72
192.168.1.83  192.168.1.2   366
192.168.1.2   192.168.1.83  72
192.168.1.2   192.168.1.83  72
192.168.1.83  192.168.1.2   72
192.168.1.2   192.168.1.83  72
 02:32:22  ~/pkgs/capinfos  tshark -n -r http-2.pcap -T fields -E header=y -e 'ip.src' -e 'ip.dst' -e 'frame.len'|column -t|awk 'NR>1{sum+=$NF}END{print sum}'
Running as user "root" and group "root". This could be dangerous.
726
 02:32:24  ~/pkgs/capinfos
```

4）显示数据包大小限制（-l）

此选项会显示包文件抓包时的限制大小（file hdr）和包文件中数据帧的推断长度（inferred）：

```
capinfos -l <文件名>
```



```
 02:35:22  ~/pkgs/capinfos  capinfos -l *
File name:           1.pcap
Packet size limit:   file hdr: 2048 bytes
Packet size limit:   inferred: 192 bytes

File name:           2.pcap
Packet size limit:   file hdr: 2048 bytes
Packet size limit:   inferred: 192 bytes

File name:           http-2.pcap
Packet size limit:   file hdr: (not set)
 02:36:01  ~/pkgs/capinfos
```

输出含义如下：

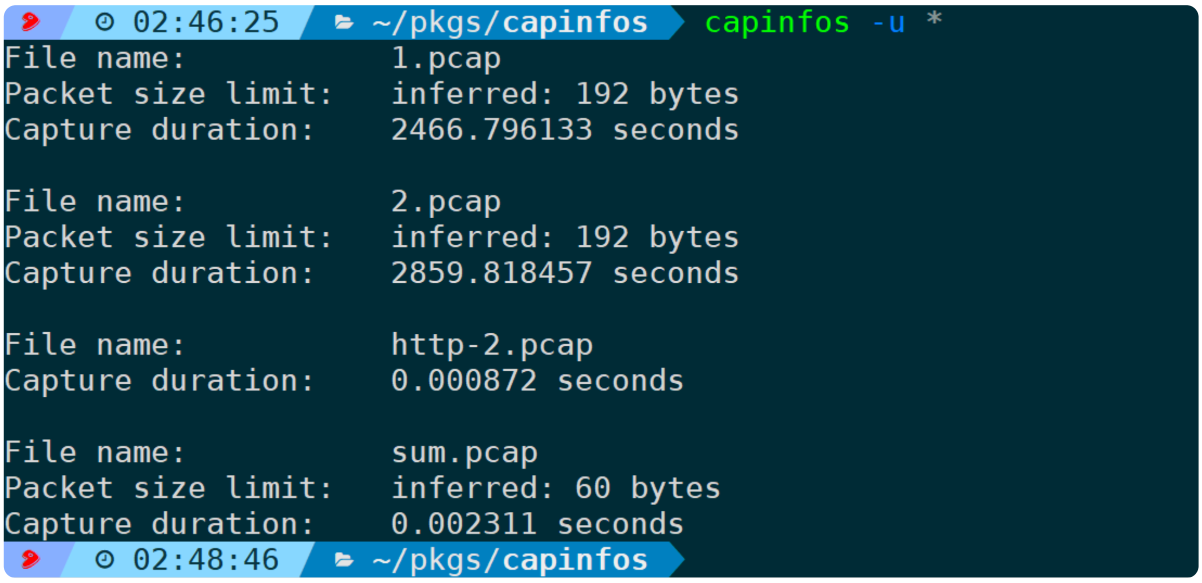
```
File name:           1.pcap
Packet size limit:   file hdr: 2048 bytes #抓包设置的每个帧最大抓包Length
Packet size limit:   inferred: 192 bytes  #根据包文件里的帧推断的Length

File name:           2.pcap
Packet size limit:   file hdr: 2048 bytes  #抓包设置的每个帧最大抓包Length
Packet size limit:   inferred: 192 bytes   #根据包文件里的帧推断的Length

File name:           http-2.pcap
Packet size limit:   file hdr: (not set)  #没有设限
```

3.时间信息选项

1）统计捕获持续时间（-u）

以秒为单位，显示统计抓包时的持续时间：

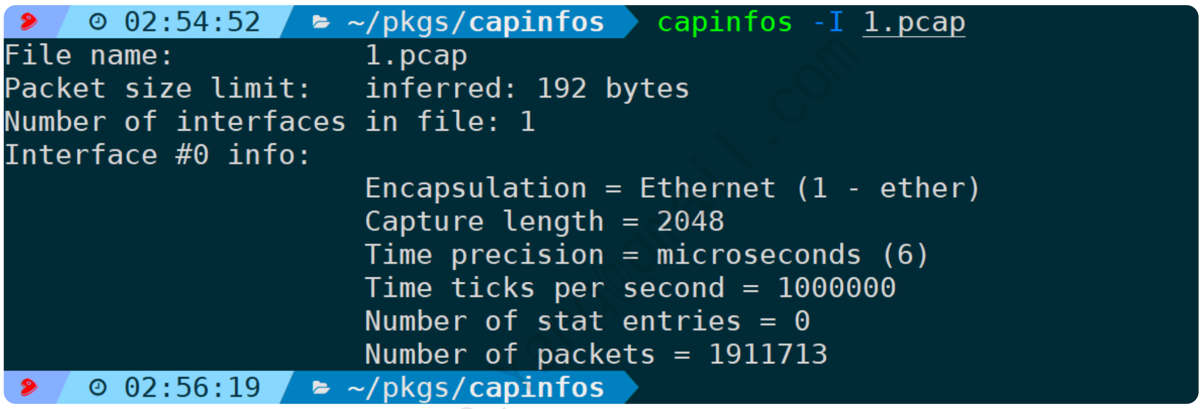
```
capinfos -u <文件名>
```



```
02:46:25  ~/pkgs/capinfos  capinfos -u *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Capture duration:    2466.796133 seconds

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Capture duration:    2859.818457 seconds

File name:           http-2.pcap
Capture duration:    0.000872 seconds

File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Capture duration:    0.002311 seconds
02:48:46  ~/pkgs/capinfos
```

以1.pcap为例，如上图，-u统计的时间间隔为2466.796133秒，我们先通过-I选项拿到包文件的总包量：

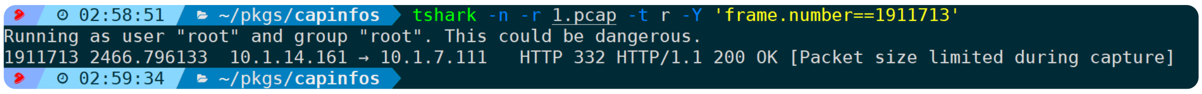
```
capinfos -I <文件名>
```



```
02:54:52  ~/pkgs/capinfos  capinfos -I 1.pcap
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 2048
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 1911713
02:56:19  ~/pkgs/capinfos
```

包量为1911713，也就是说最后一帧的帧序号为1911713；此时通过tshark，来看最后一帧相对于第一帧的时间间隔：

```
tshark -n -r <文件名> -t r -Y 'frame.number==xxx'
```



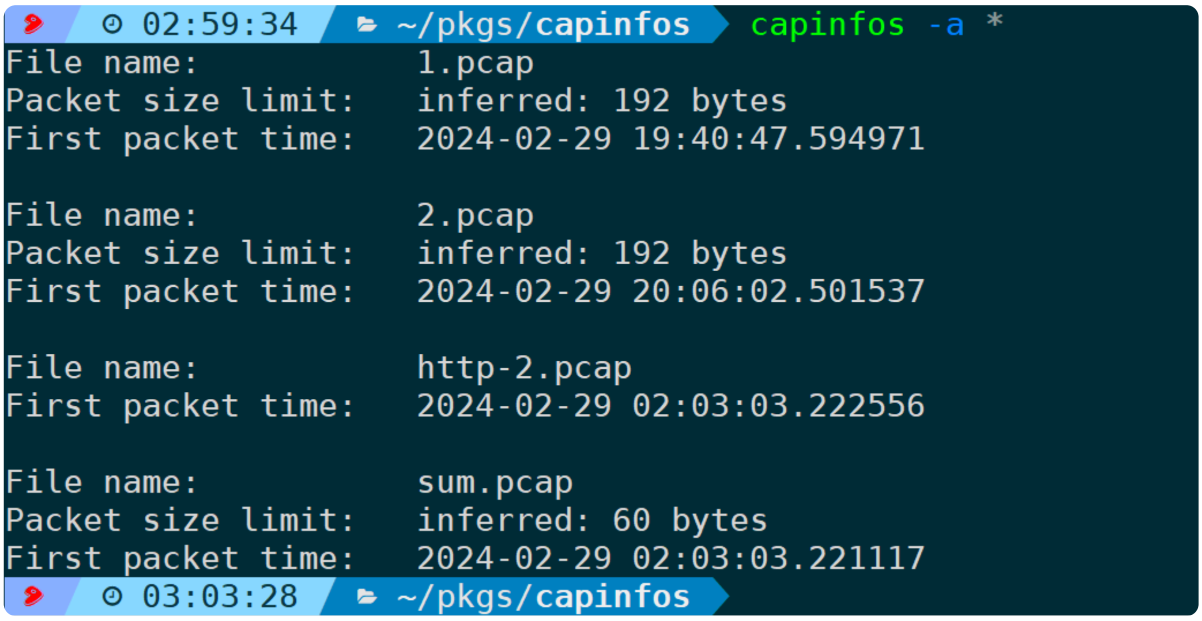
```
02:58:51  ~/pkgs/capinfos  tshark -n -r 1.pcap -t r -Y 'frame.number==1911713'
Running as user "root" and group "root". This could be dangerous.
1911713 2466.796133  10.1.14.161 → 10.1.7.111    HTTP 332 HTTP/1.1 200 OK [Packet size limited during capture]
02:59:34  ~/pkgs/capinfos
```

输出结果为2466.796133秒，符合预期。所以可以清晰知道，-u统计方法实际就是尾包减去首包的时间差。

2）显示抓包的开始时间日期（-a）

此参数不用做过多介绍：

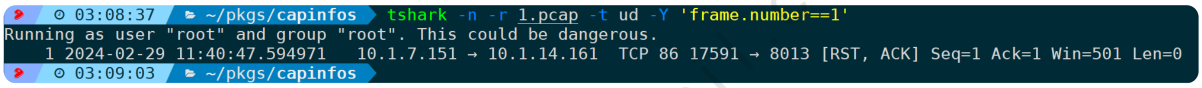
```
capinfos -a <文件名>
```



```
 02:59:34  ~/pkgs/capinfos  capinfos -a *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
First packet time:   2024-02-29 19:40:47.594971

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
First packet time:   2024-02-29 20:06:02.501537

File name:           http-2.pcap
First packet time:   2024-02-29 02:03:03.222556

File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
First packet time:   2024-02-29 02:03:03.221117
 03:03:28  ~/pkgs/capinfos
```

还有很多种方式可以查看抓包开始时间，比如通过tshark输出第一帧的时间：

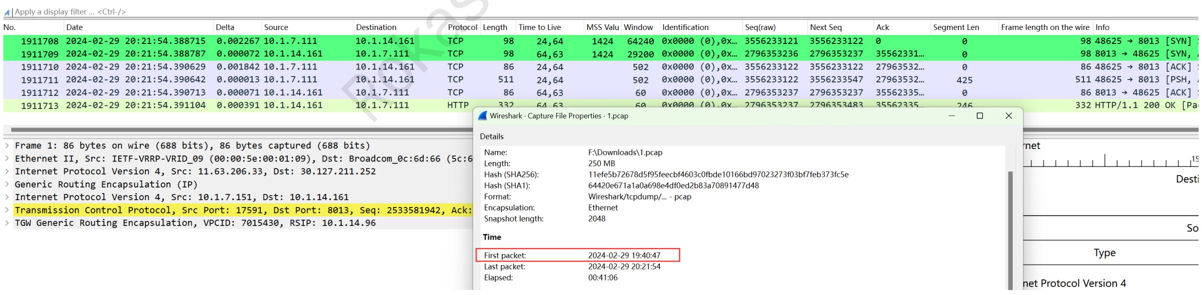
```
tshark -n -r <文件名> -t ud -Y 'frame.number==1'
```



```
 03:08:37  ~/pkgs/capinfos  tshark -n -r 1.pcap -t ud -Y 'frame.number==1'
Running as user "root" and group "root". This could be dangerous.
    1 2024-02-29 11:40:47.594971   10.1.7.151 → 10.1.14.161  TCP 86 17591 → 8013 [RST, ACK] Seq=1 Ack=1 Win=501 Len=0
 03:09:03  ~/pkgs/capinfos
```

> -t ud统计的是UTC时间，需要在此基础上+8才是北京时间。

在Wireshark的捕获文件属性里，也能看到首尾包时间：



```
Apply a display filter ... <Ctrl-/>
No.      Date                       Delta    Source      Destination  Protocol Length Time to Live MSS Valu Window Identification   Seq(raw)   Next Seq   Ack         Segment Len Frame length on the wire Info
 1911708 2024-02-29 20:21:54.388715 0.002267 10.1.7.111  10.1.14.161  TCP      98     24,64        1424     64240  0x0000 (0),0x…  3556233121 3556233122 0           0           98 48625 → 8013 [SYN]
 1911709 2024-02-29 20:21:54.388787 0.000072 10.1.14.161 10.1.7.111   TCP      98     64,63        1424     29200  0x0000 (0),0x…  2796353236 2796353237 35562331…  0           98 8013 → 48625 [SYN,
 1911710 2024-02-29 20:21:54.390629 0.001842 10.1.7.111  10.1.14.161  TCP      86     24,64                 502    0x0000 (0),0x…  3556233122 3556233122 27963532…  0           86 48625 → 8013 [ACK]
 1911711 2024-02-29 20:21:54.390642 0.000013 10.1.7.111  10.1.14.161  TCP      511    24,64                 502    0x0000 (0),0x…  3556233122 3556233547 27963532…  425         511 48625 → 8013 [PSH,
 1911712 2024-02-29 20:21:54.390713 0.000071 10.1.14.161 10.1.7.111   TCP      86     64,63                 60     0x0000 (0),0x…  2796353237 2796353237 35562335…  0           86 8013 → 48625 [ACK]
 1911713 2024-02-29 20:21:54.391104 0.000391 10.1.14.161 10.1.7.111   HTTP     332    64,63                 60     0x0000 (0),0x…  2796353237 2796353483 35562335…  246         332 HTTP/1.1 200 OK [Pa

> Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
> Ethernet II, Src: IETF-VRRP-VRID_09 (00:00:5e:00:01:09), Dst: Broadcom_0c:6d:66 (5c:6
> Internet Protocol Version 4, Src: 11.63.206.33, Dst: 30.127.211.252
> Generic Routing Encapsulation (IP)
> Internet Protocol Version 4, Src: 10.1.7.151, Dst: 10.1.14.161
> Transmission Control Protocol, Src Port: 17591, Dst Port: 8013, Seq: 2533581942, Ack:
> TGW Generic Routing Encapsulation, VPCID: 7015430, RSIP: 10.1.14.96
```

Wireshark · Capture File Properties · 1.pcap

Details

Name:	F:\Downloads\1.pcap
Length:	250 MB
Hash (SHA256):	11efe5b72678d5f95feecbf4603c0fbde10166bd97023273f03bf7feb373fc5e
Hash (SHA1):	64420e671a1a0a698e4df0ed2b83a70891477d48
Format:	Wireshark/tcpdump/... - pcap
Encapsulation:	Ethernet
Snapshot length:	2048

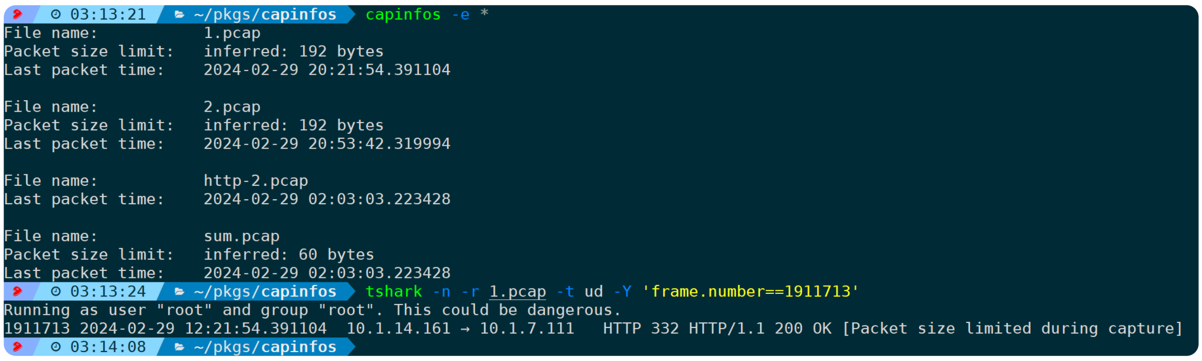
Time

First packet:	2024-02-29 19:40:47
Last packet:	2024-02-29 20:21:54
Elapsed:	00:41:06

3）显示抓包的结束时间日期（-e）

与开始（-a）相对的则为-e显示抓包结束时间，实际就是统计尾包的时间日期：

```
capinfos -e <包文件>
```

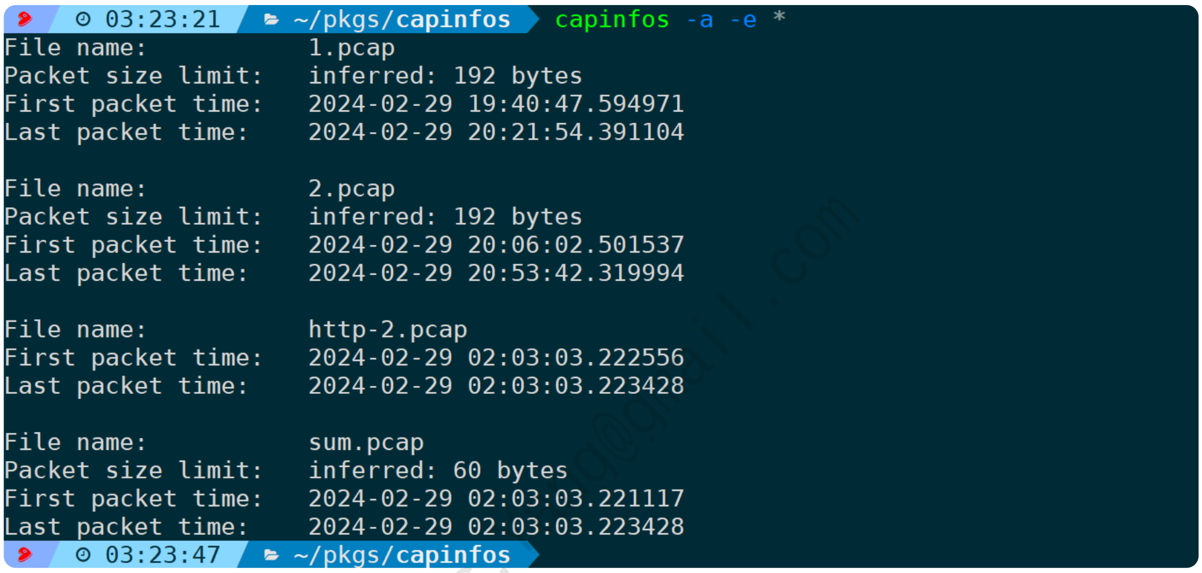


```
 03:13:21  ~/pkgs/capinfos  capinfos -e *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Last packet time:    2024-02-29 20:21:54.391104

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Last packet time:    2024-02-29 20:53:42.319994

File name:           http-2.pcap
Last packet time:    2024-02-29 02:03:03.223428

File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Last packet time:    2024-02-29 02:03:03.223428
 03:13:24  ~/pkgs/capinfos  tshark -n -r 1.pcap -t ud -Y 'frame.number==1911713'
Running as user "root" and group "root". This could be dangerous.
1911713 2024-02-29 12:21:54.391104  10.1.14.161 → 10.1.7.111   HTTP 332 HTTP/1.1 200 OK [Packet size limited during capture]
 03:14:08  ~/pkgs/capinfos
```

如图，通过tshark统计尾包的UTC时间再+8，也能得到相同的结果。

-a和-e可以同时使用，既显示开始时间又显示结束时间：

```
capinfos -a -e <包文件>
```



```
 03:23:21  ~/pkgs/capinfos  capinfos -a -e *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
First packet time:   2024-02-29 19:40:47.594971
Last packet time:    2024-02-29 20:21:54.391104

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
First packet time:   2024-02-29 20:06:02.501537
Last packet time:    2024-02-29 20:53:42.319994

File name:           http-2.pcap
First packet time:   2024-02-29 02:03:03.222556
Last packet time:    2024-02-29 02:03:03.223428

File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
First packet time:   2024-02-29 02:03:03.221117
Last packet time:    2024-02-29 02:03:03.223428
 03:23:47  ~/pkgs/capinfos
```

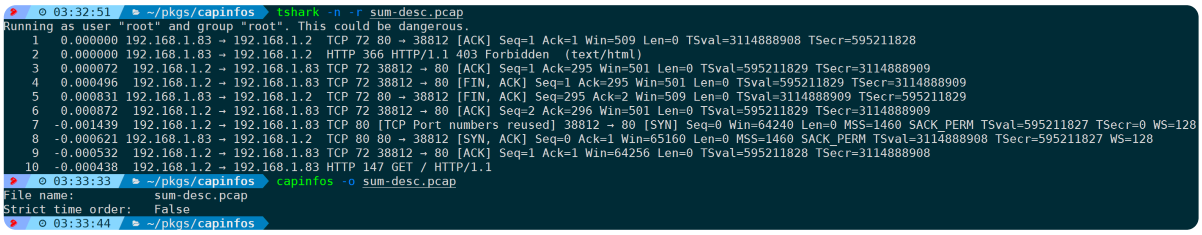
4）显示抓包文件的时间顺序真假（-o）

当数据帧的顺序没有严格按照时间顺序进行排列时，则会判定为False，反之判定为True：

```
capinfos -o <文件名>
```

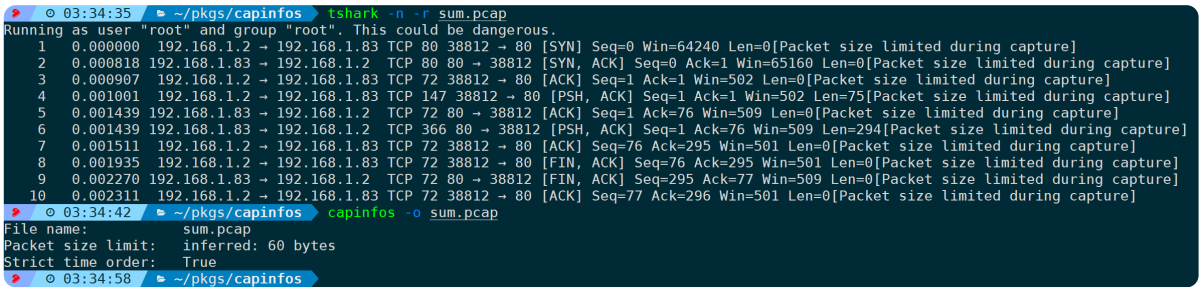
以下面这个例子为例：

sum-desc.pcap的包序，没有严格按照绝对时间进行排序， **-o**选项识别为False：



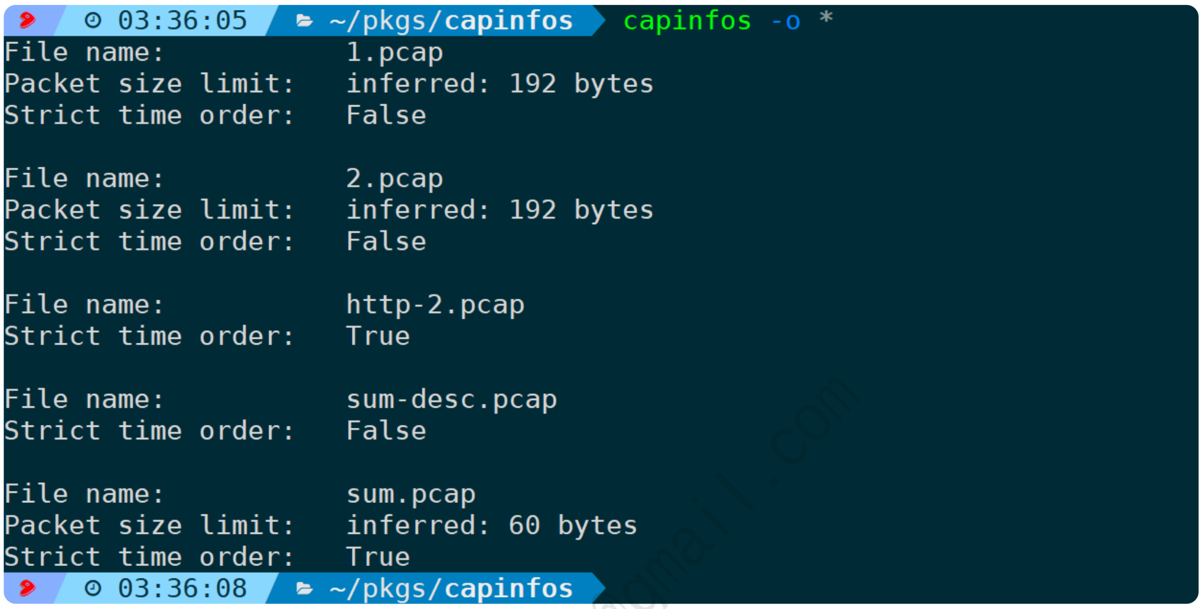
```
 03:32:51  ~/pkgs/capinfos  tshark -n -r sum-desc.pcap
Running as user "root" and group "root". This could be dangerous.
    1   0.000000 192.168.1.83 → 192.168.1.2  TCP 72 80 → 38812 [ACK] Seq=1 Ack=1 Win=509 Len=0 TSval=3114888908 TSecr=595211828
    2   0.000000 192.168.1.83 → 192.168.1.2  HTTP 366 HTTP/1.1 403 Forbidden  (text/html)
    3   0.000072  192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [ACK] Seq=1 Ack=295 Win=501 Len=0 TSval=595211829 TSecr=3114888909
    4   0.000496  192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [FIN, ACK] Seq=1 Ack=295 Win=501 Len=0 TSval=595211829 TSecr=3114888909
    5   0.000831 192.168.1.83 → 192.168.1.2  TCP 72 80 → 38812 [FIN, ACK] Seq=295 Ack=2 Win=509 Len=0 TSval=3114888909 TSecr=595211829
    6   0.000872  192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [ACK] Seq=2 Ack=296 Win=501 Len=0 TSval=595211829 TSecr=3114888909
    7  -0.001439  192.168.1.2 → 192.168.1.83 TCP 80 [TCP Port numbers reused] 38812 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=595211827 TSecr=0 WS=128
    8  -0.000621 192.168.1.83 → 192.168.1.2  TCP 80 80 → 38812 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3114888908 TSecr=595211827 WS=128
    9  -0.000532  192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=595211828 TSecr=3114888908
   10  -0.000438  192.168.1.2 → 192.168.1.83 HTTP 147 GET / HTTP/1.1
 03:33:33  ~/pkgs/capinfos  capinfos -o sum-desc.pcap
File name:           sum-desc.pcap
Strict time order:   False
 03:33:44  ~/pkgs/capinfos
```

而反观sum.pcap的包序，已经严格按照绝对时间排序，识别为True：



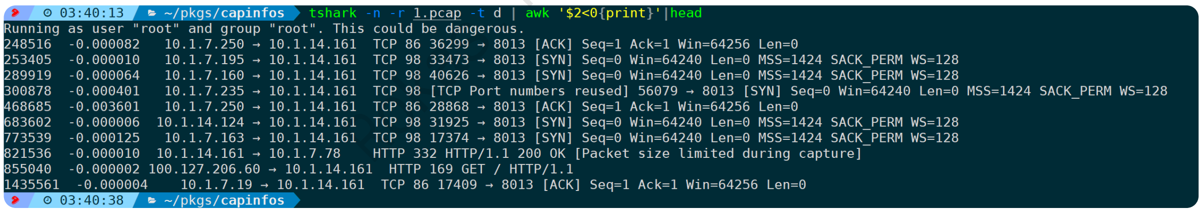
```
 03:34:35  ~/pkgs/capinfos  tshark -n -r sum.pcap
Running as user "root" and group "root". This could be dangerous.
    1   0.000000  192.168.1.2 → 192.168.1.83 TCP 80 38812 → 80 [SYN] Seq=0 Win=64240 Len=0[Packet size limited during capture]
    2   0.000818 192.168.1.83 → 192.168.1.2  TCP 80 80 → 38812 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0[Packet size limited during capture]
    3   0.000907  192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [ACK] Seq=1 Ack=1 Win=502 Len=0[Packet size limited during capture]
    4   0.001001  192.168.1.2 → 192.168.1.83 TCP 147 38812 → 80 [PSH, ACK] Seq=1 Ack=1 Win=502 Len=75[Packet size limited during capture]
    5   0.001439 192.168.1.83 → 192.168.1.2  TCP 72 80 → 38812 [ACK] Seq=1 Ack=76 Win=509 Len=0[Packet size limited during capture]
    6   0.001439 192.168.1.83 → 192.168.1.2  TCP 366 80 → 38812 [PSH, ACK] Seq=1 Ack=76 Win=509 Len=294[Packet size limited during capture]
    7   0.001511  192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [ACK] Seq=76 Ack=295 Win=501 Len=0[Packet size limited during capture]
    8   0.001935  192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [FIN, ACK] Seq=76 Ack=295 Win=501 Len=0[Packet size limited during capture]
    9   0.002270 192.168.1.83 → 192.168.1.2  TCP 72 80 → 38812 [FIN, ACK] Seq=295 Ack=77 Win=509 Len=0[Packet size limited during capture]
   10   0.002311  192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [ACK] Seq=77 Ack=296 Win=501 Len=0[Packet size limited during capture]
 03:34:42  ~/pkgs/capinfos  capinfos -o sum.pcap
File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Strict time order:   True
 03:34:58  ~/pkgs/capinfos
```

路径下还有1.pcap、2.pcap识别为False：



```
 03:36:05  ~/pkgs/capinfos  capinfos -o *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Strict time order:   False

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Strict time order:   False

File name:           http-2.pcap
Strict time order:   True

File name:           sum-desc.pcap
Strict time order:   False

File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Strict time order:   True
 03:36:08  ~/pkgs/capinfos
```

通过时间戳也可以判断，tshark时间格式设定为**-t d**（delta时间，相对于上一个frame的时间间隔），如果出现负值，则说明包序不对（即：明明更早就收到了，但排序在后面）：



```
 03:40:13  ~/pkgs/capinfos  tshark -n -r 1.pcap -t d | awk '$2<0{print}'|head
Running as user "root" and group "root". This could be dangerous.
248516  -0.000082   10.1.7.250 → 10.1.14.161  TCP 86 36299 → 8013 [ACK] Seq=1 Ack=1 Win=64256 Len=0
253405  -0.000010   10.1.7.195 → 10.1.14.161  TCP 98 33473 → 8013 [SYN] Seq=0 Win=64240 Len=0 MSS=1424 SACK_PERM WS=128
289919  -0.000064   10.1.7.160 → 10.1.14.161  TCP 98 40626 → 8013 [SYN] Seq=0 Win=64240 Len=0 MSS=1424 SACK_PERM WS=128
300878  -0.000401   10.1.7.235 → 10.1.14.161  TCP 98 [TCP Port numbers reused] 56079 → 8013 [SYN] Seq=0 Win=64240 Len=0 MSS=1424 SACK_PERM WS=128
468685  -0.003601   10.1.7.250 → 10.1.14.161  TCP 86 28868 → 8013 [ACK] Seq=1 Ack=1 Win=64256 Len=0
683602  -0.000006  10.1.14.124 → 10.1.14.161  TCP 98 31925 → 8013 [SYN] Seq=0 Win=64240 Len=0 MSS=1424 SACK_PERM WS=128
773539  -0.000125   10.1.7.163 → 10.1.14.161  TCP 98 17374 → 8013 [SYN] Seq=0 Win=64240 Len=0 MSS=1424 SACK_PERM WS=128
821536  -0.000010  10.1.14.161 → 10.1.7.78    HTTP 332 HTTP/1.1 200 OK [Packet size limited during capture]
855040  -0.000002 100.127.206.60 → 10.1.14.161  HTTP 169 GET / HTTP/1.1
1435561  -0.000004    10.1.7.19 → 10.1.14.161  TCP 86 17409 → 8013 [ACK] Seq=1 Ack=1 Win=64256 Len=0
 03:40:38  ~/pkgs/capinfos
```

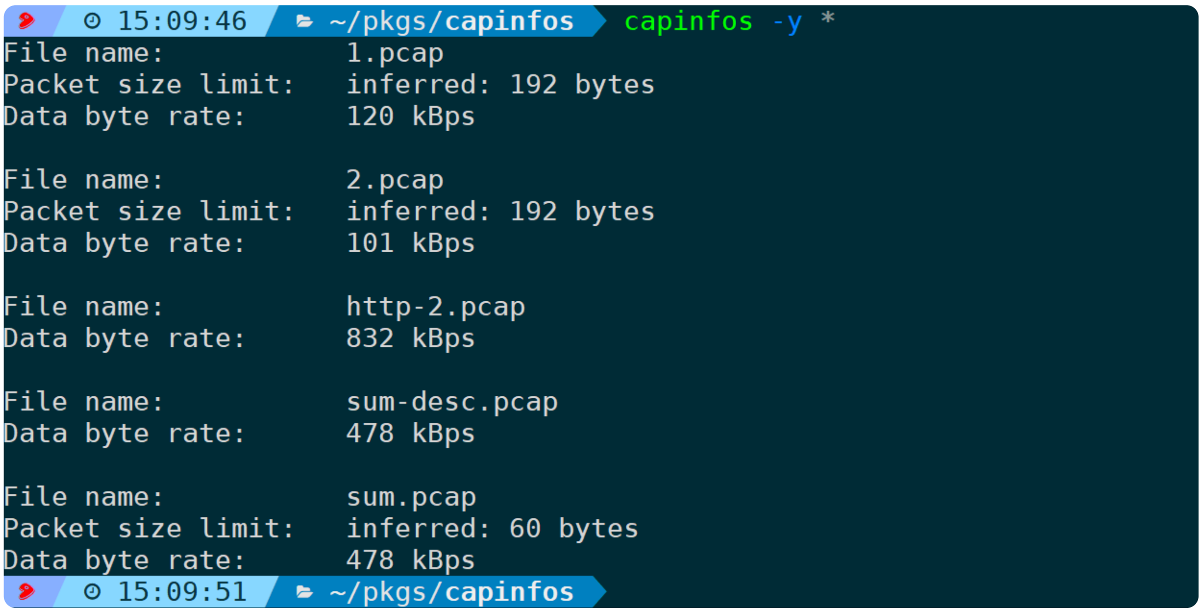
所以-o判定为False。

4.统计分析选项

1）统计数据传输平均速率（-y/-i）

输出单位为字节每秒（ Bytes/sec）：

```
capinfos -y <文件名>
```

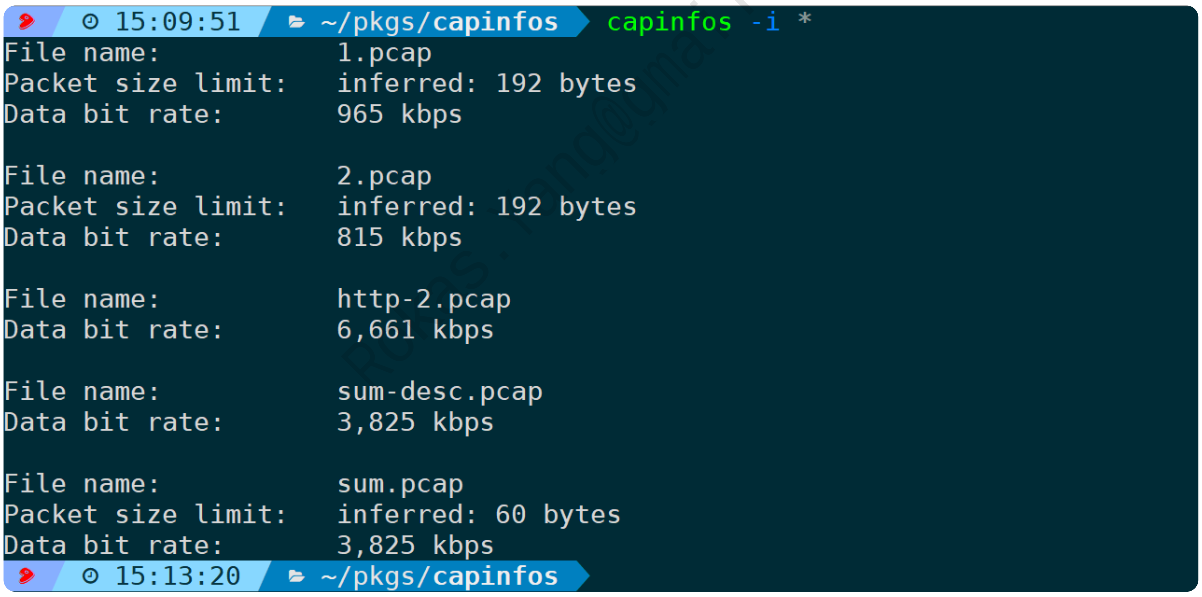


```
15:09:46  ~/pkgs/capinfos  capinfos -y *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Data byte rate:      120 kBps

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Data byte rate:      101 kBps

File name:           http-2.pcap
Data byte rate:      832 kBps

File name:           sum-desc.pcap
Data byte rate:      478 kBps

File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Data byte rate:      478 kBps
15:09:51  ~/pkgs/capinfos
```

Data byte rate字段即为数据传输的平均速率。

输出单位以比特每秒（bit/sec），则为-i选项：

```
capinfos -i <文件名>
```

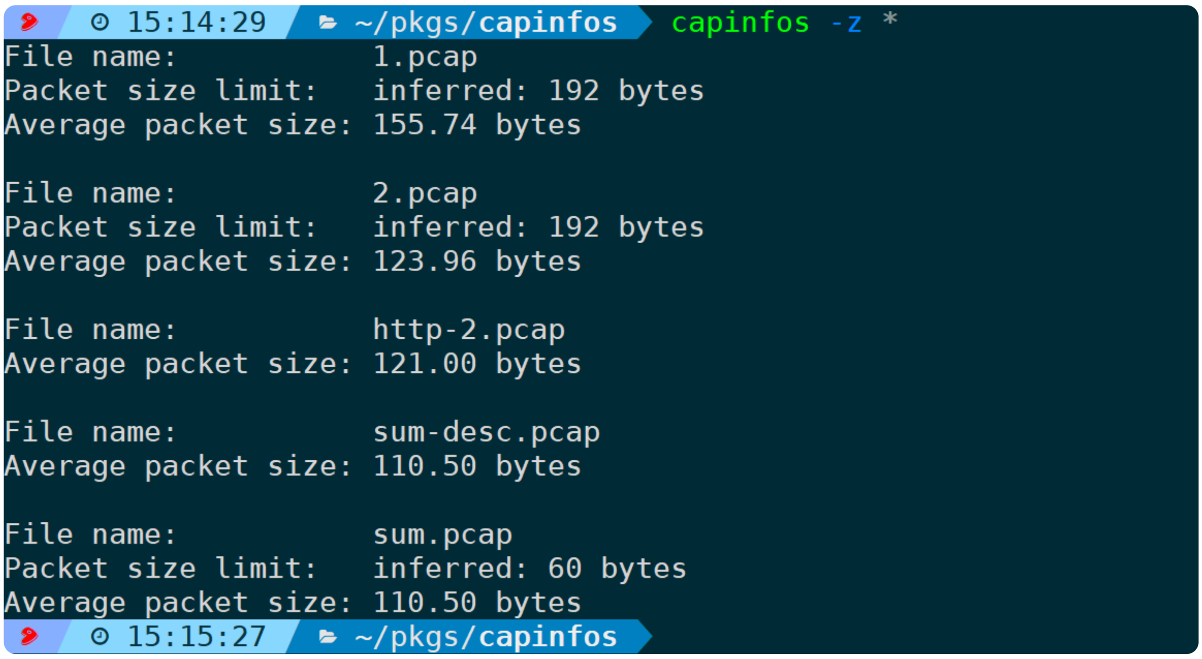


```
15:09:51  ~/pkgs/capinfos  capinfos -i *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Data bit rate:       965 kbps

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Data bit rate:       815 kbps

File name:           http-2.pcap
Data bit rate:       6,661 kbps

File name:           sum-desc.pcap
Data bit rate:       3,825 kbps

File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Data bit rate:       3,825 kbps
15:13:20  ~/pkgs/capinfos
```

2）统计每个帧的平均大小（-z）

默认以字节为单位：

```
capinfos -z <文件名>
```



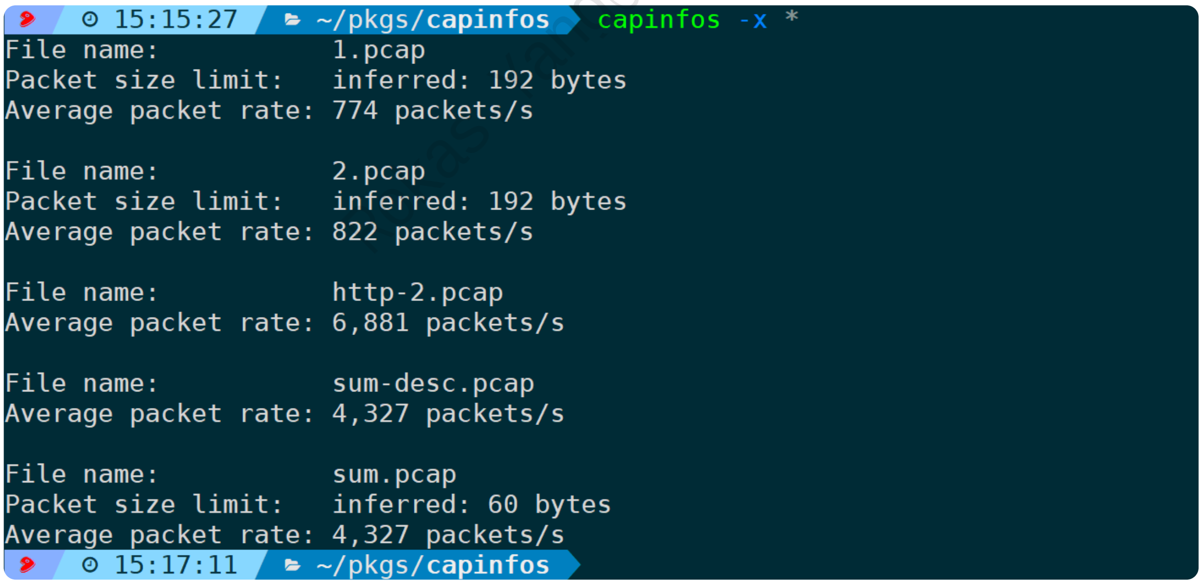
```
 15:14:29  ~/pkgs/capinfos  capinfos -z *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Average packet size: 155.74 bytes

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Average packet size: 123.96 bytes

File name:           http-2.pcap
Average packet size: 121.00 bytes

File name:           sum-desc.pcap
Average packet size: 110.50 bytes

File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Average packet size: 110.50 bytes
 15:15:27  ~/pkgs/capinfos
```

Average packet size字段即为每个帧的平均大小。

3）统计平均收发包速率（-x）

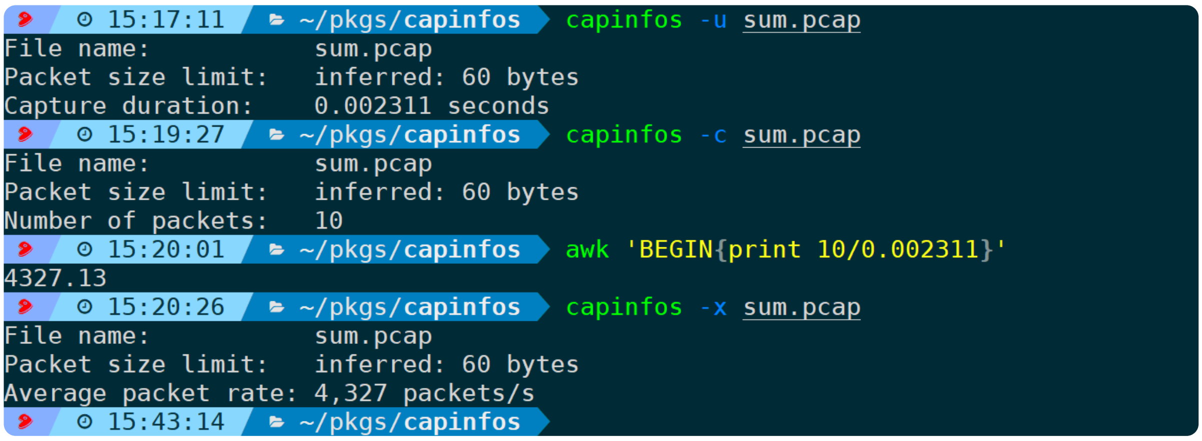
单位为包量每秒：

```
capinfos -x <文件名>
```



```
 15:15:27  ~/pkgs/capinfos  capinfos -x *
File name:           1.pcap
Packet size limit:   inferred: 192 bytes
Average packet rate: 774 packets/s

File name:           2.pcap
Packet size limit:   inferred: 192 bytes
Average packet rate: 822 packets/s

File name:           http-2.pcap
Average packet rate: 6,881 packets/s

File name:           sum-desc.pcap
Average packet rate: 4,327 packets/s

File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Average packet rate: 4,327 packets/s
 15:17:11  ~/pkgs/capinfos
```

统计逻辑也很简单，平均包速率 = 总包量 / 总时间，比如sum.pcap：



```
 ⊙ 15:17:11  ~/pkgs/capinfos  capinfos -u sum.pcap
File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Capture duration:    0.002311 seconds
 ⊙ 15:19:27  ~/pkgs/capinfos  capinfos -c sum.pcap
File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Number of packets:   10
 ⊙ 15:20:01  ~/pkgs/capinfos  awk 'BEGIN{print 10/0.002311}'
4327.13
 ⊙ 15:20:26  ~/pkgs/capinfos  capinfos -x sum.pcap
File name:           sum.pcap
Packet size limit:   inferred: 60 bytes
Average packet rate: 4,327 packets/s
 ⊙ 15:43:14  ~/pkgs/capinfos 
```

5.输出格式选项

选项	含义
-L	生成长报告，默认行为
-T	以表格形式生成
-M	在长报告中显示机器可读的值

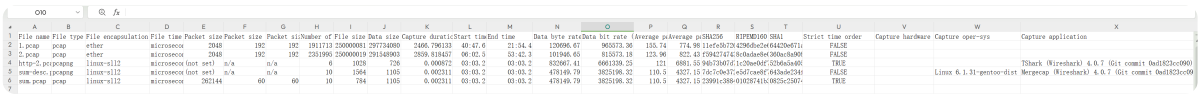
值得一讲的是-T参数，-T参数下面还包含一系列子选项：

选项	含义
-R	生成头记录，默认行为
-r	不生成头记录
-B	使用TAB字符分隔字段，默认行为
-m	使用逗号（,）分隔字段
-b	使用空格分隔字段
-N	不要引用信息，默认行为
-q	使用单引号引用信息
-Q	使用双引号引用信息

-T后面接什么按需调整，如果一次性需要读取包信息的内容比较多，可以考虑把输出内容重定向到Excel文件，比如：

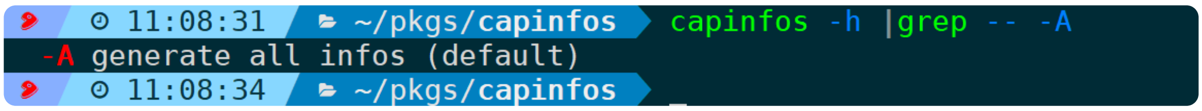
```
capinfos -T <文件名> > output.xlsx
```

用Excel打开的效果：



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	File name	File type	File encapsulation	File time	Packet size	Packet size	Packet si	Number of	File size	Data size	Capture duratic	Start time	End time	Data byte rate	Data bit rate (	Average pa	Average pa	SHA256	RIPEMD160	SHA1	Strict time order	Capture hardware	Capture oper-sys	Capture application
2	1.pcap	pcap	ether	microsecor	2048	192	192	1911713	250000081	297734080	2466.796133	40:47.6	21:54.4	120696.67	965573.36	155.74	774.98	11efe5b726	4296dbe2e6	64420e671a	FALSE			
3	2.pcap	pcap	ether	microsecor	2048	192	192	2351995	250000019	291548903	2859.818457	06:02.5	53:42.3	101946.65	815573.18	123.96	822.43	f59427474	8c0adae8e0	360ac8a908	FALSE			
4	http-2.pca	pcapng	linux-sll2	microsecor	(not set)	n/a	n/a	6	1028	726	0.000872	03:03.2	03:03.2	832667.41	6661339.25	121	6881.55	94b73b07d7	1c20ae0df7	52b6a5a405	TRUE			TShark (Wireshark) 4.0.7 (Git commit 0ad1823cc090)
5	sum-desc.p	pcapng	linux-sll2	microsecor	(not set)	n/a	n/a	10	1564	1105	0.002311	03:03.2	03:03.2	478149.79	3825198.32	110.5	4327.15	7dc7c0e372	e5d7cae8f7	643ade234f	FALSE		Linux 6.1.31-gentoo-dist	Mergecap (Wireshark) 4.0.7 (Git commit 0ad1823cc09
6	sum.pcap	pcap	linux-sll2	microsecor	262144	60	60	10	784	1105	0.002311	03:03.2	03:03.2	478149.79	3825198.32	110.5	4327.15	23991c388	01028741b3	0825c2507	TRUE			
7																								

输出的字段包含所有信息，因为没有接任何其他选项，默认采用-A，即输出所有信息：

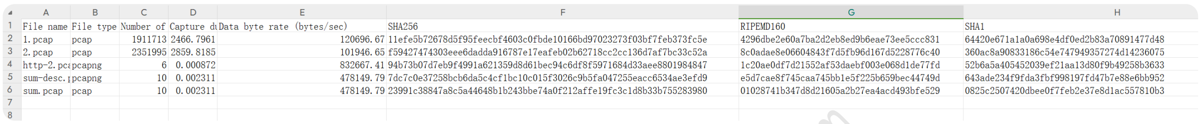


```
 11:08:31  ~/pkgs/capinfos  capinfos -h |grep -- -A
  -A generate all infos (default)
 11:08:34  ~/pkgs/capinfos 
```

结合前面所讲的参数，你可以任意搭配使用，比如显示包量、文件类型、hash值、抓包持续时间、传输平均速率，可以是：

```
capinfos -c -t -H -u -y -T <文件名> > output.xlsx
```

此时输出的字段则为我们想要的内容：



	A	B	C	D	E	F	G	H
1	File name	File type	Number of	Capture du	Data byte rate (bytes/sec)	SHA256	RIPEMD160	SHA1
2	1.pcap	pcap	1911713	2466.7961	120696.67	11efe5b72678d5f95feecbf4603c0fbde10166bd97023273f03bf7feb373fc5e	4296dbe2e60a7ba2d2eb8ed9b6eae73ee5ccc831	64420e671a1a0a698e4df0ed2b83a70891477d48
3	2.pcap	pcap	2351995	2859.8185	101946.65	f59427474303eee6dadda916787e17eafeb02b62718cc2cc136d7af7bc33c52a	8c0adae8e06604843f7d5fb96d167d5228776c40	360ac8a90833186c54e747949357274d14236075
4	http-2.pca	pcapng	6	0.000872	832667.41	94b73b07d7eb9f4991a621359d8d61bec94c6df8f5971684d33aee8801984847	1c20ae0df7d21552af53daebf003e068d1de77fd	52b6a5a405452039ef21aa13d80f9b49258b3633
5	sum-desc.p	pcapng	10	0.002311	478149.79	7dc7c0e37258bcb6da5c4cf1bc10c015f3026c9b5fa047255eacc6534ae3efd9	e5d7cae8f745caa745bb1e5f225b659bec44749d	643ade234f9fda3fbf998197fd47b7e88e6bb952
6	sum.pcap	pcap	10	0.002311	478149.79	23991c38847a8c5a44648b1b243bbe74a0f212affe19fc3c1d8b33b755283980	01028741b347d8d21605a2b27ea4acd493bfe529	0825c2507420dbee0f7feb2e37e8d1ac557810b3
7								
8								

四、总结

本文介绍了capinfos的使用方法及其在实际应用中的案例，也包含了所有重要参数的用法分析，如果没有特殊需求，不加任何参数是最快最高效率的方式。同时，capinfos是Wireshark套件中一个实用的命令行工具，方便快速查看抓包文件（包括但不限于pcap、pcapng等）的元数据信息，包括文件类型、数据链路层类型、数据包数量、文件大小、捕获持续时间等，利于快速定位抓包文件是否覆盖到异常时间点。