

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。



Linux

二、安装	ail. com
Linux	- dedille
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	emergeask wireshark

Windows

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

C □ > 此电脑 > 软件(D:) > Wireshark >

① □ ④ ② □ 1 排序 → 三 查看 → •••

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:

🖭 C:\Windows\System32\cmd.e 🛛 🗙 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>

👂 🖉 0 11:09:36 / 🖕	~/pkgs/capinfos capinfos <u>1.pcap</u>
File name:	1.pcap
File type:	Wireshark/tcpdump/ pcap
File encapsulation:	Ethernet
File timestamp precis	sion: 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
▶ 0 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:

n	Protocol L	ength	Time to Live	MSS Valu Wi	ndow Identific	ation	Seq(raw)	Next Seq	Ack	Segment Len	Frame length on the wire	Info	
.161	TCP	86	24,64	1	502 0x000) (0),0x	. 3047677759	304767775	59 111743341	0	8	6 501	14 -
.161	TCP	437	24,64	1	502 0x000) (0),0x	. 3047677759	304767811	111743341	351	43	7 501	14 -
19	TCP	86	64,63	3	60 0x000) (0),0x	. 111743341	111743341	304767811	0 0	8	6 801	3 →
19 📕 Wire	shark · Capture I	File Prope	rties · 1.pcap								_		>
.1													
Details													
File													
19													
Name				F:\Downloads\1.pcap)								
Lengt	1:			240 IVIB 11ofoEb72678dEf0E	oochf4602c0fbr	o10166bd07	022272602667666	72fcEo					
19 Hash	SHA1):			64420e671a1a0a698	e4df0ed2b83a7	891477d48	023273103017160.	in sitese					
1c Forma	t:			Wireshark/tcpdump	pcap	00011111010							
Encap	sulation:			Ethernet									
Snaps	hot length:			2048									
Te Time													
. I lime													
📕 🛛 First p	acket:			2024-02-29 19:40:47									
Last p	acket:			2024-02-29 20:20:24									
.1 Elapse	d:			00:39:37									
19 Captu	re												
19													
1 Hardy	/are:			Unknown									
16 OS:	ation			Unknown									
I.1 Applic	ation:			Unknown									
11 Interf	aces												
1.1 Interf				Droppod packate		Contur	filtor		Link type		Packet cize limit (coapleo)		
L.1 Unkno	wn			Unknown		Unknov	vn		Ethernet		2048 bytes		
16						0.111101			Luioniot		2010 0 (00		
16 Statist	ics												
.1 Measu	rement			Captured			Dist	played		Marked			
1.1 Packe	is			1836574			183	6574 (100.0%)		_			
.1 Time	pan, s			2377.258			237	7.258		_			
24 Avera	ge pps			772.6			772	.6		-			
.1 Avera	ge packet size,	В		156	_		156			_			
.1 Bytes	na hustaa (a			28600971	(286	009717 (100.0%)		0			
24 Avera	je bytes/s			120 K			120	κ ν		_			
24	90 2003/3			JOL K			502						
-													-
Capture	e file comment	s											
s)													
e													
08 Refr	esh									Save Comments	Close Copy To Clipboard	H	elp
						_							

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>

Ø 23:18:58	<pre>> ~/p/mergecap > capinfos -t http-1.txt</pre>
File name:	http-1.txt
File type:	Wireshark/ pcapng
Ø 23:19:38	<pre>> ~/p/mergecap > capinfos -t *</pre>
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
Ø 23:19:45	<pre>> ~/pkgs/mergecap file http-1.txt</pre>
http-1.txt: pcapn	g 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>
                ~/pkgs/3.1/10.1.14.96 capinfos
    o 23:31:27
File name:
                    1.pcap
File encapsulation: Ethernet
                    inferred: 192 bytes
Packet size limit:
File name:
                    2.pcap
ile 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 (-1)

- 1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>

Ø 23:55:06	~/pkgs/3.1/10.1.14.96
File name:	1.pcap
Packet size limit:	inferred: 192 bytes
Number of interfaces	in file: 1
Interface #0 info:	
	Encapsulation = Ethernet (I - ether)
	Lapture Length = 2048
	Time precision = microseconds (6)
	Number of stat optrios - A
	Number of packats $= 1011713$
	Number of packets – 1911/15
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 ncan
Number of interfaces	in file 1
Interface #0 info:	
	<pre>Encapsulation = Linux cooked-mode capture v2 (210 - linux-sll2)</pre>
	Capture length = 262144
	Time precision = microseconds (6)
	Time ticks per second = 1000000
	Number of stat entries = 0
	Number of packets = 6
👂 🖉 0 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>
```

```
File name: 1.pcap
File timestamp precision: microseconds (6)
Packet size limit: inferred: 192 bytes
                      2.pcap
File name:
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
2
```

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>

Ø 00:15:13	~/pkgs/3.1/10.1.14.96 capinfos -H *
File name:	I.pcap
Packet size limit:	Interred: 192 bytes
	12001b2070001951000001951000010001000970232751050171005751C50
	4290006260047042026086090064673669200051 6440062716120560844560370801477448
SHAI:	0442000/1d1d0d09004010002D03d/00914//040
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:	52b6a5a405452039e†21aa13d80†9b49258b3633
File name:	test.pcap
SHA256:	4aab7c78b50445480eb4dc2f67deaf2dd3e2816010d484c05a8638388b9c32e5
RIPEMD160:	55ab9e6616d9f4882ddc44bfd806f3983bfbe837
SHA1:	c3ca49814e595ebf505fd4a56f81c30f747d6e0a
🤌 🖉 00:15:20 / ⊵	~/pkgs/3.1/10.1.14.96
	0
っ文仕大小	
	$\mathcal{A}_{\mathcal{O}}$
1) 日二句 (
・)水小らて(

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 / 🖻 -	<pre>~/pkgs/capinfos > capinfos -c *</pre>
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:



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 🖊 🍃 /	<pre>~/pkgs/capinfos > capinfos -s *</pre>
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
ا © 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 File name: 1.pcap Packet size limit: Data size: inferred: 192 bytes 297 MB File name: 2.pcap Packet size limit: inferred: 192 bytes Data size: 291 MB File name: http-2.pcap 726 bytes Data size: 0 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:



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 -1 <文件名>

🦻 🖉 02:35:22 / ⊳ /	<pre>~/pkgs/capinfos > capinfos -l *</pre>
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)
Ø 02:36:01	~/pkgs/capinfos

输出含义如下:

File name:1.pcapPacket size limit:file hdr: 2048 bytes #抓包设置的每个帧最大抓包Length
inferred: 192 bytes #根据包文件里的帧推断的LengthFile name:2.pcapPacket size limit:file hdr: 2048 bytes #抓包设置的每个帧最大抓包Length
inferred: 192 bytes #根据包文件里的帧推断的LengthFile name:http-2.pcapPacket size limit:http-2.pcapFile name:http-2.pcapFile 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 / 🖻 ∕	<pre>~/pkgs/capinfos > capinfos -u *</pre>
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
Ø 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>



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'

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
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:

	t ud -Y	'fra	me.numb	er==1'			A.				
	fee tobo	ale a		a da und	V. L.f. nome	-	11				
O 05:00:57 = ~/pkgs/capin		ск - п	-r <u>pca</u>	<u>-</u> - c ua -		e.numbe	r==1.				
Running as user "root" and group	"root". If	nis co	buld be da	angerous.							
<u>1 2024-02-29</u> 11:40:47.594971	10.1.7.1	151 →	10.1.14.1	161 TCP	86 17591	L → 801:	3 [RST,	ACK]	Seq=1	Ack=1	Win=501 Len=0
Ø 03:09:03 / > ~/pkgs/capin	fos										,
-tud统计的是UTC时间。	霊要在比り	甚础	ト+8才是	北京时	NAT.						
					01-11						
			- \'A'								
TIMirocharle的描述文件常件	+ +++++++++++++++++++++++++++++++++++++	- / · II-									
1 + VV + E > 1 = 1 + E + E + E + E + E + E + E + E + E +	丰, 113.864	有判旧	白尾包ボ	t间:							
11 WITESTIAL KUTTING 人工 IT IS IT IS	主,也形1	首 到	自尾包的	间:							
1上1111125110110131用3人又计周江	主,也形约	首 到[自尾包的	间:							
	主,也形的	「判」	自尾包的	间:							
A	主, U月已4	自到 Protocol	自尾包的 Length Time to Live	前: MSS Valu Window	Identification	Seq(raw)	Next Seq	Ack	Segment Len	Frame lengt	th on the wire Info
▲ Lepty a display 開ビー < Cht / > ▲ Lepty a display 開ビー < Cht / > No. Date Deta Source 191708 2024-92-29 20:21:54, 388775 0, 002267 10.1.7.111 191708 2024-92-29 20:21:54, 388775 0, 002277 10.1.7.111 191708 2024-92-29 20:21:54, 388775 0, 002277 10.1.7.111 191708 2024-92-29 20:21:54, 388775 0, 002277 10.1.7.111 191708 2024-92-29 20:21:54, 38775 0, 00227 10.1.7.111 191708 2024-92-29 20:21:54, 38775 0, 00227 10.1.7.111 191708 2024-92-29 20:21:54, 38775 0, 00257 10.1.7.111 191708 2024 191708 2024 1917	世, U月E4	自到! Protocol TCP	自尾包的 Length Time to Live 98 24,64	MSS Valu Window 1424 64240	Identification 8x8608 (0),9x. 8x8608 (0) 9x	Seq(raw) 3556233121 2796353236	Next Seq 3556233122 2796553237	Ack 0	Segment Len	Frame lengt	th on the wire Info 98 48625 → 8013 [SVN] 98 8013 → 48675 [SVN]
ALPEPtradiction Child Chili Child Child	Destination 10.1.14.161 10.1.7.111 10.1.14.161	自到 Protocol TCP TCP	自尾包的 Length Time to Live 98 24,64 98 624,64 86 24,64	MSS Valu Window 1424 64240 1424 29208 582	Identification 0x0000 (0),0x 0x0000 (0),0x 0x0000 (0),0x	Seq(raw) 3556233121 2796353236 3556233122	Next Seq 3556233122 2796353237 3556233122	Ack 0 35562331 27963532	Segment Len Ø Ø	Frame lengt	thon the wire info 98 48625 → 8913 (SYN) 98 8913 → 48625 (SYN) 86 48625 → 8913 (AcX)
A LAPPA display filter - <ctbr></ctbr> No. Date Ctbr/> 1911708 2024-02-29 20:21:54.388775 0.00272 18.1.1.4.11 191171 0.2024-02-29 20:21:54.388757 0.00272 18.1.14.10 1911711 2024-02-29 20:21:54.38957 0.001452 10.1.7.111 1911711 2024-02-29 20:21:54.38957 0.001452 10.1.7.111 191171 2024-02-29 20:21:54.389567 0.001452 10.1.7.111	Destination 10.1.14.161 10.1.7.111 10.1.14.161 10.1.14.161	Frotocol TCP TCP TCP TCP	自尾包的 Length Time to Live 98 24,64 98 64,63 86 24,64 511 24,64	MSS Valu Window 1424 64248 1424 29208 502 502	identification 0x0000 (0),0x 0x0000 (0),0x 0x0000 (0),0x	Seq(raw) 3556233121 2796353236 3556233122 3556233122	Next Seq 3556233122 2796353237 3556233122 3556233547	Ack 0 35562331 27963532 27963532	Segment Len Ø Ø 425	Frame lengt	th on the wire info 98 48625 + 8913 [SYN] 98 6013 + 48625 [SYN], 86 48625 + 8813 [ACK] 511 48625 + 8813 [PSH],
Alphysical display filter Cht /> No. Date Date Source 1312768 2024-02-22 20121154,388735 0,609072 10.1.1.4.161 1312768 2024-02-22 20121154,388735 0,609072 10.1.1.4.161 1312716 2024-02-22 20121154,388735 0,609072 10.1.1.4.161 1312716 2024-02-22 20121154,399625 0,600072 10.1.1.4.161 1312712 2024-02-22 20121154,399625 0,600071 10.1.1.4.161 1312712 2024-02-22 20121154,399625 0,600071 10.1.1.4.161 1312712 2024-02-22 20121154,399625 0,600071 10.1.1.4.161	Destination 10.1.14.161 10.1.7.111 10.1.14.161 10.1.14.161 10.1.7.111	Protocol TCP TCP TCP TCP TCP TCP	自尾包的 1000 - 10000 - 10000 - 1000 - 1000 - 1000 - 1000 - 1000 - 1000 - 1000 -	MSS Valu Window 1424 64248 1424 29208 502 502 60	Identification 0x0000 (0),0x 0x0000 (0),0x 0x0000 (0),0x 0x0000 (0),0x 0x0000 (0),0x	Seq(raw) 3556233121 2796353236 3556233122 3556233122 2796353237	Next Seq 3556233122 2796353237 3556233122 3556233547 2796353237	Ack 0 35562331 27963532 35562335	Segment Len 0 0 425 0	Frame lengt	th on the wire into 98 48625 - 8013 (SYNI) 98 8013 - 48625 (SYNI) 86 48625 - 8013 (ACK) 511 48625 - 8013 (PSH) 86 8013 - 48625 (ACK)
ALL WITESTIGIKUJJ田公人又「十/丙」エ ALL WITESTIGIKUJJ田公人又「十/丙」エ No. Date Source 1911768 2024-02-29 20:21:54.388778 0.000226 10:1.7.111 191179 2024-02-29 20:21:54.388787 0.00027 10:1.14.161 191171 2024-02-29 20:21:54.398673 0.00021 30:1.7.111 191171 2024-02-29 20:21:54.398673 0.00027 10:1.14.161 191171 2024-02-29 20:21:54.398673 0.00027 10:1.14.161	Destination 10.1.14.161 10.1.7.111 10.1.7.111 10.1.4.161 10.1.7.111 10.1.7.111	Protocol TCP TCP TCP TCP TCP TCP	自尾包的 98 24,64 98 64,63 86 24,64 511 24,64 86 64,63 332 64 63	MSS Valu Window 1424 64240 1424 29200 502 502 502 60 60	Identification 0x0000 (0),0x.	Seq(raw) 3556233121 2796353236 3556233122 3556233122 2796353237 2796353237	Next Seq 3556233122 2796353237 3556233122 3556233122 3556233547 2796353237 2796353483	Ack 9 35562331 27963532 35562335 35562335	Segment Len 0 0 425 0 246	Frame lengt	th on the wire info 98 48625 - 8813 [SYN] 98 8813 - 48625 [SYN], 86 48625 + 8813 [SYN], 86 8813 - 48625 [Ack] 32 4179/1.1 2000 (F)
Date Date Source 191708 2024-02-2 20212154,38875 0.00227 10.1.7.111 191709 2024-02-2 20212154,38875 0.00227 10.1.7.111 191709 2024-02-2 2021354,398675 0.00227 10.1.7.111 191719 2024-02-2 2021354,399675 0.00027 10.1.7.111 191712 2024-02-2 2021354,399675 0.00027 10.1.4.61 191712 2024-02-2 2021354,399675 0.00027 10.1.4.161 191712 2024-02-2 2021354,399718 0.00027 10.1.4.161	Destination 10.1.14.361 10.1.7.111 10.1.7.111 10.1.14.361 10.1.7.111 10.1.7.111	Protocol TCP TCP TCP TCP TCP	自尾包的 98 24,64 98 64,63 86 24,64 511 24,64 86 64,63 332 66 63 332 66 63	MSS Valu Window 1424 64240 1424 292000 502 502 502 60 40 reperties - 1.pr.ps	Identification 0x0000 (0),0x	Seq(raw) 3556233121 2796352336 3556233122 2556233122 2796353237 2796353237	Next Seq 3556233122 2796353237 3556233142 3556233547 2796353237 2796353483	Ack 9 35562331 27963532 35562335 35562335	Segment Len 0 0 425 0 -	Frame lengt	th on the wire into 98 48625 + 8013 [SYN] 98 8013 + 48625 [SYN] 86 48625 + 8013 [CSN] 511 48625 + 8013 [CSN] 86 8013 + 84625 [ACC] 352 HTTP/1-1 200 OK [Pa
Alpepte a display filter _ < Alpepte a display filter _ <	Destination 10.1.14.401 10.1.14.161 10.1.14.161 10.1.14.161 10.1.17.111 10.1.17.111	Protocol TCP TCP TCP TCP TCP	自尾包的 98 24,64 98 64,63 86 24,64 511 24,64 86 64,63 322 64 63 322 64 64 326 64 63 326 64 64 36 64 63 326 64 64 36 64 63 36 64 64 36 64 64 64 76 64 64 76 64 76 64 76 64 76 64 76 64 76 64 76 76 76 76 76 76 76 76 76 76 76 76 76 7	が見ていた。 MSS Valu Window 1424 64240 1424 29200 502 502 502 60 40 Yroperties・1,0cap	Identification 0x0600 (0),0x. 0x0600 (0),0x. 0x0600 (0),0x. 0x0600 (0),0x. 0x0600 (0),0x. 0x0600 (0),0x.	Seq(raw) 3556233121 2796353236 3556233122 2796353237 2796353237	Next Seq 3556233122 2796353237 3556233122 3556233122 3556233547 2796353483	Ack 0 35562331 27963532 35562335 35562335	Segment Len 0 0 425 0 246 -	Frame lengt	th on the wire info 98 44525 - 8013 [SYN] 98 8013 - 44625 [SYN], 86 44525 - 8013 [SYN], 511 46252 - 8033 [SYN], 86 8013 - 48625 [ACK] 332 HTTP/1.1 200 OK [Pa
Date Date Source 1912769 3024-02-29 3021543,388715 0.002267 10.1.7.113 1912709 3024-02-29 3021543,388715 0.002267 10.1.7.113 1912709 3024-02-29 3021543,389625 0.002267 10.1.7.113 191270 3024-02-29 20212154,389625 0.00207 10.1.1.4.101 191271 3024-02-29 20212154,399625 0.00207 10.1.1.4.101 191271 3024-02-29 20212154,399613 0.00207 10.1.1.4.101 191271 3024-02-29 20212154,399613 0.00207 10.1.1.4.101 191271 3024-02-29 20212154,399613 0.00207 10.1.1.4.101 191271 3024-02-29 20212154,399130 0.00207 10.1.1.4.101 191271 3024-02-29 20212154,399130 0.00207 10.1.1.4.101 191271 3024-02-29 20212154,399140 0.00207 10.1.1.4.101 191271 3024-02-29 20212154,391140 0.00207 10.1.1.4.101 191271 3024-02	Destination 10.1.14.161 10.1.7.111 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.7.111 10.1.7.111 10.1.7.111	Protocol TCP TCP TCP TCP TCP	自尾包的 98 24,64 98 64,63 86 24,64 511 24,66 86 64,63 832 cs.62 Wreshark-Capture File P 244418 Name:	MSS Valu Window 1424 64240 1424 5920 592 592 592 592 592 592 592 592	Identification 0x0600 (0),0x. 0x0600 (0),0x. 0x060 (0),0	Seq(raw) 3556233121 2796353236 3556233122 2796353237 2796353237	Next Seq 3556233122 2796352337 3556233122 3556233547 279635323547 2796353237 2796353483	Ack 0 35562331 27963532 35562335 35562335 35562335	Segment Len Ø Ø 425 Ø 246 –	Frame lengt	th on the wire info 36 44625 - 8033 [SYN] 36 44625 - 8031 [SY], 51 46525 - 8031 [SY], 56 6013 - 8632 [SYN] 56 6013 - 8632 [SYN] 56 2013 - 8632 [SYN] 50 2017 - 1 200 00 [Pa
ALL VUIT CSLIGAT KLL3]HBAA & KLT//BLL2 a[Apply a display filter <cht></cht> b No Date Data Source 1911708<0204-02-29	Destination 10.1.14.401 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111	Protocol TCP TCP TCP TCP TCP TCP TCP TCP	自尾包的 98 24,64 98 64,63 86 24,64 98 64,63 86 64,63 332 66 64,63 332 66 64,63 332 66 64,63 332 66 8 Weethark-Capture File P Veethark-Capture File P	MSS Valu Window 1424 64240 1424 29200 502 502 60 toperties - 1 pcap	Identification 9x0008 (0),9x, 9x0008 (0),9x, 9x0000 (0),9x, 9x, 9x000 (0),9x, 9x, 9x, 9x, 9x, 9x, 9x, 9x, 9x, 9x,	Seq(raw) 3556233121 2796353236 3556233122 279635237 2796353237	Next Seq 3556233122 2796353237 3556233122 3556233547 2796353237 2796353483	Ack 8 35562331 27963532 35562335 35562335	Segment Len 0 0 425 0 - -	Frame lengt	th on the wire Info 98 48625 + 8913 [SYN] 98 8013 + 48625 [SYN, 86 48625 + 8013 [ACI] 511 48525 + 8013 [SN, 86 8013 + 84825 [ACI] 332 HTTP/1.1 200 OK [Pa
Date Date Source 191708 3024-02-20 2012154,38873 6,06237 16,12.7.11 191708 3024-02-20 2012154,38873 6,06237 16,1.7.11 191708 3024-02-20 2012154,39873 6,06237 16,1.7.11 191701 3024-02-20 2012154,399624 0,06023 16,1.7.111 191711 3024-02-20 2012154,399614 0,06023 16,1.7.111 191712 3024-02-20 2012154,399124 0,06023 16,1.7.111 191712 3024-02-20 2012154,399124 0,06023 16,1.7.111 191712 3024-02-20 2012154,399124 0,06023 16,1.7.111 191712 3024-02-20 2012154,399124 0,06023 16,1.7.111 191712 3024-02-20 2012154,399124 0,06023 16,1.7.111 191712 3024-02-20 2012154,399124 0,060231 16,1.7.111 191712 3024-02-20 2012154,399124 0,060231 16,1.7.111 191712 3024-02-20 2012154,399124<	Destination 10:.1.14.161 10:.1.7.111 10:.1.7.111 10:.1.14.161 10:.1.7.111 10:.1.7.111 10:.1.7.111 88: bits) st: Broadcom_0c:6d:07 7.211.252	Рионской ТСР ТСР ТСР ТСР ТСР ТСР ТСР ТСР ТСР ТСР	Length Time to Live 98 24,64 99 64,63 86 24,64 85 24,64 85 24,64 85 64,63 86 64,63 86 64,63 86 64,63 87 64 86 64,63 87 64,63 88 62,64 88 62,64	MSS Valu Window 1424 64240 1424 65240 1602 592 592 592 592 592 592 592 592 592 592 592 592 592 592 592 592 592 593 11etes572670539 11etes5726705395 14 alobes	Identification 0x0600 (0),0X. 0x0600 (0),0X	Seq(raw) 3556233121 2796353236 3556233122 3556233122 2796353237 2796353237 2796353237	Next Seq 3556233122 2796353237 3556233522 3556233547 2796353237 2796353483	Ack 0 35562331 27963532 35562335 35562335	Segment Len 0 425 0 -	Frame lengt	th on the wire lob. 98 48625 - 8013 [SYN] 98 48625 - 4003 [SYN] 98 64822 - 4003 [SYN] 514 48625 - 8013 [SYN] 514 48625 - 8013 [SYN] 332 HTTP/1.1 200 OK [Pa
A Charles Deta Source a Paper a display filter - Charles Deta Source b Date Deta Source Source 1912/08 2024-02-29 20:21:54,3883715 0.000271 06.1.7.111 1917/10 2024-02-29 20:21:54,389175 0.000071 06.1.14.161 1917/10 2024-02-29 20:21:54,390629 0.000071 106.1.14.161 1917/13 2024-02-29 20:21:54,390124 0.000071 106.1.14.161 917/17 2024-02-29 20:21:54,390124 0.000071 106.1.14.161 0.0000971 106.1.14.161 917/17 2024-02-29 20:21:54,390124 0.0000971 106.1.14.161 0.0000971 106.1.14.161 917/17 2024-02-29 20:21:54,390124 0.0000971 106.1.14.161 0.0000971 106.1.14.161 917/17 2024-02-29 20:21:54,390124 0.0000971 106.1.14.161 0.0000971 106.1.14.161 917/17 2024-02-29 20:21:54,390124 0.0000971 106.1.14.161 0.0000971 106.1.14.161 917/17 2024-02-29 20:21:54,390124 0.0000971 106.1.14.161 0.0000971 106.1.14.161 917/17 2024-02-29 20:21:54,390124 0.0000971 106.1.14.161 0.0000971 106.1.14.161 917/17 2024-02-29 20:21:54,390124 0.0000971 106.1.14.161 0.0000971 106.1.14.161 917/17 2024-02-29 20:21:54,3	Destination 10.1.14.161 10.1.7.11 10.1.7.11 10.1.14.161 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111 10.1.7.111	Protocol TCP TCP TCP TCP TCP TCP TCP TCP TCP TCP	Imme to Live 98 24,64 98 24,64 56 24,64 511 24,64 533 26,63 333 26,63 Wreshak: Cipture File P Nume: Length Length Hah (SM256): Hah (SM12): Format:	MSS Valu Window 1424 64248 1424 29208 502 502 60 roperties - 1 2008 FADownloads/1.pcr. 250 MB TiefeS7278d359 64420671 at 8008	Identification 8x80608 (8),9x. 8x80608 (8),9x. 8x808 (8),9x. 8x80608 (Seq(raw) 3556233121 2796353236 3556233122 3556233122 2796353237 2796353237 2796353237 66bd97023273f03b	Next Seq 3556233122 2796353237 3556233122 2796353237 2796353237 2796353237 (7feb373fc5e	Ack 9 35562331 27963532 35562335 35562335	Segment Len 0 425 0 246 –	Frame lengt	thon the wire Info 98 48625 + 8013 [SYN] 98 8013 + 48055 [SYN], 86 48625 + 8033 [Act] S11 48052 + 8033 [PSN], 86 8013 + 48625 [Act] 332 HTTP/1.1 200 OK [Pa
Alphysical display filter = < < Alphysical display filter = <<	Destination 10:1.14.161 10:1.7.111 10:1	Protocol TCP TCP TCP TCP TCP HTTP HTTP 42, Ack:	Length Time to Live 98 24, 64 98 64, 63 86 24, 64 86 24, 64 86 24, 64 86 64, 63 83 24, 64 86 64, 64 86 64, 64 Wethate: Capture File P Wethate: Capture File P Wethate: Capture File P Wethate: Capture File P Hath (SHAT): format: formapulation: formapulation:	MSS Valu Window 1424 64240 1424 29200 5902 590 5902 590 5902 590 590 290 590 290 590 100 590 100 500 1000 500 100 500 500 100 500 500 100 500 500 500 500 500 500 500 500 500	Identification 0x0000 (0),0x. 0x0000 (0),0x. 0x000 (0),0x	Seq(raw) 3556233121 2796353236 3556233122 2796353232 2796353237 2796353237 2796353237	Next Seq 3556233122 279635223 3556233122 3556233547 27963523547 27963523547 27963523547 27963523547	Ack 0 35562331 27963532 35562335 35562335 35562335	Segment Len 0 0 425 0 -246 -	Frame lengt	th on the wire into 98 48252 - 8013 [5YN] 98 4803 - 84823 [SYN] 98 48023 - 84823 [SYN] 91 48025 : 8013 [SYN] 96 8013 → 48025 [SCK] 32 HTTP/1.1 200 OK [Pa 1 1 1 1 1 1 1 1
Image: Source and the second	Destination 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.7.111 10.1.7.111 10.1.7.111 88 bits) st: Broadcom_@c:6d(1 7.111.252 .161	Protocol TCP TCP TCP TCP TCP HTTP HTTP 42, Ack:	Imme to Live 98 24,64 98 64,63 86 24,64 98 64,63 383 c6,46 383 c6,46 384 c6,463 383 c6,463 384 c6,463 184 c14,2565 Fash (SH4256) Fash (SH4256)	MSS Valu Window 1424 64240 1424 64240 1424 5922 592 592 592 592 592 592 592	Identification 0x0000 (0),0x. 0x0000 (0),0x. 0x000 (0),0x.	Seq(raw) 3556(23)121 2796353236 355633122 2796353237 355633122 2796353237 37964353237 37964353237 666697023273036	Next Seq 27665523122 2556233122 355623342 255623354 2796353237 2796353287	Ack 9 35562331 27963532 35562335 35562335 35562335	Segment Len 0 0 425 - 246 - -	Frame lengt	h on the wire Info 96 44625 + 8013 [SYN] 96 8013 - 48622 [SYN] 86 44625 + 8013 [CSN] S11 46625 + 8013 [CSN] 352 HTTP/1.1 200 OK [Pa UNION CONTINUE OF CONTIN
Alternative Control Date Source 191708<0204-02-23	Destination 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.161 10.1.14.162 10.1.14.162 10.1.14.162 10.1	Protocol TCP TCP TCP TCP HTTP HTTP 42, Ack:	Length Terre to Live 98 64, 63 98 64, 63 86 62, 64 85 62, 64 85 85 84 85 85 85 85 85 85 85 85 85 85	MSS Valu Window MSS Valu Window 1424 64249 1424 29200 592 592 69 29300 114557763559 1145507763559 1145507763559 1145507763559 Wireshark/Copding Ethernet 2046	Identification 8x8696 (0),9x. 8x8696 (0),9x.	Seq(raw) 355623121 2796353236 3556233122 2796353237 2796353237 2796353237 4964597023273030	Next Seq 3556233122 2796353237 3556233123 3556233122 2796353237 7766353483	Ack 6 35562331 27963532 35562333 35562335 35562335	Segment Len 0 0 425 - 246 - 0 246	Frame lengt	10 on The wire Info 98 44525 = 48013 [5Wi] 98 8435 = 48625 [5Wi] 98 8435 = 48625 [3Ki] 511 48625 = 48013 [5Wi], 86 8013 ⇒ 48625 [ACI] 322 HTTP/1.1 200 OK [Pa Dest 50
Prame 1: 86 bytes on wire (688 bits), 86 bytes captured (6 > Thermet TL, Src: 1167-WRC WRC WRC WRC WRC WRC WRC WRC WRC WRC	Destination 10.1.14.161 10.1.7.111 10.1	Protocol TCP TCP TCP TCP TCP TCP TCP TCP	Image: Control of the second	MSS Valu Window 1424 64248 1424 64248 1424 64248 1424 64248 1424 64248 1424 64248 1424 64248 1424 64248 1424 64248 1424 64248 1424 64288 1424 64288 1425 14408 14428 64288 14428 14488 14428 14488 14428 14488 14428 14488 14428 14488 14428 14488 14428 14488 14488 14488 14488 14488 14488 14488 14488 14488 14488 14488 14488 14488 14488 14488 14488 14488 14488 144888 14488 <	Identification 0x0000 (0),0x. 0x0000 (0),0x. 0x000 (0),0x	Seq(raw) 3556233121 2796353226 3556233122 3556233122 279635237 279635237 279635237 666697023273036	Next Seq 3556233122 2796353237 355623354 2796353237 2796353237 2796353237	Ack 0 35562331 27963532 35562332 35562335 35562335	Segment Len 0 425 2346 -	Frame lengt	th on the wire with 99 44625 → 8033 [SYN], 99 8045 → 86823 [SYN], 95 80452 → 8083 [SYN], 511 46925 → 86823 [SYN], 56 8013 → 48933 [ACK] 332 HTTP/1.1 200 OK [Pa
Participation Control Deta Source 1912708 2024-02-23 20121154, 388715 0.00277 10.1.1.4.161 1912708 2024-02-23 20121154, 388715 0.00277 10.1.1.4.161 1912708 2024-02-23 20121154, 388715 0.00277 10.1.1.4.161 1912708 2024-02-23 20121154, 398623 0.00277 10.1.1.4.161 191271 2024-02-23 20121154, 399623 0.00277 10.1.1.4.161 191271 2024-02-23 20121154, 399713 0.002077 10.1.1.4.161 191271 2024-02-23 20121154, 399713 0.002077 10.1.1.4.161 191271 2024-02-23 20121154, 399713 0.002077 10.1.1.4.161 917171 2024-02-23 20121154, 399713 0.002077 10.1.1.4.161 917171 2024-02-23 20121154, 399713 0.002077 10.1.1.4.161 917181 57.001 57.001 57.001 57.001 57.001 9117181 57.001 57.001 57.001 57.001 57.001 <td>Desitorion 10.1.14.65 10.1.14.65 10.1.7.11 10.1.14.61 10.1.7.11 10.1.7.</td> <td>Protocol TCP TCP TCP TCP TCP TCP TCP TCP TCP TCP</td> <td>Length Time to Live 98 24,64 98 24,64 98 24,64 98 24,64 50 24</td> <td>MSS Valay Window 1122 66240 122 662 122 562 562 562 562 562 562 562 562 562 563 562 564 562 570 563 750 563 750 765 750 765 750 765 750 765 750 765 750 767 750 767 750 767 750 767 750 767 750 767 750 767 750 767 750 767 767 767 767 767 767 767 767 767 767 767 767 767 767 767 767</td> <td>Meterilication 0:00000 (0),0x 0x0000 (0),0x 0x000 (</td> <td>Seq(raw) 3556233121 279635226 355623122 356633122 279635227 279635227 279635227 279635227</td> <td>Next Seq 3556233122 2796352327 3556233154 3556233154 279635237 27963523483 77/eb373/c5e</td> <td>A:k 6 35562331 27963532 27963532 35562335 35562335</td> <td>Segment Len 0 0 425 0 246 - C</td> <td>Frame lengt</td> <td>th on the wire into 98 48625 - 8013 [SVII] 98 8813 - 48625 [SVI] 98 8813 - 48625 [SVI] 511 48625 - 8031 [SVI] 522 HTTP/1.1 200 OK [Pa Dest So Type</td>	Desitorion 10.1.14.65 10.1.14.65 10.1.7.11 10.1.14.61 10.1.7.11 10.1.7.	Protocol TCP TCP TCP TCP TCP TCP TCP TCP TCP TCP	Length Time to Live 98 24,64 98 24,64 98 24,64 98 24,64 50 24	MSS Valay Window 1122 66240 122 662 122 562 562 562 562 562 562 562 562 562 563 562 564 562 570 563 750 563 750 765 750 765 750 765 750 765 750 765 750 767 750 767 750 767 750 767 750 767 750 767 750 767 750 767 750 767 767 767 767 767 767 767 767 767 767 767 767 767 767 767 767	Meterilication 0:00000 (0),0x 0x0000 (0),0x 0x000 (Seq(raw) 3556233121 279635226 355623122 356633122 279635227 279635227 279635227 279635227	Next Seq 3556233122 2796352327 3556233154 3556233154 279635237 27963523483 77/eb373/c5e	A:k 6 35562331 27963532 27963532 35562335 35562335	Segment Len 0 0 425 0 246 - C	Frame lengt	th on the wire into 98 48625 - 8013 [SVII] 98 8813 - 48625 [SVI] 98 8813 - 48625 [SVI] 511 48625 - 8031 [SVI] 522 HTTP/1.1 200 OK [Pa Dest So Type

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 <mark>-e</mark> <包文件>

🌘 🖉 🖉 🖉 🕗 🖉	<pre>~/pkgs/capinfos capinfos -e *</pre>
File name:	1.pcap
Packet size limit:	inferred: 192 bytes
Last packet time:	2024-02-29 20:21:54.391104
Tile nome.	
File Hame:	z. prop
Packet Size timit:	interred: 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
Eilo namo:	
	Sum. Pcap
Packet size limit:	Interred: 60 bytes
Last packet time:	2024-02-29 02:03:03.223428
ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا	<mark>~/pkgs/capinfos ></mark> tshark -n -r <u>1.pcap</u> -t ud -Y 'frame.number==1911713'
Running as user "roo [.]	t" and group "root". This could be dangerous.
1911713 2024-02-29 13	2:21:54.391104 10.1.14.161 → 10.1.7.111 HTTP 332 HTTP/1.1 200 OK [Packet size limited during capture]
Ø 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 -<mark>a</mark> -e <包文件>

Ø 03:23:21	<pre>~/pkgs/capinfos > capinfos -a -e *</pre>
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 nacket time:	2024-02-29 02.03.03 222556
last nacket time.	2024 02 29 02:03:03:222330
	2024 02 23 02:03:03:223420
File name:	
Packet size limit.	inferred: 60 hytes
First nacket time:	$2024 - 02 - 20 = 02 \cdot 03 \cdot 03 = 221117$
last nacket time.	2024 02 23 02:03:03:221117
2 0 03.23.47	$\sim/nkgs/caninfos$
003.23.47	

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:

- 6	> (03:32:51	/ > ~/pkgs/capintos / tsnark -n -r sum-desc.pcap
R	unnir	ıg as user	"root" and group "root". This could be dangerous.
		0.00000	0 192.168.1.83 → 192.168.1.2 TCP 72 80 → 38812 [ACK] Seq=1 Ack=1 Win=509 Len=0 TSval=3114888908 TSecr=595211828
		0.00000) 192.168.1.83 → 192.168.1.2 HTTP 366 HTTP/1.1 403 Forbidden (text/html)
		0.000072	2 192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [ACK] Seq=1 Ack=295 Win=501 Len=0 TSval=595211829 TSecr=3114888909
		0.000496	5 192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [FIN, ACK] Seq=1 Ack=295 Win=501 Len=0 TSval=595211829 TSecr=3114888909
		0.00083	l 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
		0.000872	2 192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [ACK] Seq=2 Ack=296 Win=501 Len=0 TSval=595211829 TSecr=3114888909
		-0.001439	9 192.168.1.2 → 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=12
		-0.000621	1 192.168.1.83 → 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
		-0.000532	2 192.168.1.2 → 192.168.1.83 TCP 72 38812 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=595211828 TSecr=3114888908
	10	-0.000438	3 192.168.1.2 → 192.168.1.83 HTTP 147 GET / HTTP/1.1
	۶ (03:33:33	> ~/pkgs/capinfos > capinfos -o sum-desc.pcap
F	ile r	name:	sum-desc.pcap
5	trict	: time orde	er: False
	^	07.77.44	

而反观sum.pcap的包序,已经严格按照绝对时间排序,识别为True:

2	0	03:34:35	✓ ≥ ~/pkgs/capinfos > tshark -n -r sum.pcap
Run	nning	g as user '	'root" and group "root". This could be dangerous.
		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]
		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]
		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]
		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]
		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]
		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]
		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]
		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]
8	Ø	03:34:42	🖕 ~/pkgs/capinfos >> capinfos -o <u>sum.pcap</u>
Fil	.e na	ame:	sum.pcap
Pac	ket	size limi†	t: inferred: 60 bytes
Str	rict	time orde	r: True
	0	03.34.58	► -/pkgs/capinfos

路径下还有1.pcap、2.pcap识别为False:

0 03·36·05	<pre>~/nkgs/capinfos -0 *</pre>
Filo namo:	1 ncan
Dockot cizo limit.	informadi 102 hytoc
	Interfed: 192 bytes
Strict time order:	False
File name:	2.pcap
Packet size limit:	inferred: 192 bytes
Strict time order:	False
File name:	http-2 ncan
Strict time order:	Trun
	i i de
File name:	sum-desc.pcap
Strict time order:	False
File name:	sum.pcap
Packet size limit:	inferred: 60 bytes
Strict time order:	True
2 0 03·36·08	<pre>~/nkgs/caninfos</pre>

通过时间戳也可以判断, tshark时间格式设定为-td (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):

> 0	03:40:13	/ ⊳ ~/pkgs/capinfos) tshark -n -r 1.pcap -t d awk '\$2<0{print}' head
Runnin	ig as user	"root" and group "root". This could be dangerous.
248516	-0.00008	32 10.1.7.250 → 10.1.14.161 TCP 86 36299 → 8013 [ACK] Seq=1 Ack=1 Win=64256 Len=0
253405	-0.00001	10 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	-0.00006	54 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	-0.00040	01 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	-0.00360	01 10.1.7.250 → 10.1.14.161 TCP 86 28868 → 8013 [ACK] Seq=1 Ack=1 Win=64256 Len=0
683602	-0.00000	96 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	-0.00012	25 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	-0.00001	10 10.1.14.161 → 10.1.7.78 HTTP 332 HTTP/1.1 200 0K [Packet size limited during capture]
855040	-0.00000	02 100.127.206.60 → 10.1.14.161 HTTP 169 GET / HTTP/1.1
143556	1 -0.0000	004 10.1.7.19 → 10.1.14.161 TCP 86 17409 → 8013 [ACK] Seq=1 Ack=1 Win=64256 Len=0
٥ 📀	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>

🦻 🖉 🥺 👂 👂	<pre>~/pkgs/capinfos > capinfos -y *</pre>
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>

● 0 15:09:51 ►	<pre>~/pkgs/capinfos > capinfos -i *</pre>
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>

0 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 Ø 15:15:27 A ~/pkgs/capinfos
 Ø

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>

👂 🖉 0 15:15:27 🖉 🍃 -	<pre>~/pkgs/capinfos > capinfos -x *</pre>
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
● _ 0 15:17:11 / ⊳ -	<pre>~/pkgs/capinfos</pre>

统计逻辑也很简单,平均包速率=总包量/总时间,比如sum.pcap:

▶ 0 15:17:11	~/pkgs/capinfos > capinfos -u sum.pcap
File name:	sum.pcap
Packet size limit	: inferred: 60 bytes
Capture duration:	0.002311 seconds
▶ 0 15:19:27	🖕 ~/pkgs/capinfos) capinfos -c <u>sum.pcap</u>
File name:	sum.pcap
Packet size limit	: inferred: 60 bytes
Number of packets	: 10
▶ ◎ 15:20:01	<pre>> ~/pkgs/capinfos > awk 'BEGIN{print 10/0.002311}'</pre>
4327.13	
▶ ◎ 15:20:26	🖕 ~/pkgs/capinfos > capinfos -x <u>sum.pcap</u>
File name:	sum.pcap
Packet size limit	: inferred: 60 bytes
Average packet ra	te: 4,327 packets/s
≥ 0 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:

	010		©, fx																					
	A	в	C	D	E	F	G	н	1	1	ĸ	L	м	N	0	Р	Q	R	s	т	U	v	W	х
	File non	e File type	File encapsulati	on File time Pack	et size P	acket size F	acket siz	Number of	File size I	ata size (Capture duratio	Start time	ind time	Data byte rat	eData bit rate	(Average p	Average p	eSHA256	RIPENDI60 SH	1	Strict time order	Capture hardware	Capture oper-sys	Capture application
2	1. pcap	pcap	ether	microsecos	2048	192	192	1911713	250000081 :	297734080	2466.796133	40:47.6	21:54.4	4 120696.67	965573.30	155.74	774.98	11efe5b720	(4296dbe2et644	120e871	FALSE			
3	2. pcap	pcap	ether	microsecos	2048	192	192	2351995	250000019 :	291548903	2859.818457	06:02.5	53:42.3	3 101946.65	815573.18	123.95	822.43	1 159427474:	Sc0adaeSet360	lac8a90	FALSE			
1	http:2.p	capcaper	linux-s112	microsecos (not	set) n	/a r	u/a	6	1028	726	0.000872	03:03.2	03:03.1	2 832667.41	6661339.23	5 121	6881.55	9457350783	1c20ae0df1528	685840	TRUE			TShark (Wireshark) 4.0.7 (Git commit Oad1823cc090)
5	sun-desc	. (peaper	linux-s112	microsecos (not	set) n	/a r	u/a	10	1564	1105	0.002311	03:03.2	03:03.3	2 478149.75	3825198.33	110.5	4327.15	i 7dc7c0e37;	e5d7cae8f1643	3ade234	FALSE		Linux 6.1.31-gentoo-dist	Mergecap (Wireshark) 4.0.7 (Git commit Oad1823cc06
5	sun peap	pcap	linux-s112	microsecos	262144	60	60	10	784	1105	0.002311	03:03.2	03:03.3	2 478149.75	3825198.33	110.5	4327.15	i 23991c388-	010287415:083	25:2507	TRUE			

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:

2	© 11:08:31	🕒 > ~/pkgs/capinfos > capinfos -h grepA	
- /	generate all	. infos (default)	
2	© 11:08:34	<mark>′⊳ ~/pkgs/capinfos _</mark>	

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	В	С	D		E	F	G	н
1 1	File name	File type	Number of	Capture d	hData byte rate	(bytes/sec)	SHA256	RIPEMD160	SHA1
2	l. pcap	pcap	1911713	2466. 7961		120696. 6	11efe5b72678d5f95feecbf4603c0fbde10166bd97023273f03bf7feb373fc5e	4296dbe2e60a7ba2d2eb8ed9b6eae73ee5ccc831	64420e671a1a0a698e4df0ed2b83a70891477d48
3	2. pcap	pcap	2351995	2859.8185	5	101946.6	5 f59427474303eee6dadda916787e17eafeb02b62718cc2cc136d7af7bc33c52a	8c0adae8e06604843f7d5fb96d167d5228776c40	360ac8a90833186c54e747949357274d14236075
4	nttp-2.pc	pcapng	6	0.000872	2	832667.4	94b73b07d7eb9f4991a621359d8d61bec94c6df8f5971684d33aee8801984847	1c20ae0df7d21552af53daebf003e068d1de77fd	52b6a5a405452039ef21aa13d80f9b49258b3633
5	sum-desc.	pcapng	10	0.002311		478149.7	7dc7c0e37258bcb6da5c4cf1bc10c015f3026c9b5fa047255eacc6534ae3efd9	e5d7cae8f745caa745bb1e5f225b659bec44749d	643ade234f9fda3fbf998197fd47b7e88e6bb952
6	вин. рсар	pcap	10	0.002311	1	478149.7	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。