

# Hping3

---

hping3是一个能够发送自定义TCP/IP的网络工具，信息包和显示目标应答，就像ping一样ICMP回复。hping3可以处理碎片，并且几乎任意数据包大小和内容，使用命令行界面，并且在Hping3开始，已经开始支持脚本使用了。

——[antirez@invece.org](mailto:antirez@invece.org)

---

## 一，帮助手册

用法：hping3 host[选项]

-h—帮助显示此帮助

-v—版本显示版本

-c—计数数据包计数

-i—间隔等待（uX为X微秒，例如-i u1000）

---i u10000的快速别名（10包每秒）

--更快的别名-i u1000（100包每秒）

--以最快的速度发送数据包。不显示答复。

-n—数值输出

-q—安静

-I—接口名称（否则为默认路由接口）

-V—详细详细模式

-D—调试信息

-z—将ctrl+z绑定到ttl（默认为dst端口）

-Z—取消绑定取消绑定ctrl+z

--接收到的每个匹配数据包的蜂鸣音

模式

默认模式TCP

-0—rawip原始IP模式

-1—icmp icmp模式

-2—udp-udp模式

-8—扫描扫描模式。

示例：hping--scan 1-30,70-90-S [www.target.host](http://www.target.host)

-9—监听模式

知识产权

-a—欺骗源地址

- 随机目标地址模式。看那个人。
- 随机源地地址模式。看那个人。
- t—ttl ttl (默认64)
- N—id id (默认随机)
- W—win id使用win\*id字节排序
- r—rel relativize id字段 (用于估计主机流量)
- f—将数据包分成多个碎片。(可能通过弱acl)
- x—morefrag设置更多片段标志
- y—dontfrag set不分段标志
- g—fragoff设置片段偏移量
- m—mtu设置虚拟mtu, 表示—frag if packet size>mtu
- o—tos服务类型 (默认0x00), try--tos帮助
- G--ROUTE包含记录路由选项并显示路由缓冲区
- lsrr松散源路由和记录路由
- ssrr严格源路由和记录路由
- H--ipproto设置IP协议字段, 仅在原始IP模式下
- ICMP公司
- C—icmp type icmp类型 (默认回显请求)
- K—icmp code icmp代码 (默认为0)
- 强制icmp发送所有icmp类型 (默认仅发送支持的类型)
- icmp gw设置icmp重定向的网关地址 (默认为0.0.0.0)
- icmp ts别名--icmp--icmp type 13 (icmp时间戳)
- icmp addr别名--icmp--icmp type 17 (icmp地址子网掩码)
- icmp帮助显示其他icmp选项的帮助
- UDP/TCP协议
- s—基本端口基本源端口 (默认随机)
- p--destport[+][+]目标端口 (默认为0) ctrl+z inc/dec
- k—保持源端口不变
- w—winsize (默认64)
- O—tcpoff设置假tcp数据偏移量 (而不是tcphdrlen/4)
- Q—seqnum只显示tcp序列号
- b—badcksum (尝试) 发送IP校验和错误的数据包
- 许多系统将修复发送数据包的IP校验和
- 所以你会得到坏的UDP/TCP校验和。
- M—setseq设置TCP序列号
- L—设置TCP ack
- F—fin集合fin标志
- S—syn设置syn标志
- R—重新设置重新设置标志
- P—推集推标志
- A—ack集合ack标志
- U—urg设置urg旗
- X—xmas设置X未使用标志 (0x40)
- Y—ymas设置Y未使用标志 (0x80)
- tcp exit code使用最后一个tcp->th\_标志作为退出代码
- tcp mss启用具有给定值的tcp mss选项
- tcp timestamp启用tcp timestamp选项来猜测HZ/uptime
- 普通的
- d—数据数据大小 (默认为0)
- E—文件中的文件数据
- e—签名添加“签名”
- j—转储十六进制数据包

-J—打印转储可打印字符  
-B——安全启用“安全”协议  
-u--end告诉您--file何时到达EOF并防止倒带  
-T—traceroute traceroute模式（表示--bind和--ttl 1）  
--在traceroute模式下接收第一个非ICMP时tr stop Exit  
--tr keep ttl keep the source ttl fixed，仅用于监视一个跃点  
--tr no rtt在traceroute模式下不计算/显示rtt信息  
ARS数据包描述（新的，不稳定的）  
--apd发送用apd描述的数据包（见docs/apd.txt）

usage: hping3 host [options]

-h --help show this help  
-v --version show version  
-c --count packet count  
-i --interval wait (uX for X microseconds, for example -i u1000)  
--fast alias for -i u10000 (10 packets for second)  
--faster alias for -i u1000 (100 packets for second)  
--flood sent packets as fast as possible. Don't show replies.  
-n --numeric numeric output  
-q --quiet quiet  
-I --interface interface name (otherwise default routing interface)  
-V --verbose verbose mode  
-D --debug debugging info  
-z --bind bind ctrl+z to ttl (default to dst port)  
-Z --unbind unbind ctrl+z  
--beep beep for every matching packet received

Mode

default mode TCP  
-0 --rawip RAW IP mode  
-1 --icmp ICMP mode  
-2 --udp UDP mode  
-8 --scan SCAN mode.

Example: hping --scan 1-30,70-90 -S [www.target.host](http://www.target.host)

-9 --listen listen mode

IP

-a --spoof spoof source address  
--rand-dest random destination address mode. see the man.  
--rand-source random source address mode. see the man.  
-t --ttl ttl (default 64)  
-N --id id (default random)  
-W --winid use win\* id byte ordering  
-r --rel relativize id field (to estimate host traffic)  
-f --frag split packets in more frag. (may pass weak acl)  
-x --morefrag set more fragments flag  
-y --dontfrag set don't fragment flag  
-g --fragoff set the fragment offset  
-m --mtu set virtual mtu, implies --frag if packet size > mtu  
-o --tos type of service (default 0x00), try --tos help  
-G --rroute includes RECORD\_ROUTE option and display the route buffer  
--lsrr loose source routing and record route  
--ssrr strict source routing and record route

-H --ipproto set the IP protocol field, only in RAW IP mode

#### ICMP

-C --icmptype icmp type (default echo request)

-K --icmpcode icmp code (default 0)

--force-icmp send all icmp types (default send only supported types)

--icmp-gw set gateway address for ICMP redirect (default 0.0.0.0)

--icmp-ts Alias for --icmp --icmptype 13 (ICMP timestamp)

--icmp-addr Alias for --icmp --icmptype 17 (ICMP address subnet mask)

--icmp-help display help for others icmp options

#### UDP/TCP

-s --baseport base source port (default random)

-p --destport [+][+] destination port(default 0) ctrl+z inc/dec

-k --keep keep still source port

-w --win winsize (default 64)

-O --tcpoff set fake tcp data offset (instead of tcphdrlen / 4)

-Q --seqnum shows only tcp sequence number

-b --badcksum (try to) send packets with a bad IP checksum  
many systems will fix the IP checksum sending the packet  
so you'll get bad UDP/TCP checksum instead.

-M --setseq set TCP sequence number

-L --setack set TCP ack

-F --fin set FIN flag

-S --syn set SYN flag

-R --rst set RST flag

-P --push set PUSH flag

-A --ack set ACK flag

-U --urg set URG flag

-X --xmas set X unused flag (0x40)

-Y --ymas set Y unused flag (0x80)

--tcpxitcode use last tcp->th\_flags as exit code

--tcp-mss enable the TCP MSS option with the given value

--tcp-timestamp enable the TCP timestamp option to guess the HZ/uptime

#### Common

-d --data data size (default is 0)

-E --file data from file

-e --sign add 'signature'

-j --dump dump packets in hex

-J --print dump printable characters

-B --safe enable 'safe' protocol

-u --end tell you when --file reached EOF and prevent rewind

-T --traceroute traceroute mode (implies --bind and --ttl 1)

--tr-stop Exit when receive the first not ICMP in traceroute mode

--tr-keep-ttl Keep the source TTL fixed, useful to monitor just one hop

--tr-no-rtt Don't calculate/show RTT information in traceroute mode

#### ARS packet description (new, unstable)

--apd-send Send the packet described with APD (see docs/APD.txt)

---

## 二，命令实例

hping3 -v

显示当前版本

```
[root@parrot: ~]# hping3 -v
hping3 version 3.0.0-alpha-2 ($Id: release.h,v 1.4 2004/04/09 23:38:56 antirez Exp $)
This binary is TCL scripting capable
```

hping3 -c 10 192.168.11.136

仅对目标发送10次请求包

```
[root@parrot]~[/home/kun]
#hping3 -c 10 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
len=40 ip=192.168.11.136 ttl=64 DF id=10450 sport=0 flags=RA seq=0 win=0 rtt=7.8 ms
len=40 ip=192.168.11.136 ttl=64 DF id=10451 sport=0 flags=RA seq=1 win=0 rtt=3.6 ms
len=40 ip=192.168.11.136 ttl=64 DF id=10452 sport=0 flags=RA seq=2 win=0 rtt=7.5 ms
len=40 ip=192.168.11.136 ttl=64 DF id=10453 sport=0 flags=RA seq=3 win=0 rtt=7.1 ms
len=40 ip=192.168.11.136 ttl=64 DF id=10454 sport=0 flags=RA seq=4 win=0 rtt=7.3 ms
len=40 ip=192.168.11.136 ttl=64 DF id=10455 sport=0 flags=RA seq=5 win=0 rtt=6.8 ms
len=40 ip=192.168.11.136 ttl=64 DF id=10456 sport=0 flags=RA seq=6 win=0 rtt=6.5 ms
len=40 ip=192.168.11.136 ttl=64 DF id=10457 sport=0 flags=RA seq=7 win=0 rtt=6.3 ms
len=40 ip=192.168.11.136 ttl=64 DF id=10458 sport=0 flags=RA seq=8 win=0 rtt=2.3 ms
len=40 ip=192.168.11.136 ttl=64 DF id=10459 sport=0 flags=RA seq=9 win=0 rtt=6.4 ms

--- 192.168.11.136 hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 2.3/6.2/7.8 ms
```

## 洪水攻击

在泛定的广义上来讲，子要网络数据包发送量过大时，就会造成洪水攻击，常见的洪水攻击如DDoS，ARP等。

### 相关原理

攻击者往往会使用多台PC主机或服务器对目标进行攻击，而这个攻击则就是向目标发发送微秒级的数据请求，这可能会造成目标主机的无法访问或服务器资源耗尽等，而这个攻击是无法防御的，对方服务器可做操作（常见）有：

#### 1.对目标IP进行封禁

一般有些网站或服务器会设置一个在一段时间内的访问次数，比如十分钟内凡是来自你的数据包汗你特征的，比如IP，MAC这些，会把你短暂的封禁，导致你的无法访问。

#### 2.服务器进行分流

服务器分流是一种DDoS攻击常见的方法，使用多个服务器以提高整体的性能，当对方服务器只有10台的时候，那么你发送的数据包将会分散在这十台服务器之间，比如你发送的是5个请求包，Server\_id1 接受的请求包为3，Server\_id2接受的请求包为2，那么你的数据包就会完全分散，除非攻击者增量攻击，否则攻击者将会失败

#### 3.提高服务器性能

提高服务器整体性能是一个非常耗材的办法，比如此时USER1购买的是某云学生机，那么攻击者只需要等待积分在即可达到目的。

但是如果目标购买的是某云高防服务器则攻击者很难达到攻击，只是因为服务器的性能不同和宽带数量而决定的。

什么是宽带？

宽带，英文名词为“Broadband”，在电子通信等领域上，主要用来描述路线能够同时处理宽带的范围和频率。

我们拿一个最简单的例子来描述，加入你是一个车主，有一个五星级加油站和三星级的加油站，而五星级的加油站有5个加油口，但是三星级的加油站有2个口。你作为车主想马上踏上行程的话，去三星级的加油站需要排队，但是价格更低，实惠。

那么你也可以选择五星级的加油站，五星级的加油站可以为你提供优质的服务，并且不用等待，开过去就可以加油。

那么你作为车主，你会选择那个？

hping3 -i u100 192.168.11.136

设置请求间隔（单位：100）

旗下还附带了三条命令

可搭配 --rand-source（随机使用一个IP地址）

--fast -u10000 //在10000微秒内每秒发送10个

--faster -i u1000 //在1000微秒内快速发送，每秒发送100个包

hping3 -i u --flood 192.168.11.135 //以最可能的最快的速度发送数据包，并且不打印输出

138 0.000196416	192.168.11.1	192.168.11.136	TCP	54 2880 - 0 [<None>] Seq=1 Win=512 Len=0
139 0.000257733	192.168.11.136	192.168.11.1	TCP	54 0 - 2879 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
140 0.000311335	192.168.11.136	192.168.11.1	TCP	54 0 - 2880 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
141 0.000314818	192.168.11.1	192.168.11.136	TCP	54 2881 - 0 [<None>] Seq=1 Win=512 Len=0
142 0.000331210	192.168.11.136	192.168.11.1	TCP	54 0 - 2881 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
143 0.000431160	192.168.11.1	192.168.11.136	TCP	54 2882 - 0 [<None>] Seq=1 Win=512 Len=0
144 0.000551061	192.168.11.1	192.168.11.136	TCP	54 2883 - 0 [<None>] Seq=1 Win=512 Len=0
145 0.000607597	192.168.11.136	192.168.11.1	TCP	54 0 - 2882 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
146 0.000648514	192.168.11.136	192.168.11.1	TCP	54 0 - 2883 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
147 0.000724416	192.168.11.1	192.168.11.136	TCP	54 2884 - 0 [<None>] Seq=1 Win=512 Len=0
148 0.000804066	192.168.11.136	192.168.11.1	TCP	54 0 - 2884 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
149 0.000855976	192.168.11.1	192.168.11.136	TCP	54 2885 - 0 [<None>] Seq=1 Win=512 Len=0
150 0.000979385	192.168.11.1	192.168.11.136	TCP	54 2886 - 0 [<None>] Seq=1 Win=512 Len=0
151 0.000988064	192.168.11.136	192.168.11.1	TCP	54 0 - 2885 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
152 0.000978950	192.168.11.136	192.168.11.1	TCP	54 0 - 2886 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
153 0.000911028	192.168.11.1	192.168.11.136	TCP	54 2887 - 0 [<None>] Seq=1 Win=512 Len=0
154 0.000923706	192.168.11.1	192.168.11.136	TCP	54 2888 - 0 [<None>] Seq=1 Win=512 Len=0
155 0.000929268	192.168.11.136	192.168.11.1	TCP	54 0 - 2887 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
156 0.000930213	192.168.11.136	192.168.11.1	TCP	54 0 - 2888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
157 0.000940777	192.168.11.1	192.168.11.136	TCP	54 2889 - 0 [<None>] Seq=1 Win=512 Len=0
158 0.000460969	192.168.11.1	192.168.11.136	TCP	54 2890 - 0 [<None>] Seq=1 Win=512 Len=0
159 0.000492020	192.168.11.136	192.168.11.1	TCP	54 0 - 2891 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
160 0.000576074	192.168.11.136	192.168.11.1	TCP	54 0 - 2890 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
161 0.000587365	192.168.11.1	192.168.11.136	TCP	54 2891 - 0 [<None>] Seq=1 Win=512 Len=0
162 0.000709324	192.168.11.1	192.168.11.136	TCP	54 2892 - 0 [<None>] Seq=1 Win=512 Len=0
163 0.000830862	192.168.11.1	192.168.11.136	TCP	54 2893 - 0 [<None>] Seq=1 Win=512 Len=0
164 0.000907254	192.168.11.136	192.168.11.1	TCP	54 0 - 2891 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
165 0.000918813	192.168.11.136	192.168.11.1	TCP	54 0 - 2892 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
166 0.000950356	192.168.11.1	192.168.11.136	TCP	54 2894 - 0 [<None>] Seq=1 Win=512 Len=0
167 0.000950603	192.168.11.136	192.168.11.1	TCP	54 0 - 2893 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
168 0.010626839	192.168.11.136	192.168.11.1	TCP	54 0 - 2894 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
169 0.010070331	192.168.11.1	192.168.11.136	TCP	54 2895 - 0 [<None>] Seq=1 Win=512 Len=0
170 0.010190300	192.168.11.1	192.168.11.136	TCP	54 2896 - 0 [<None>] Seq=1 Win=512 Len=0
171 0.010240841	192.168.11.136	192.168.11.1	TCP	54 0 - 2895 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
172 0.010284112	192.168.11.136	192.168.11.1	TCP	54 0 - 2896 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
173 0.010300510	192.168.11.1	192.168.11.136	TCP	54 2897 - 0 [<None>] Seq=1 Win=512 Len=0
174 0.010420014	192.168.11.1	192.168.11.136	TCP	54 2898 - 0 [<None>] Seq=1 Win=512 Len=0
175 0.010510288	192.168.11.136	192.168.11.1	TCP	54 0 - 2897 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

hping3 -n 192.168.11.136

以目标需要的形式输出

```
#hping3 -n 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
len=40 ip=192.168.11.136 ttl=64 DF id=33737 sport=0 flags=RA seq=0 win=0 rtt=3.8 ms

> Frame 5: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface vmnet8, id 0
> Ethernet II, Src: VMware_ec:86:13 (00:0c:29:ec:86:13), Dst: VMware_c0:00:08 (00:50:56:c0:00:08)
> Internet Protocol Version 4, Src: 192.168.11.136, Dst: 192.168.11.1
> Transmission Control Protocol, Src Port: 0, Dst Port: 1094, Seq: 1, Ack: 1, Len: 0
```

hping3 -q 192.168.11.136

不打印结果但正常发送数据包

```
#hping3 -q 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
3 1.000129491 192.168.11.1 192.168.11.136 TCP 54 1096 - 0
4 1.000490568 192.168.11.136 192.168.11.1 TCP 54 0 - 1066
5 2.000209950 192.168.11.1 192.168.11.136 TCP 54 1667 - 0
```

hping3 -I vmnet8 192.168.11.136

使用vmnet8网卡接口对192.168.11.136进行扫描

```
#hping3 -I vmnet8 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
```

hping3 -V 192.168.11.136

详细的打印192.168.11.136的扫描结果

```
[root@parrot ~]# hping3 -V 192.168.11.136
using vmnet8, addr: 192.168.11.1, MTU: 1500
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
len=40 ip=192.168.11.136 ttl=64 DF id=33940 tos=0 iplen=40
sport=0 flags=RA seq=0 win=0 rtt=7.9 ms
seq=0 ack=104384307 sum=db41 urp=0
```

hping3 -D 192.168.11.136

将最终结果设置为调试模式

```
DEBUG: Output interface address: 0.0.0.0
DEBUG: if lo: The address doesn't match
DEBUG: if wlan0: The address doesn't match
DEBUG: if virbr0: The address doesn't match
DEBUG: if docker0: The address doesn't match
DEBUG: if vmnet1: The address doesn't match
DEBUG: if vmnet8: OK
using vmnet8, addr: 192.168.11.1, MTU: 1500
DEBUG: pcap_open_live(vmnet8, 99999, 0, 1, 0x559203dec3a0)
DEBUG: dlttype is 1
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
45 00 00 28 CD 82 00 00 04 06 00 00 C0 A8 0B 01 C0 A8 0B 88 05 BE 00 00 5D 8E 81 ED 5C 74 6A D9 50 00 02 00 69 83
00 00
DEBUG: send_packet成功(1)
```

hping3 -z 192.168.11.136

使用“ctrl+z”键来绑定该数据包的生存时间“ttl”

```
[root@parrot ~]# hping3 -z 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data
len=40 ip=192.168.11.136 ttl=64 DF id=34478 sport=0 flags=RA seq=0 win=0 rtt=3.9 ms
65: len=40 ip=192.168.11.136 ttl=64 DF id=34479 sport=0 flags=RA seq=1 win=0 rtt=7.7
66: len=40 ip=192.168.11.136 ttl=64 DF id=34480 sport=0 flags=RA seq=2 win=0 rtt=3.9
len=40 ip=192.168.11.136 ttl=64 DF id=34481 sport=0 flags=RA seq=3 win=0 rtt=3.8 ms
67: len=40 ip=192.168.11.136 ttl=64 DF id=34482 sport=0 flags=RA seq=4 win=0 rtt=7.4
69: len=40 ip=192.168.11.136 ttl=64 DF id=34483 sport=0 flags=RA seq=5 win=0 rtt=7.3
70: len=40 ip=192.168.11.136 ttl=64 DF id=34484 sport=0 flags=RA seq=6 win=0 rtt=3.2
71: len=40 ip=192.168.11.136 ttl=64 DF id=34485 sport=0 flags=RA seq=7 win=0 rtt=7.5
73: len=40 ip=192.168.11.136 ttl=64 DF id=34486 sport=0 flags=RA seq=8 win=0 rtt=2.9
74: len=40 ip=192.168.11.136 ttl=64 DF id=34487 sport=0 flags=RA seq=9 win=0 rtt=6.8
75: len=40 ip=192.168.11.136 ttl=64 DF id=34488 sport=0 flags=RA seq=10 win=0 rtt=6.
rc
--- 192.168.11.136 hping statistic ---
11 packets transmitted, 11 packets received, 0% packet loss
round-trip min/avg/max = 2.9/5.5/7.7 ms
[root@parrot ~]#
```

No.	Time	Source	Destination	Protocol	Length	Info
15	17.273755270	192.168.11.1	192.168.11.136	TCP	54	1630 → 0 [←None] Seq=1 Win=512 Len=0
16	17.274077244	192.168.11.136	192.168.11.1	TCP	54	0 → 1630 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
17	17.284503213	Vmware, ec:86:13	Vmware, c0:00:08	ARP	42	Who has 192.168.11.1? Tell 192.168.11.136
18	17.284526456	Vmware, c0:00:08	Vmware, ec:86:13	ARP	42	192.168.11.1 is at 00:50:56:c0:00:08
19	17.412953566	Vmware, c0:00:08	Vmware, ec:86:13	ARP	42	Who has 192.168.11.120? Tell 192.168.11.1
20	17.413648871	Vmware, ec:86:13	Vmware, c0:00:08	ARP	42	192.168.11.136 is at 00:5c:29:ec:86:13
21	18.273855276	192.168.11.1	192.168.11.136	TCP	54	1640 → 0 [←None] Seq=1 Win=512 Len=0
22	18.274154547	192.168.11.136	192.168.11.1	TCP	54	0 → 1640 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
23	19.273979056	192.168.11.1	192.168.11.136	TCP	54	1641 → 0 [←None] Seq=1 Win=512 Len=0
24	19.243232250	192.168.11.136	192.168.11.1	TCP	54	0 → 1641 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
25	26.274873922	192.168.11.1	192.168.11.136	TCP	54	1642 → 0 [←None] Seq=1 Win=512 Len=0
26	26.274373824	192.168.11.136	192.168.11.1	TCP	54	0 → 1642 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
27	21.274164500	192.168.11.1	192.168.11.136	TCP	54	1643 → 0 [←None] Seq=1 Win=512 Len=0
28	21.274507748	192.168.11.136	192.168.11.1	TCP	54	0 → 1643 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
29	22.274264776	192.168.11.1	192.168.11.136	TCP	54	1644 → 0 [←None] Seq=1 Win=512 Len=0
30	22.274593653	192.168.11.136	192.168.11.1	TCP	54	0 → 1644 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

```
.....00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 40
Identification: 0x2c7a (11386)
Flags: 0x0000
  0... .. = Reserved bit: Not set
  0... .. = Don't fragment: Not set
  0... .. = More fragments: Not set
Fragment offset: 0
[Window size: 0]
Protocol: TCP (6)
Header checksum: 0xad7c [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.11.1
Destination: 192.168.11.136
P V E
```

hping3 -Z 192.168.11.136

取消绑定 ctrl + z

```
#hping3 -Z 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
len=40 ip=192.168.11.136 ttl=64 DF id=34493 sport=0 flags=RA seq=0 win=0 rtt=3.5 ms
len=40 ip=192.168.11.136 ttl=64 DF id=34494 sport=0 flags=RA seq=1 win=0 rtt=3.4 ms
len=40 ip=192.168.11.136 ttl=64 DF id=34495 sport=0 flags=RA seq=2 win=0 rtt=3.3 ms
^Z
[5]+  Stopped                  hping3 -Z 192.168.11.136
```

heping3 --beep 192.168.11.136

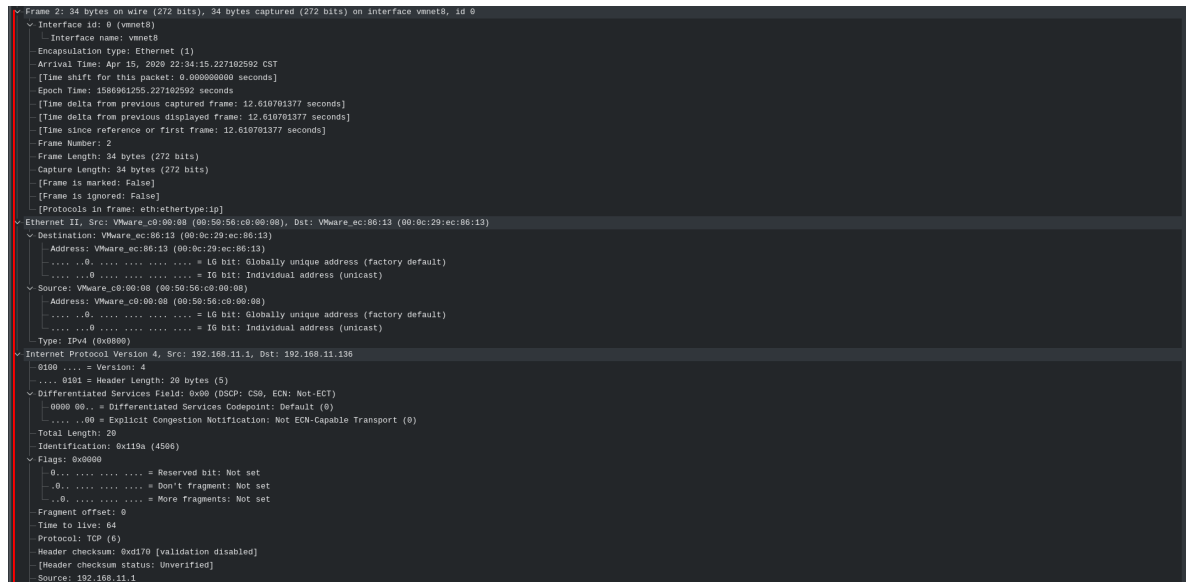
当切换或其他窗口时则提醒结果



Mode (模式)

hping3 -0 192.168.11.136

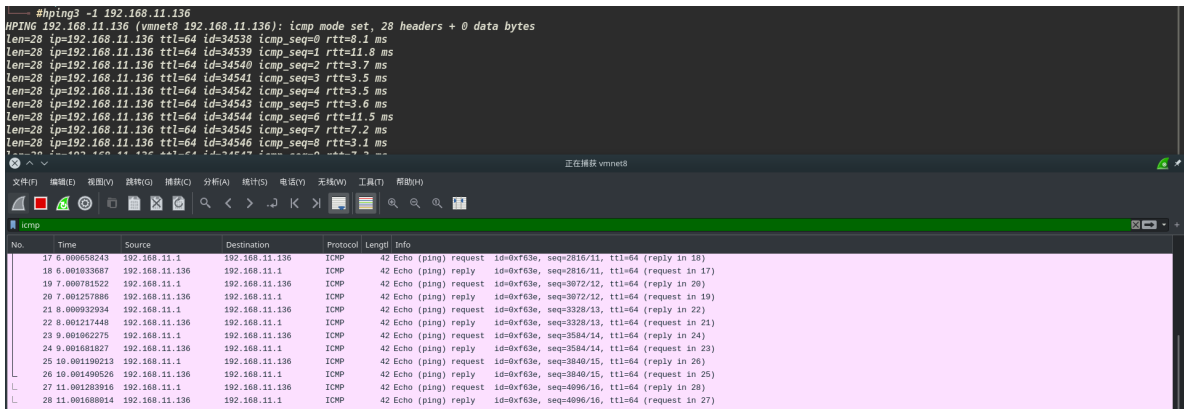
使用源IP模式对目标进行扫描



hping3 -1 192.168.11.136

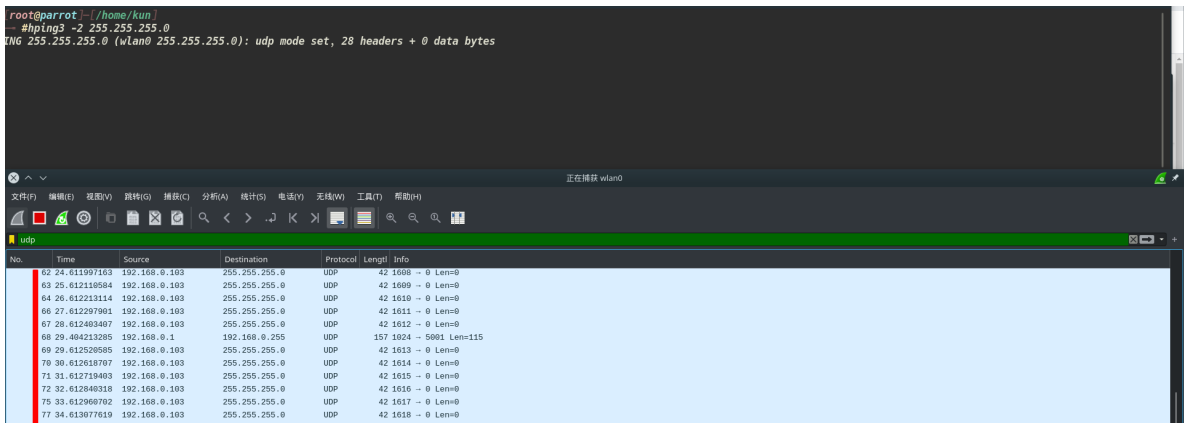
使用ICMP协议对192.168.11.136进行检索





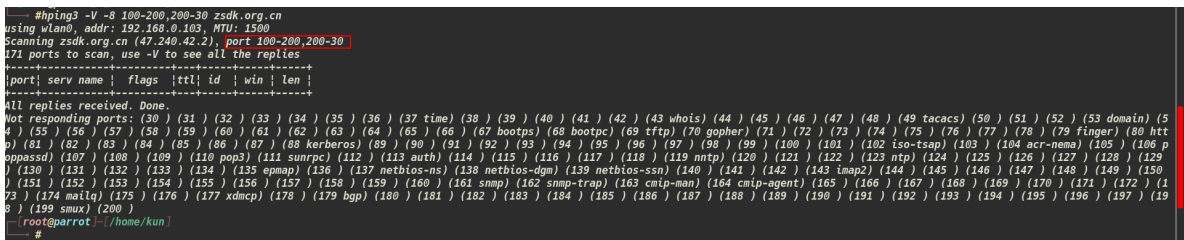
hping3 -2 192.168.11.136

使用UDP模式对目标进行检索



hping3 -8 100-200,200-30 zstdk.org.cn

扫描zstdk.org.cn端口范围为100~200,200~300[^演示部分撒谎哦写了一个0]



hping3 -9 192.168.0.102

监听192.168.0.102发送是SYN数据包

```
root@ubuntu:/home/kun# hping3 -9 192.168.0.100
Warning: Unable to guess the output interface
hping3 listen mode
[main] memlockall(): Success
Warning: can't disable memory paging!
```

## IP

hping3 -a 192.168.11.132 255.255.255.0

对源IP地址进行欺骗



伪造虚假的IP地址对目标发送数据包，使得攻击者可隐藏自身IP地址，冒充任何计算机（IP）

其原理是路由只转发IP地址，并不对源地址进行验证

```
#hping3 -a 192.168.11.132 192.168.11.255
NG 192.168.11.255 (vmnet8 192.168.11.255): NO FLAGS are set, 40 headers + 0 data bytes
```

38	31.006003706	192.168.11.132	192.168.11.255	TCP	54 1280 - 0 [<None>] Seq=1 Win=512 Len=0
37	32.006381604	192.168.11.132	192.168.11.255	TCP	54 1281 - 0 [<None>] Seq=1 Win=512 Len=0

hping3 -I wlan0 --rand-dest 192.168.11.136

随机目标地址模式

```
#hping3 -I wlan0 --rand-dest 192.168.11.132
HPING 192.168.11.132 (wlan0 192.168.11.132): NO FLAGS are set, 40 headers + 0 data bytes
```

43	44.737028931	192.168.0.100	224.0.0.22	ICMPV	
44	45.628577995	192.168.0.100	172.104.164.108	TLSV	
45	45.951364229	172.104.164.108	192.168.0.100	TLSV	
46	45.951426977	192.168.0.100	172.104.164.108	TCP	
47	46.000000000	192.168.0.100	239.255.255.250	SSDP	
48	51.618453769	192.168.0.100	239.255.255.250	SSDP	
49	52.115009038	192.168.0.100	5.61.49.13	TCP	
50	52.619899112	192.168.0.100	239.255.255.250	SSDP	
51	53.621179717	192.168.0.100	239.255.255.250	SSDP	
52	54.62226423	192.168.0.100	239.255.255.250	SSDP	
53	55.757026814	192.168.0.100	5.61.49.13	TCP	
54	60.069052969	192.168.0.1	192.168.0.255	UDP	

hping3 -rand-source 192.168.11.136

使用一个随机的IP对目标发送数据包

32 12.002471290	192.168.11.136	190.169.134.185	TCP	54 0 → 1998 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0	HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data
33 13.002112390	192.168.11.136	192.168.11.136	TCP	54 1997 → 0 [None] Seq=1 Win=512 Len=0	len=40 ip=192.168.11.136 ttl=64 DF id=52861 sport=0 flags=RA seq=0 win=0 rtt=7.7 ms
34 13.002202010	192.168.11.136	192.168.11.136	TCP	54 1998 → 0 [None] Seq=1 Win=512 Len=0	len=40 ip=192.168.11.136 ttl=64 DF id=52877 sport=0 flags=RA seq=1 win=0 rtt=3.6 ms
35 14.002245132	192.168.11.136	192.168.11.136	TCP	54 1999 → 0 [None] Seq=1 Win=512 Len=0	len=40 ip=192.168.11.136 ttl=64 DF id=51147 sport=0 flags=RA seq=2 win=0 rtt=3.5 ms
36 15.002389996	192.168.11.136	192.168.11.136	TCP	54 1999 → 0 [None] Seq=1 Win=512 Len=0	len=40 ip=192.168.11.136 ttl=64 DF id=51147 sport=0 flags=RA seq=3 win=0 rtt=3.4 ms

hping3 -t 10 192.168.11.136

设置发送输出包的生存时间为“10”[默认认为64]

82 35.134055136	192.168.11.136	192.168.11.1	TCP	54 0 → 2546 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0	HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
83 36.13404484	192.168.11.1	192.168.11.136	TCP	54 2546 → 0 [None] Seq=1 Win=512	len=40 ip=192.168.11.136 ttl=64 DF id=14639 sport=0 flags=RA seq=0 win=0 rtt=7.9 ms
84 36.135075377	192.168.11.136	192.168.11.1	TCP	54 0 → 2546 [RST, ACK] Seq=1 Ack=1	len=40 ip=192.168.11.136 ttl=64 DF id=14639 sport=0 flags=RA seq=1 win=0 rtt=8.0 ms
...0... More fragments: Not set					
Fragment offset: 0					
Time to live: 10					
Protocol: TCP (6)					

hping3 -f 192.168.11.136

将数据包分割为更多的数据包[可能会通过ACL ( Access Control List ,访问控制协议 ) ]

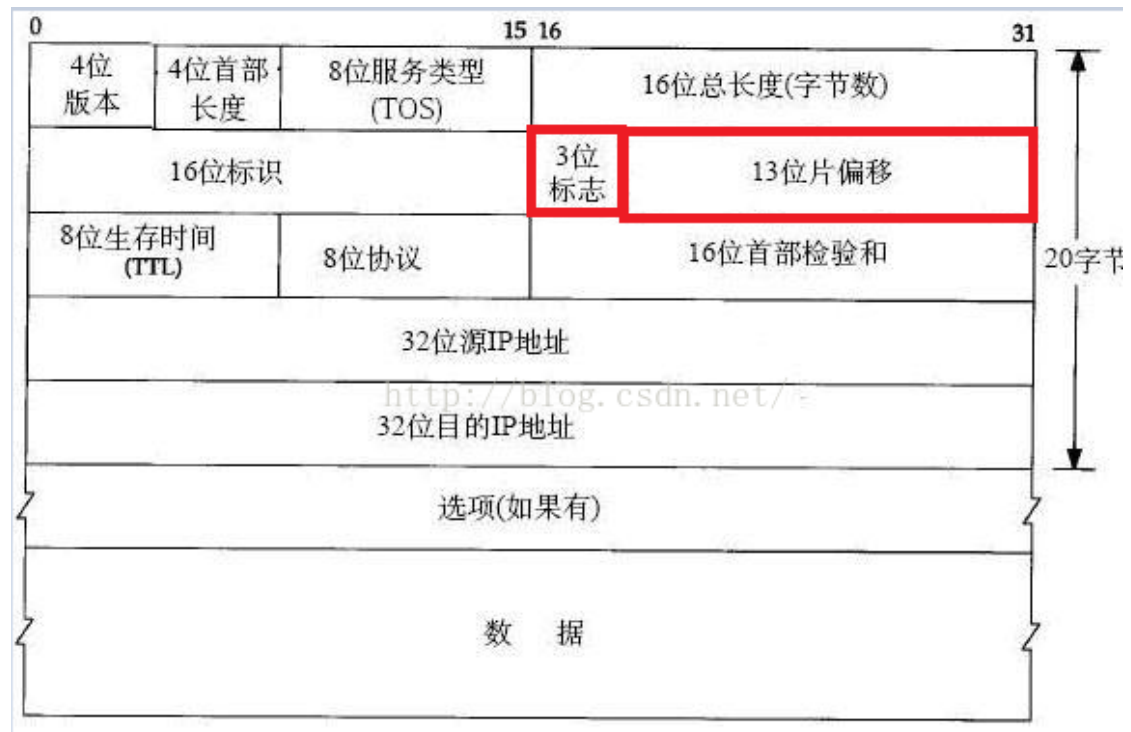
87 52.150226487	192.168.0.100	192.169.11.136	TCP	54 1589 → 0 [None] Seq=1 Win=512 Len=0	
88 53.087577359	192.168.0.103	255.255.255.255	UDP	544 4466 → 4466 Len=512	
89 53.150402154	192.168.0.100	192.169.11.136	TCP	54 1590 → 0 [None] Seq=1 Win=512 Len=0	
90 54.107421954	192.168.0.103	255.255.255.255	UDP	539 4466 → 4466 Len=488	

hping3 -x 192.168.11.136

设置更多的碎片标志

hping3 -y 192.168.11.136 //碎片标志

178 175.070027486	Huawei10_75:95:a7	Broadcast	ARP	42 Who has 192.168.0.1? Tell 192.168.0.103	
179 176.644578189	192.168.0.100	239.255.255.250	SSDP	209 M-SEARCH * HTTP/1.1	
180 176.707210764	192.168.0.103	255.255.255.255	UDP	558 4466 → 4466 Len=516	
181 177.646070271	192.168.0.100	239.255.255.250	SSDP	209 M-SEARCH * HTTP/1.1	
182 178.647656410	192.168.0.100	239.255.255.250	SSDP	209 M-SEARCH * HTTP/1.1	
183 179.648390304	192.168.0.100	239.255.255.250	SSDP	209 M-SEARCH * HTTP/1.1	



hping3 -g 10 192.168.11.136

设置一个偏移量为10[可能是我的思路或者技术问题，没有设置成功]

4 28.493180759	192.168.11.2	192.168.11.136	DNS	143 Standard query response 0x6f3 No such name A videosearch.ubuntu.com 50A nsl.canonical.com	
...0... More fragments: Not set					
Fragment offset: 0					

hping3 -o telnet 192.168.11.136

使用telnet服务类型发送数据包 [^hping3官方为此提供了一个帮助手册，可使用命令 hping3 try -tos help 进行查看]

7 3.680281331	192.168.11.1	192.168.11.136	TCP	54 1688 - 0 [<none>] Seq=1 Win=512 Len=0
8 2.880770160	192.168.11.1	192.168.11.136	TCP	54 0 - 1688 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
9 4.680378612	192.168.11.1	192.168.11.136	TCP	54 1688 - 0 [<none>] Seq=1 Win=512 Len=0
10 4.680746462	192.168.11.136	192.168.11.1	TCP	54 0 - 1688 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
11 5.680456466	192.168.11.1	192.168.11.136	TCP	54 1688 - 0 [<none>] Seq=1 Win=512 Len=0
12 8.680868940	192.168.11.136	192.168.11.1	TCP	54 0 - 1688 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
13 5.680210381	VMware ac:09:13	VMware c9:09:08	ARP	42 who has 192.168.11.1? tell 192.168.11.136

Protocol: TCP (6)

Header checksum: 0xfbd8 [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.11.1

Destination: 192.168.11.136

Transmission Control Protocol, Src Port: 1688, Dst Port: 0, Seq: 1, Len: 0

Source Port: 1688

hping3 -G 192.168.11.136

包括RECORD\_ROUTE（记录路由）选项并显示路由缓冲区

Record route

记录路由（Record route）：数据包离开时为每台机器提供空间记录数据包的出站接口地址，便于保存数据包经过的所有路由器的记录，类似与路由追踪，但是和路由追踪不同

路由记录（Record route）：当IP离开路由器的时候记录路由器的出站IP接口地址

路由记载（Record route）：fast路由ip地址，当IP包脱节到每个路由器的时间记载路由器的出站接口地址。

记录路由选项（Record route）当报文离开时为每个路由器提供空间记录报文的出站接口地址，以便保存保温经过的所有路由器记录

路由缓冲区

在数据包中添加了相应的驱动和缓冲模块，避免了数据的冲突，完美的保障了数据的完整性。

```
#hping3 -G 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
len=80 ip=192.168.11.136 ttl=64 DF id=16584 sport=0 flags=RA seq=0 win=0 rtt=11.9 ms
RR: 1.2.3.4
192.168.11.136
len=80 ip=192.168.11.136 ttl=64 DF id=16585 sport=0 flags=RA seq=1 win=0 rtt=7.8 ms
(same route)
len=80 ip=192.168.11.136 ttl=64 DF id=16586 sport=0 flags=RA seq=2 win=0 rtt=3.9 ms
(same route)
len=80 ip=192.168.11.136 ttl=64 DF id=16587 sport=0 flags=RA seq=3 win=0 rtt=7.8 ms
```

hping3 --less 192.168.11.255 192.168.11.136

使用松散源路由和记录路由对目标发送数据包

hping3 --ssrr 192.168.11.255 192.168.11.136

松散源路由（loose source route）

松散源路由，通俗的来说是你给出一个目标（IP地址），不管用什么方法，只需要经过这个路由就可以了。

严格源路由（strict route）

严格源路由，通俗上说，他严格的规定了路由要经过的路径上的每一个路由器，经过路由的顺序也不可更改，与“松散源路由（loose source route）”的区别是松散源路由只需要经过指定的路由即可。

而严格路由，需要你把该过的流程都过了，都过一遍，才可以”

```
hping3 -H 192.168.11.255 192.168.11.136
```

仅在原始IP模式下设置IP协议字段

1	0.000000000	192.168.11.1	192.168.11.136	TCP	54 2419 → 0 [←None] Seq=1 Win=512 Len=0
2	0.000000001	192.168.11.136	192.168.11.1	TCP	54 0 → 2419 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

ICMP

```
hping3 -C vmnet8 192.168.11.136
```

使用ICMP协议模式向目标发送数据包[^vmnet为该网段所使用的网卡，可使用“-V”参数进行查看]

10	3.0000319979	192.168.11.1	192.168.11.136	ICMP	42 Echo (ping) reply	id=0x9310, s
11	4.0000417214	192.168.11.1	192.168.11.136	ICMP	42 Echo (ping) reply	id=0x9310, s
12	5.0000509691	192.168.11.1	192.168.11.136	ICMP	42 Echo (ping) reply	id=0x9310, s

```
hping3 -K 192.168.11.136
```

设置ICMP代码为“10”

```
#hping3 -K 10 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): icmp mode set, 28 headers + 0 data bytes
len=28 ip=192.168.11.136 ttl=64 id=64773 icmp_seq=0 rtt=7.9 ms
len=28 ip=192.168.11.136 ttl=64 id=64774 icmp_seq=1 rtt=11.7 ms
len=28 ip=192.168.11.136 ttl=64 id=64775 icmp_seq=2 rtt=3.6 ms
len=28 ip=192.168.11.136 ttl=64 id=64776 icmp_seq=3 rtt=7.5 ms
^C
--- 192.168.11.136 hping statistic ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 3.6/7.7/11.7 ms
root@parrot: ~/home/kun

-- bash -- kun: bash --

Source: 192.168.11.1
Destination: 192.168.11.136
Internet Control Message Protocol
Type: 0 (Echo (ping) request)
Code: 10
Checksum: Wk7a3 [correct]
[Checksum Status: Good]
Identifier (BE): 18 (0x0012)
Identifier (LE): 4080 (0x1000)
Sequence number (BE): 0 (0x0000)
Sequence number (LE): 0 (0x0000)
0000 00 0c 29 ec a6 13 00 50 56 c9 00 00 00 00 45 00  )..SP.Vxxx..E
0040 00 1c 48 0c 00 40 01 7a 0b c9 a9 00 01 c9 a9  )..H..0..z.....
0080 00 88 00 0a f7 03 00 12 00 00                                     )..H..0..z.....
```

```
hping3 -K --force-icmp 192.168.11.136
```

发送所有ICMP类型（Hping目前所支持的）

```
#hping3 -K --force-icmp 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): icmp mode set, 28 headers + 0 data byte
len=28 ip=192.168.11.136 ttl=64 id=65116 icmp_seq=0 rtt=3.9 ms
len=28 ip=192.168.11.136 ttl=64 id=65117 icmp_seq=1 rtt=11.0 ms
len=28 ip=192.168.11.136 ttl=64 id=65118 icmp_seq=2 rtt=3.6 ms
len=28 ip=192.168.11.136 ttl=64 id=65119 icmp_seq=3 rtt=3.5 ms
len=28 ip=192.168.11.136 ttl=64 id=65120 icmp_seq=4 rtt=7.4 ms
len=28 ip=192.168.11.136 ttl=64 id=65121 icmp_seq=5 rtt=3.3 ms
len=28 ip=192.168.11.136 ttl=64 id=65122 icmp_seq=6 rtt=3.2 ms
len=28 ip=192.168.11.136 ttl=64 id=65123 icmp_seq=7 rtt=3.2 ms
len=28 ip=192.168.11.136 ttl=64 id=65124 icmp_seq=8 rtt=7.5 ms
len=28 ip=192.168.11.136 ttl=64 id=65125 icmp_seq=9 rtt=3.0 ms
len=28 ip=192.168.11.136 ttl=64 id=65126 icmp_seq=10 rtt=2.9 ms
len=28 ip=192.168.11.136 ttl=64 id=65127 icmp_seq=11 rtt=2.8 ms
len=28 ip=192.168.11.136 ttl=64 id=65128 icmp_seq=12 rtt=6.7 ms
```

No.	Time	Source	Destination	Protocol	Length	Info
02	25.982671632	192.168.11.136	192.168.11.1	ICMP	42	Echo (ping) reply id=0xa913, seq=6405
03	26.002470358	192.168.11.1	192.168.11.136	ICMP	42	Echo (ping) request id=0xa913, seq=6654
04	26.002813051	192.168.11.136	192.168.11.1	ICMP	42	Echo (ping) reply id=0xa913, seq=6654
05	27.002549990	192.168.11.1	192.168.11.136	ICMP	42	Echo (ping) request id=0xa913, seq=6915
06	27.002900466	192.168.11.136	192.168.11.1	ICMP	42	Echo (ping) reply id=0xa913, seq=6915
07	28.002635064	192.168.11.1	192.168.11.136	ICMP	42	Echo (ping) request id=0xa913, seq=7166
08	28.003000976	192.168.11.136	192.168.11.1	ICMP	42	Echo (ping) reply id=0xa913, seq=7166
09	29.002717230	192.168.11.1	192.168.11.136	ICMP	42	Echo (ping) request id=0xa913, seq=7424
10	29.003003996	192.168.11.136	192.168.11.1	ICMP	42	Echo (ping) reply id=0xa913, seq=7424

重定向网关地址到192.168.11.202[^hping3默认重定向网关到0.0.0.0]

```
#hping3 --icmp-gw 192.168.11.136 192.168.11.123
HPING 192.168.11.123 (vmnet8 192.168.11.123): NO FLAGS are set, 40 headers + 0 data b
```

```
hping3 --icmp-ts 192.168.11.136
```

## 设置ICMP类型为“13”的时间戳请求

```
hping3 --icmp-addr 192.168.11.136
```

## 设置ICMP类型为“17”的时间戳请求

```

1 0.000000000 192.168.11.1 192.168.11.136 ICMP 46 Address mask request id=0x6319, seq
Time to live: 64
Protocol: ICMP (1)
Header checksum: 0x0b77 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.11.1
Destination: 192.168.11.136
Internet Control Message Protocol
Type: 17 (Address mask request)
Code: 0
Checksum: 0x8be6 [correct]
[Checksum Status: Good]
Identifier (BE): 25369 (0x6319)
Identifier (LE): 6499 (0x1963)
Sequence number (BE): 0 (0x0000)
Sequence number (LE): 0 (0x0000)
Address Mask: 0.0.0.0

```

```
hping3 --icmp-ipver 2 192.168.11.136
```

修改发送数据包的IP协议版本为“2”[^默认“4”]

```
[root@parrot: ~/home/kun.]# #hping3 --icmp-ipver 2 192.168.11.136
HPING 192.168.11.136 (vnmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data b
ytes => 192.168.11.136 ttl=64 DF len=64816 sport=0 flags=RA seq=0 win=0 rtt=7.6 ms
len=40 ip=192.168.11.136 ttl=64 DF len=64817 sport=0 flags=RA seq=1 win=0 rtt=7.4 ms
IC
192.168.11.136 hping statistic ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 7.4/7.5/7.6 ms
```

```
hping3 --icmp-iplen 10 192.168.11.136
```

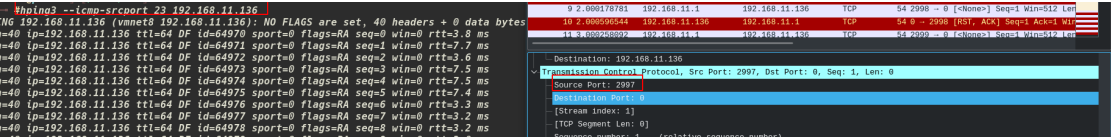
设置IP总长度为“10”

```
hping3 --icmp-ipid 10 192.168.11.136
```

设置ip总长度[默认随机]

hping3 --icmp-srcport 23

设置源端口为“23” [^默认随机]



hping --icmp-dstport 43 192.168.11.136

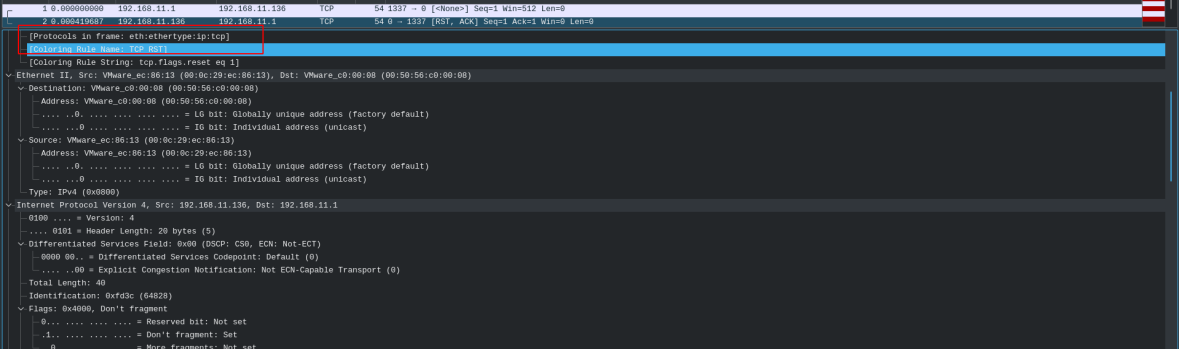
设置目标端口为“43”



hping3 --icmp-iproto 192.168.11.255 192.168.11.136

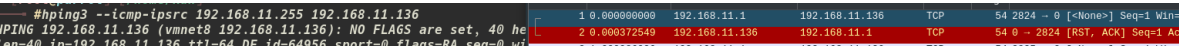
设置IP协议[^默认为"IPPROTO\_TCP"]

“IPPROTO\_TCP”与“IPPROTO\_IP”分别代表了TCP协议和IP协议。



hping3 --icmp-ipsrc 192.168.11.255 192.168.11.136

设置IP地址源[^默认0.0.0.0]



hping3 --icmp-ipdst 192.168.11.255 192.168.11.134

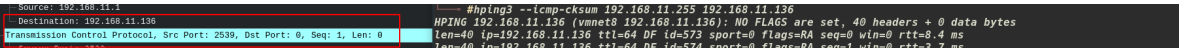
设置目标地址为192.168.11.134



hping3 --icmp-cksum 192.168.11.136

集合icmp校验和[^默认为正确的cksum]

cksum是一种排错检查方式，由CCITT所制定的演算标准。至少可以检测到99.998%的已知错误。





## UDP/TCP

hping3 -s 73 192.168.11.136

设置基本源端口地址为73,之后随着数据包的增加而增加端口数

源端口就是一个数据包的出口地址,比如你从A门出去,则你的源地址是A,如果你的女朋友从B们出去,那么你的女朋友的源地址就是B。

而设置源IP地址就是设置你出入的门,此处我们吧数据包的出口地址设置为了73,而“-s”参数有一个非常便捷的功能,加入我们设置的源端口为“73”,那么第一个数据包序号为0 的源端口一定是73,而第二个数据包序号为1的源端口一定为74,第三个数据包序号3的一定为75,以此类推。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.11.1	192.168.11.136	TCP	54	73 → 0 [RST, ACK] Seq=1 Win=512 Len=0
2	0.000373501	192.168.11.136	192.168.11.1	TCP	54	0 → 73 [RST, ACK] Seq=1 Ack=1 Win=0
3	1.000092406	192.168.11.1	192.168.11.136	TCP	54	74 → 0 [RST, ACK] Seq=1 Win=512 Len=0
4	1.000416855	192.168.11.136	192.168.11.1	TCP	54	0 → 74 [RST, ACK] Seq=1 Ack=1 Win=0
5	2.000180585	192.168.11.1	192.168.11.136	TCP	54	75 → 0 [RST, ACK] Seq=1 Win=512 Len=0

Time to live: 64

Protocol: TCP (6)

Header checksum: 0x4d4b [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.11.1

Destination: 192.168.11.136

Transmission Control Protocol, Src Port: 73, Dst Port: 0, Seq: 1, Len: 0

Source Port: 73

Destination Port: 0

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 1 (relative sequence number)

Sequence number (raw): 122892315

hping3 -p 73 192.168.11.136

目标端口为73,并向目标发送数据包,通过 ctrl+z 键进行增/减端口

Inc, 全称“progressive increase”中文译名为 递增和递减。

dec, 全称“decrease progressively”中文译名为 逐渐增加

No.	Time	Source	Destination	Protocol	Length	Info
46	19.002127836	192.168.11.136	192.168.11.1	TCP	54	73 → 2317 [RST, ACK] Seq=1 Ack=1 Win=0
47	20.002196056	192.168.11.1	192.168.11.136	TCP	54	2318 → 73 [RST, ACK] Seq=1 Ack=1 Win=0
48	20.002338063	192.168.11.136	192.168.11.1	TCP	54	74 → 2318 [RST, ACK] Seq=1 Ack=1 Win=0
49	21.002068280	192.168.11.1	192.168.11.136	TCP	54	2319 → 74 [RST, ACK] Seq=1 Ack=1 Win=0
50	21.002425610	192.168.11.136	192.168.11.1	TCP	54	74 → 2319 [RST, ACK] Seq=1 Ack=1 Win=0
51	22.002190071	192.168.11.1	192.168.11.136	TCP	54	2320 → 75 [RST, ACK] Seq=1 Ack=1 Win=0
52	23.002526313	192.168.11.136	192.168.11.1	TCP	54	75 → 2320 [RST, ACK] Seq=1 Ack=1 Win=0

hping3 -k 192.168.11.136

保持源端口不动[假如源地址为2748,则数据包所有hping3请求的源地址都为2748端口]

No.	Time	Source	Destination	Protocol	Length	Info
10	3.000626744	192.168.11.136	192.168.11.1	TCP	54	0 → 2748 [RST, ACK] Seq=1 Ack=14964
11	4.000331573	192.168.11.1	192.168.11.136	TCP	54	2748 → 0 [RST, ACK] Seq=375330363 Win=0
12	4.000745660	192.168.11.136	192.168.11.1	TCP	54	0 → 2748 [RST, ACK] Seq=1 Ack=37533
13	5.000521643	192.168.11.1	192.168.11.136	TCP	54	2748 → 0 [RST, ACK] Seq=1583493020 Win=0
14	5.000893710	192.168.11.136	192.168.11.1	TCP	54	0 → 2748 [RST, ACK] Seq=1 Ack=15834

Destination: 192.168.11.136

Transmission Control Protocol, Src Port: 2748, Dst Port: 0, Seq: 1583493020, Len: 0

Source Port: 2748

Destination Port: 0

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 1583493020 (relative sequence number)



设置窗口大小为520[^Windows是窗口的意思，而size就是大小的意思，组合为“窗口大小”]

```
hping3 -O 10 192.168.11.136
```

hping3 -Q 192.168.11.136

```
hping3 -q 192.168.11.136
```

### hping3 -M 5201314 192.168.11.136

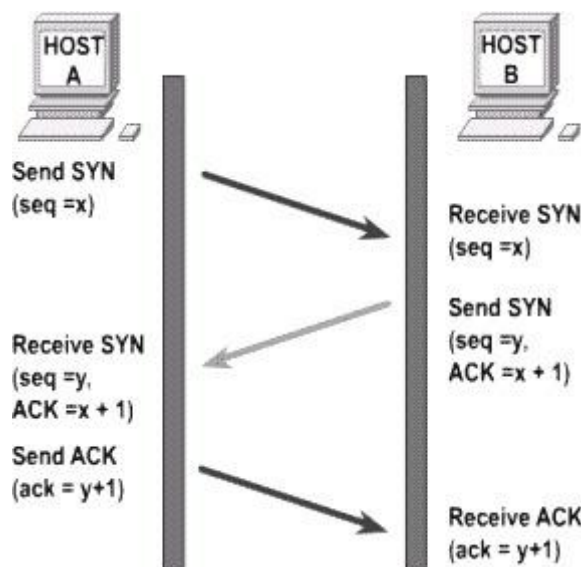
将TCP序列号自定义为"5201314"["^和女朋友的硬核表白"]

```
File Edit View Bookmarks Settings Help
root@parrot: /home/kun
#hping3 -M 5201314 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data
len=40 ip=192.168.11.136 ttl=64 DF id=55540 sport=0 flags=RA seq=0 win=0 rtt=7.8 ms
len=40 ip=192.168.11.136 ttl=64 DF id=55541 sport=0 flags=RA seq=1 win=0 rtt=3.7 ms
len=40 ip=192.168.11.136 ttl=64 DF id=55542 sport=0 flags=RA seq=2 win=0 rtt=7.9 ms
--- 192.168.11.136 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3.7/6.5/7.9 ms
root@parrot: /home/kun
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.11.1	192.168.11.136	TCP	54	2420 → 0 [<None>] Seq=1 Win=512 Len=0
2	0.000443772	192.168.11.136	192.168.11.1	TCP	54	0 → 2420 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
3	1.009142179	192.168.11.1	192.168.11.136	TCP	54	2421 → 0 [<None>] Seq=1 Win=512 Len=0

```
Header checksum: 0x157b [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.11.1
Destination: 192.168.11.136
Transmission Control Protocol, Src Port: 2420, Dst Port: 0, Seq: 1, Len: 0
Source Port: 2420
Destination Port: 0
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 1 (relative sequence number)
Sequence number (raw): 5201314
```

## TCP三次握手



HOST A需要与HOST B建立连接，则需要发送一个 SYN请求（seq = x，顺序号码）发放至HOST B，HOST B收到HOST A的SYN数据后发送数据为（seq = y，顺序号码，ACK(来表示确定收到数据包)），之后HOST A发送数据（ACK = y+1，来表示已经确定收到数据）之后HOST B 已经知道了HOST A收到了“我”发送的数据。

## Flags

hping3 -L 520 192.168.11.136

设置TCP ACK (Acknowledgement，确认字符)

TCP ACK全称为 (Acknowledgement) 即确认字符，在数据通信中，接受站发给发送站的一种传输类控制字符。用于表示已经发来的数据以确认接受无误。

而ACK信号一般为为ASCII字符，在不同协议之中，ACK信号并不是一样的。

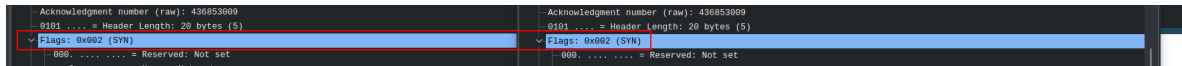
```
root@parrot: /home/kun
#hping3 -L 520 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data
len=40 ip=192.168.11.136 ttl=64 DF id=55543 sport=0 flags=RA seq=0 win=0 rtt=7.8 ms
len=40 ip=192.168.11.136 ttl=64 DF id=55544 sport=0 flags=RA seq=1 win=0 rtt=3.8 ms
len=40 ip=192.168.11.136 ttl=64 DF id=55545 sport=0 flags=RA seq=2 win=0 rtt=11.6 ms
len=40 ip=192.168.11.136 ttl=64 DF id=55546 sport=0 flags=RA seq=3 win=0 rtt=3.5 ms
--- 192.168.11.136 hping statistic ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 3.5/6.7/11.6 ms
root@parrot: /home/kun
```

```
Sequence number: 1 (relative sequence number)
Sequence number (raw): 52015017
Next sequence number: 1 (relative sequence number)
Acknowledgment number: 520
[Expert Info (Note/Protocol): The acknowledgment number field is nonzero while the ACK flag is not set]
  [The acknowledgment number field is nonzero while the ACK flag is not set]
  [Severity level: Note]
  [Group: Protocol]
Acknowledgment number (raw): 520
9191 ... = Header Length: 20 bytes (5)
```

hping3 -S 192.168.11.136

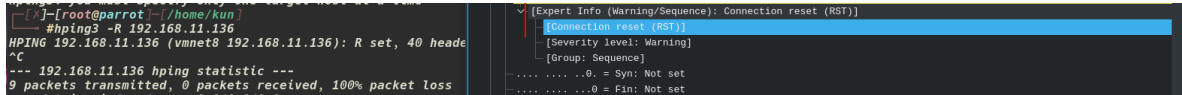
设置SYN (Synchronize Sequence Numbers) 标志[^主要建立连接]

SYN全称 ( Synchronize Sequence Numbers ) 中文译名为同步序列编号, 是TCP/IP建立连接是所使用的握手信号。在客户机和服务器之间建立正常的TCP网络连接时, 客户机首先需要发送一个SYN信息, 服务器收到了在使用SYN/ACK来应答此消息



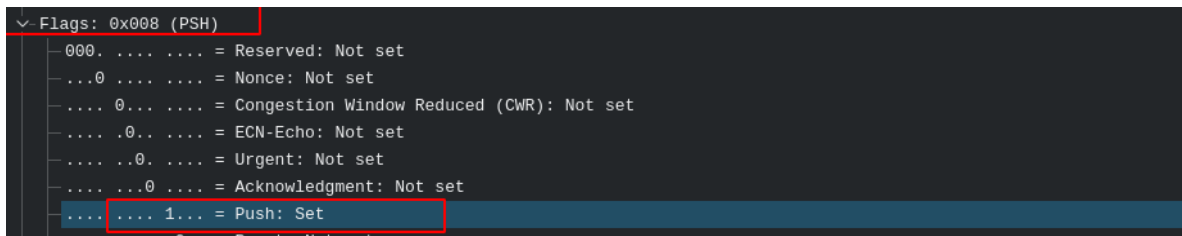
hping3 -R 192.168.11.136

设置RST ( Reset ) 重置连接标志[^主要用于重置连接]



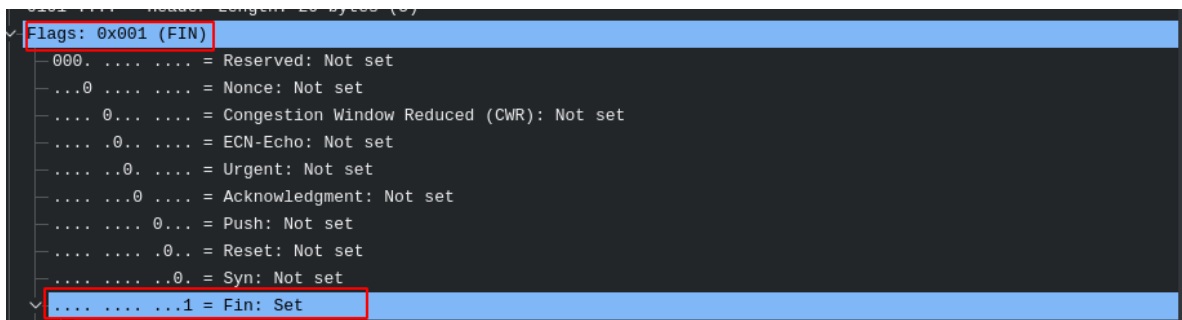
hping3 -P 192.168.11.136

设置Push标志[^表示有数据传输]



hping3 -F 192.168.11.136

设置Fin标识[^表示关闭连接]



hping3 -A 192.168.11.136

设置ACK ( Acknowledgment ) 模式[^表示响应]

ACK ( Acknowledgment ), 中文译名为“却热字符”, 在数据的通信中, 用于接受对方发送的请求, 用于确认数据已经成功发放, 通常ACK数据为ASII信号。

```

1 0.000000000 192.168.11.1 192.168.11.136 TCP 54 2410 - 0 [ACK] Seq=1 Ack=1 Win=512
0101 .... = Header Length: 20 bytes (5)
  ✓ Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... ....0... = Push: Not set
    .... .....0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set

```

hping -U 192.168.11.136

设置URG (Urgent, 紧急) 标志

```

Acknowledgment number: 1267121190
  ✓ [Expert Info (Note/Protocol): The acknowledgment number field is nonzero while the ACK flag is not set]
    [The acknowledgment number field is nonzero while the ACK flag is not set]
    [Severity level: Note]
    [Group: Protocol]
Acknowledgment number (raw): 1267121190
0101 .... = Header Length: 20 bytes (5)
  ✓ Flags: 0x020 (URG)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ...1 .... = Urgent: Set
    .... ....0... = Acknowledgment: Not set
    .... ....0... = Push: Not set
    .... .....0.. = Reset: Not set
    .... .... ..0. = Syn: Not set

```

hping3 -X 192.168.11.136

设置X个为使用标志

```

0101 .... = Header Length: 20 bytes (5)
  ✓ Flags: 0x040 (ECN)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .1.. = ECN-Echo: Set
    .... ..0. = Urgent: Not set

```

hping3 -Y 192.168.11.136

设置y个为未用标志

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.11.1	192.168.11.136	TCP	54	2719 - 0 [CWR] Seq=1 Win=512 Len=0

```

  ✓ Flags: 0x080 (CWR)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 1... = Congestion Window Reduced (CWR): Set

```

hping3 --tcpexitcode 192.168.11.136

使用TCP\_Flags作为推出码

```

.... ..0 = Fin: Not set
[TCP Flags: 00000000]

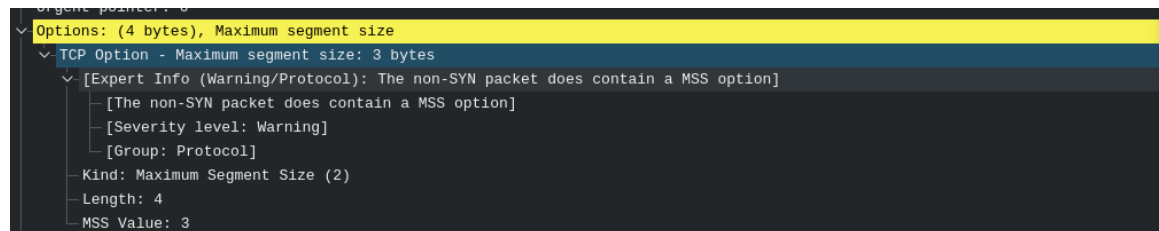
```

hping3 --tcp-mss 192.168.11.136

设置MSS ( Maximum segment size ) 最大值，并启用MSS选项

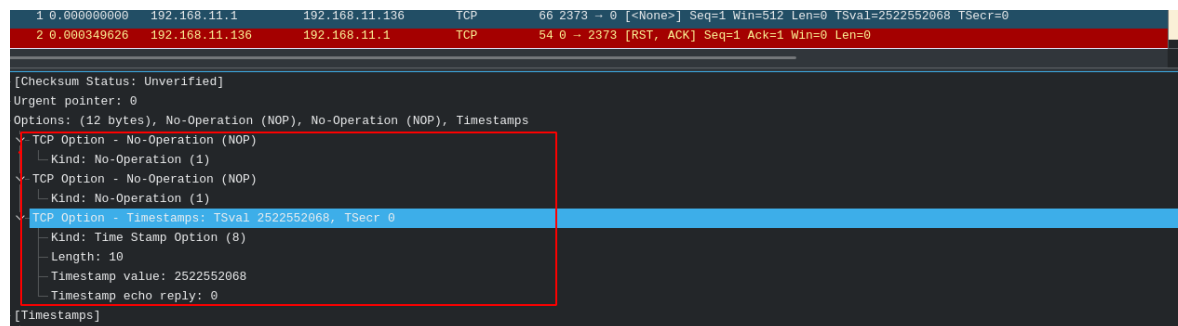
MSS，全称为“Maximum segment size”即最大报文长度。在发送SYN连接时，同时将MSS发送给对方，而MSS紧紧只会出现在SYN之中，而MSS主要的作用就是告诉对方最大的报文长度。

在网络传输数据时，数据最终需要交付到链路层协议，最后需要伪装成一个“帧”。而帧的大小如果超过了1500字节，则需要进行分片，而且这个是必须要做的。



hping3 --tcp-timestamp 192.168.11.136

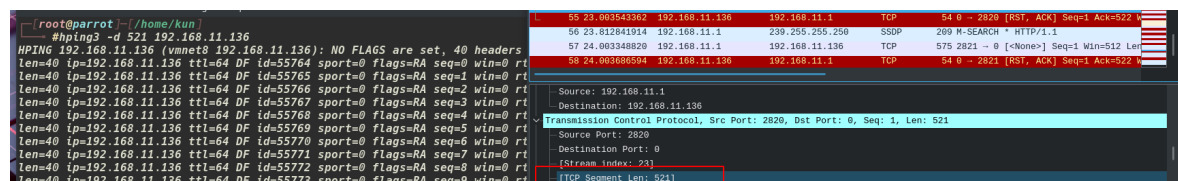
允许TCP时间戳选项猜测



Common ( 模式 )

hping3 -d 521 192.168.11.136

设置数据包TCP段 ( TCP Segment Len ) 大小为“521”



-E

数据文件

-e

添加TCP签名

-B

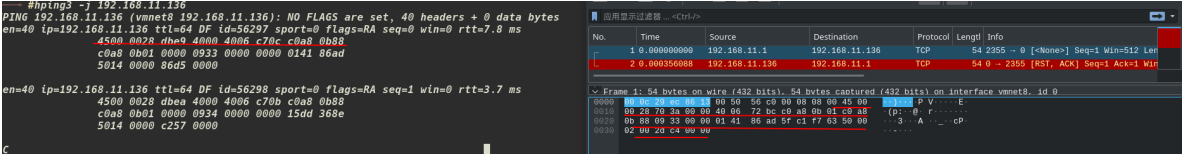
启用安全协议

-U

告诉你文件什么时候到达，并防止文件倒带

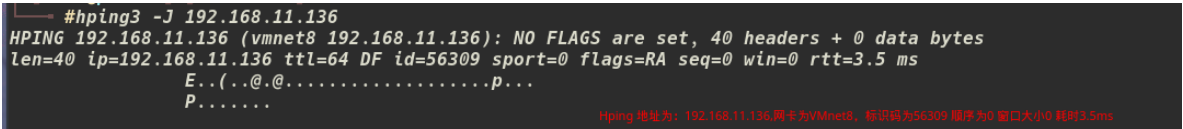
hping3 -j 192.168.11.136

转换为十六进制数据包[^O.o']



hping3 -J 192.168.11.136

输出为可打印字符



hping3 -T

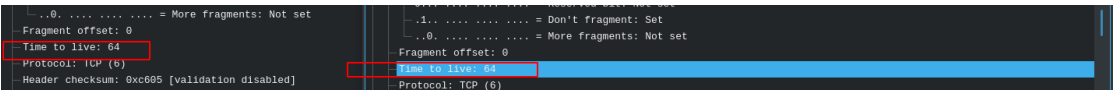
暗指--bind和--ttl

--tr-stop

在traceroute模式下，如果接收到了ICMP协议的情况下，自动推出 --tr-stop

--tr-keep-ttl

保持一个源ttl固定



## 附件1 命令组合

查看数据包

hping3 -j -j 192.168.11.136

可显示发送时数据包的16进制和大部分信息

```
#hping3 -j -J 192.168.11.136
HPING 192.168.11.136 (vmnet8 192.168.11.136): NO FLAGS are set, 40 headers + 0 data bytes
len=40 ip=192.168.11.136 ttl=64 DF id=56562 sport=0 flags=RA seq=0 win=0 rtt=7.5 ms
4500 0028 dcf2 4000 4006 c603 c0a8 0b88
c0a8 0b01 0000 0486 0000 0000 4277 8b9c
5014 0000 455d 0000

E..(..@.....Bw..
P...E]..
```

\*vmnet8

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

udp

Time	Source	Destination	Protocol	Length	Info
1 0.000000000	192.168.11.1	192.168.11.136	TCP	54	1158 → 0 [<None>] Seq=1 Win=512 Len=0
2 0.000299873	192.168.11.136	192.168.11.1	TCP	54	0 → 1158 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

- 0000 00.. = Differentiated Services Codepoint: Default (0)
- .... 00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

Total Length: 40

```
0000 00 0c 29 ec 86 13 00 50 56 c0 00 08 08 00 45 00 ..)....P.V....E.
0010 00 28 38 42 00 00 40 06 aa b4 c0 a8 0b 01 c0 a8 8B...@.....
0020 00 b8 04 86 00 00 42 77 8b 9c 41 52 21 7e 50 00 ....Bw..AR!~P.
0030 02 00 e0 a0 00 00 .....
```