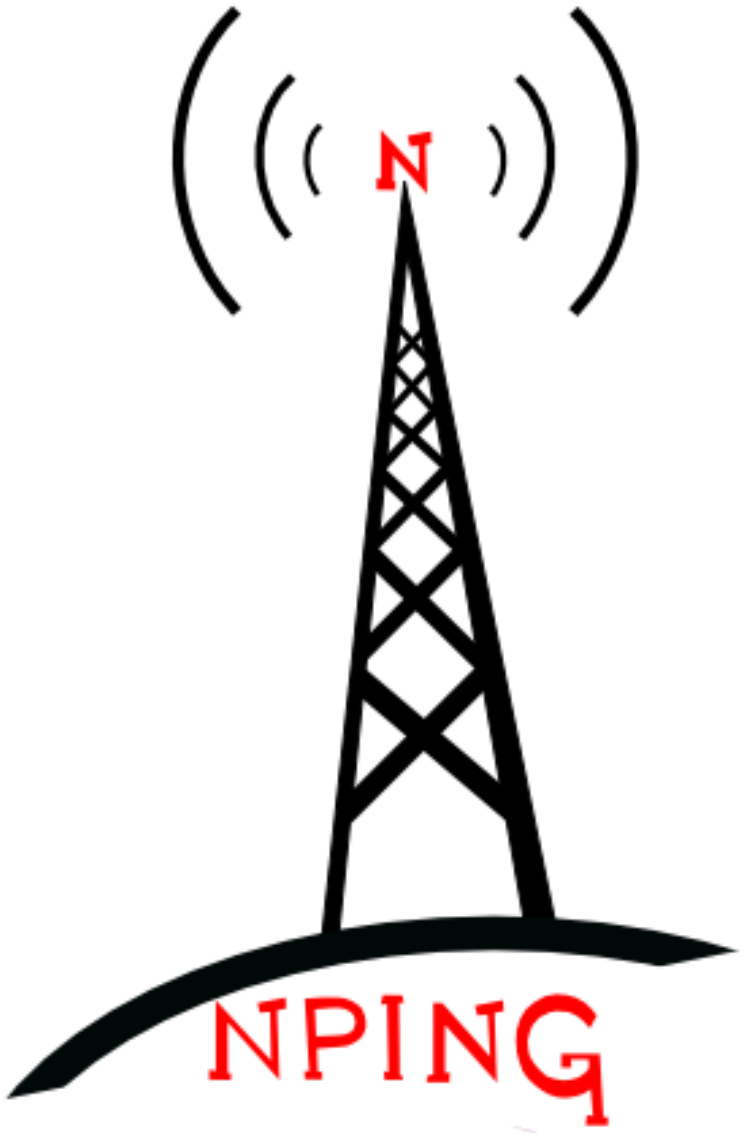
网络探测的中流砥柱-Nping使用指南

笔者：Rokas.Yang@gmail.com



一、前言

nping为nmap的子命令，和nmap一样为免费开源的探测器，只要安装好nmap就能使用nping，支持高度自定义的报文定制及探测，本文将从nping五大探测模式及各个参数用法详细展开介绍。

二、探测模式(PROBE MODES)

附上nping man文档 支持的几大探测模式说明：

参数	说明
--tcp-connect	无特权的tcp连接探测
--tcp	tcp探测
--udp	udp探测
--icmp	icmp探测
--arp	arp/rarp探测
--tr/--traceroute	路由跟踪模式(需配合--tcp/--udp/--icmp一起使用)

以下将从以上探测模式展开说明主流及高级用法。

1.tcp连接模式(--tcp-connect)

此模式包含-p、-g两大基础参数，分别为指定dest port、src port

参数	说明
-p/--dest-port	指定目的端口
-g/--source-port	指定源端口

1）指定目的端口探测(-p/--dest-port)

```
nping --tcp-connect -p 80 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --tcp-connect -p 80 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 06:52 CST
SENT (0.0018s) Starting TCP Handshake > 192.168.1.1:80
RCVD (0.0021s) Handshake with 192.168.1.1:80 completed
SENT (1.0033s) Starting TCP Handshake > 192.168.1.1:80
RCVD (1.0036s) Handshake with 192.168.1.1:80 completed
SENT (2.0049s) Starting TCP Handshake > 192.168.1.1:80
RCVD (2.0051s) Handshake with 192.168.1.1:80 completed
SENT (3.0063s) Starting TCP Handshake > 192.168.1.1:80
RCVD (3.0067s) Handshake with 192.168.1.1:80 completed
SENT (4.0080s) Starting TCP Handshake > 192.168.1.1:80
RCVD (4.0082s) Handshake with 192.168.1.1:80 completed

Max rtt: 0.330ms | Min rtt: 0.195ms | Avg rtt: 0.258ms
TCP connection attempts: 5 | Successful connections: 5 | Failed: 0 (0.00%)
Nping done: 1 IP address pinged in 4.01 seconds
```

有连接的探测，可以看到屏幕输出正常完成80端口的tcp建联，且不指定探测次数默认只发起5次，指定次数可用-c参数。

-p参数可以指定单个或多个端口，比如指定特定的几个端口：

```
nping --tcp-connect -p 22,80,443 192.168.1.1
```

指定端口范围：

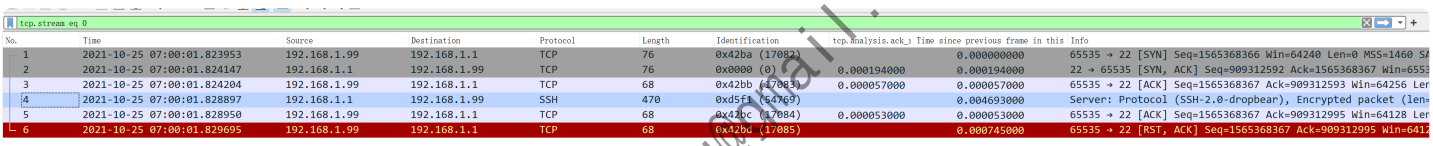
```
nping --tcp-connect -p 22-443 192.168.1.1
```

2）指定源端口探测(-g/--source-port)

使用65535高端口对目标主机22端口发起探测，-c指定次数为1次：

```
nping --tcp-connect -g 65535 -p 22 192.168.1.1 -c 1
```

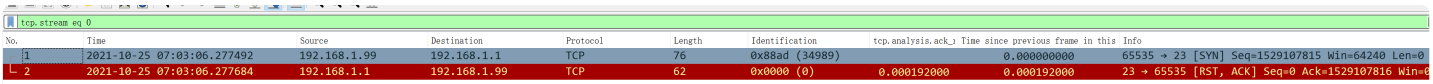
```
┌──(root@kali)-[~]
└─# nping --tcp-connect -g 65535 -p 22 192.168.1.1 -c 1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 07:00 CST
SENT (0.0019s) Starting TCP Handshake > 192.168.1.1:22
RCVD (0.0075s) Handshake with 192.168.1.1:22 completed

Max rtt: 5.602ms | Min rtt: 5.602ms | Avg rtt: 5.602ms
TCP connection attempts: 1 | Successful connections: 1 | Failed: 0 (0.00%)
Nping done: 1 IP address pinged in 0.01 seconds
```



No.	Time	Source	Destination	Protocol	Length	Identification	tcp.analysis.ack_	Time since previous frame in this	Info
1	2021-10-25 07:00:01.823953	192.168.1.99	192.168.1.1	TCP	76	0x42ba (17082)		0.000000000	65535 → 22 [SYN] Seq=1565368366 Win=64240 Len=0 MSS=1460 SA
2	2021-10-25 07:00:01.824147	192.168.1.1	192.168.1.99	TCP	76	0x0000 (0)	0.000194000	0.000194000	22 → 65535 [SYN, ACK] Seq=909312592 Ack=1565368367 Win=655
3	2021-10-25 07:00:01.824204	192.168.1.99	192.168.1.1	TCP	68	0x42bb (17083)	0.000057000	0.000057000	65535 → 22 [ACK] Seq=1565368367 Ack=909312593 Win=64256 Len
4	2021-10-25 07:00:01.828897	192.168.1.1	192.168.1.99	SSH	470	0xd5f1 (54769)		0.004693000	Server: Protocol (SSH-2.0-dropbear), Encrypted packet (len=
5	2021-10-25 07:00:01.828950	192.168.1.99	192.168.1.1	TCP	68	0x42bc (17084)	0.000053000	0.000053000	65535 → 22 [ACK] Seq=1565368367 Ack=909312995 Win=64128 Len
6	2021-10-25 07:00:01.829695	192.168.1.99	192.168.1.1	TCP	68	0x42bd (17085)		0.000745000	65535 → 22 [RST, ACK] Seq=1565368367 Ack=909312995 Win=641

可以看到客户端和服务端正常完成握手(65535 -> 22)，如果是非开放端口，则收到的显示是这样的：

```
┌──(root@kali)-[~]
└─# nping --tcp-connect -g 65535 -p 23 -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 07:03 CST
SENT (0.0018s) Starting TCP Handshake > 192.168.1.1:23
RCVD (0.0020s) Possible TCP RST received from 192.168.1.1:23 --> Connection refused

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
TCP connection attempts: 1 | Successful connections: 0 | Failed: 1 (100.00%)
Nping done: 1 IP address pinged in 0.00 seconds
```



No.	Time	Source	Destination	Protocol	Length	Identification	tcp.analysis.ack_	Time since previous frame in this	Info
1	2021-10-25 07:03:06.277492	192.168.1.99	192.168.1.1	TCP	76	0x88ad (34989)		0.000000000	65535 → 23 [SYN] Seq=1529107815 Win=64240 Len=0
2	2021-10-25 07:03:06.277684	192.168.1.1	192.168.1.99	TCP	62	0x0000 (0)	0.000192000	0.000192000	23 → 65535 [RST, ACK] Seq=0 Ack=1529107816 Win=0

客户端发起SYN申请握手，被对端RST,ACK回绝了。

2.tcp探测模式(--tcp)

参数	说明
-p/--dest-port	指定目的端口
-g/--source-port	指定源端口
--seq	指定序列号
--flags	指定tcp标识
--ack	指定ack数
--win	指定窗口大小
--badsum	使用随机无效checksum

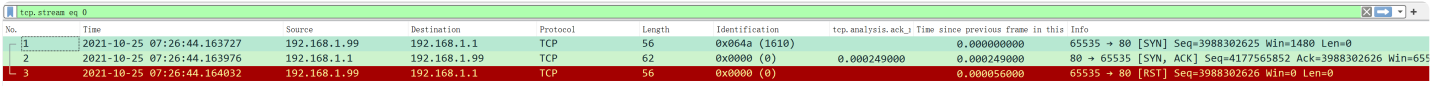
tcp探测模式和tcp-connect探测模式最大的不同是，前者不需要完成建联，效率相对更高，以下将从常用的几个参数中依次展开说明。

1）指定源/目的端口(-p/-g)

```
nping --tcp -p 80 -c 1 192.168.1.1
nping --tcp -g 65535 -p 80 -c 1 192.168.1.1
```

--tcp模式下显示的参数较--tcp-connect模式下的报文字段更详细:

```
┌──(root@kali)-[~]
└─# nping --tcp -p 80 -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 07:23 CST
SENT (0.0604s) TCP 192.168.1.99:38945 > 192.168.1.1:80 S ttl=64 id=40162 iplen=40  seq=2505637678 win=1480
RCVD (0.0608s) TCP 192.168.1.1:80 > 192.168.1.99:38945 SA ttl=64 id=0 iplen=44  seq=4169810818 win=65535 <mss 1460>

Max rtt: 0.187ms | Min rtt: 0.187ms | Avg rtt: 0.187ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.11 seconds

┌──(root@kali)-[~]
└─# nping --tcp -g 65535 -p 80 -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 07:25 CST
SENT (0.0583s) TCP 192.168.1.99:65535 > 192.168.1.1:80 S ttl=64 id=45267 iplen=40  seq=3464495581 win=1480
RCVD (0.0588s) TCP 192.168.1.1:80 > 192.168.1.99:65535 SA ttl=64 id=0 iplen=44  seq=2612628233 win=65535 <mss 1460>

Max rtt: 0.275ms | Min rtt: 0.275ms | Avg rtt: 0.275ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.09 seconds
```



和--tcp-connect显而易见的区别是--tcp模式并不需要完成建联，收到SYN,ACK后则判断为端口开放，之后发RST中断连接，和nmap的-sS半开扫描探测逻辑一致，这样可以省去不必要的交互，节省流量的同时提高探测效率。

2）指定tcp标志位(--flags)

指定标识可以发起任意标识的tcp探测，例如发起一个标志位为SYN的请求包：

```
nping --tcp -g 65535 -p 80 -c 1 --flags syn 192.168.1.1
nping --tcp -g 65535 -p 80 -c 1 --flags s 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --tcp -g 65535 -p 80 -c 1 --flags syn 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 07:44 CST
SENT (0.0343s) TCP 192.168.1.99:65535 > 192.168.1.1:80 S ttl=64 id=29699 iplen=40  seq=2903932639 win=1480
RCVD (0.0348s) TCP 192.168.1.1:80 > 192.168.1.99:65535 SA ttl=64 id=0 iplen=44  seq=3929136220 win=65535 <mss 1460>

Max rtt: 0.308ms | Min rtt: 0.308ms | Avg rtt: 0.308ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.07 seconds

┌──(root@kali)-[~]
└─# nping --tcp -g 65535 -p 80 -c 1 --flags s 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 07:45 CST
SENT (0.0563s) TCP 192.168.1.99:65535 > 192.168.1.1:80 S ttl=64 id=52622 iplen=40  seq=323145542 win=1480
RCVD (0.0568s) TCP 192.168.1.1:80 > 192.168.1.99:65535 SA ttl=64 id=0 iplen=44  seq=4196632228 win=65535 <mss 1460>

Max rtt: 0.267ms | Min rtt: 0.267ms | Avg rtt: 0.267ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.10 seconds
```


写成syn或者缩写s都行，甚至不区分大小写，也可以配合使用，比如SA即SYN,ACK，PA即PSH,ACK；当然你也可以写成16进制，范围为[0x00-0xFF]，例如指定0x012发SYN,ACK请求：

```
Flags: 0x012 (SYN, 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
    .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
```

```
┌──(root@kali)-[~]
└─# nping --tcp -g 65535 -p 80 -c 1 --flags 0x012 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 07:54 CST
SENT (0.0519s) TCP 192.168.1.99:65535 > 192.168.1.1:80 SA ttl=64 id=29446 iplen=40  seq=3284155260 win=1480
RCVD (0.0526s) TCP 192.168.1.1:80 > 192.168.1.99:65535 R ttl=64 id=0 iplen=40  seq=3679463963 win=0

Max rtt: 0.437ms | Min rtt: 0.437ms | Avg rtt: 0.437ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.10 seconds
```

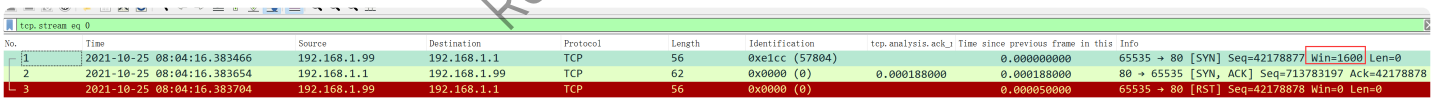
毫不意外的被目的主机发RST拒绝连接，因为在目的端看来，这是一个不完整的非法握手请求。

3.指定窗口大小(--win)

指定客户端Window size为1600字节：

```
nping --tcp -g 65535 -p 80 -c 1 --win 1600 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --tcp -g 65535 -p 80 -c 1 --win 1600 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 08:04 CST
SENT (0.0451s) TCP 192.168.1.99:65535 > 192.168.1.1:80 S ttl=64 id=57804 iplen=40  seq=42178877 win=1600
RCVD (0.0455s) TCP 192.168.1.1:80 > 192.168.1.99:65535 SA ttl=64 id=0 iplen=44  seq=713783197 win=65535 <mss 1460>

Max rtt: 0.227ms | Min rtt: 0.227ms | Avg rtt: 0.227ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.10 seconds
```

窗口大小即本端向对端说明目前本端缓冲区还能处理的最大字节数，希望对端发出的包不要超过这个值，同时如果从单个包大小的维度去看则和MTU有关，每次发包大小都由两端MTU最小的一方决定每个报文最大size，超过则需要分片发送。

3.udp探测模式(--udp)

参数	说明
-g/--source-port	指定源端口
-p/--dest-port	指定目的端口

1）指定目的端口探测(-p/--dest-port)

```
nping --udp -p 53 -c 1 192.168.1.197
```

```
┌──(root@kali)-[~]
└─# nping --udp -p 53 -c 1 192.168.1.197

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 09:01 CST
SENT (0.0580s) UDP 192.168.1.99:53 > 192.168.1.197:53 ttl=64 id=34763 iplen=28
RCVD (0.0584s) ICMP [192.168.1.197 > 192.168.1.99 Port unreachable (type=3/code=3) ] IP [ttl=64 id=55330 iplen=56 ]

Max rtt: 0.189ms | Min rtt: 0.189ms | Avg rtt: 0.189ms
Raw packets sent: 1 (28B) | Rcvd: 1 (56B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.10 seconds
```

对端回复了Port unreachabel端口不可达，即说明端口为关闭状态或者被防火墙拦截了。

2）指定源端口及目的端口(-g/--source-port)

```
┌──(root@kali)-[~]
└─# nping --udp -g 65535 -p 53 -c 1 192.168.1.197

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 08:59 CST
SENT (0.0456s) UDP 192.168.1.99:65535 > 192.168.1.197:53 ttl=64 id=61404 iplen=28
RCVD (0.0460s) ICMP [192.168.1.197 > 192.168.1.99 Port unreachable (type=3/code=3) ] IP [ttl=64 id=42270 iplen=56 ]

Max rtt: 0.250ms | Min rtt: 0.250ms | Avg rtt: 0.250ms
Raw packets sent: 1 (28B) | Rcvd: 1 (56B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.08 seconds
```


3）配合payload选项使用(--data-string)

```
nping --udp -g 65535 -p 8080 -c 1 --data-string test 192.168.1.197
```

```
┌──(root@kali)-[~]
└─# nping --udp -g 65535 -p 8080 -c 1 --data-string test 192.168.1.197

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 08:56 CST
SENT (0.0411s) UDP 192.168.1.99:65535 > 192.168.1.197:8080 ttl=64 id=58821 iplen=32
RCVD (0.0433s) UDP 192.168.1.197:8080 > 192.168.1.99:65535 ttl=64 id=50910 iplen=32

Max rtt: 1.961ms | Min rtt: 1.961ms | Avg rtt: 1.961ms
Raw packets sent: 1 (32B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.08 seconds
```

设置发起探测时的udp paylaod，前提是目的主机能识别payload并作出回应，这里用socat在目的主机上做了udp监听及回显功能：

```
root@Server ~ socat -v udp-l:8080,fork exec:'/bin/cat'
> 2021/10/25 08:56:19.337237  length=4 from=0 to=3
test< 2021/10/25 08:56:19.338180  length=4 from=0 to=3
test2021/10/25 08:56:19 socat[5463] E read(5, 0x55baba7e43f0, 8192): Connection refused
```

4.icmp探测模式(--icmp)

参数	说明
--icmp-type	设置icmp type
--icmp-code	设置icmp code
--icmp-id	设置标识id
--icmp-seq	设置序列号
--icmp-redirect-addr	设置重定向地址
--icmp-param-pointer	设置参数问题指针
--icmp-advert-entry	添加路由实体
--icmp-orig-time	设置初始时间戳
--icmp-recv-time	设置接收时间戳
--icmp-trans-time	设置传输时间戳

支持自定义的icmp参数相对较多，将详细讲解常用且不鸡肋的参数。

1）指定icmp type发起探测(--icmp-type)

nping允许用户发起自定义的icmp type、icmp code，高度自定义请求，但对于主动式的探测场景来讲，常用类型并不多，对于被动服务式则可以派上用场。

发起icmp request:

```
nping --icmp --icmp-type 8 -c 1 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --icmp --icmp-type 8 -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 19:16 CST
SENT (0.0681s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=64637 seq=1] IP [ttl=64 id=21908 iplen=28 ]
RCVD (0.0685s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=64637 seq=1] IP [ttl=64 id=29718 iplen=28 ]

Max rtt: 0.149ms | Min rtt: 0.149ms | Avg rtt: 0.149ms
Raw packets sent: 1 (28B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.11 seconds
```

这其实是一个正常的icmp请求，同时也会统计rtt时延，如果不指定--icmp-type则默认也是8。

如果是icmp reply:

```
nping --icmp --icmp-type 0 -c 1 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --icmp --icmp-type 0 -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 19:18 CST
SENT (0.0484s) ICMP [192.168.1.99 > 192.168.1.1 Echo reply (type=0/code=0) id=28487 seq=1] IP [ttl=64 id=17203 iplen=28 ]

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (28B) | Rcvd: 0 (0B) | Lost: 1 (100.00%)
Nping done: 1 IP address pinged in 1.09 seconds
```

这是一个没有意义的请求，在目的主机看来，自己并没有发icmp request却收到了icmp reply，对与莫名其妙的请求对端肯定是置之不理的，并不符合协议交互逻辑，但这个包确实可以单独发出来。

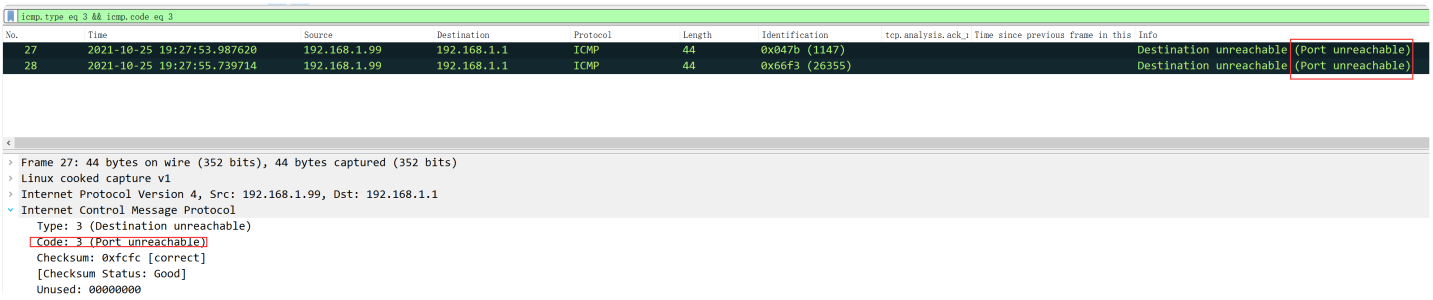
2）指定icmp code发起探测(--icmp-code)

每个icmp type都会对应一组icmp code，用于细化状态。

比如目的不可达的icmp type为3，同时icmp code为3，则会模拟端口不可达请求(Port unreachabel):

```
nping --icmp --icmp-type 3 --icmp-code 3 -c 1 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --icmp --icmp-type 3 --icmp-code 3 -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 19:27 CST
SENT (0.0333s) ICMP 192.168.1.99 > 192.168.1.1 Destination unreachable (type=3/code=3) ttl=64 id=26355 iplen=28

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (28B) | Rcvd: 0 (0B) | Lost: 1 (100.00%)
Nping done: 1 IP address pinged in 1.08 seconds
```



每个icmp type对应一组icmp code含义，可参照IANA官方对ICMP协议的type和code的分配说明。

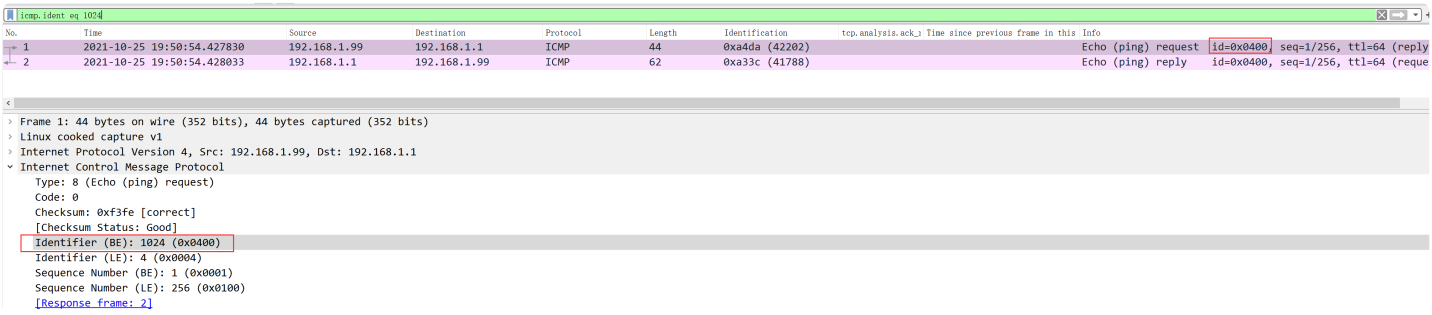
3）指定icmp id探测(--icmp-id)

顾名思义，此参数可以指定icmp identifier，用于过滤特定ident报文，特别是在大流量场景下，对端主机可以进准过滤出来自源端发起的某个包的探测，设置范围为：0<=N<2^16，即N∈[0-65536)，比如指定icmp id为1024：

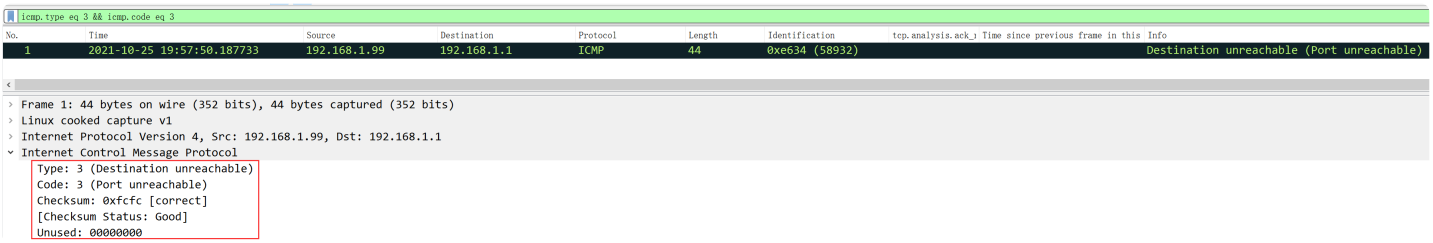
```
nping --icmp --icmp-code 0 --icmp-id 1024 -c 1 -vv 192.168.1.1  #-vv参数可以详细展示输出
```



```
┌──(root@kali)-[~]
└─# nping --icmp --icmp-code 0 --icmp-id 1024 -c 1 -vv 192.168.1.1                                                    1 ×

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 19:50 CST
SENT (0.0482s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=1024 seq=1] IP [ver=4 ihl=5 tos=0x00 iplen=28 id=422
02 foff=0 ttl=64 proto=1 csum=0x5252]
RCVD (0.0486s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=1024 seq=1] IP [ver=4 ihl=5 tos=0x00 iplen=28 id=41788
 foff=0 ttl=64 proto=1 csum=0x53f0]

Max rtt: 0.261ms | Min rtt: 0.261ms | Avg rtt: 0.261ms
Raw packets sent: 1 (28B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Tx time: 0.00131s | Tx bytes/s: 21406.73 | Tx pkts/s: 764.53
Rx time: 1.00165s | Rx bytes/s: 45.92 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.08 seconds
```



需要注意的是，当发起的是icmp不可达报文时，则icmp头部里面并不会包含id信息。

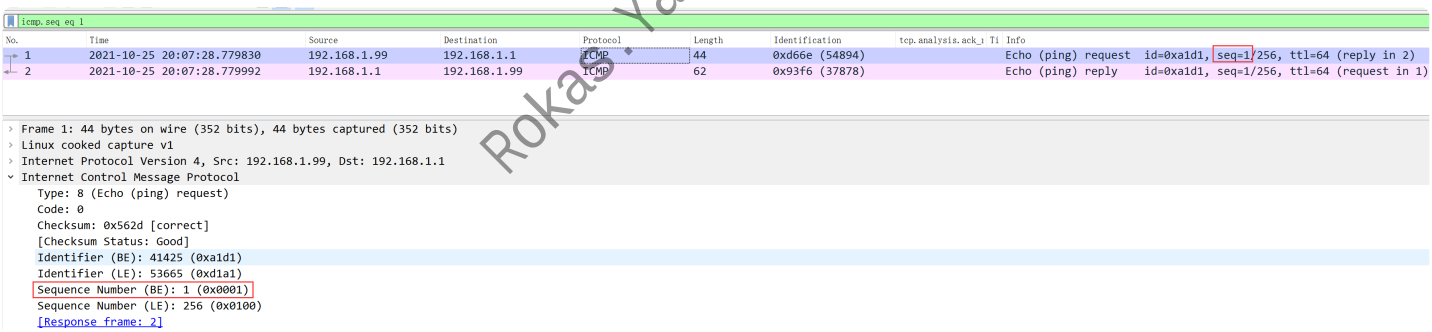


4）指定icmp seq探测(--icmp-seq)

和id功能类似，指定seq序列号：

```
nping --icmp --icmp-seq 1 -c 1 -v 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --icmp --icmp-seq 1 -c 1 -v 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 20:07 CST
SENT (0.0446s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=41425 seq=1] IP [ttl=64 id=54894 proto=1 csum=0x20be
 iplen=28 ]
RCVD (0.0450s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=41425 seq=1] IP [ttl=64 id=37878 proto=1 csum=0x6336 i
plen=28 ]

Max rtt: 0.157ms | Min rtt: 0.157ms | Avg rtt: 0.157ms
Raw packets sent: 1 (28B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Tx time: 0.00134s | Tx bytes/s: 20864.38 | Tx pkts/s: 745.16
Rx time: 1.00158s | Rx bytes/s: 45.93 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.08 seconds
```

通过icmp.seq可以进准过滤指定的序列号。

很显然，--icmp-id和--icmp-seq也可以同时使用：

```
nping --icmp --icmp-id 1024 --icmp-seq 1 -c 1 -v 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --icmp --icmp-id 1024 --icmp-seq 1 -c 1 -v 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-25 20:11 CST
SENT (0.0525s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=1024 seq=1] IP [ttl=64 id=8299 proto=1 csum=0xd6c1 i
plen=28 ]
RCVD (0.0529s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=1024 seq=1] IP [ttl=64 id=26493 proto=1 csum=0x8faf ip
len=28 ]

Max rtt: 0.214ms | Min rtt: 0.214ms | Avg rtt: 0.214ms
Raw packets sent: 1 (28B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Tx time: 0.00131s | Tx bytes/s: 21357.74 | Tx pkts/s: 762.78
Rx time: 1.00161s | Rx bytes/s: 45.93 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.10 seconds
```

5.arp/rarp探测模式(--arp)

参数	说明
--arp-type	arp探测类型，可接：arp、arp-reply、rarp、rarp-reply
--arp-sender-mac	指定发送者的MAC地址
--arp-sender-ip	指定发送者的IP地址
--arp-target-mac	指定目标主机的MAC地址
--arp-target-ip	指定目标主机的IP地址

1）指定探测类型(--arp-type)

指定ARP的探测类型，可以是ARP也可以是RARP以及对应的reply报文。

通过向对端主机发送arp报文，拿到对应的mac地址：

```
nping --arp --arp-type arp -c 1 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --arp --arp-type arp -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-31 01:45 CST
SENT (0.0559s) ARP who has 192.168.1.1? Tell 192.168.1.99
RCVD (0.0564s) ARP reply 192.168.1.1 is at 00:0C:29:BE:5A:26

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (42B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.09 seconds
```

同理也可以发起arp-reply，主动向对端发送自己的MAC地址：

```
nping --arp --arp-type arp-reply -c 1 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --arp --arp-type arp-reply -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-31 01:52 CST
SENT (0.0473s) ARP reply 192.168.1.99 is at 00:90:27:E2:74:38

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (42B) | Rcvd: 0 (0B) | Lost: 1 (100.00%)
Nping done: 1 IP address pinged in 1.09 seconds

┌──(root@kali)-[~]
└─#
```

```
Openwrt ×
root@OpenWrt:~# tcpdump -i any -nn -s 0 host 192.168.1.99 and arp -v
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
01:52:59.031438 ARP, Ethernet (len 6), IPv4 (len 4), Reply 192.168.1.99 is-at 00:90:27:e2:74:38, length 46
01:52:59.031446 ARP, Ethernet (len 6), IPv4 (len 4), Reply 192.168.1.99 is-at 00:90:27:e2:74:38, length 46
```

2）指定发送者的MAC地址(--arp-sender-mac)

利用此参数可以指定mac发送给对端主机，因此可以利用此行为来刷新对端的arp地址表，达到arp欺骗的目的:


```
nping --arp --arp-type arp-reply --arp-sender-mac 00:00:00:00:00:01
-c 1 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --arp --arp-type arp-reply --arp-sender-mac 00:00:00:00:00:01 -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-31 02:02 CST
SENT (0.0484s) ARP reply 192.168.1.99 is at 00:00:00:00:00:01
RCVD (0.3800s) ARP who has 192.168.1.197? Tell 192.168.1.100
RCVD (0.3800s) ARP who has 192.168.1.10? Tell 192.168.1.100
RCVD (0.3800s) ARP who has 192.168.1.99? Tell 192.168.1.100

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (42B) | Rcvd: 3 (138B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 0.41 seconds

┌──(root@kali)-[~]
└─#
```

```
Openwrt ×
root@OpenWrt:~# arp -a|grep 1.99
192.168.1.99     0x1         0x2         00:00:00:00:00:01     *        br-lan
root@OpenWrt:~#
```

3）指定发送者的IP地址(--arp-sender-ip)

同理，此参数可以伪造arp头部里的send ip address字段，让对端主机认为arp地址请求是此源地址发送的，因此也会reply给此源地址：

```
nping --arp --arp-type arp --arp-sender-ip 192.168.1.197  -c 1 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --arp --arp-type arp --arp-sender-ip 192.168.1.197  -c 1 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-31 02:10 CST
SENT (0.0394s) ARP who has 192.168.1.1? Tell 192.168.1.197
RCVD (0.0400s) ARP reply 192.168.1.1 is at 00:0C:29:BE:5A:26

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (42B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.07 seconds

┌──(root@kali)-[~]
└─#
```

```
Debian-home ×
root@Server ~ tcpdump -iany -nn -s 0 -v host 192.168.1.1 and arp -c 1
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
02:10:53.834754 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.1 tell 192.168.1.197, length 46
1 packet captured
1 packet received by filter
0 packets dropped by kernel
root@Server ~ _
```

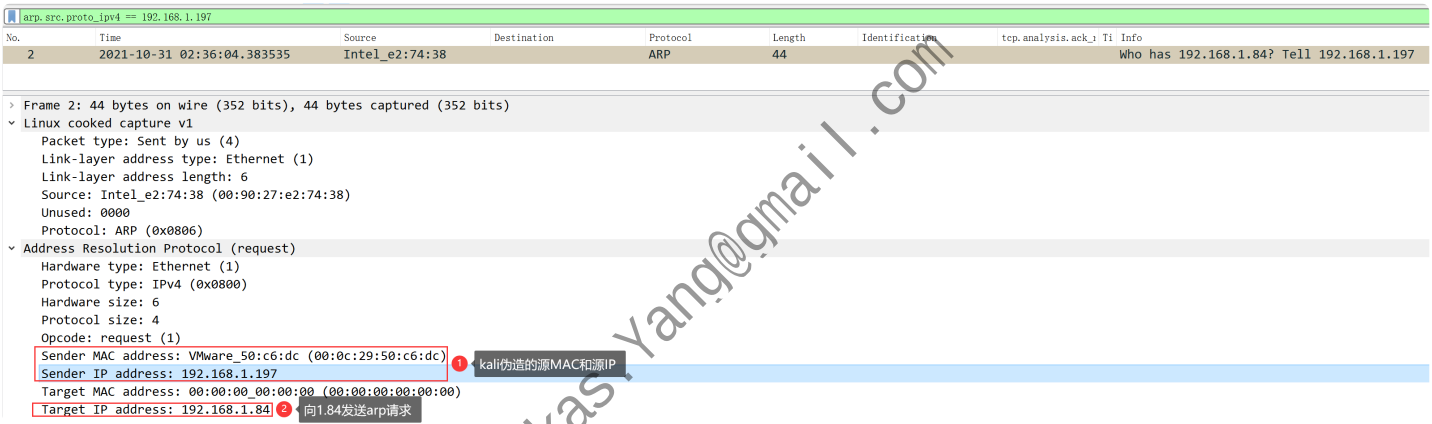

和--arp-sender-ip搭配使用，可以达到伪造IP+伪造MAC，向广播域发送arp请求，并且将请求指定回应给设置的伪造地址：

```
nping --arp --arp-type arp --arp-sender-ip 192.168.1.197 --arp-sender-mac 00:0c:29:50:c6:dc -c 1 192.168.1.84
```

```
┌──(root@kali)-[~]
└─# nping --arp --arp-type arp --arp-sender-ip 192.168.1.197 --arp-sender-mac 00:0c:29:50:c6:dc -c 1 192.168.1.84

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-31 02:26 CST
SENT (0.0563s) ARP who has 192.168.1.84? Tell 192.168.1.197

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (42B) | Rcvd: 0 (0B) | Lost: 1 (100.00%)
Nping done: 1 IP address pinged in 1.09 seconds

┌──(root@kali)-[~]
└─#
```

```
Debian-home ✕  Debian-home (1)
 root@Server  ~  tcpdump -iany -nn -s 0 -v host 192.168.1.84 and arp -v -c 2
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
02:26:32.133452 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.84 tell 192.168.1.197, length 46
02:26:32.133505 ARP, Ethernet (len 6), IPv4 (len 4), Reply 192.168.1.84 is-at 00:0c:29:92:26:bb, length 46
2 packets captured
2 packets received by filter
0 packets dropped by kernel
 root@Server  ~  ip add s ens192|grep -Po '(?i)(?<=ether\s)[\w+:]+(?=\sbrd)'
00:0c:29:50:c6:dc
 root@Server  ~
```

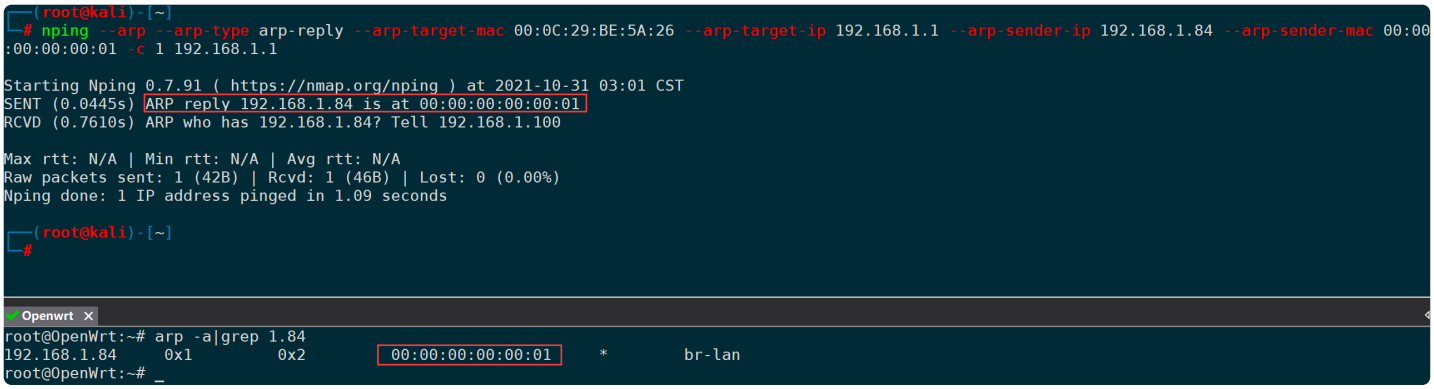
arp请求头如下：



4）指定目的主机MAC地址及IP地址(--arp-target-mac/--arp-target-ip)/ARP泛洪

配合前面几个参数使用，指定arp头部可伪造的所有IP/MAC字段，比如将1.84捆绑到一个不正确的MAC地址上，向网关主机发送arp reply，刷新、填充其ARP表：

```
nping --arp --arp-type arp-reply --arp-target-mac 00:0C:29:BE:5A:26 --arp-target-ip 192.168.1.1 --arp-sender-ip 192.168.1.84 --arp-sender-mac 00:00:00:00:00:01 -c 1 192.168.1.1
```



当捆绑伪造的IP-MAC地址对足够庞大时，使网关的arp条目耗尽，合法用户不能维持正确的arp地址表，则会导致通信中断，这种行为通常称之为arp泛洪攻击。

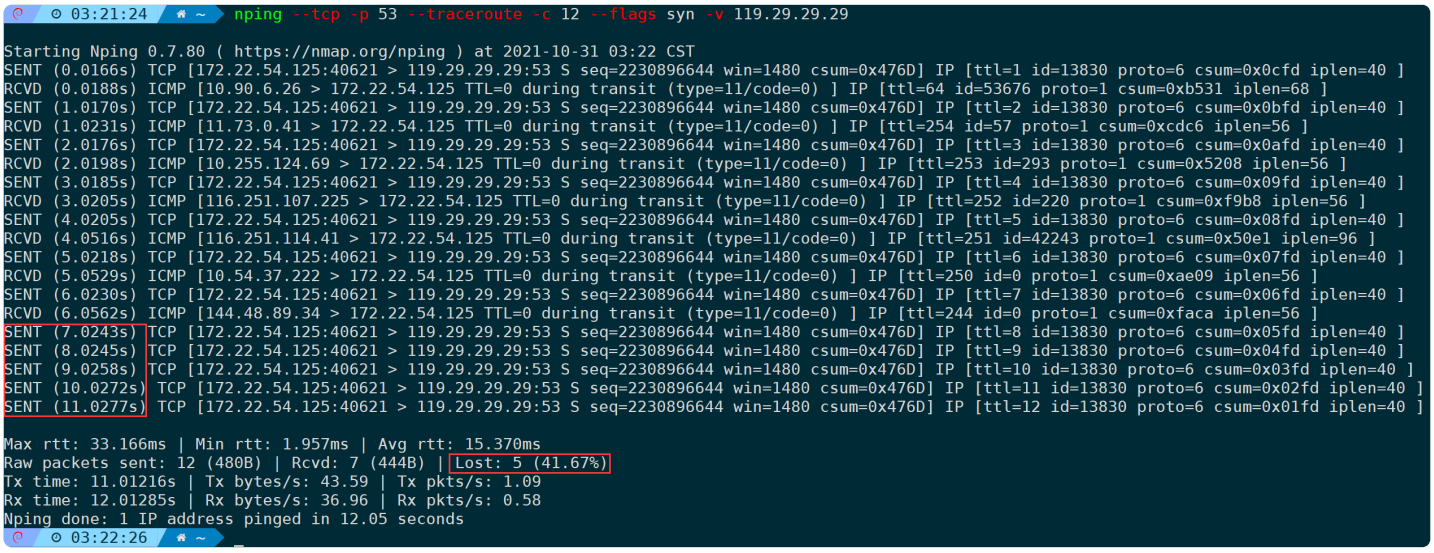
6.探测模式的路由跟踪(--tr/--traceroute)

此模式可适用于TCP/UDP/ICMP探测模式，不适用于TCP全连接和ARP模式。

当想确认链路是否有丢包，对端又禁ping了，只放开了特定协议，那么此模式配合TCP/UDP/ICMP可以精准控制路由跟踪使用的协议，--tr参数会将沿途经过的所有路由节点都探测一遍，有些路由只转发不响应(俗称拒绝回显)在client端看来其实也是超时不响应的。

对tcp/53端口进行路由跟踪:

```
nping --tcp -p 53 --traceroute -c 12 --flags syn -v 119.29.29.29
```



因为内网用openwrt做了劫持转发，--tr只会看到一跳，这里使用另一台机器做测试。

- 这里的-c 12也就是整个命令只发起12个探测包，可以理解为只探测沿途经过的12个节点，默认每个节点探测一次。
- 最后显示丢包5个(41.67%)，并不是真正意义上的丢包，探测的是对端的53/dns服务端口，没有发起dns query请求，在对端看来没有响应的必要，因此对探测机来讲，只要对端不响应，就会视为丢包。

> 前面说过，中间节点有不响应的可能，因为出于安全原因会设置某些策略禁止回显，但nping也会把这部分不响应的数据也计算到丢包率里面去做一个综合性统计，根据实际输出判断即可，不用太依赖最终的丢包率，实际工作中仅作为特定协议的路径跟踪也是不错的选择。

ICMP/UDP同理：

```
nping --udp -p 53 --traceroute -c 12 -v 119.29.29.29
nping --icmp --traceroute -c 9 -v 114.114.114.114
```

```
⊙ 03:31:05 ~ nping --udp -p 53 --traceroute -c 12 -v 119.29.29.29

Starting Nping 0.7.80 ( https://nmap.org/nping ) at 2021-10-31 03:31 CST
SENT (0.0163s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=1 id=18180 proto=17 csum=0xfbff iplen=28 ]
RCVD (0.0191s) ICMP [10.90.5.26 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=64 id=36956 proto=1 csum=0xf78d iplen=56 ]
SENT (1.0166s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=2 id=18180 proto=17 csum=0xfaff iplen=28 ]
RCVD (1.0223s) ICMP [11.73.0.85 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=254 id=508 proto=1 csum=0xcbd7 iplen=56 ]
SENT (2.0187s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=3 id=18180 proto=17 csum=0xf9ff iplen=28 ]
RCVD (2.0219s) ICMP [10.255.123.217 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=253 id=224 proto=1 csum=0x52b9 iplen=56 ]
SENT (3.0198s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=4 id=18180 proto=17 csum=0xf8ff iplen=28 ]
RCVD (3.0216s) ICMP [10.255.124.49 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=252 id=566 proto=1 csum=0x520b iplen=56 ]
SENT (4.0218s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=5 id=18180 proto=17 csum=0xf7ff iplen=28 ]
RCVD (4.0559s) ICMP [140.205.25.37 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=251 id=7237 proto=1 csum=0x1ad2 iplen=96 ]
SENT (5.0229s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=6 id=18180 proto=17 csum=0xf6ff iplen=28 ]
RCVD (5.0572s) ICMP [10.54.37.214 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=250 id=0 proto=1 csum=0xae11 iplen=56 ]
SENT (6.0239s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=7 id=18180 proto=17 csum=0xf5ff iplen=28 ]
RCVD (6.0579s) ICMP [144.48.89.33 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=244 id=0 proto=1 csum=0xfacb iplen=56 ]
SENT (7.0254s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=8 id=18180 proto=17 csum=0xf4ff iplen=28 ]
SENT (8.0259s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=9 id=18180 proto=17 csum=0xf3ff iplen=28 ]
SENT (9.0268s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=10 id=18180 proto=17 csum=0xf2ff iplen=28 ]
SENT (10.0280s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=11 id=18180 proto=17 csum=0xf1ff iplen=28 ]
SENT (11.0290s) UDP [172.22.54.125:53 > 119.29.29.29:53 csum=0x88A6] IP [ttl=12 id=18180 proto=17 csum=0xf0ff iplen=28 ]

Max rtt: 34.272ms | Min rtt: 1.765ms | Avg rtt: 16.514ms
Raw packets sent: 12 (336B) | Rcvd: 7 (432B) | Lost: 5 (41.67%)
Tx time: 11.01384s | Tx bytes/s: 30.51 | Tx pkts/s: 1.09
Rx time: 12.01465s | Rx bytes/s: 35.96 | Rx pkts/s: 0.58
Nping done: 1 IP address pinged in 12.06 seconds
⊙ 03:31:23 ~
```

```
⊙ 03:34:13 ~ nping --icmp --traceroute -c 9 -v 114.114.114.114

Starting Nping 0.7.80 ( https://nmap.org/nping ) at 2021-10-31 03:34 CST
SENT (0.0214s) ICMP [172.22.54.125 > 114.114.114.114 Echo request (type=8/code=0) id=35584 seq=1] IP [ttl=1 id=46038 proto=1 csum=0x3e93 iplen=28 ]
RCVD (0.0231s) ICMP [10.90.7.26 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=64 id=39459 proto=1 csum=0xebc6 iplen=56 ]
SENT (1.0223s) ICMP [172.22.54.125 > 114.114.114.114 Echo request (type=8/code=0) id=35584 seq=2] IP [ttl=2 id=46038 proto=1 csum=0x3d93 iplen=28 ]
RCVD (1.2124s) ICMP [11.73.0.73 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=254 id=226 proto=1 csum=0xccfd iplen=56 ]
SENT (2.0240s) ICMP [172.22.54.125 > 114.114.114.114 Echo request (type=8/code=0) id=35584 seq=3] IP [ttl=3 id=46038 proto=1 csum=0x3c93 iplen=28 ]
RCVD (2.0264s) ICMP [10.255.124.89 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=253 id=465 proto=1 csum=0x5148 iplen=56 ]
SENT (3.0248s) ICMP [172.22.54.125 > 114.114.114.114 Echo request (type=8/code=0) id=35584 seq=4] IP [ttl=4 id=46038 proto=1 csum=0x3b93 iplen=28 ]
RCVD (3.0260s) ICMP [10.255.124.49 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=252 id=455 proto=1 csum=0x527a iplen=56 ]
SENT (4.0252s) ICMP [172.22.54.125 > 114.114.114.114 Echo request (type=8/code=0) id=35584 seq=5] IP [ttl=5 id=46038 proto=1 csum=0x3a93 iplen=28 ]
RCVD (4.0616s) ICMP [140.205.25.37 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=249 id=17076 proto=1 csum=0xf61a iplen=168 ]
SENT (5.0269s) ICMP [172.22.54.125 > 114.114.114.114 Echo request (type=8/code=0) id=35584 seq=6] IP [ttl=6 id=46038 proto=1 csum=0x3993 iplen=28 ]
RCVD (5.0628s) ICMP [103.52.72.225 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=250 id=28 proto=1 csum=0x2e00 iplen=56 ]
SENT (6.0281s) ICMP [172.22.54.125 > 114.114.114.114 Echo request (type=8/code=0) id=35584 seq=7] IP [ttl=7 id=46038 proto=1 csum=0x3893 iplen=28 ]
RCVD (6.0643s) ICMP [180.97.126.169 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=248 id=116 proto=1 csum=0xac9e iplen=56 ]
SENT (7.0294s) ICMP [172.22.54.125 > 114.114.114.114 Echo request (type=8/code=0) id=35584 seq=8] IP [ttl=8 id=46038 proto=1 csum=0x3793 iplen=28 ]
RCVD (7.0622s) ICMP [180.97.126.5 > 172.22.54.125 TTL=0 during transit (type=11/code=0) ] IP [ttl=246 id=226 proto=1 csum=0xaed4 iplen=56 ]
^C
Max rtt: 190.001ms | Min rtt: 1.091ms | Avg rtt: 42.006ms
Raw packets sent: 8 (224B) | Rcvd: 8 (560B) | Lost: 0 (0.00%)
Tx time: 7.05447s | Tx bytes/s: 31.75 | Tx pkts/s: 1.13
Rx time: 7.05447s | Rx bytes/s: 79.38 | Rx pkts/s: 1.13
Nping done: 1 IP address pinged in 7.07 seconds
⊙ 03:34:46 ~
```

三、探测选项详解(OPTIONS)

1.IPv4 OPTIONS

将从如下常用参数讲解V4参数选项：

选项	说明
-S/--source-ip	设置源地址
--dest-ip	设置目的地址
--id	设置identification字段
--df	不允许分片
--ttl	设置生存周期[0-255]
--mtu	设置MTU大小

1）指定源主机(-S/--source-ip)

此参数望文生义，指定源地址的作用，可以指定其他网卡的IP，也可以指定任意IP来做伪造，比如指定网关IP向目的主机1.197发送80端口的syn探测：

```
nping --tcp -S 192.168.1.1 --flags syn -p 80 -c 1 192.168.1.197
```

```
┌──(root@kali)-[~]
└─# nping --tcp -S 192.168.1.1 --flags syn -p 80 -c 1 192.168.1.197

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-31 03:35 CST
SENT (0.0530s) TCP 192.168.1.1:12180 > 192.168.1.197:80 S ttl=64 id=52691 iplen=40  seq=2852990786 win=1480

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (40B) | Rcvd: 0 (0B) | Lost: 1 (100.00%)
Nping done: 1 IP address pinged in 1.08 seconds

┌──(root@kali)-[~]
└─#
```

```
Debian-home ×
root@Server ~ tcpdump -i any tcp port 80 and host 192.168.1.1 -nn
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
03:35:38.946428 IP 192.168.1.1.12180 > 192.168.1.197.80: Flags [S], seq 2852990786, win 1480, length 0
03:35:38.946500 IP 192.168.1.197.80 > 192.168.1.1.12180: Flags [S.], seq 2852990787, win 29200, options [mss
03:35:38.946686 IP 192.168.1.1.12180 > 192.168.1.197.80: Flags [R], seq 2852990787, win 0, length 0
```

1 197向对端响应了SYN,ACK

2 对端主机发RST拒绝连接，在对端看来，是莫名其妙的SYN,ACK，自己没有发SYN

此参数适用于所有具备IP层的探测模式。

2）指定目的主机(--dest-ip)

即指定目的主机，加不加都一样，默认会携带此参数

```
nping --icmp --icmp-type 8 -c 1 --dest-ip 192.168.1.1 192.168.1.197
 #向1.1和1.197发起icmp探测
nping --icmp --icmp-type 8 -c 1 --dest-ip 192.168.1.0/24  #向整个C端发起icmp探测
```

```
┌──(root@kali)-[~]
└─# nping --icmp --icmp-type 8 -c 1 --dest-ip 192.168.1.1 192.168.1.197

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-10-31 03:43 CST
SENT (0.0611s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=9980 seq=1] IP [ttl=64 id=6563 iplen=28 ]
RCVD (0.0615s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=9980 seq=1] IP [ttl=64 id=62840 iplen=28 ]
SENT (1.0616s) ICMP [192.168.1.99 > 192.168.1.197 Echo request (type=8/code=0) id=46717 seq=1] IP [ttl=64 id=6563 iplen=28 ]
RCVD (1.0620s) ICMP [192.168.1.197 > 192.168.1.99 Echo reply (type=0/code=0) id=46717 seq=1] IP [ttl=64 id=4464 iplen=28 ]

Statistics for host 192.168.1.1:
  |  Probes Sent: 1 | Rcvd: 1 | Lost: 0  (0.00%)
  |_ Max rtt: 0.188ms | Min rtt: 0.188ms | Avg rtt: 0.188ms
Statistics for host 192.168.1.197:
  |  Probes Sent: 1 | Rcvd: 1 | Lost: 0  (0.00%)
  |_ Max rtt: 0.301ms | Min rtt: 0.301ms | Avg rtt: 0.301ms
Raw packets sent: 2 (56B) | Rcvd: 2 (92B) | Lost: 0 (0.00%)
Nping done: 2 IP addresses pinged in 1.10 seconds

┌──(root@kali)-[~]
└─#
```


3）指定identification(--id)

identification字段，对于指定包的分析非常好用，上面说过，icmp可通过--icmp-id及--icmp-seq来标识报文，便于后期精准过滤出我们发送的特定包，同理其他探测模式也有--id字段，--id为IP层的identification，因此arp没有。

指定tcp探测的id字段，则可以写成：

```
nping --tcp -p 22 --id 1024 192.168.1.1 -v -c 1  #指定id为102
```

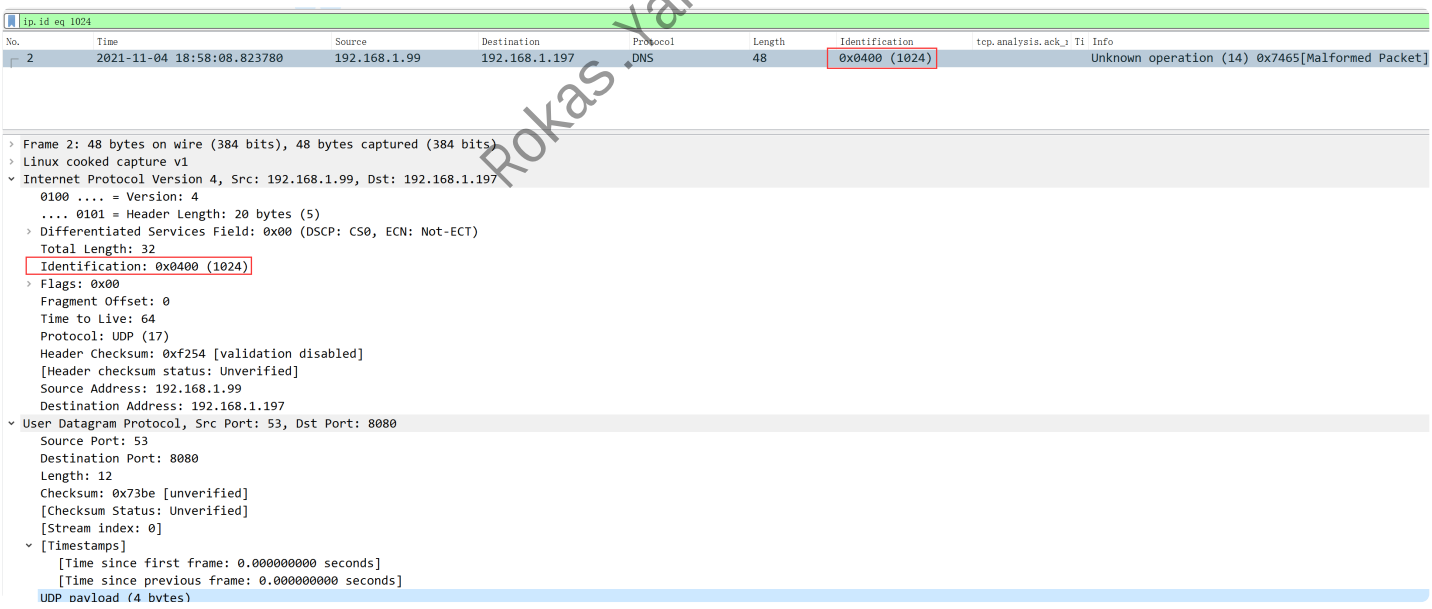
```
┌──(root@kali)-[~]
└─# nping --tcp -p 22 --id 1024 192.168.1.1 -v -c 1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 18:51 CST
SENT (0.0403s) TCP [192.168.1.99:65230 > 192.168.1.1:22 S seq=96950455 win=1480 csum=0xC902] IP [ttl=64 id=1024 proto=6 csum=0xf31b
 iplen=40 ]
RCVD (0.0407s) TCP [192.168.1.1:22 > 192.168.1.99:65230 SA seq=2471731162 win=65535 csum=0x83CF <mss 1460>] IP [ttl=64 id=0 proto=6
 csum=0xb707 iplen=44 ]

Max rtt: 0.220ms | Min rtt: 0.220ms | Avg rtt: 0.220ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Tx time: 0.00133s | Tx bytes/s: 30143.18 | Tx pkts/s: 753.58
Rx time: 1.00164s | Rx bytes/s: 45.92 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.08 seconds
```

udp探测：

```
nping --udp -p 8080 --id 1024 --data-string test 192.168.1.197 -c 1 -v
```

```
┌──(root@kali)-[~]
└─# nping --udp -p 8080 --id 1024 --data-string test 192.168.1.197 -c 1 -v

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 18:58 CST
SENT (0.0551s) UDP [192.168.1.99:53 > 192.168.1.197:8080 csum=0x73BE] IP [ttl=64 id=1024 proto=17 csum=0xf254 iplen=32 ]
RCVD (0.0573s) UDP [192.168.1.197:8080 > 192.168.1.99:53 csum=0x73BE] IP [ttl=64 id=21460 proto=17 csum=0x6280 iplen=32 ]

Max rtt: 1.957ms | Min rtt: 1.957ms | Avg rtt: 1.957ms
Raw packets sent: 1 (32B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Tx time: 0.00132s | Tx bytes/s: 24316.11 | Tx pkts/s: 759.88
Rx time: 1.00135s | Rx bytes/s: 45.94 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.09 seconds
```

可通过ip.id筛选我们发送的特定报文，之后跟踪udp.stream即可看到一条完整流：

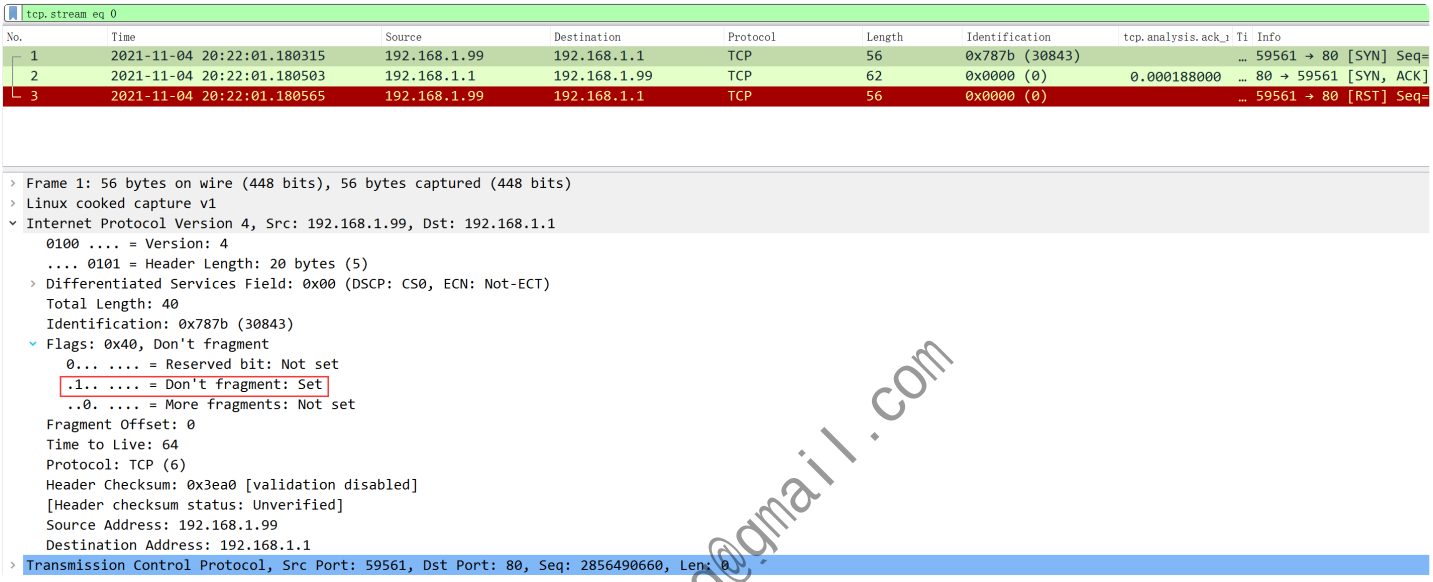


4）设置不分片标志位(--df)

即Don't Fragment不允许分片的意思，当报文大小超过client和server端协商的最小MTU时，默认会进行分片传输，此参数则指定不允许进行分片操作，所发即所得，即使对端处理不了DROP掉也还是不允许分片，此标志位在IP层，因此所有在IP层或之上的协议都能设置此字段，显而易见支持TCP、UDP、ICMP三大探测模式。

```
nping --tcp -p 80 --df 192.168.1.1 -c 1
```

```
┌──(root@kali)-[~]
└─# nping --tcp -p 80 --df 192.168.1.1 -c 1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 20:22 CST
SENT (0.0576s) TCP 192.168.1.99:59561 > 192.168.1.1:80 S ttl=64 id=30843 iplen=40  seq=2856490660 win=1480
RCVD (0.0580s) TCP 192.168.1.1:80 > 192.168.1.99:59561 SA ttl=64 id=0 iplen=44  seq=961219897 win=65535 <mss 1460>

Max rtt: 0.236ms | Min rtt: 0.236ms | Avg rtt: 0.236ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.09 seconds
```



可以看到，IP层的flag标志位的DF字段设置为了1，表示不允许分片。

那么，其他探测模式同理：


```
nping --udp -p 53 --df 192.168.1.1 -c 1
nping --icmp --icmp-type 8 --df 192.168.1.1 -c 1
```

5）设置生存周期(--ttl)

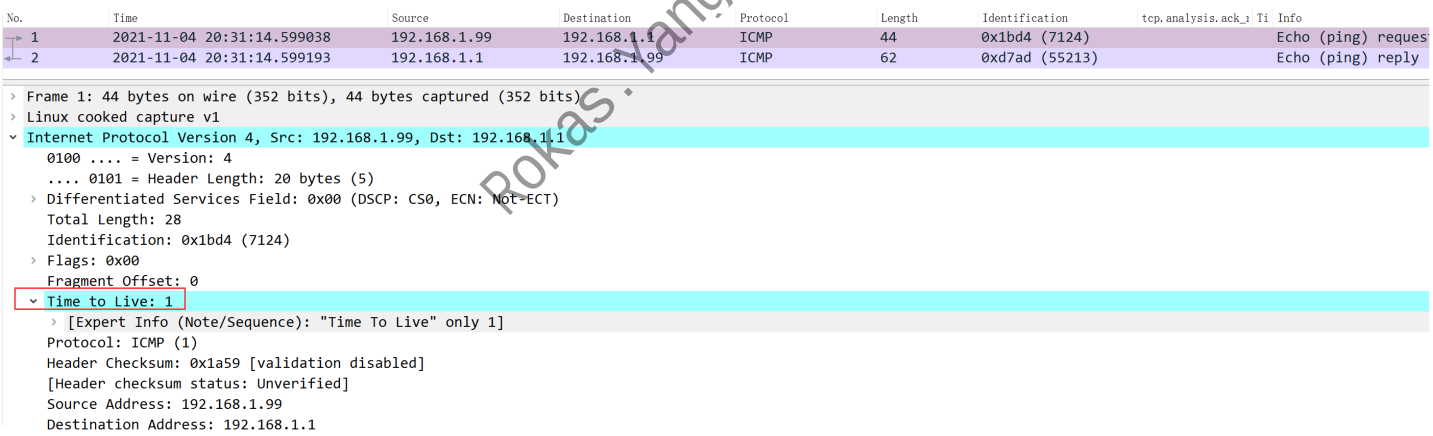
指定time to live，即报文生存周期，每经过一个路由节点，ttl减一，如果不设置ttl，则会读取系统默认ttl值，不同OS默认ttl也会不一样。

ttl在IP层头部，那么理所当然，ICMP/TCP/UDP均能被支持。

如指定ttl为1:

```
nping --icmp --ttl 1 -c 1 -v 192.168.1.1
nping --udp -p 53 --ttl 1 -c 1 -v 192.168.1.1
nping --tcp -p 80 --ttl 1 -c 1 -v 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --udp -p 53 --ttl 1 -c 1 -v 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 20:30 CST
SENT (0.0462s) UDP [192.168.1.99:53 > 192.168.1.1:53 csum=0x7BBF] IP [ttl=1 id=22642 proto=17 csum=0xddaa iplen=28 ]

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (28B) | Rcvd: 0 (0B) | Lost: 1 (100.00%)
Tx time: 0.00131s | Tx bytes/s: 21325.21 | Tx pkts/s: 761.61
Rx time: 1.00141s | Rx bytes/s: 0.00 | Rx pkts/s: 0.00
Nping done: 1 IP address pinged in 1.08 seconds

┌──(root@kali)-[~]
└─# nping --tcp -p 80 --ttl 1 -c 1 -v 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 20:30 CST
SENT (0.0425s) TCP [192.168.1.99:41974 > 192.168.1.1:80 S seq=1333014177 win=1480 csum=0x040A] IP [ttl=1 id=45335 proto=6 csum=0x8504 iplen=40
]
RCVD (0.0429s) TCP [192.168.1.1:80 > 192.168.1.99:41974 SA seq=1870731598 win=65535 csum=0x6935 <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0xb717
 iplen=44 ]

Max rtt: 0.224ms | Min rtt: 0.224ms | Avg rtt: 0.224ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Tx time: 0.00131s | Tx bytes/s: 30487.80 | Tx pkts/s: 762.20
Rx time: 1.00164s | Rx bytes/s: 45.92 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.08 seconds

┌──(root@kali)-[~]
└─# nping --icmp --ttl 1 -c 1 -v 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 20:31 CST
SENT (0.0491s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=13098 seq=1] IP [ttl=1 id=7124 proto=1 csum=0x1a59 iplen=28 ]
RCVD (0.0495s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=13098 seq=1] IP [ttl=64 id=55213 proto=1 csum=0x1f7f iplen=28 ]

Max rtt: 0.280ms | Min rtt: 0.280ms | Avg rtt: 0.280ms
Raw packets sent: 1 (28B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Tx time: 0.00129s | Tx bytes/s: 21671.83 | Tx pkts/s: 773.99
Rx time: 1.00163s | Rx bytes/s: 45.93 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.10 seconds
```



因为源端和对端是直连的，中间不需要经过路由节点，所以ttl设置为1也能正常传输数据。

6）设置MTU大小(--mtu)

指定发送方最大传输单元(Maximum Transmission Unit)，如果超出设定的值则需要分片发送。

```
nping --tcp -p 80 --mtu 16 --data-length 1000 -v 192.168.1.197 -c 1
 #--data-length指定payload大小，随机填充
nping --icmp --mtu 16 --data-length 1000 -v 192.168.1.197 -c 1
nping --udp --mtu 16 -v 192.168.1.197 -c 1
```

```
┌──(root@kali)-[~]
└─# nping --tcp -p 80 --mtu 16 --data-length 1000 -v 192.168.1.197 -c 1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 21:06 CST
SENT (0.0595s) TCP [192.168.1.99:58847 > 192.168.1.197:80 S seq=8465382 win=1480 csum=0xDD13] IP [ttl=64 id=45457 proto=6 csum=0x40de iplen=104
0 ]
RCVD (0.0610s) TCP [192.168.1.197:80 > 192.168.1.99:58847 SA seq=1089709203 win=29200 csum=0xA96E <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0xb6
53 iplen=44 ]

Max rtt: 0.424ms | Min rtt: 0.424ms | Avg rtt: 0.424ms
Raw packets sent: 1 (1.040KB) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Tx time: 0.00225s | Tx bytes/s: 461606.75 | Tx pkts/s: 443.85
Rx time: 1.00209s | Rx bytes/s: 45.90 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.09 seconds

┌──(root@kali)-[~]
└─# nping --icmp --mtu 16 --data-length 1000 -v 192.168.1.197 -c 1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 21:06 CST
SENT (0.0435s) ICMP [192.168.1.99 > 192.168.1.197 Echo request (type=8/code=0) id=31796 seq=1] IP [ttl=64 id=48124 proto=1 csum=0x3684 iplen=10
28 ]
RCVD (0.0448s) ICMP [192.168.1.197 > 192.168.1.99 Echo reply (type=0/code=0) id=31796 seq=1] IP [ttl=64 id=37617 proto=1 csum=0x5f8f iplen=1028
 ]

Max rtt: 0.414ms | Min rtt: 0.414ms | Avg rtt: 0.414ms
Raw packets sent: 1 (1.028KB) | Rcvd: 1 (1.028KB) | Lost: 0 (0.00%)
Tx time: 0.00203s | Tx bytes/s: 506653.52 | Tx pkts/s: 492.85
Rx time: 1.00174s | Rx bytes/s: 1026.22 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.08 seconds

┌──(root@kali)-[~]
└─# nping --udp -p 53 --mtu 16 --data-length 1000 -v 192.168.1.197 -c 1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 21:06 CST
SENT (0.0547s) UDP [192.168.1.99:53 > 192.168.1.197:53 csum=0x6FC0] IP [ttl=64 id=45968 proto=17 csum=0x3ee0 iplen=1028 ]
RCVD (0.0561s) ICMP [192.168.1.197 > 192.168.1.99 Port 53 unreachable (type=3/code=3) ] IP [ttl=64 id=38107 proto=1 csum=0x5ea9 iplen=576 ]

Max rtt: 0.295ms | Min rtt: 0.295ms | Avg rtt: 0.295ms
Raw packets sent: 1 (1.028KB) | Rcvd: 1 (576B) | Lost: 0 (0.00%)
Tx time: 0.00217s | Tx bytes/s: 472643.68 | Tx pkts/s: 459.77
Rx time: 1.00175s | Rx bytes/s: 574.99 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 1.11 seconds
```


2.回显模式(ECHO CLIENT/SERVER)

选项	说明
--ec/--echo-client	客户端模式
--es/--echo-server	服务端模式
--ep/--echo-port	使用自定义端口来监听或连接
--nc/--no-crypto	关闭加密和验证
--once	一次连接后停止服务器

1）client和server端(--ec/--echo-client、--es/--echo-server)

这两个参数需要一起配合使用，一个作为服务端，一个客户端，和iperf3的c/s测速模式差不多，但nping此参数则是用来作为探测用的，主要用来持续性测试点到点之间的延时。

server端：

```
nping --echo-server "public" -e ens192 -vvv  #public为任意字符串即可，要和client对应
```

```
root@Server ~ nping --echo-server "public" -e ens192 -v3

Starting Nping 0.7.40 ( https://nmap.org/nping ) at 2021-11-04 21:48 CST
Packet capture will be performed using network interface ens192.
Waiting for connections...
Server bound to 0.0.0.0:9929
[1636033709] Connection received from 192.168.1.99:34879
[1636033709] Good packet specification received from client #0 (Specs=9,IP=4,Proto=6,Cnt=5)
[1636033709] NEP handshake with client #0 (192.168.1.99:34879) was performed successfully
IPv4:TCP:Raw Data:
IPv4:TCP:Raw Data:
IPv4:TCP:Raw Data:
IPv4:TCP:Raw Data:
IPv4:TCP:Raw Data:
[1636033714] Client #0 (192.168.1.99:34879) disconnected
^C
Raw packets captured: 3701 (709.221KB) | Echoed: 5 (230B) | Not Matched: 3696 (708.991KB) (99.86%)
Tx time: 63.53232s | Tx bytes/s: 0.00 | Tx pkts/s: 0.00
Rx time: 63.53232s | Rx bytes/s: 11163.15 | Rx pkts/s: 58.25
Nping done: 1 client served in 63.53 seconds
root@Server ~
```

client端：

```
nping --echo-client "public" 192.168.1.197 --tcp -p 80
```

```
┌──(root@kali)-[~]
└─# nping --echo-client "public" 192.168.1.197 --tcp -p 80

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 21:48 CST
SENT (0.6521s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
CAPT (0.7347s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
RCVD (0.6525s) TCP 192.168.1.197:80 > 192.168.1.99:10570 SA ttl=64 id=0 iplen=44  seq=1135713294 win=29200 <mss 1460>
SENT (1.6526s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
CAPT (1.6773s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
RCVD (1.6549s) TCP 192.168.1.197:80 > 192.168.1.99:10570 SA ttl=64 id=0 iplen=44  seq=1150163029 win=29200 <mss 1460>
SENT (2.6567s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
CAPT (2.7311s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
RCVD (2.6572s) TCP 192.168.1.197:80 > 192.168.1.99:10570 SA ttl=64 id=0 iplen=44  seq=1164625777 win=29200 <mss 1460>
SENT (3.6605s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
CAPT (3.6749s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
RCVD (3.6609s) TCP 192.168.1.197:80 > 192.168.1.99:10570 SA ttl=64 id=0 iplen=44  seq=1179104262 win=29200 <mss 1460>
SENT (4.6617s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
CAPT (4.7280s) TCP 192.168.1.99:10570 > 192.168.1.197:80 S ttl=64 id=42904 iplen=40  seq=1982241044 win=1480
RCVD (4.6622s) TCP 192.168.1.197:80 > 192.168.1.99:10570 SA ttl=64 id=0 iplen=44  seq=1193544744 win=29200 <mss 1460>

Max rtt: 2.175ms | Min rtt: 0.207ms | Avg rtt: 0.662ms
Raw packets sent: 5 (200B) | Rcvd: 5 (230B) | Lost: 0 (0.00%)| Echoed: 5 (230B)
Nping done: 1 IP address pinged in 5.71 seconds
```

2）指定监听端口(--ep/--echo-port)

此参数可加可不加，即指定echo server的监听端口，不加默认为9929端口

server端：

```
nping --echo-server "public" -e ens192 -v3 --ep 10 --nc #--nc不做加密即验证，加上此参数后则client端也要对应加上
```

```
root@Server ~ nping --echo-server "public" -e ens192 -v3 --ep 10 --nc

Starting Nping 0.7.40 ( https://nmap.org/nping ) at 2021-11-04 22:05 CST
Packet capture will be performed using network interface ens192.
Waiting for connections...
Server bound to 0.0.0.0:10
[1636034759] Connection received from 192.168.1.99:43217
[1636034759] Good packet specification received from client #0 (Specs=9,IP=4,Proto=6,Cnt=2)
[1636034759] NEP handshake with client #0 (192.168.1.99:43217) was performed successfully
IPv4:TCP:Raw Data:
IPv4:TCP:Raw Data:
[1636034761] Client #0 (192.168.1.99:43217) disconnected
```

client端：

```
nping --echo-client "pubulic" 192.168.1.197 --tcp -p 80 --ep 10 --nc -c 2
```

```
┌──(root@kali)-[~]
└─# nping --echo-client "pubulic" 192.168.1.197 --tcp -p 80 --ep 10 --nc -c 2

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 22:07 CST
SENT (0.2151s) TCP 192.168.1.99:16298 > 192.168.1.197:80 S ttl=64 id=26127 iplen=40  seq=2549545244 win=1480
CAPT (0.3086s) TCP 192.168.1.99:16298 > 192.168.1.197:80 S ttl=64 id=26127 iplen=40  seq=2549545244 win=1480
RCVD (0.2157s) TCP 192.168.1.197:80 > 192.168.1.99:16298 SA ttl=64 id=0 iplen=44  seq=2140909652 win=29200 <mss 1460>
SENT (1.2176s) TCP 192.168.1.99:16298 > 192.168.1.197:80 S ttl=64 id=26127 iplen=40  seq=2549545244 win=1480
CAPT (1.2518s) TCP 192.168.1.99:16298 > 192.168.1.197:80 S ttl=64 id=26127 iplen=40  seq=2549545244 win=1480
RCVD (1.2181s) TCP 192.168.1.197:80 > 192.168.1.99:16298 SA ttl=64 id=0 iplen=44  seq=2155367872 win=29200 <mss 1460>

Max rtt: 0.401ms | Min rtt: 0.303ms | Avg rtt: 0.352ms
Raw packets sent: 2 (80B) | Rcvd: 2 (92B) | Lost: 0 (0.00%)| Echoed: 2 (92B)
Nping done: 1 IP address pinged in 2.25 seconds
```

3）一次性监听(--once)

此参数用于server端，当client端一次完整探测结束后，就关闭server端的监听。

server端：

```
nping --echo-server "public" -e ens192 -v3 --nc --once
```

```
root@Server ~ nping --echo-server "public" -e ens192 -v3 --nc --once

Starting Nping 0.7.40 ( https://nmap.org/nping ) at 2021-11-04 22:14 CST
Packet capture will be performed using network interface ens192.
Waiting for connections...
Server bound to 0.0.0.0:9929
[1636035295] Connection received from 192.168.1.99:44103
[1636035295] Good packet specification received from client #0 (Specs=6,IP=4,Proto=17,Cnt=2)
[1636035295] NEP handshake with client #0 (192.168.1.99:44103) was performed successfully
IPv4:UDP:Raw Data:
IPv4:UDP:Raw Data:
[1636035297] Client #0 (192.168.1.99:44103) disconnected

Raw packets captured: 2623 (481.635KB) | Echoed: 2 (92B) | Not Matched: 2621 (481.543KB) (99.92%)
Tx time: 5.94112s | Tx bytes/s: 0.00 | Tx pkts/s: 0.00
Rx time: 5.94122s | Rx bytes/s: 81066.68 | Rx pkts/s: 441.49
Nping done: 1 client served in 5.94 seconds
root@Server ~
```

client端：

```
nping --echo-client "pubulic" 192.168.1.197 --udp -p 8080 --nc -c 2
--once
```

```
┌──(root@kali)-[~]
└─# nping --echo-client "pubulic" 192.168.1.197 --udp -p 8080 --nc -c 2 --once

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-04 22:17 CST
SENT (1.0458s) UDP 192.168.1.99:53 > 192.168.1.197:8080 ttl=64 id=34397 iplen=28
CAPT (1.1367s) UDP 192.168.1.99:53 > 192.168.1.197:8080 ttl=64 id=34397 iplen=28
SENT (2.0555s) UDP 192.168.1.99:53 > 192.168.1.197:8080 ttl=64 id=34397 iplen=28
CAPT (2.0835s) UDP 192.168.1.99:53 > 192.168.1.197:8080 ttl=64 id=34397 iplen=28

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 2 (56B) | Rcvd: 0 (0B) | Lost: 2 (100.00%)| Echoed: 2 (92B)
Nping done: 1 IP address pinged in 3.10 seconds
```

3.延时和速率(Timing&Performance)

选项	说明
--delay	指定探测延时(单位：ms、s、m、h)
--rate	每秒发送包量

1）指定探测延时/间隔(--delay)

--delay参数可以指定每次探测包与包之间的延时间隔，当对端限制单位时间内访问频率时可用此参数来绕过检测。

每2s秒发一次SYN探测则可以写成：

```
nping --tcp -p 80,443 --flag syn -v --delay 2s -c 2 www.qq.com
```

```
┌──(root@kali)-[~]
└─# nping --tcp -p 80,443 --flag syn -v --delay 2s -c 2 www.qq.com

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-05 19:06 CST
SENT (0.0710s) TCP [192.168.1.99:28528 > 23.56.24.92:80 S seq=1768456831 win=1480 csum=0x5CD3] IP [ttl=64 id=5804 proto=6 csum=0x7285 iplen=40 ]
RCVD (0.0715s) TCP [23.56.24.92:80 > 192.168.1.99:28528 SA seq=1715927201 win=65535 csum=0xEBE5 <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0x492d iplen=44 ]
SENT (2.0729s) TCP [192.168.1.99:28528 > 23.56.24.92:443 S seq=1768456831 win=1480 csum=0x5B68] IP [ttl=64 id=5804 proto=6 csum=0x7285 iplen=40 ]
RCVD (2.0733s) TCP [23.56.24.92:443 > 192.168.1.99:28528 SA seq=2611939549 win=65535 csum=0xA4D6 <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0x492d iplen=44 ]
SENT (4.0786s) TCP [192.168.1.99:28528 > 23.56.24.92:80 S seq=1768456831 win=1480 csum=0x5CD3] IP [ttl=64 id=5804 proto=6 csum=0x7285 iplen=40 ]
RCVD (4.0791s) TCP [23.56.24.92:80 > 192.168.1.99:28528 SA seq=1773728308 win=65535 csum=0xEEE0 <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0x492d iplen=44 ]
SENT (6.0809s) TCP [192.168.1.99:28528 > 23.56.24.92:443 S seq=1768456831 win=1480 csum=0x5B68] IP [ttl=64 id=5804 proto=6 csum=0x7285 iplen=40 ]
RCVD (6.0813s) TCP [23.56.24.92:443 > 192.168.1.99:28528 SA seq=1319224483 win=65535 csum=0x361E <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0x492d iplen=44 ]

Max rtt: 0.339ms | Min rtt: 0.244ms | Avg rtt: 0.284ms
Raw packets sent: 4 (160B) | Rcvd: 4 (184B) | Lost: 0 (0.00%)
Tx time: 6.01114s | Tx bytes/s: 26.62 | Tx pkts/s: 0.67
Rx time: 6.01139s | Rx bytes/s: 30.61 | Rx pkts/s: 0.67
Nping done: 1 IP address pinged in 6.13 seconds
```

每10s发起一次udp探测：

```
nping --udp -p 53 -v -c 2 --delay 10s --data-length 16 192.168.1.197
```

```
┌──(root@kali)-[~]
└─# nping --udp -p 53 -v -c 2 --delay 10s --data-length 16 192.168.1.197

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-05 19:05 CST
SENT (0.0713s) UDP [192.168.1.99:53 > 192.168.1.197:53 csum=0xBF41] IP [ttl=64 id=24453 proto=17 csum=0x96c3 iplen=44 ]
RCVD (0.0736s) UDP [192.168.1.197:53 > 192.168.1.99:53 csum=0xBF41] IP [ttl=64 id=28940 proto=17 csum=0x453c iplen=44 ]
SENT (10.0775s) UDP [192.168.1.99:53 > 192.168.1.197:53 csum=0xBF41] IP [ttl=64 id=24453 proto=17 csum=0x96c3 iplen=44 ]
RCVD (10.0827s) UDP [192.168.1.197:53 > 192.168.1.99:53 csum=0xBF41] IP [ttl=64 id=30928 proto=17 csum=0x3d78 iplen=44 ]

Max rtt: 5.187ms | Min rtt: 2.191ms | Avg rtt: 3.689ms
Raw packets sent: 2 (88B) | Rcvd: 2 (92B) | Lost: 0 (0.00%)
Tx time: 10.00741s | Tx bytes/s: 8.79 | Tx pkts/s: 0.20
Rx time: 10.01259s | Rx bytes/s: 9.19 | Rx pkts/s: 0.20
Nping done: 1 IP address pinged in 10.12 seconds
```

icmp则可以是:

```
nping --icmp --icmp-type 8 -c 2 --delay 10s 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --icmp --icmp-type 8 -c 2 --delay 10s 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-05 19:09 CST
SENT (0.0416s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=15528 seq=1] IP [ttl=64 id=55106 iplen=28 ]
RCVD (0.0420s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=15528 seq=1] IP [ttl=64 id=40595 iplen=28 ]
SENT (10.0545s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=15528 seq=2] IP [ttl=64 id=55106 iplen=28 ]
RCVD (10.0550s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=15528 seq=2] IP [ttl=64 id=40888 iplen=28 ]

Max rtt: 0.376ms | Min rtt: 0.247ms | Avg rtt: 0.311ms
Raw packets sent: 2 (56B) | Rcvd: 2 (92B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 10.09 seconds
```


此参数适用于所有探测模式。

2）指定每秒发包量(--rate)

默认情况下，nping单位时间1s只发一次包，通过--rate可以指定任意值。

承接上面的icmp探测，指定发包量可以写成:

```
nping --icmp --icmp-type 8 --rate 5 --delay 0.01s 192.168.1.1
```

```
┌──(root@kali)-[~]
└─# nping --icmp --icmp-type 8 --rate 5 --delay 0.01s 192.168.1.1

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-05 19:19 CST
SENT (0.0445s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=13685 seq=1] IP [ttl=64 id=29584 iplen=28 ]
RCVD (0.0449s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=13685 seq=1] IP [ttl=64 id=10547 iplen=28 ]
SENT (0.0548s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=13685 seq=3] IP [ttl=64 id=29584 iplen=28 ]
RCVD (0.0551s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=13685 seq=3] IP [ttl=64 id=10548 iplen=28 ]
SENT (0.0649s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=13685 seq=4] IP [ttl=64 id=29584 iplen=28 ]
RCVD (0.0651s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=13685 seq=4] IP [ttl=64 id=10549 iplen=28 ]
SENT (0.0749s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=13685 seq=5] IP [ttl=64 id=29584 iplen=28 ]
RCVD (0.0751s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=13685 seq=5] IP [ttl=64 id=10550 iplen=28 ]
SENT (0.0850s) ICMP [192.168.1.99 > 192.168.1.1 Echo request (type=8/code=0) id=13685 seq=5] IP [ttl=64 id=29584 iplen=28 ]
RCVD (0.0852s) ICMP [192.168.1.1 > 192.168.1.99 Echo reply (type=0/code=0) id=13685 seq=5] IP [ttl=64 id=10551 iplen=28 ]

Max rtt: 0.196ms | Min rtt: 0.150ms | Avg rtt: 0.177ms
Raw packets sent: 5 (140B) | Rcvd: 5 (230B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 0.12 seconds
```

--rate指定了单位时间发5个包，--delay将发包间隔设定为0.01s，可以快速拿到返回结果。

同理，tcp的syn探测：

```
nping --tcp -p 80 -v --rate 10 192.168.1.197
```

```
┌──(root@kali)-[~]
└─# nping --tcp -p 80 -v --rate 10 192.168.1.197

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-11-05 19:25 CST
SENT (0.0424s) TCP [192.168.1.99:42108 > 192.168.1.197:80 S seq=2406201980 win=1480 csum=0x36ED] IP [ttl=64 id=37166 proto=6 csum=0x6529 iplen=40 ]
RCVD (0.0430s) TCP [192.168.1.197:80 > 192.168.1.99:42108 SA seq=2940939574 win=29200 csum=0xD655 <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0xb653 iplen=44 ]
SENT (0.1439s) TCP [192.168.1.99:42108 > 192.168.1.197:80 S seq=2406201980 win=1480 csum=0x36ED] IP [ttl=64 id=37166 proto=6 csum=0x6529 iplen=40 ]
RCVD (0.1443s) TCP [192.168.1.197:80 > 192.168.1.99:42108 SA seq=2942400981 win=29200 csum=0x89A0 <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0xb653 iplen=44 ]
SENT (0.2446s) TCP [192.168.1.99:42108 > 192.168.1.197:80 S seq=2406201980 win=1480 csum=0x36ED] IP [ttl=64 id=37166 proto=6 csum=0x6529 iplen=40 ]
RCVD (0.2450s) TCP [192.168.1.197:80 > 192.168.1.99:42108 SA seq=2943853813 win=29200 csum=0x5E6A <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0xb653 iplen=44 ]
SENT (0.3449s) TCP [192.168.1.99:42108 > 192.168.1.197:80 S seq=2406201980 win=1480 csum=0x36ED] IP [ttl=64 id=37166 proto=6 csum=0x6529 iplen=40 ]
RCVD (0.3453s) TCP [192.168.1.197:80 > 192.168.1.99:42108 SA seq=2945300230 win=29200 csum=0x4C43 <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0xb653 iplen=44 ]
SENT (0.4450s) TCP [192.168.1.99:42108 > 192.168.1.197:80 S seq=2406201980 win=1480 csum=0x36ED] IP [ttl=64 id=37166 proto=6 csum=0x6529 iplen=40 ]
RCVD (0.4454s) TCP [192.168.1.197:80 > 192.168.1.99:42108 SA seq=2946742849 win=29200 csum=0x48F2 <mss 1460>] IP [ttl=64 id=0 proto=6 csum=0xb653 iplen=44 ]

Max rtt: 0.315ms | Min rtt: 0.249ms | Avg rtt: 0.288ms
Raw packets sent: 5 (200B) | Rcvd: 5 (230B) | Lost: 0 (0.00%)
Tx time: 0.40356s | Tx bytes/s: 495.59 | Tx pkts/s: 12.39
Rx time: 0.40403s | Rx bytes/s: 569.26 | Rx pkts/s: 12.38
Nping done: 1 IP address pinged in 0.48 seconds
```

此参数也适用于所有探测模式。

4.其他参数

选项	说明
-c	到达发包次数后停止
-e/--interface	指定网卡
-H/--hide-sent	不显示发送的包
-N/--no-capture	不抓响应包
-v/-v[level]	显示详细输出，一共1-4个等级
-d/-d[level]	显示debug信息，一共3个等级
--debug	显示输出信息及debug信息为最高级

以上参数说明已经写的很详细，且部分已经做过演示，这里不再一一举例说明，详细用法可参考man文档。

四、总结

以上涵盖了nping的各个探测使用指南，功能并不逊色于nmap，有些高级功能是nmap覆盖不到的，同时两者定位也不一样，nping更偏向于网络延时分析、路径跟踪等场景，nmap更适用于端口扫描。nping也是安全领域工具中不可获缺的一部分，前期的信息收集对于渗透测试至关重要，同时也是攻击者囊中利器。