Atk6-trace6

trace6:非常快速的traceroute6，支持ICMP6回应请求和TCP-SYN

——*van Hauser*

一，帮助手册

语法:atk6-trace 6[--abdtu] [-s src6]接口和目标地址 [端口]

命令：

-a 使用路由器警报选项插入逐跳报头。
-D 插入目标扩展头
-E 插入带有无效选项的目标扩展头
-F 插入一个一次性碎片头
-b 使用TooBig而不是ICMP6 Ping（您将看不到目标）
-B 使用PingReply而不是ICMP6 Ping（您将看不到目标）
-d 将IPv6地址解析为DNS。
-t 启用隧道检测
u- 如果提供了端口，则使用UDP而不是TCP
-r raw模式（对于没有以太网的适配器网络）
-src6 指定源IPv6地址

最大跳数：31

一个基本但非常快速的traceroute6程序。如果没有指定端口，则使用ICMP6 Ping请求，否则使用TCP SYN到指定端口的数据包。选项D、E和F可以多次使用。

Syntax: atk6-trace6 [-abdtu] [-s src6] interface targetaddress [port]

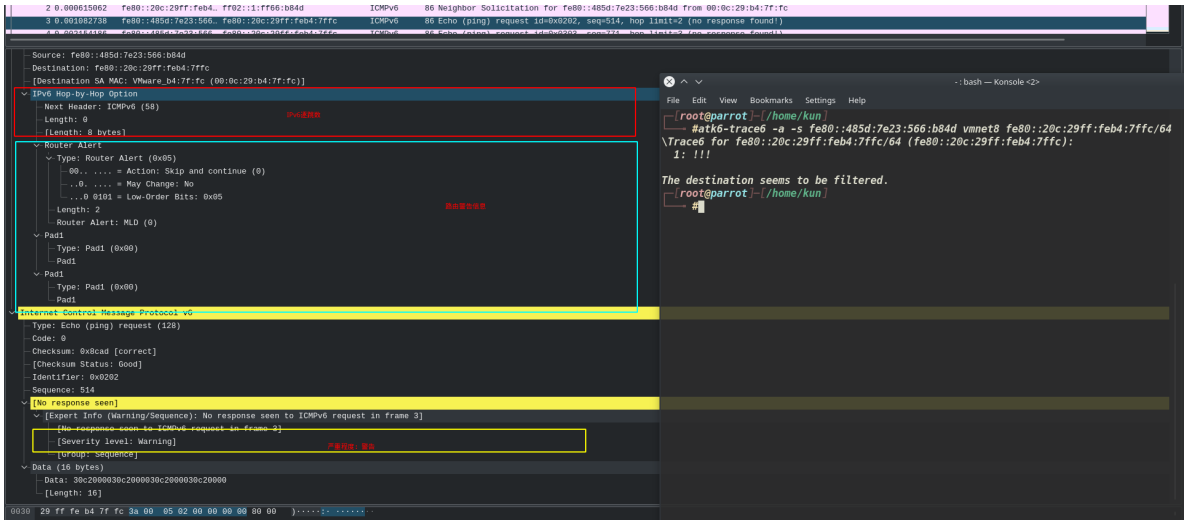
Options:
-a insert a hop-by-hop header with router alert option.
-D insert a destination extension header
-E insert a destination extension header with an invalid option
-F insert a one-shot fragmentation header
-b instead of an ICMP6 Ping, use TooBig (you will not see the target)
-B instead of an ICMP6 Ping, use PingReply (you will not see the target)
-d resolves the IPv6 addresses to DNS.
-t enables tunnel detection
-u use UDP instead of TCP if a port is supplied
-r raw mode (for adapters networks that have no ethernet)
-s src6 specifies the source IPv6 address

Maximum hop reach: 31

二，命令实例

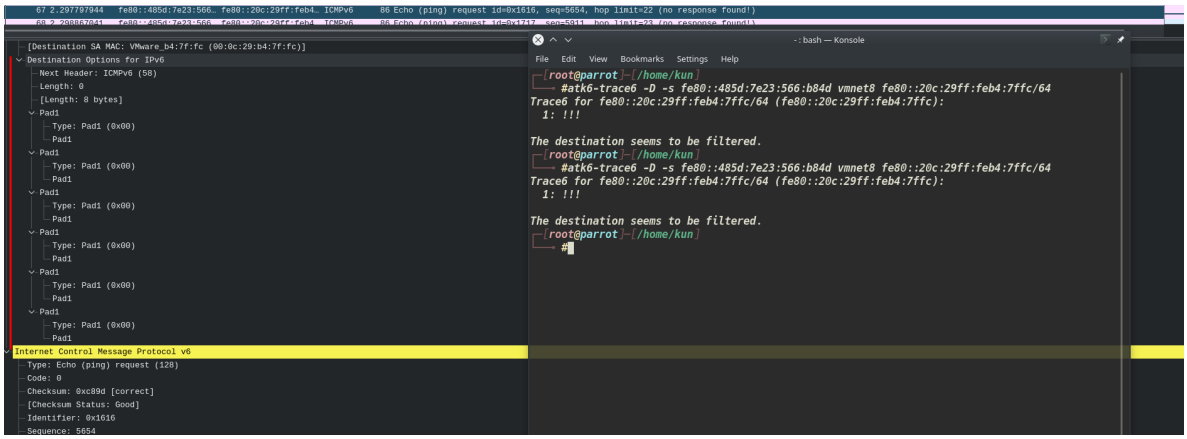
```
# ------------------------------------------------------------+
# Atk6-trace6 [-参数] [-s 原IPv6地址] [网卡] [目标IPv6地址] [端口] /
# ------------------------------------------------------------+
```

> **逐跳(Hop-by-hop)**
>
> IP网络的包转发使逐跳（Hop-by-hop）进行的。即包据哦了路由器在内的每一个节点要么将一个一个数据包直接发送给目的的节点。
>
> 要么将其数据包发送到摸底节点路径上的吓一跳节点，由下一跳继续价格数据包转发过去。
>
> 数据包必须经历所有的中间点才能到达目的，每一个路由器或是主机都是独立的。



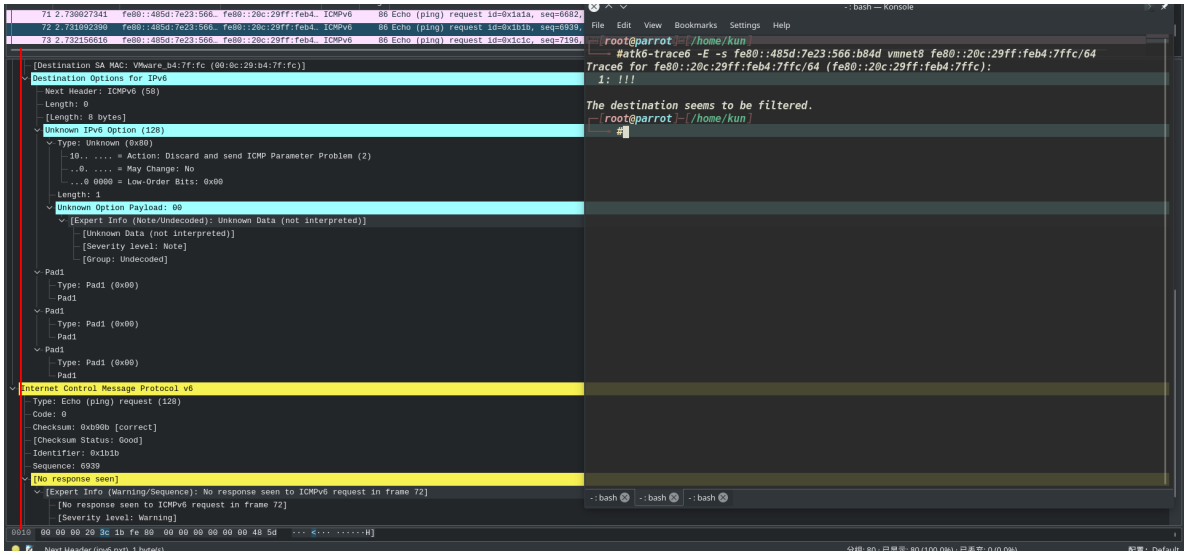
atk6-trace6 -a -s fe80::485d:7e23:566:b84d vmnet8 fe80::20c:29ff:feb4:7ffc/64

使得对方路由器报警选项并插入逐跳报头



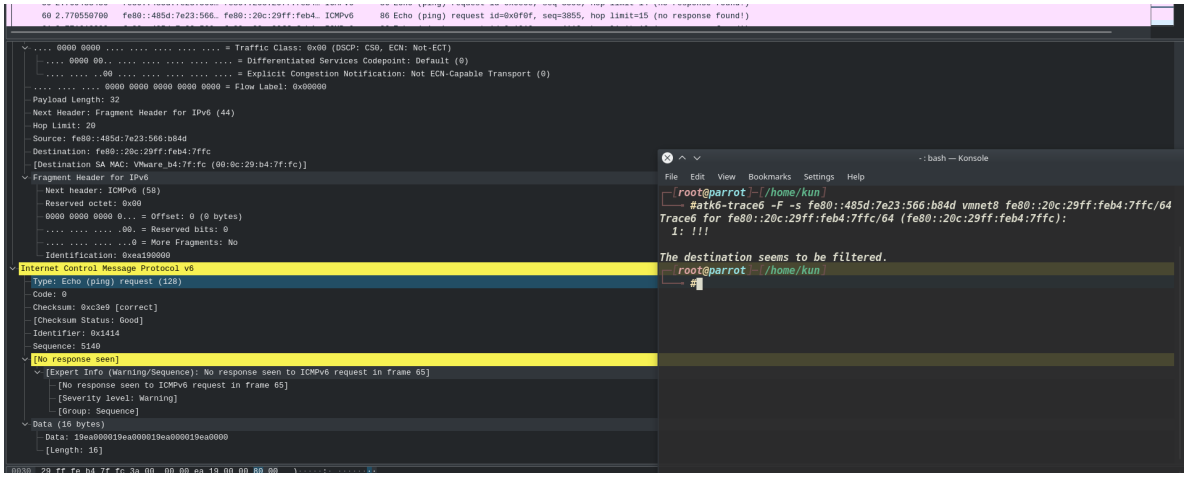
fe80::20c:29ff:feb4:7ffc/64atk6-tracke6 -D -s fe80::485d:7e23:566:b84d vmnet8

向目标数据包中插入扩展头



atk6-trace6 -E -s fe80::485d:7e23:566:b84d vmnet8 fe80::20c:29ff:feb4:7ffc/64

插入无效的选项目标扩展头



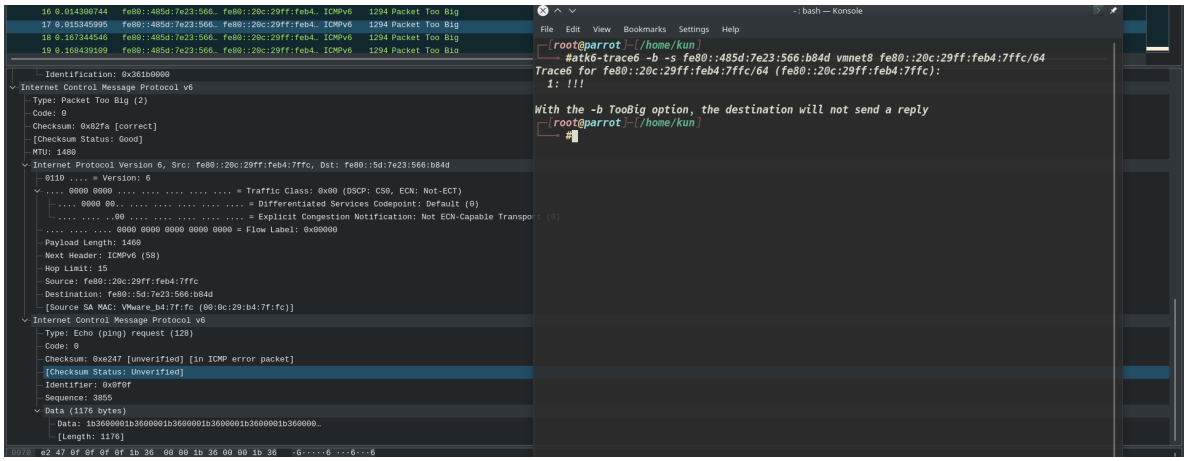
atk6-trace6 -F -s fe80::485d:7e23:566:b84d vmnet8 fe80::20c:29ff:feb4:7ffc/64

插入一个一次性的碎片头

> **TooBing**
>
> ToobBing是SNMP协议中位于get/set首部的差错状态（Error status）
>
> 由代理进程回答时填入0 ~ 5中的一个数字

差错状态	名字	说明
0	noError	一切正常
1	tooBing	代理无法将回答装入到一个SNMP报文之中

差错状态	名字	说明
2	noSuchName	操作指明了一个不存在的变量
3	badValue	一个set操作指明了一个无效值或无效语法
4	readOnly	管理进程试图改动一个仅仅读变量
5	genErr	某些其他差错



atk6-trace6 -b -s fe80::485d:7e23:566:b84d vmnet8 fe80::20c:29ff:feb4:7ffc/64

使用TooBing而不是ICMPv6 ping

Ping类尝试将internet控制信息协议（ICMP）会送请求发送到远程计算机，并通过ICMP会送答复消息从计算机接收消息。

ping类使用PingReply类实例返回有关操作信息，例如其状态和发送亲球及接收的所用时间

如果有兴趣可以使用C#写出

> 下面实例了如何使用ping同步发送ICMP回显请求和响应

```
public class PingReply
```

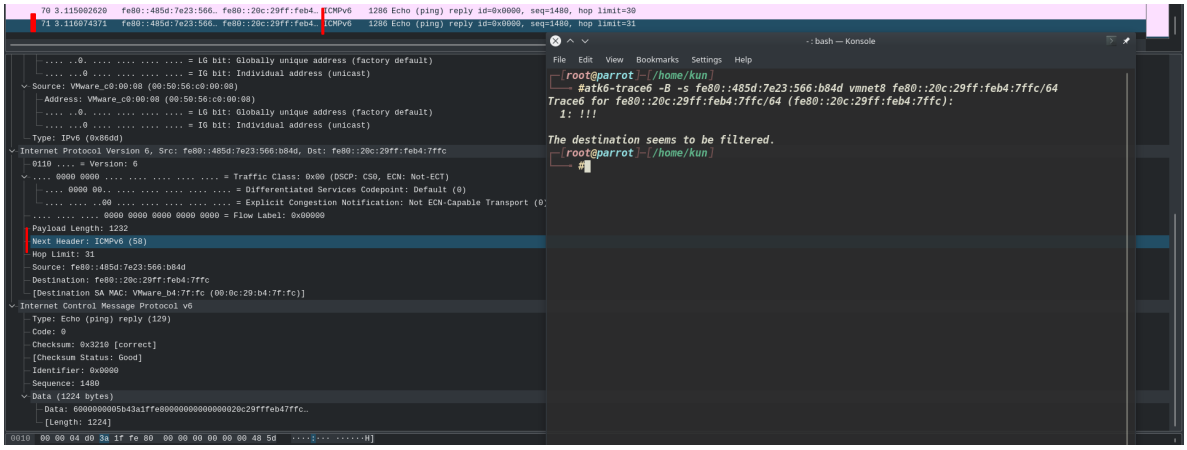
```
using System;
using System.Net;
using System.Net.NetworkInformation;
using System.Text;

namespace Examples.System.Net.NetworkInformation.PingTest
{
    public class PingExample
    {
        // args[0] can be an IPaddress or host name.
        public static void Main (string[] args)
        {
            Ping pingSender = new Ping ();
```

```
            PingOptions options = new PingOptions ();

            // Use the default Ttl value which is 128,
            // but change the fragmentation behavior.
            options.DontFragment = true;

            // Create a buffer of 32 bytes of data to be transmitted.
            string data = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
            byte[] buffer = Encoding.ASCII.GetBytes (data);
            int timeout = 120;
            PingReply reply = pingSender.Send (args[0], timeout, buffer,
options);
            if (reply.Status == IPStatus.Success)
            {
                Console.WriteLine ("Address: {0}", reply.Address.ToString
());
                Console.WriteLine ("RoundTrip time: {0}",
reply.RoundtripTime);
                Console.WriteLine ("Time to live: {0}", reply.Options.Ttl);
                Console.WriteLine ("Don't fragment: {0}",
reply.Options.DontFragment);
                Console.WriteLine ("Buffer size: {0}",
reply.Buffer.Length);
            }
        }
    }
}
```



atk6-trace6 -B -s fe80::485d:7e23:566:b84d vmnet8 fe80::20c:29ff:feb4:7ffc/64

使用PingReply而不是ICMP6 ping

> 哈哈哈哈哈哈哈哈哈哈哈哈哈哈！！！！！

> **raw socket**
>
> Raw socket即原始套子节，他和其他的套子节的不同之处在于它工作在网层或数据链路层，而其他类型的套字节工作在传输层。
>
> 在共享的网络中，所有的包都是广播的，所以的都能接受到。但是在交互式的网络中只能接收到自己的包和以广播方式发的包。

[其他命令]:

> -r 将IPV6地址解析为DNS
>
> -t 启用隧道检测
>
> -u 如果提供了端口则使用UDP而不是TCP
>
> -r raw模式（对于没有以太网的适配器网络）
>
> -s 指定源IPV6地址