Ex040 Report

Benjamin Ruddy

2022-09-24

Contents

Goal

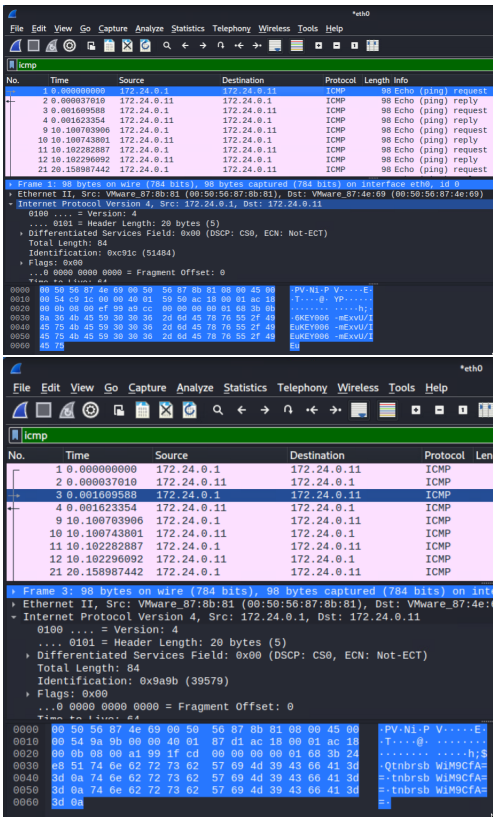
The goal in this exercise was to capture and analyze network traffic, particularly as it relates to ICMP packets.

Attack Narrative

An initial traceroute was conducted on plunder.pr0b3.com as follows:

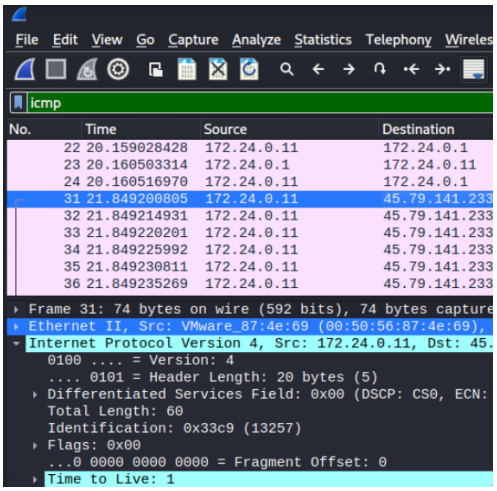
```
┌──(kali㉿kali)-[~]
└─$ sudo traceroute -I plunder.pr0b3.com
[sudo] password for kali:
traceroute to plunder.pr0b3.com (45.79.141.233), 30 hops max, 60 byte packets
 1  172.24.0.1 (172.24.0.1)  0.360 ms  0.342 ms  0.337 ms
 2  202.150.115.1 (202.150.115.1)  0.906 ms  0.901 ms  0.897 ms
 3  plunder.pr0b3.com (45.79.141.233)  1.471 ms  1.420 ms  1.406 ms
```

The `-I` option was specified to indicate the usage of ICMP Echo requests. Upon capturing traffic for around 60 seconds, the packets were filtered in Wireshark specifically for ICMP, and the first 12 showed what seemed to be KEY006 in the metadata coming from 172.24.0.1 (the network gateway):



`KEY006-mExvU/IEutnbrsbWiM9CfA==`

The ICMP packets captured from then on are mostly from our traceroute scan, which we can tell because they show up as `Echoe (ping) requests` coming from our machine's IP (172.24.0.11) with a destination of the pr0b3 domain shown earlier (45.79.141.233 is its IP address).

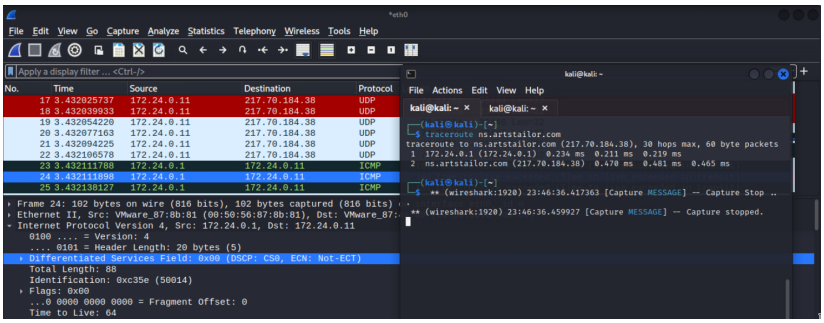


As per the traceroute specification, the network "hops" are calculated out by sending a series of packets with an increasing TTL (Time To Live), starting at 1. At each TTL number, three packets get sent out, which is reflected in Wireshark. Two hosts end up responding with TTL Exceeded messages: our local gateway (172.24.0.1) and another host (202.150.115.1) beyond the gateway but before our destination pr0b3 server.

Our destination server never returns a TTL exceeded message because once our traffic gets to it, it does not need to go further and thus it completed its journey. In other words, a TTL of 3 was all that we needed in order to reach our destination. Interestingly enough, the ICMP packet data shows that packets of up to a TTL of 8 were sent to the destination, likely because they were already being sent out before traceroute recognized that it already got a reply back from our final destination host.

Besides the KEY006 traffic, there is other ICMP traffic that our machine tries to make to the Debian NTP (Network Time Protocol) server, but cannot be completed because of a lack of internet connectivity on the NDG machines.

Finally, we made another traceroute request, this time to the nameserver `ns.artstailor.com`, which consisted of only two hops this time. As seen below, this request involved the use of UDP packets (as part of the DNS queries we're makingg) as opposed to just ICMP:



Nonetheless, the same concept of TTL flags still applies to these UDP packets, and the route to this host was similarly mapped out.

If a host were to not reply to ICMP Echo requests, some "additional methods" are implemented that use particular protocol and source/destination ports in order to bypass firewalls. Such methods include the "tcp method" wherein if some filters are preesent in the path, it crafts packets depending on the host type that specifically try to seem like regular traffic (e.g. targeting port 25 if a mail server is being traced).