

## Create Traceroute using ICMP

```
(no subject) - srivensrila... • tracerouticmp.py - De... *eth0
• tracerouticmp.py - Desktop - Visual Studio Co

minal Help

python • UDPPACKETCREATION.py SENDUDPPACKET.py PACKETCAPTURE.py ud

tracerouticmp.py > ...
3 def traceroute(target_ip, max_hops=30, timeout=2):
8     packet = IP(dst=target_ip, ttl=ttl) / ICMP()
9
10    # Send the packet and wait for a response
11    reply = srl(packet, timeout=timeout, verbose=0)
12
13    if reply is None:
14        print(f"{ttl}. No reply")
15        continue
16
17    if reply.type == 3: # Type 3 means "Destination Unreachable"
18        print(f"{ttl}. Reached destination {reply.src}")
19        break
20    elif reply.type == 11: # Type 11 means "Time Exceeded"
21        print(f"{ttl}. Hop {reply.src}")
22    else:
23        print(f"{ttl}. Unexpected reply {reply.type} from {reply.src}")
24
25    if __name__ == "__main__":
26        target = "192.168.56.1"
27        traceroute(target)
28

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
6. Unexpected reply 0 from 192.168.56.1
7. Unexpected reply 0 from 192.168.56.1
8. Unexpected reply 0 from 192.168.56.1
9. Unexpected reply 0 from 192.168.56.1
10. Unexpected reply 0 from 192.168.56.1
11. Unexpected reply 0 from 192.168.56.1
12. Unexpected reply 0 from 192.168.56.1
13. Unexpected reply 0 from 192.168.56.1
14. Unexpected reply 0 from 192.168.56.1
15. Unexpected reply 0 from 192.168.56.1
16. Unexpected reply 0 from 192.168.56.1
17. Unexpected reply 0 from 192.168.56.1
18. Unexpected reply 0 from 192.168.56.1
```

No.	Time	Source	Destination	Protocol	Length	Info
29277	3281.5960406...	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
29278	3281.6274088...	10.0.2.15	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=2 (reply in 29279)
29279	3281.6316476...	192.168.56.1	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (request in 29278)
29280	3281.6675893...	10.0.2.15	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=3 (reply in 29281)
29281	3281.6681999...	192.168.56.1	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (request in 29280)
29282	3281.7079421...	10.0.2.15	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=4 (reply in 29283)
29283	3281.7085074...	192.168.56.1	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (request in 29282)
29284	3281.7366123...	10.0.2.15	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=5 (reply in 29285)
29285	3281.7372181...	192.168.56.1	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (request in 29284)
29286	3281.7717733...	10.0.2.15	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=6 (reply in 29287)
29287	3281.7724048...	192.168.56.1	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (request in 29286)
29288	3281.8033856...	10.0.2.15	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=7 (reply in 29289)
29289	3281.8041693...	192.168.56.1	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (request in 29288)
29290	3281.8355079...	10.0.2.15	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=8 (reply in 29291)
29291	3281.8361678...	192.168.56.1	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (request in 29290)
29292	3281.8674164...	10.0.2.15	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=9 (reply in 29293)
29293	3281.8679543...	192.168.56.1	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (request in 29292)
29294	3281.8996007...	10.0.2.15	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=10 (reply in 29295)
Code: 0 (Time to live exceeded in transit)						
Checksum: 0xf4ff [correct]						
[Checksum Status: Good]						
Unused: 00000000						
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.56.1						
0100 .... = Version: 4						
..., 0101 = Header Length: 20 bytes (5)						
Differentially Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
Total Length: 28						
Identification: 0x0001 (1)						
Flags: 0x00						
0000	08 00 27 0e 34 8d 52 54	00 12 35 02 08 00 45 c0	... 4 RT ... 5 ... E			
0010	00 38 94 36 00 00 ff 01	0e be 0a 00 02 02 0a 00	... 8 6 ...			
0020	02 0f 0b 00 f4 ff 00 00	00 00 45 00 00 1c 00 01	... .. E ...			
0030	00 00 01 01 b5 28 0a 00	02 0f c0 a8 38 01 08 00	... .. 8 ...			
0040	f7 ff 00 00 00 00 00					

We can see that from the above wireshark requests and replays, the TTL is created.