How to Use Traceroute on a Security Test

When performing a security, it is a common requirement to be asked to produce a network diagram of the targeted infrastructure. The most common way of achieving this is via the **`traceroute/tracert`** commands on both Linux/Kali and Windows. The primary function of the **`traceroute/tracert`** commands is to display the amount of time it takes for a packet to go between a client and a target. This information can then be used to help diagnose network congestion. So, on Linux/Kali the traceroute command looks likes

```
kali@kali:~$ traceroute
Usage:
  traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ]
  [ -m max_ttl ] [ -N squeries ] [ -p port ] [ -t tos ] [ -l flow_label ]
  [ -w MAX,HERE,NEAR ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ]
  [ --fwmark=num ] host [ packetlen ]
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
kali:~$
```

And on Microsoft Windows the tracert command looks like

```
C:\Windows\System32>tracert
Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name
Options:
    -d                   Do not resolve addresses to hostnames.
    -h maximum_hops      Maximum number of hops to search for target.
    -j host-list         Loose source route along host-list (IPv4-only).
    -w timeout           Wait timeout milliseconds for each reply.
    -R                   Trace round-trip path (IPv6-only).
    -S srcaddr           Source address to use (IPv6-only).
    -4                   Force using IPv4.
    -6                   Force using IPv6.
C:\Windows\System32>
```

Both the traceroute and tracert commands perform the same functions. The record and display the route from your current computer system to the target computer system. The standard/default protocol that both **`traceroute`** and **`tracert`** use to map out a route is UDP.

```
C:\Windows\System32>tracert www.google.com

Tracing route to www.google.com [142.250.200.4]
over a maximum of 30 hops:

  1     7 ms     2 ms     3 ms  eehub.home [192.168.1.254]
  2    85 ms     7 ms     8 ms  172.16.16.63
  3     *        *        *     Request timed out.
  4    52 ms    52 ms    38 ms  213.121.98.128
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  9    19 ms    14 ms    13 ms  lhr48s29-in-f4.1e100.net [142.250.200.4]
```

When trace route does not get a respond from a hop in the network it marks that hope in a **`*`**. However, **`traceroute`** also support the use of ICMP for mapping out a route via the **`-I`** option.

```
kali@kali:~$ traceroute -I www.google.com
traceroute to www.google.com (216.58.213.4), 64 hops max, 72 byte packets
 1   eehub (192.168.1.254)  8.326 ms  2.140 ms  4.796 ms
 2   172.16.16.63 (172.16.16.63)  8.501 ms  6.029 ms  6.189 ms
 3   213.121.98.129 (213.121.98.129)  26.756 ms  28.816 ms  17.143 ms
 4   213.121.98.128 (213.121.98.128)  27.622 ms  12.948 ms  12.071 ms
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 9   lhr25s25-in-f4.1e100.net (216.58.213.4)  15.444 ms  11.969 ms  12.609 ms
```

Hence via cycling through a range of IP address to be tested it is possible to construct a complex network diagram showing the network route and topology.