CSCD 330 – Computer Networks

Lab 7, TCP Traceroute

Overview:

Today you will use the `scapy` skills you developed last time to create your own TCP `traceroute` tool. In addition, you will add a feature that let's you see all the Autonomous Systems (ASs) your packet traverses.

Instructions:

Complete the above program following the steps in the next section. Call this file `lab7.py` and include your name at the top of the file in a comment – #author: YOUR NAME. Your output should match the style of the example output. You must also create a README explaining how to run your program and answering the questions below. Furthermore, you must create a test script in Bash. Call this file test.sh.

Note that you must use a bridged network adaptor in Virtual Box otherwise you will only see 1 hop.

You may only use the following imports:

```
from scapy.all import *
from socket import gethostbyname
from subprocess import getstatusoutput
from sys import argv
```

Steps:

Step 0: Setup

Your program should take 2 arguments. The first argument will be the target website and the second argument will be the maximum number of hops.

Step 1: Send packets

Craft and send TCP packets, using `sr1`, with increasing TTLs from 1 up to the maximum number of hops or until you reach the target website. If you reach the target website, you must stop sending packets. For each hop, print the hop

number alongside the response IP. If you do not receive a response, print the hop number followed by "* * *". Note that you must use the options `verbose=0` and `timeout=3` for `sr1`. `verbose=0` disables `sr1's` default output and `timeout=3` stops `sr1` from waiting more than 3 seconds for a response.

As an example: `sr1(packet, verbose=0, timeout=3)`

Step 2: Print AS numbers

For each IP in the traceroute you must also lookup it's AS number using `whois`. After all the hops print, output the chain of traversed ASs removing any duplicates.

Step 3: Reverse DNS (5 points of extra credit)

At each hop do a reverse DNS lookup on the IP of said hop. Print the returned device name in addition to the hop number and IP address.

Questions:

1. Besides TCP, what other protocols can be used for a `traceroute` tool?
2. When traversing to a website, does the path remain constant every time?
3. If a packet dies before reaching the target website, what type of packet is returned?
4. Can the `whois` command be used to discover the owner of an AS number?

Turn in:

Submit a tarball with the following:

- Your source code (in Python) called lab7.py
- Your test script (in Bash) called test.sh – test at least 3 inputs.
- Your README answering the above questions and explaining how to run your program.

In case you have forgotten: `tar -czvf lab7_YOURNAME.tar.gz *.py *.sh README`

Example output:

python3 lab7.py google.com 30

route to google.com (142.250.69.206), 30 hops max
1 - 192.168.1.1
2 - 100.64.0.1
3 - 172.16.250.50
4 - * * *
5 - 206.224.66.147
6 - 142.250.163.222
7 - 142.251.229.137
8 - 142.251.48.211
9 - 142.250.69.206
Traversed AS numbers: AS31027 -> AS31042 -> AS272710 -> AS14593 -> AS15169

python3 lab7.py yahoo.com 15

route to yahoo.com (74.6.143.26), 15 hops max
1 - 192.168.1.1
2 - 100.64.0.1
3 - 172.16.250.50
4 - * * *
5 - 206.224.64.145
6 - 206.81.81.50
7 - 209.191.65.49
8 - 209.191.65.104
9 - 209.191.68.36
10 - 209.191.64.83
11 - 74.6.227.137
12 - 74.6.122.41
13 - 74.6.123.239
14 - 74.6.98.138
15 - * * *
Traversed AS numbers: AS31027 -> AS31042 -> AS272710 -> AS14593 -> AS10310 -> AS26101