

1)

a. 64.4.21.91 - Microsoft

Traceroute

tracing path from www.princeton.edu to 64.4.21.91 ...















```
traceroute to 64.4.21.91 (64.4.21.91), 30 hops max, 40 byte packets
 1 core-ns-router (128.112.128.2)  1.159 ms  0.851 ms  0.808 ms
 2 rtr-core-east-router.princeton.edu (128.112.12.225)  0.753 ms  0.855 ms  0.737 ms
 3 fw-border-87-router.princeton.edu (128.112.12.10)  1.018 ms  0.982 ms  0.883 ms
 4 rtr-border-87-router.princeton.edu (204.153.48.1)  1.339 ms  1.441 ms  1.662 ms
 5 172-96-130.unassigned.userdns.com (172.96.130.53)  4.353 ms  6.295 ms  4.044 ms
 6 172-96-130.unassigned.userdns.com (172.96.130.61)  6.468 ms  6.481 ms  172-96-130.unassigned.userdns.com (172.96.130.77)  6.075 ms
 7 bundle-ether240.202.core1.newy32aoa.net.internet2.edu (198.71.47.232)  6.765 ms  6.945 ms  6.191 ms
 8 fourhundredge-0-0-48.4079.aggr2.newy2.net.internet2.edu (163.253.2.149)  7.547 ms  fourhundredge-0-0-48.4079.aggr1.newy2.net.internet2.edu (163.253.2.123)  5.314 ms  fourhundredge
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
```

Done.

i.

ii. This traceroute seemed like it went through neighboring networks to try to find the source but did not. It started out in the princeton network and went to an unnamed DNS provider. It then went to a couple of new york routers and stopped after that. This shows that microsoft potentially has some network infrastructure close to the New York area.

b. 64.233.167.147 - Google

Hop	IP / Host Name	ISP	Netblock	Country	Loss	Response
1	172.17.0.1				0.0%	0.26ms
2	10.206.5.139				0.0%	0.35ms
3	10.206.35.8				0.0%	0.53ms
4	10.206.32.1				0.0%	1.08ms
5	100-0.gw2.cjj1.us.linode.com 173.255.239.102	AKAMAI-LINODE-AP Akamai Connected Cloud, SG	173.255.239.0/24		0.0%	0.70ms
6	ae31.r01.lga01.iem.netarch.akamai.com 23.203.156.16	AKAMAI-ASN1, NL	23.203.156.0/24		0.0%	2.06ms
7	a23-203-156-153.deploy.static.akamaitechnologies.com 23.203.156.153	AKAMAI-ASN1, NL	23.203.156.0/24		0.0%	1.80ms
8	142.251.78.59	GOOGLE, US	142.250.0.0/15		0.0%	3.56ms
9	192.178.108.20	GOOGLE, US	192.178.0.0/15		0.0%	2.41ms
10	209.85.254.239	GOOGLE, US	209.85.128.0/17		0.0%	3.75ms
11	172.253.65.167	GOOGLE, US	172.253.0.0/16		0.0%	71.71ms
12	209.85.252.120	GOOGLE, US	209.85.128.0/17		0.0%	76.07ms
13	209.85.248.5	GOOGLE, US	209.85.128.0/17		0.0%	76.00ms
14	108.170.231.145	GOOGLE, US	108.170.192.0/18		0.0%	76.71ms
15	???					

- i.
- ii. We can see that the website used is hosted on linode as the hops exit private address directly into linode. This shows that google has direct access to akamai's network and can direct traffic for the websites hosted by them into their network directly.

b. 128.138.238.18 - CU Boulder

Traceroute

tracing path from www.net.princeton.edu to 128.138.238.18 ...

```

traceroute to 128.138.238.18 (128.138.238.18), 30 hops max, 40 byte packets
 1 core-ns-router (128.112.128.2)  1.096 ms  0.908 ms  0.630 ms
 2 rtr-core-east-router.princeton.edu (128.112.12.225)  1.087 ms  0.573 ms  0.508 ms
 3 fw-border-87-router.princeton.edu (128.112.12.10)  1.001 ms  1.064 ms  0.932 ms
 4 rtr-border-87-router.princeton.edu (204.153.48.1)  1.179 ms  1.393 ms  1.356 ms
 5 172-96-130.unassigned.userdns.com (172.96.130.53)  2.793 ms  5.763 ms  3.995 ms
 6 bundle-ether1.102.core1.phil.net.internet2.edu (163.253.5.8)  4.964 ms  5.029 ms  6.029 ms
 7 fourhundredge-0-0-2.4079.core2.ashb.net.internet2.edu (163.253.1.136)  44.252 ms  44.642 ms  43.365 ms
 8 fourhundredge-0-0-1.4079.core2.clev.net.internet2.edu (163.253.1.139)  45.827 ms  44.430 ms  44.508 ms
 9 fourhundredge-0-0-2.4079.core2.eqch.net.internet2.edu (163.253.2.17)  45.076 ms  44.439 ms  44.626 ms
10 fourhundredge-0-0-2.4079.core2.chic.net.internet2.edu (163.253.2.18)  44.071 ms  44.141 ms  42.945 ms
11 fourhundredge-0-0-1.4079.core1.kans.net.internet2.edu (163.253.1.245)  45.022 ms  44.079 ms  44.664 ms
12 fourhundredge-0-0-1.4079.core1.denv.net.internet2.edu (163.253.1.242)  43.381 ms  44.819 ms  44.658 ms
13 163.253.5.43 (163.253.5.43)  42.489 ms  42.408 ms  42.486 ms
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
```

Done.

iii.

iv. As we cannot see past the internet2 network from this traceroute we can't find the connection for CU to the internet, but we can make some educated guesses. Internet2 provides networking, and direct connection, for many major university services such as canvas and eduroam. We can make the assumption that both CU and princeton are connected to this ISP and as such have a connected route through it.

b. 216.17.128.2 - Amazon

Traceroute

tracing path from www.net.princeton.edu to 216.17.128.2 ...

```
traceroute to 216.17.128.2 (216.17.128.2), 30 hops max, 40 byte packets
 1  core-ns-router (128.112.128.2)  1.111 ms  0.738 ms  0.587 ms
 2  rtr-core-east-router.princeton.edu (128.112.12.225)  0.885 ms  0.757 ms  0.532 ms
 3  fw-border-87-router.princeton.edu (128.112.12.10)  0.956 ms  0.923 ms  0.960 ms
 4  rtr-border-87-router.princeton.edu (204.153.48.1)  1.391 ms  1.392 ms  1.459 ms
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * 172-96-130.unassigned.userdns.com (172.96.130.53)  4.323 ms !N *
10  * * *
11  * * *
12  * * *
13  172-96-130.unassigned.userdns.com (172.96.130.53)  5.979 ms !N * *
14  * * *
15  * * 172-96-130.unassigned.userdns.com (172.96.130.53)  5.189 ms !N
16  * * 172-96-130.unassigned.userdns.com (172.96.130.53)  4.092 ms !N
17  * * *
18  * * *
19  * * *
20  172-96-130.unassigned.userdns.com (172.96.130.53)  4.360 ms !N * *
21  * * *
22  * * *
23  172-96-130.unassigned.userdns.com (172.96.130.53)  3.665 ms !N * *
24  * * *
25  * * 172-96-130.unassigned.userdns.com (172.96.130.53)  4.640 ms !N
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Done.

v.
vi. For the Amazon IP, it starts at the princeton network, and then follows its way to a provider called userdns.com. It actually gets stuck there for a couple of hops. I would assume that once it reaches the Amazon network, it drops the packet due to firewall rules

f) Microsoft, Google, CU Boulder, Amazon. We can determine this by doing a DNS lookup or determine where it is going based on the domain names in the traceroute.

g) I feel like it provides a worse view than the traceroute servers. From the snippets above, we can see that it shows a set of routers that are responding to the source and often

display a domain name and IP. It can be more efficient to trace where in the internet the Ips are and how they got there.

h) Some problems with this approach could be the possibility of AS hijackers. They could be advertising a network with a larger subnet which will catch this bgp route table request and spit out misinformation. Also AS path prepending can also show misinformation.

i) I could get variable results depending on the network I was looking from. If we think of a network with border BGP routers with an internal OSPF network. The ASBR BGP routers can be geographically in a different place and depending on where we initiate the ip route, it will show the best route to the destination with the least hop count.

2)

a) To set up the routing with UUNet, and it being a Tier 1 connection, I would set up a BGP route to UUNet with an increased hop count. Because a T1 connection is older and not as fast, this would be a secondary means of reaching the internet if there was not a faster connection.

b) With a T3 connection, I would set a BGP route with less hops to UUNet and assign a majority of my users to the T3 connection with a high priority.

c) Now with a second T3 connection, I would set half of my customer IP space to one ISP and the other half to the other. While there would still be failover to that of the T1 and T3 network, the T1 network would not be needed anymore. My UUNet T1 and T3 connection both originate from the ISP and if there was an outage to my T3 connection, then there would probably also be an outage for the T1 connection.

d) I would simply reduce move some of the customer /20 IP spaces to the uncongested AboveNet T-3. I would set hop count for the /20 customer IP spaces to be less for the AboveNet T-3 and only when the link fails, the traffic will transfer over to UUNet. I would set the inbound routes to the /20 IP space to be more favorable to the AboveNet.

e) I would reconfigure the inbound routes of AboveNet to now prefer UUNet. With this set this would allow for the users to prefer UUNet.

3) A router operated by AS7007 leaked its own routing table and created a networking black hole that dropped all inbound and outbound traffic. The leaked routes deaggregated to /24s and ultimately made most of the internet routers to prefer the leaked routes.

4)

a) 5.0.0.1 because the localpref is the highest

b) 5.0.0.9 because 5.0.0.1 is invalid, localpref for the other 2 is the same, and the AS_PATH is lower for 5.0.0.9 and will be chosen.

c) 5.0.0.5 because the localpref is the same, it prefers the smallest AS path

d) 5.0.0.1 because localpref is the same, AS_PATH is the same, ORIGIN is the same, MED is the same, closest IGP neighbor is the same, so the route with the lowest ROUTER_ID is 5.0.0.1

e) 5.0.0.1 because localpref is the same, AS_PATH is the same, so the route with the most preferred origin is 5.0.0.1: IGP -> EGP -> UNK.