



BGP Exploration and Attack



April 2024
Abdulfatah Abdillahi

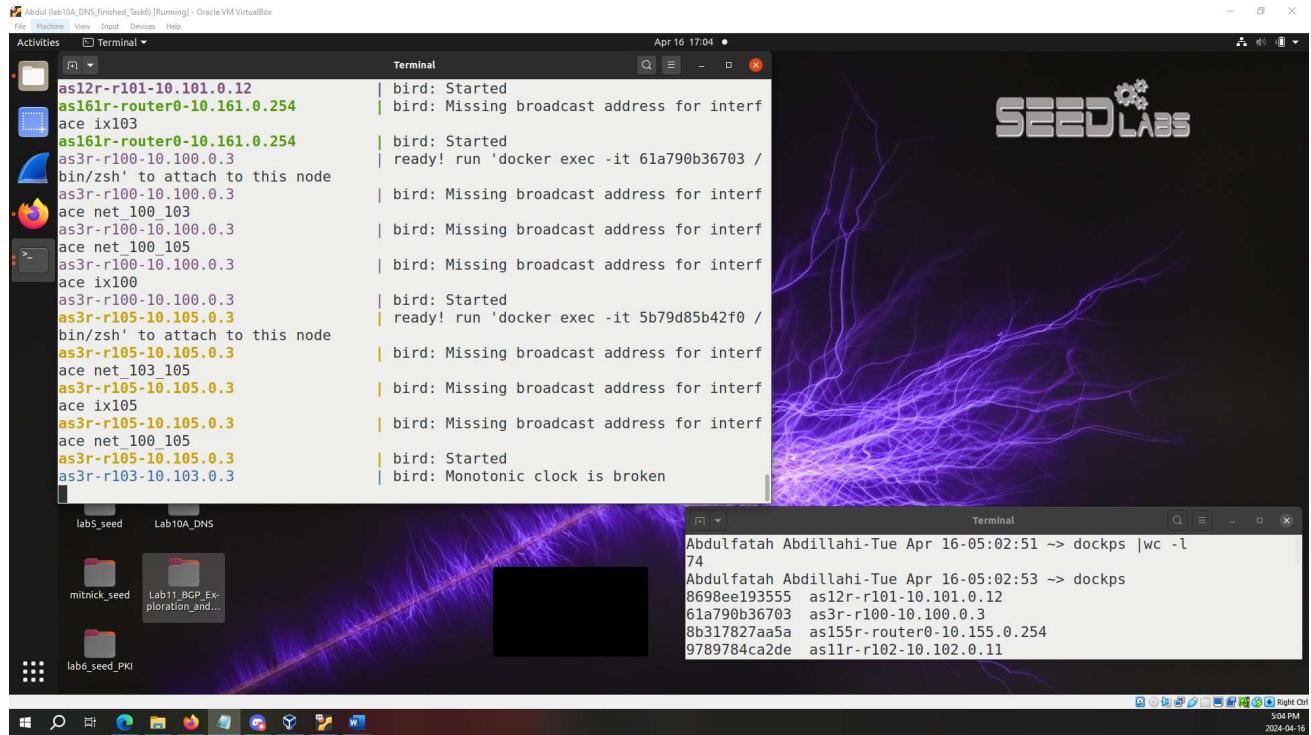
Contents

SEED: BGP Lab	3
Task 1: Stub Autonomous System (AS)	3
Task 1.a: Understanding AS-155's BGP Configuration.....	3
Task 1.b: Observing BGP UPDATE Messages.....	6
Task 1.c: Experimenting with Large Communities	9
Task 1.d: Configuring AS-180.....	12
Task 2: Transit Autonomous System.....	16
Task 2.a: Experimenting with IBGP	17
Task 2.b: Experimenting with IGP	20
Task 3: Path Selection.....	22
Task 3.a.....	22
Task 3.b.....	22
Task 4: IP Anycast	23
Task 5: BGP Prefix Attacks.....	25
Task 5.a: Launching the Prefix Hijacking Attack from AS-161	25
Task 5.b: Fighting back from AS-154	26
References	29

SEED: BGP Lab

Setting up the Environment

Creating docker containers



Task 1: Stub Autonomous System (AS)

In this task, we focus on stub ASes, see how it peers with others.

Task 1.a: Understanding AS-155's BGP Configuration

Task 1.a.1

Here, you can see me searching for AS-155 and accessing that container to view the *bird.conf* file.

```

Terminal      Terminal      Terminal      Terminal
Abdulfatah Abdillahi-Tue Apr 16-05:05:13 ~/.../output> dockps | grep as155
8b317827aa5a as155r-router0-10.155.0.254
116945c76d40 as155h-host_0-10.155.0.71
64a3c85df233 as155h-webservice_1-10.155.0.72
Abdulfatah Abdillahi-Tue Apr 16-05:20:03 ~/.../output> docksh as155r-router0-10.155.0.254
root@8b317827aa5a / # vim /etc/bird/bird.conf
root@8b317827aa5a / #

```

Below you can see what the *bird.conf* file looks like. After examining the file, it was concluded that AS-155 peers with AS-2 a neighbors IP “10.102.0.2”, AS-4 a neighbors IP “10.102.0.4”, and AS-156 a neighbors IP “10.102.0.156” and each of which have a local IP of “10.102.0.155”.

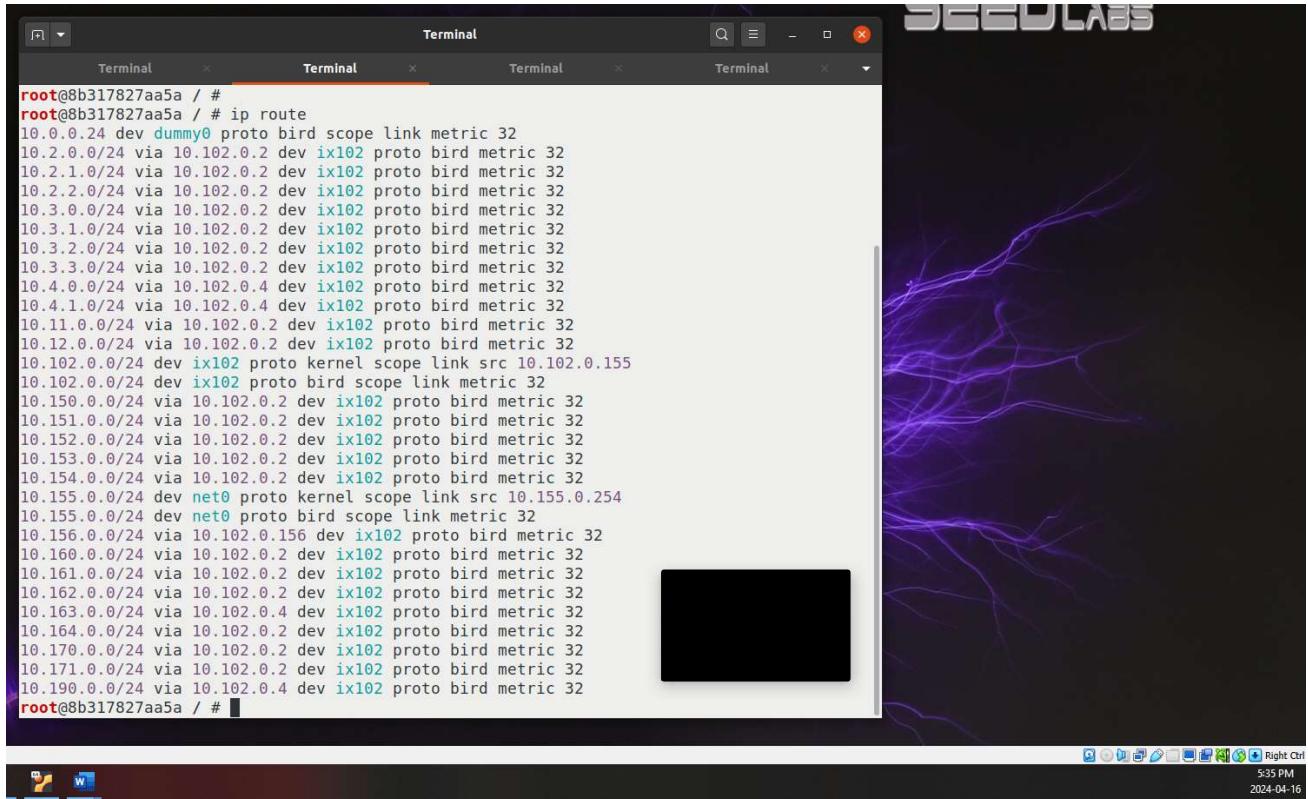
```

protocol pipe {
    table t_direct;
    peer table t_bgp;
    import none;
    export filter { bgp_large_community.add(LOCAL_COMM); bgp_local_pref = 40; accept; };
}
protocol bgp u_as2 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 10;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM];
        next hop self;
    };
    local 10.102.0.155 as 155;
    neighbor 10.102.0.2 as 2;
}
protocol bgp u_as4 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 10;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM];
        next hop self;
    };
    local 10.102.0.155 as 155;
    neighbor 10.102.0.4 as 4;
}
protocol bgp p_as156 {
    ipv4 {
        table t_bgp;
        import filter {

```

Task 1.a.2

Here, you can see the routes set on AS-155 before attempting to disable peering with AS-2 (10.102.0.2).

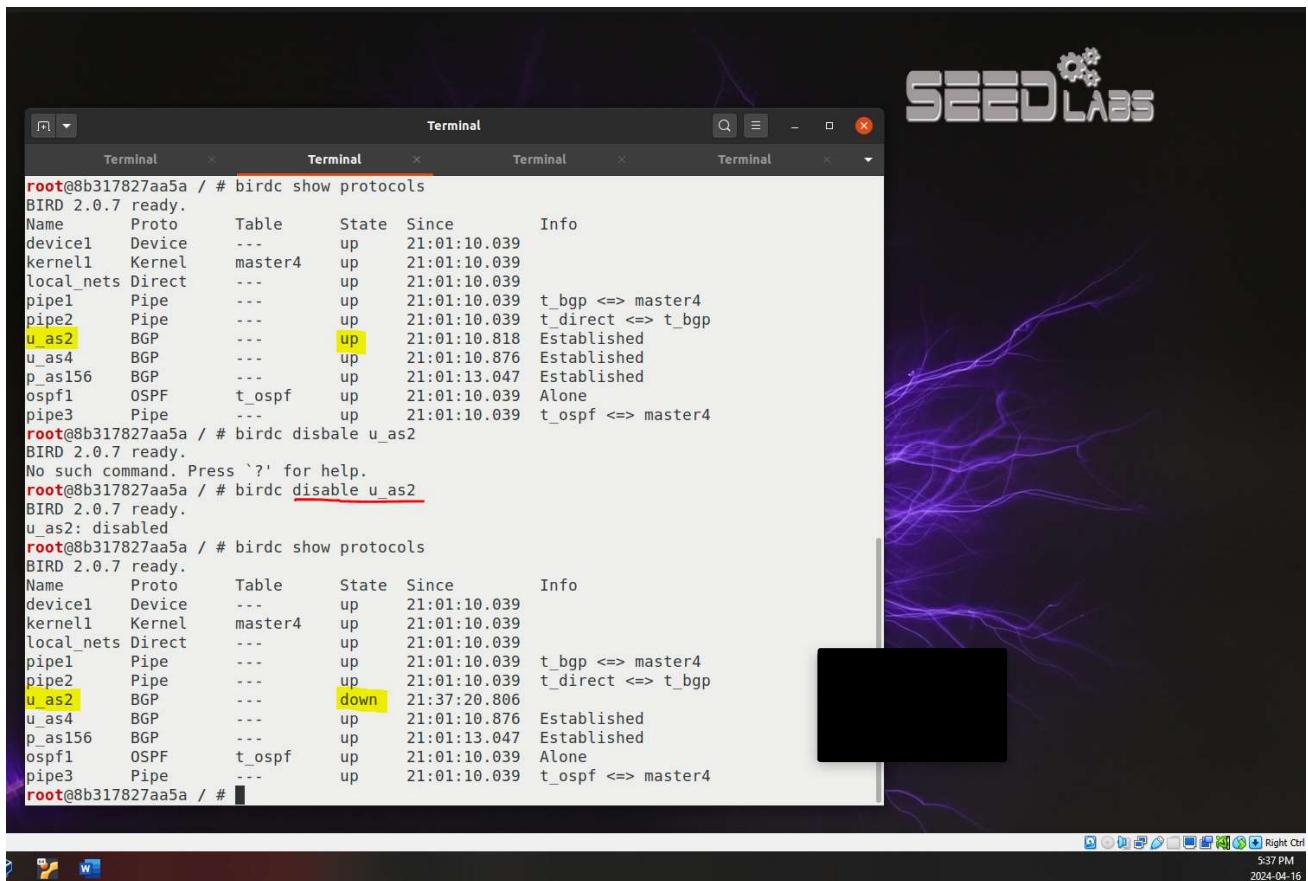


```

root@8b317827aa5a / #
root@8b317827aa5a / # ip route
10.0.0.24 dev dummy0 proto bird scope link metric 32
10.2.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.2.1.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.2.2.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.3.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.3.1.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.3.2.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.3.3.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.4.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.4.1.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.11.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.12.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.102.0.0/24 dev ix102 proto kernel scope link src 10.102.0.155
10.102.0.0/24 dev ix102 proto bird scope link metric 32
10.150.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.151.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.152.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.153.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.154.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.155.0.0/24 dev net0 proto kernel scope link src 10.155.0.254
10.155.0.0/24 dev net0 proto bird scope link metric 32
10.156.0.0/24 via 10.102.0.156 dev ix102 proto bird metric 32
10.160.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.161.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.162.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.163.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.164.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.170.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.171.0.0/24 via 10.102.0.2 dev ix102 proto bird metric 32
10.190.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
root@8b317827aa5a / #

```

The below screenshot shows a comparison of the status of the BGP session before and after disabling AS-2 as a peer for AS-155 router.



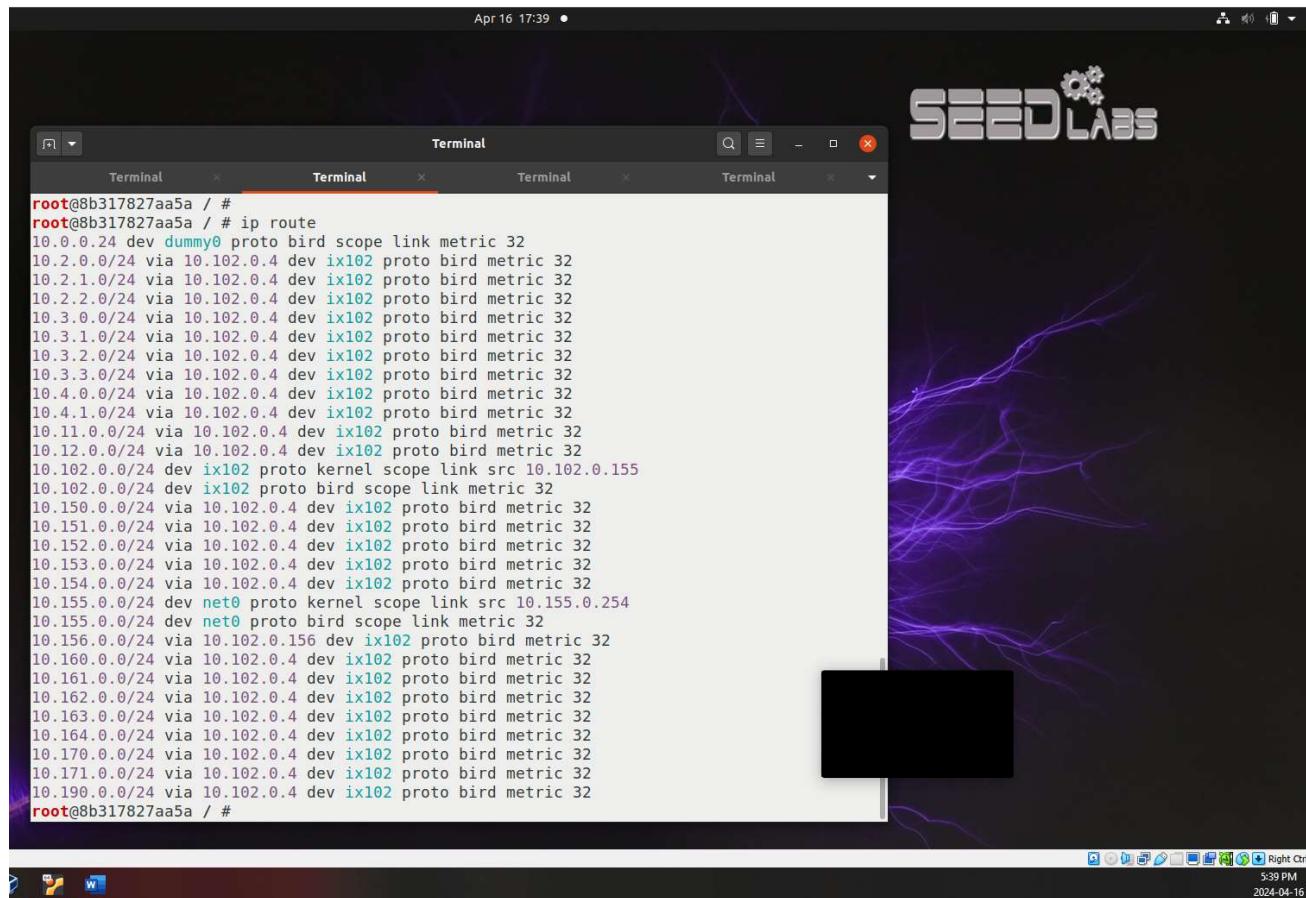
```

root@8b317827aa5a / # birdc show protocols
BIRD 2.0.7 ready.
Name Proto Table State Since Info
device1 Device --- up 21:01:10.039
kernel1 Kernel master4 up 21:01:10.039
local_nets Direct --- up 21:01:10.039
pipe1 Pipe --- up 21:01:10.039 t_bgp <=> master4
pipe2 Pipe --- up 21:01:10.039 t_direct <=> t_bgp
u_as2 BGP --- up 21:01:10.818 Established
u_as4 BGP --- up 21:01:10.876 Established
p_as156 BGP --- up 21:01:13.047 Established
ospf1 OSPF t_ospf up 21:01:10.039 Alone
pipe3 Pipe --- up 21:01:10.039 t_ospf <=> master4
root@8b317827aa5a / # birdc disable u_as2
BIRD 2.0.7 ready.
No such command. Press '?' for help.
root@8b317827aa5a / # birdc disable u_as2
BIRD 2.0.7 ready.
u_as2: disabled
root@8b317827aa5a / # birdc show protocols
BIRD 2.0.7 ready.
Name Proto Table State Since Info
device1 Device --- up 21:01:10.039
kernel1 Kernel master4 up 21:01:10.039
local_nets Direct --- up 21:01:10.039
pipe1 Pipe --- up 21:01:10.039 t_bgp <=> master4
pipe2 Pipe --- up 21:01:10.039 t_direct <=> t_bgp
u_as2 BGP --- down 21:37:20.806
u_as4 BGP --- up 21:01:10.876 Established
p_as156 BGP --- up 21:01:13.047 Established
ospf1 OSPF t_ospf up 21:01:10.039 Alone
pipe3 Pipe --- up 21:01:10.039 t_ospf <=> master4
root@8b317827aa5a / #

```

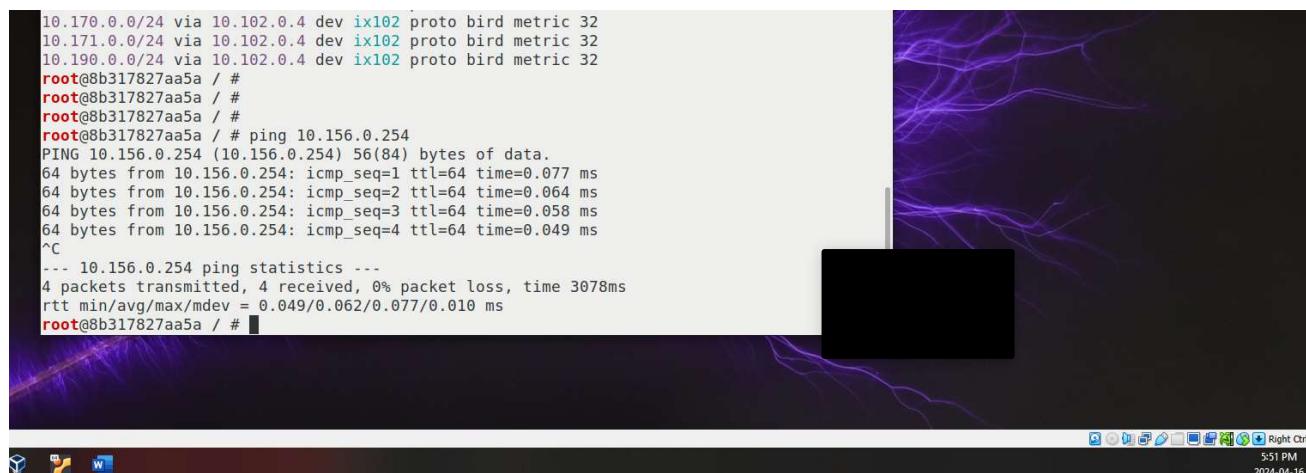
Here, you can see that as a result of disabling AS-2 as a peer, the gateway is now set to AS-4

(10.102.0.4) instead of AS-2.



```
root@8b317827aa5a / #
root@8b317827aa5a / # ip route
10.0.0.24 dev dummy0 proto bird scope link metric 32
10.2.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.2.1.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.2.2.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.3.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.3.1.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.3.2.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.3.3.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.4.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.4.1.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.11.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.12.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.102.0.0/24 dev ix102 proto kernel scope link src 10.102.0.155
10.102.0.0/24 dev ix102 proto bird scope link metric 32
10.150.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.151.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.152.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.153.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.154.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.155.0.0/24 dev net0 proto kernel scope link src 10.155.0.254
10.155.0.0/24 dev net0 proto bird scope link metric 32
10.156.0.0/24 via 10.102.0.156 dev ix102 proto bird metric 32
10.160.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.161.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.162.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.163.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.164.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.170.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.171.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.190.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
root@8b317827aa5a / #
```

To prove that AS-155 indeed does still have access to the internet, we can ping another router. This still works because of the dynamic nature of BGP as AS-155 currently relies on AS-4 instead of AS-2.



```
10.170.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.171.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
10.190.0.0/24 via 10.102.0.4 dev ix102 proto bird metric 32
root@8b317827aa5a / #
root@8b317827aa5a / #
root@8b317827aa5a / #
root@8b317827aa5a / # ping 10.156.0.254
PING 10.156.0.254 (10.156.0.254) 56(84) bytes of data.
64 bytes from 10.156.0.254: icmp_seq=1 ttl=64 time=0.077 ms
64 bytes from 10.156.0.254: icmp_seq=2 ttl=64 time=0.064 ms
64 bytes from 10.156.0.254: icmp_seq=3 ttl=64 time=0.058 ms
64 bytes from 10.156.0.254: icmp_seq=4 ttl=64 time=0.049 ms
^C
--- 10.156.0.254 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3078ms
rtt min/avg/max/mdev = 0.049/0.062/0.077/0.010 ms
root@8b317827aa5a / #
```

Task 1.b: Observing BGP UPDATE Messages

Here, you can see me searching for AS-155 and accessing that container to run the `tcpdump` command provided in the instructions (`tcpdump -i any -w /tmp/bgp.pcap "tcp port 179"`).

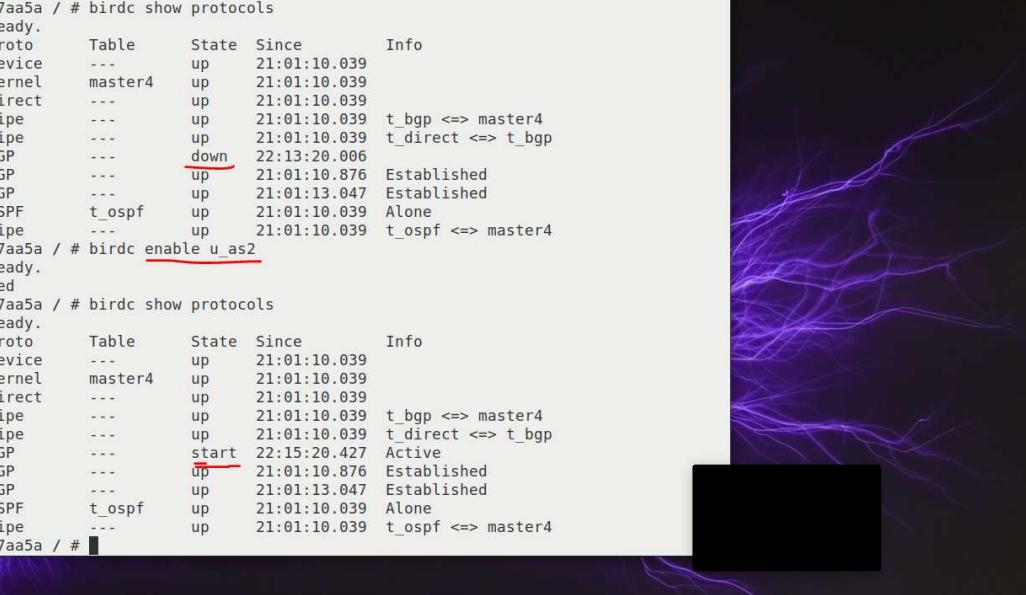
```
Abdulfatah Abdillahi-Tue Apr 16-06:01:20 ~.../output> dockps | grep as150
feb59914bbf1  as150r-router0-10.150.0.254
8884d7d237a1  as150h-webservice_0-10.150.0.71
7a57239a9b2b  as150h-host_1-10.150.0.72
Abdulfatah Abdillahi-Tue Apr 16-06:01:24 ~.../output> docksh as150r-router0-10.150.0.254
root@feb59914bbf1 / # tcpdump -i any -w /tmp/bgp.pcap "tcp port 179"
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
```

Now, I will be attempting to generate route withdrawal and route advertisement messages by going back to AS-155 and disabling and enabling AS-2 as a peer, respectively.

Here, you can see me disabling AS-2

```
127 root@8b317827aa5a / # birdc show protocols
BIRD 2.0.7 ready.
Name Proto Table State Since Info
device1 Device --- up 21:01:10.039
kernel1 Kernel master4 up 21:01:10.039
local_nets Direct --- up 21:01:10.039
pipe1 Pipe --- up 21:01:10.039 t_bgp <=> master4
pipe2 Pipe --- up 21:01:10.039 t_direct <=> t_bgp
u_as2 BGP --- up 22:04:38.765 Established
u_as4 BGP --- up 21:01:10.876 Established
p_as156 BGP --- up 21:01:13.047 Established
ospf1 OSPF t_ospf up 21:01:10.039 Alone
pipe3 Pipe --- up 21:01:10.039 t_ospf <=> master4
root@8b317827aa5a / # birdc disable u_as2
BIRD 2.0.7 ready.
u_as2: disabled
root@8b317827aa5a / # birdc show protocols
BIRD 2.0.7 ready.
Name Proto Table State Since Info
device1 Device --- up 21:01:10.039
kernel1 Kernel master4 up 21:01:10.039
local_nets Direct --- up 21:01:10.039
pipe1 Pipe --- up 21:01:10.039 t_bgp <=> master4
pipe2 Pipe --- up 21:01:10.039 t_direct <=> t_bgp
u_as2 BGP --- down 22:13:20.066
u_as4 BGP --- up 21:01:10.876 Established
p_as156 BGP --- up 21:01:13.047 Established
ospf1 OSPF t_ospf up 21:01:10.039 Alone
pipe3 Pipe --- up 21:01:10.039 t_ospf <=> master4
root@8b317827aa5a / #
```

Here, you can see me enabling AS-2 once again.



```
Terminal Terminal Terminal Terminal Terminal
root@8b317827aa5a / # birdc show protocols
BIRD 2.0.7 ready.
Name Proto Table State Since Info
device1 Device --- up 21:01:10.039
kernel1 Kernel master4 up 21:01:10.039
local_nets Direct --- up 21:01:10.039
pipe1 Pipe --- up 21:01:10.039 t_bgp <=> master4
pipe2 Pipe --- up 21:01:10.039 t_direct <=> t_bgp
u_as2 BGP --- down 22:13:20.006
u_as4 BGP --- up 21:01:10.876 Established
p_as156 BGP --- up 21:01:13.047 Established
ospf1 OSPF t_ospf up 21:01:10.039 Alone
pipe3 Pipe --- up 21:01:10.039 t_ospf <=> master4
root@8b317827aa5a / # birdc enable u_as2
BIRD 2.0.7 ready.
u_as2: enabled
root@8b317827aa5a / # birdc show protocols
BIRD 2.0.7 ready.
Name Proto Table State Since Info
device1 Device --- up 21:01:10.039
kernel1 Kernel master4 up 21:01:10.039
local_nets Direct --- up 21:01:10.039
pipe1 Pipe --- up 21:01:10.039 t_bgp <=> master4
pipe2 Pipe --- up 21:01:10.039 t_direct <=> t_bgp
u_as2 BGP --- start 22:15:20.427 Active
u_as4 BGP --- up 21:01:10.876 Established
p_as156 BGP --- up 21:01:13.047 Established
ospf1 OSPF t_ospf up 21:01:10.039 Alone
pipe3 Pipe --- up 21:01:10.039 t_ospf <=> master4
root@8b317827aa5a / #
```

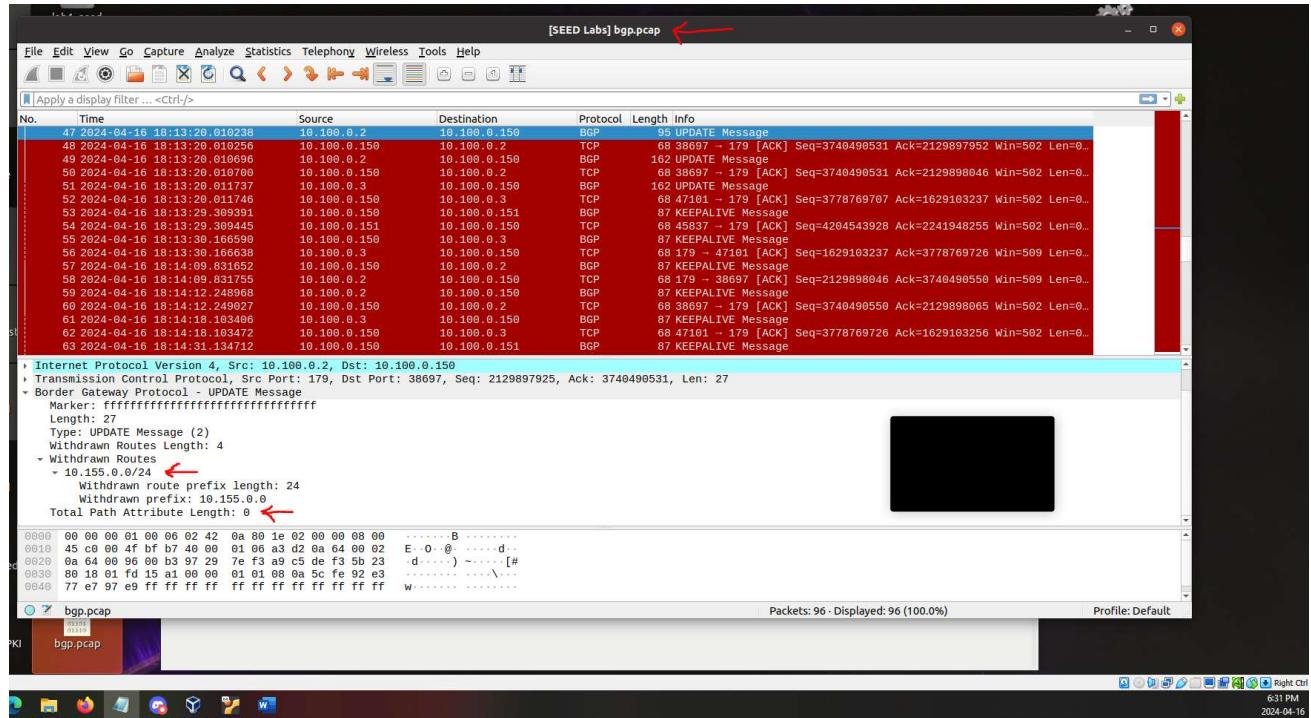
Then I stop the capture on AS-150 and copy the resulting pcap to the host computer using the "docker cp" command (from the host), to load it into Wireshark.

The screenshot shows a Linux desktop environment with a dark theme. On the left is a file manager sidebar displaying various files and folders. In the center, there are two terminal windows. The top terminal window shows command-line output related to BGP configuration and packet capture. The bottom terminal window shows the user copying a file named 'bgp.pcap' from a Docker container to the local desktop. The desktop bar at the bottom includes icons for file, search, and system status.

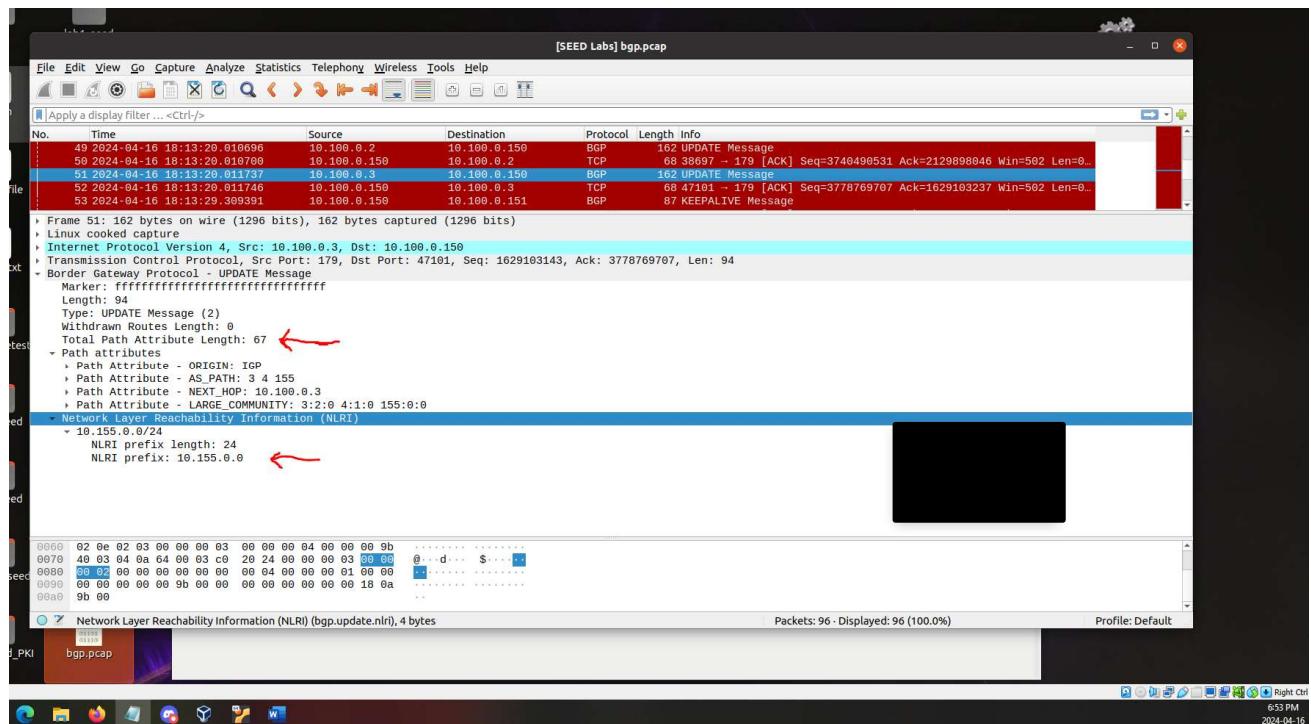
```
Abdulfatah Abdillahi-Tue Apr 16-06:01:20 ~/.../output> dockps | grep as150
feb59914bbf1 as150r-router0-10.150.0.254
8884d7d237al as150h-webservice_0-10.150.0.71
7a57239a9b2b as150h-host_1-10.150.0.72
Abdulfatah Abdillahi-Tue Apr 16-06:01:24 ~/.../output> docksh as150r-router0-10.150.0.254
root@feb59914bbf1 / # tcpdump -i any -w /tmp/bgp.pcap "tcp port 179"
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 byte
s
^C96 packets captured
96 packets received by filter
0 packets dropped by kernel
root@feb59914bbf1 / #
```

```
Abdulfatah Abdillahi-Tue Apr 16-06:23:20 ~> docker cp feb59914bbf1:/tmp/bgp.pcap
/home/seed/Desktop
Abdulfatah Abdillahi-Tue Apr 16-06:24:57 ~>
```

Here, you can see that I was successfully able to open the resulting pcap file. The first packet at the top displaying the UPDATE Message. If we expand below, we can also see that a route Withdrawal message was created specifically pointing to 10.155.0.0/24 as the withdrawn route. We can also see that no advertisements we made as the total path attribute length is set to 0.



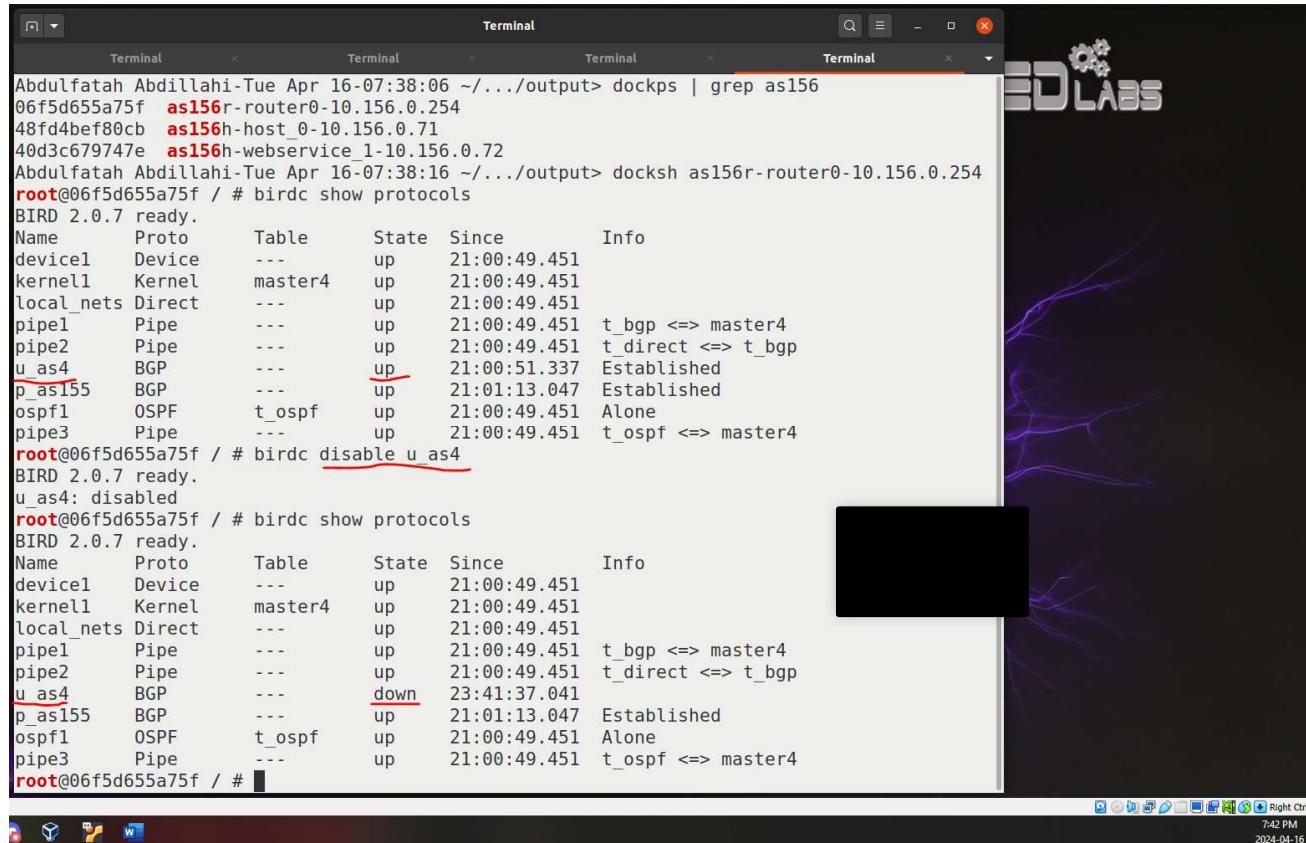
Here, we can see the next packet which is also another type of UPDATE Message. This time however, if we expand below, we will see that it displays route advertisement message specifically for the same network prefix in the previous screenshot (10.155.0.0/24). We can also see that unlike in the previous screenshot, this message is a route advertisement update as the total path attribute length is set to 67.



Task 1.c: Experimenting with Large Communities

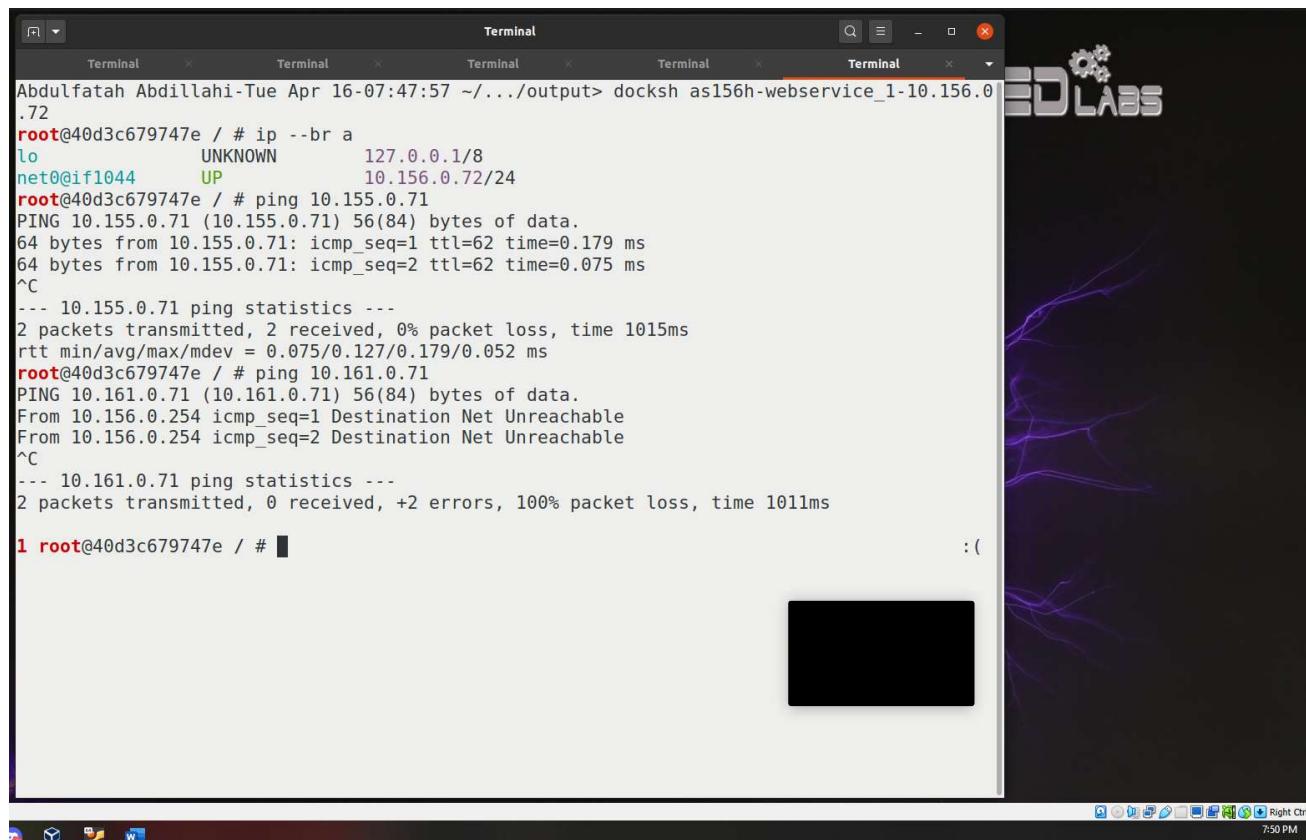
Here, you can see me searching for AS-156 and accessing that container to disable the peering between AS-4 and AS-156. Since AS-4 is the only service provider for AS-156, this essentially

disconnects AS-156 from the Internet.



```
Terminal Terminal Terminal Terminal
Abdulfatah Abdillahi-Tue Apr 16-07:38:06 ~/.../output> dockps | grep as156
06f5d655a75f as156r-router0-10.156.0.254
48fd4bef80cb as156h-host 0-10.156.0.71
40d3c679747e as156h-webservice_1-10.156.0.72
Abdulfatah Abdillahi-Tue Apr 16-07:38:16 ~/.../output> docksh as156r-router0-10.156.0.254
root@06f5d655a75f / # birdc show protocols
BIRD 2.0.7 ready.
Name Proto Table State Since Info
device1 Device --- up 21:00:49.451
kernel1 Kernel master4 up 21:00:49.451
local_nets Direct --- up 21:00:49.451
pipe1 Pipe --- up 21:00:49.451 t_bgp <=> master4
pipe2 Pipe --- up 21:00:49.451 t_direct <=> t_bgp
u_as4 BGP --- up 21:00:51.337 Established
p_as155 BGP --- up 21:01:13.047 Established
ospf1 OSPF t_ospf up 21:00:49.451 Alone
pipe3 Pipe --- up 21:00:49.451 t_ospf <=> master4
root@06f5d655a75f / # birdc disable u_as4
BIRD 2.0.7 ready.
u_as4: disabled
root@06f5d655a75f / # birdc show protocols
BIRD 2.0.7 ready.
Name Proto Table State Since Info
device1 Device --- up 21:00:49.451
kernel1 Kernel master4 up 21:00:49.451
local_nets Direct --- up 21:00:49.451
pipe1 Pipe --- up 21:00:49.451 t_bgp <=> master4
pipe2 Pipe --- up 21:00:49.451 t_direct <=> t_bgp
u_as4 BGP --- down 23:41:37.041
p_as155 BGP --- up 21:01:13.047 Established
ospf1 OSPF t_ospf up 21:00:49.451 Alone
pipe3 Pipe --- up 21:00:49.451 t_ospf <=> master4
root@06f5d655a75f / #
```

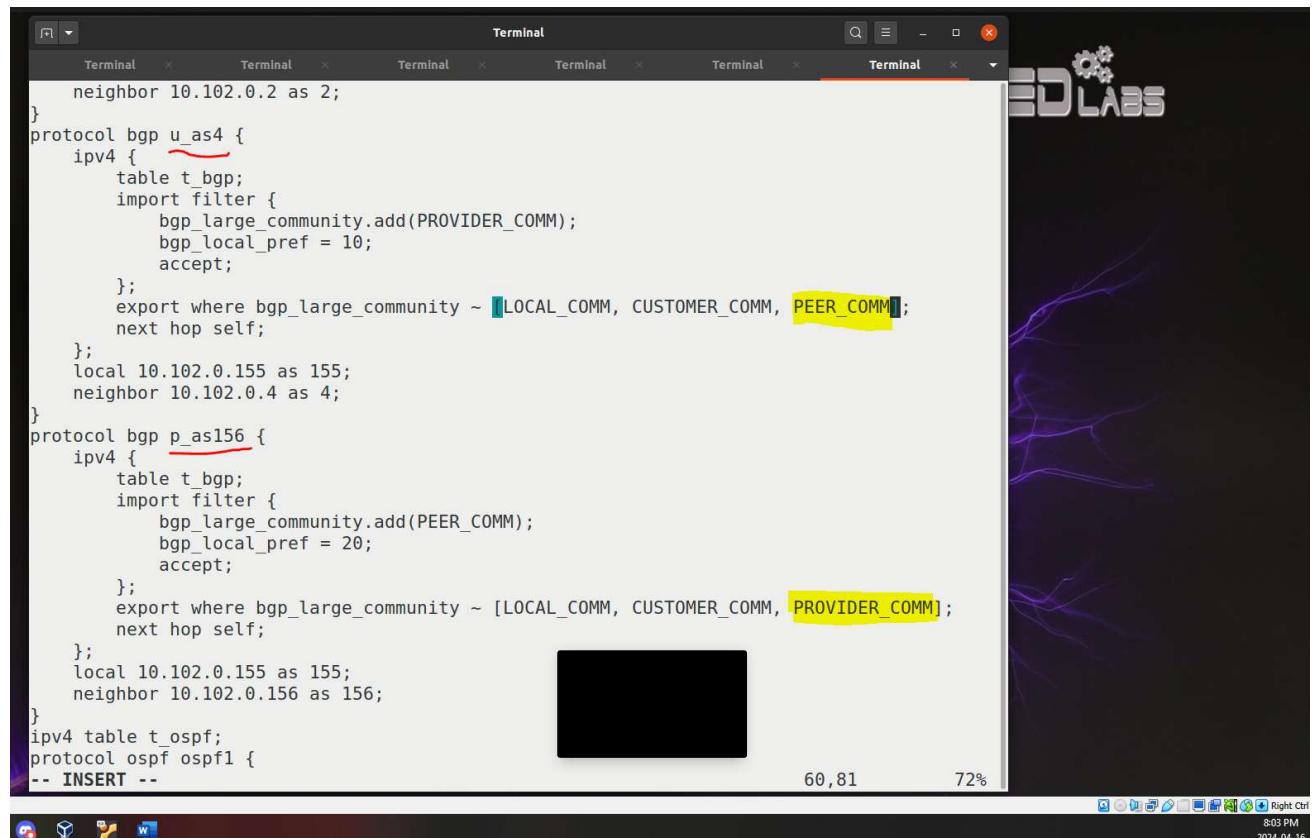
If we ping another host from one of the hosts in AS-156, we can see the following results:



```
Terminal Terminal Terminal Terminal Terminal
Abdulfatah Abdillahi-Tue Apr 16-07:47:57 ~/.../output> docksh as156h-webservice_1-10.156.0.72
root@40d3c679747e / # ip --br a
lo UNKNOWN 127.0.0.1/8
net0@if1044 UP 10.156.0.72/24
root@40d3c679747e / # ping 10.155.0.71
PING 10.155.0.71 (10.155.0.71) 56(84) bytes of data.
64 bytes from 10.155.0.71: icmp_seq=1 ttl=62 time=0.179 ms
64 bytes from 10.155.0.71: icmp_seq=2 ttl=62 time=0.075 ms
^C
--- 10.155.0.71 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1015ms
rtt min/avg/max/mdev = 0.075/0.127/0.179/0.052 ms
root@40d3c679747e / # ping 10.161.0.71
PING 10.161.0.71 (10.161.0.71) 56(84) bytes of data.
From 10.156.0.254 icmp_seq=1 Destination Net Unreachable
From 10.156.0.254 icmp_seq=2 Destination Net Unreachable
^C
--- 10.161.0.71 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1011ms
1 root@40d3c679747e / #
```

We can see that 10.155.0.71 is still reachable, because it belongs to AS-155, which is still peered with AS-156. However, 10.161.0.71 (belonging to AS-161) cannot be reached, because nobody will route the packet for AS-156.

To fix this AS-156 needs to pay AS-155 so it can temporarily allow traffic to go through AS-155 to reach the Internet. We can do this by editing the **bird.conf** file on AS-155 and adding “**PEER_COMM**” under the **u_as4** section and adding the “**PROVIDER_COMM**” under the **p_as156** section.



```
neighbor 10.102.0.2 as 2;
}
protocol bgp u_as4 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 10;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM, PEER_COMM];
        next hop self;
    };
    local 10.102.0.155 as 155;
    neighbor 10.102.0.4 as 4;
}
protocol bgp p_as156 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PEER_COMM);
            bgp_local_pref = 20;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM, PROVIDER_COMM];
        next hop self;
    };
    local 10.102.0.155 as 155;
    neighbor 10.102.0.156 as 156;
}
ipv4 table t_ospf;
protocol ospf ospf1 {
-- INSERT --
```

After saving the changes I attempted the ping one more time and as you can see this worked.

```

Terminal Terminal Terminal Terminal Terminal Terminal
From 10.156.0.254 icmp_seq=1 Destination Net Unreachable
From 10.156.0.254 icmp_seq=2 Destination Net Unreachable
^C
--- 10.161.0.71 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1011ms

1 root@40d3c679747e / # ping 10.161.0.71
PING 10.161.0.71 (10.161.0.71) 56(84) bytes of data.
From 10.156.0.254 icmp_seq=1 Destination Net Unreachable
From 10.156.0.254 icmp_seq=2 Destination Net Unreachable
^C
--- 10.161.0.71 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1010ms

1 root@40d3c679747e / # ping 10.161.0.71
PING 10.161.0.71 (10.161.0.71) 56(84) bytes of data.
From 10.156.0.254 icmp_seq=1 Destination Net Unreachable
From 10.156.0.254 icmp_seq=2 Destination Net Unreachable
From 10.156.0.254 icmp_seq=3 Destination Net Unreachable
From 10.156.0.254 icmp_seq=4 Destination Net Unreachable
From 10.156.0.254 icmp_seq=43 Destination Net Unreachable
From 10.156.0.254 icmp_seq=71 Destination Net Unreachable
64 bytes from 10.161.0.71: icmp_seq=103 ttl=56 time=0.358 ms
64 bytes from 10.161.0.71: icmp_seq=104 ttl=56 time=0.232 ms
64 bytes from 10.161.0.71: icmp_seq=105 ttl=56 time=0.189 ms
64 bytes from 10.161.0.71: icmp_seq=106 ttl=56 time=0.224 ms
64 bytes from 10.161.0.71: icmp_seq=107 ttl=56 time=0.227 ms
64 bytes from 10.161.0.71: icmp_seq=108 ttl=56 time=0.202 ms
^C
--- 10.161.0.71 ping statistics ---
108 packets transmitted, 6 received, +6 errors, 94.4444% packet loss, time 110021ms
rtt min/avg/max/mdev = 0.189/0.238/0.358/0.055 ms
root@40d3c679747e / #

```

Ubuntu desktop environment showing multiple terminal windows and a 'LABS' logo.

Task 1.d: Configuring AS-180

Here, you can see me importing the *bird.conf* files from the containers to the host.

```

Terminal
Abdulfatah Abdillahi-Tue Apr 16-09:01:03 ~> cd Desktop/Lab11_BGP_Exploration_and_
Attack/Lab11_BGP/Labsetup/task1/
Abdulfatah Abdillahi-Tue Apr 16-09:01:31 ~/.../task1> ls
export_bird_conf.sh import_bird_conf.sh
Abdulfatah Abdillahi-Tue Apr 16-09:01:44 ~/.../task1> import_bird_conf.sh
Copy bird.conf from the container: as155r
Copy bird.conf from the container: as180r
Copy bird.conf from the container: as171r
Copy bird.conf from the container: as2r-r105
Copy bird.conf from the container: as3r-r105
Abdulfatah Abdillahi-Tue Apr 16-09:01:53 ~/.../task1> ls
as155r_bird.conf as180r_bird.conf as3r-r105_bird.conf import_bird_conf.sh
as171r_bird.conf as2r-r105_bird.conf export_bird_conf.sh
Abdulfatah Abdillahi-Tue Apr 16-09:02:03 ~/.../task1>

```

Ubuntu desktop environment showing a terminal window with command history and a 'W' icon in the dock.

Below you can see the piece of code I have appended to the bottom of the *as180r_bird.conf* file.

```
define LOCAL_COMM = (180, 0, 0);
```

```

define CUSTOMER_COMM = (180, 1, 0);
define PEER_COMM = (180, 2, 0);
define PROVIDER_COMM = (180, 3, 0);

ipv4 table t_bgp;

protocol pipe {
    table t_bgp;
    peer table master4;
    import none;
    export all;
}

protocol pipe {
    table t_direct;
    peer table t_bgp;
    import none;
    export filter {
        bgp_large_community.add(LOCAL_COMM);
        bgp_local_pref = 40;
        accept;
    };
}
}

protocol bgp b_as2 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 20;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM];
        next hop self;
    };
    local 10.180.0.171 as 180;
    neighbor 10.105.0.2 as 105;
}
}

protocol bgp b_as3 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 20;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM];
        next hop self;
    };
    local 10.180.0.171 as 180;
}

```

```

        neighbor 10.105.0.3 as 105;
    }

protocol bgp b_as171 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 20;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM];
        next hop self;
    };
    local 10.105.0.180 as 180;
    neighbor 10.105.0.171 as 171;
}

```

Below you can see the piece of code I have appended to the bottom of the *as171r_bird.conf* file.

```

protocol bgp b_as180 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 20;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM];
        next hop self;
    };
    local 10.105.0.171 as 171;
    neighbor 10.105.0.180 as 180;
}

```

Here, you can see me exporting the *bird.conf* files from the host to the containers.

```

Abdulfatah Abdillahi-Tue Apr 16-09:02:03 ~/.../task1> sudo vim as180r_bird.conf
Abdulfatah Abdillahi-Tue Apr 16-09:16:56 ~/.../task1> sudo vim as171r_bird.conf
Abdulfatah Abdillahi-Tue Apr 16-09:18:25 ~/.../task1> sudo vim as180r_bird.conf
Abdulfatah Abdillahi-Tue Apr 16-09:28:09 ~/.../task1> sudo vim as171r_bird.conf
Abdulfatah Abdillahi-Tue Apr 16-09:28:32 ~/.../task1> sudo vim as171r_bird.conf
Abdulfatah Abdillahi-Tue Apr 16-09:32:41 ~/.../task1> ./export_bird_conf.sh
== Copy bird.conf to the container: as155r
== Execute 'birdc configure' on the container
BIRD 2.0.7 ready.
Reading configuration from /etc/bird/bird.conf
Reconfigured

== Copy bird.conf to the container: as180r
== Execute 'birdc configure' on the container
BIRD 2.0.7 ready.
Reading configuration from /etc/bird/bird.conf
Reconfigured

== Copy bird.conf to the container: as171r
== Execute 'birdc configure' on the container
BIRD 2.0.7 ready.
Reading configuration from /etc/bird/bird.conf
Reconfigured

== Copy bird.conf to the container: as2r-r105
== Execute 'birdc configure' on the container
BIRD 2.0.7 ready.
Reading configuration from /etc/bird/bird.conf
Reconfigured

== Copy bird.conf to the container: as3r-r105
== Execute 'birdc configure' on the container
BIRD 2.0.7 ready.
Reading configuration from /etc/bird/bird.conf
Reconfigured

```

Here, you can see that the new peers were successfully created and added. By first looking at AS-180 we can see the new AS being reflected in the output.

```

Abdulfatah Abdillahi-Tue Apr 16-09:34:16 ~/.../output> dockps |grep as180
65d77b2cf440 as180r-router0-10.180.0.254
52fd587fcab1 as180h-webservice_0-10.180.0.71
fd3498c424ab as180h-host_1-10.180.0.72
Abdulfatah Abdillahi-Tue Apr 16-09:34:31 ~/.../output> docksh as180r-router0-10.180.0.254
root@65d77b2cf440 / # bird configure
Usage: bird [-version] [--help] [-c <config-file>] [OPTIONS]
1 root@65d77b2cf440 / # birdc configure
BIRD 2.0.7 ready.
Reading configuration from /etc/bird/bird.conf
Reconfigured
root@65d77b2cf440 / # birdc show protocols
BIRD 2.0.7 ready.
Name Proto Table State Since Info
device1 Device --- up 21:00:57.189
kernel1 Kernel master4 up 21:00:57.189
local_nets Direct --- up 21:00:57.189
ospf1 OSPF t_ospf up 21:00:57.189 Alone
pipe1 Pipe --- up 21:00:57.189 t_ospf <=> master4
pipe2 Pipe --- up 01:33:19.680 t_bgp <=> master4
pipe3 Pipe --- up 01:33:19.680 t_direct <=> t_bgp
b_as2 BGP --- start 01:33:19.680 Active Socket: Connection closed
b_as3 BGP --- start 01:33:19.680 Active Socket: Connection closed
b_as171 BGP --- up 01:33:24.316 Established
root@65d77b2cf440 / #

```

The same can also be said about AS-171.

Then to prove that this works I made sure to ping a host from the AS-171 network from AS-180 and vice versa. As you can see this was successful.

```
root@65d77b2cf440 / # ping 10.171.0.71
PING 10.171.0.71 (10.171.0.71) 56(84) bytes of data.
64 bytes from 10.171.0.71: icmp_seq=1 ttl=63 time=0.163 ms
64 bytes from 10.171.0.71: icmp_seq=2 ttl=63 time=0.047 ms
^C
--- 10.171.0.71 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.047/0.105/0.163/0.058 ms
root@65d77b2cf440 / #
```

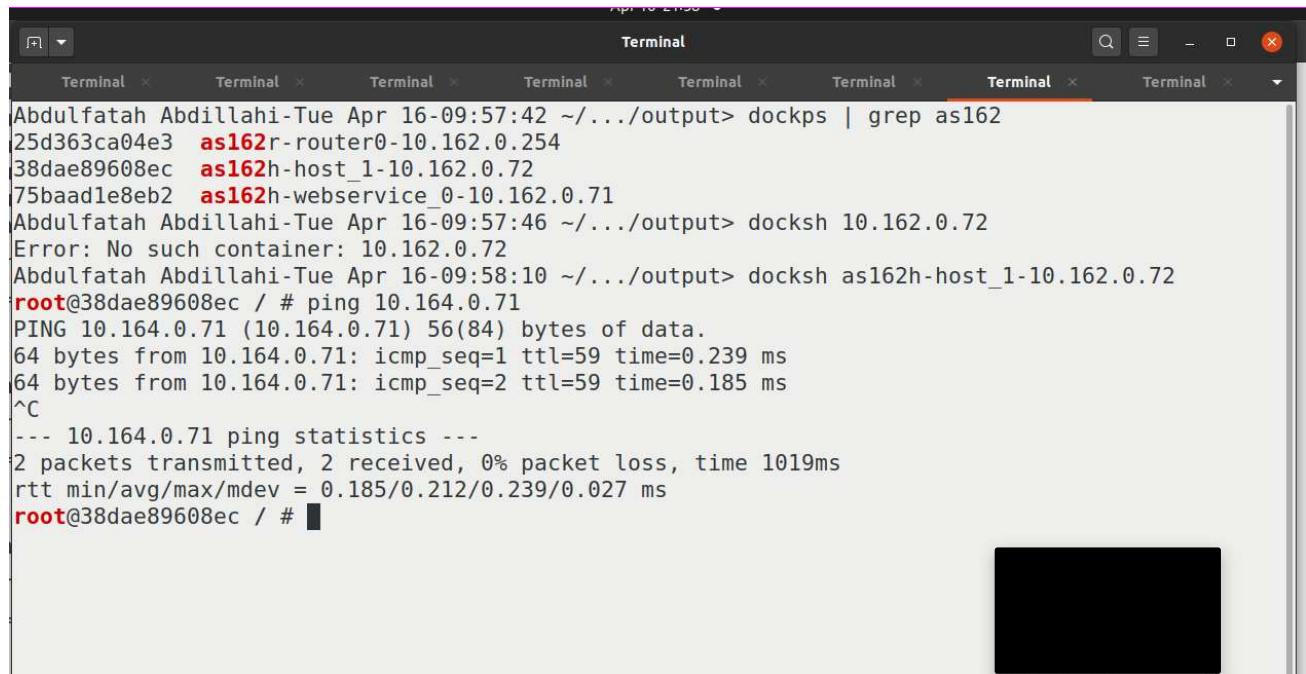
```
root@0d6f70c9cd7f / # ping 10.180.0.72
PING 10.180.0.72 (10.180.0.72) 56(84) bytes of data.
64 bytes from 10.180.0.72: icmp_seq=1 ttl=63 time=0.147 ms
64 bytes from 10.180.0.72: icmp_seq=2 ttl=63 time=0.072 ms
^C
--- 10.180.0.72 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
rtt min/avg/max/mdev = 0.072/0.109/0.147/0.037 ms
root@0d6f70c9cd7f / #
```

Then instructions also request a screenshot of traceroute, but this was not possible as traceroute was not installed on the containers before hand.

Task 2: Transit Autonomous System

Task 2.a: Experimenting with IBGP

Here, you can see me searching for AS-162 and accessing that container to ping 10.164.0.71 from a host within AS-162.



A screenshot of a terminal window titled "Terminal". The window has multiple tabs at the top, all labeled "Terminal". The main pane displays the following command-line session:

```
Abdulfatah Abdillahi-Tue Apr 16-09:57:42 ~/.../output> dockps | grep as162
25d363ca04e3  as162r-router0-10.162.0.254
38dae89608ec  as162h-host_1-10.162.0.72
75baad1e8eb2  as162h-webservice_0-10.162.0.71
Abdulfatah Abdillahi-Tue Apr 16-09:57:46 ~/.../output> docksh 10.162.0.72
Error: No such container: 10.162.0.72
Abdulfatah Abdillahi-Tue Apr 16-09:58:10 ~/.../output> docksh as162h-host_1-10.162.0.72
root@38dae89608ec / # ping 10.164.0.71
PING 10.164.0.71 (10.164.0.71) 56(84) bytes of data.
64 bytes from 10.164.0.71: icmp_seq=1 ttl=59 time=0.239 ms
64 bytes from 10.164.0.71: icmp_seq=2 ttl=59 time=0.185 ms
^C
--- 10.164.0.71 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1019ms
rtt min/avg/max/mdev = 0.185/0.212/0.239/0.027 ms
root@38dae89608ec / # █
```

Then moving on to AS-3, this screenshot shows the routes before disabling IBGP.

Terminal

```
Abdulfatah Abdillahi-Tue Apr 16-09:57:08 ~/.../output> dockps | grep as3
61a790b36703 as3r-r100-10.100.0.3
5b79d85b42f0 as3r-r105-10.105.0.3
4d6dc753a492 as3r-r103-10.103.0.3
077b3e3e506b as3r-r104-10.104.0.3
Abdulfatah Abdillahi-Tue Apr 16-09:59:19 ~/.../output> docksh as3r-r103-10.103.0.3
root@4d6dc753a492 / # ip route
10.0.0.5 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.0.0.6 dev dummy0 proto bird scope link metric 32
10.0.0.7 via 10.3.3.253 dev net_103_104 proto bird metric 32
10.0.0.8 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.2.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.2.1.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.2.2.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.3.0.0/24 dev net_100_103 proto kernel scope link src 10.3.0.253
10.3.0.0/24 dev net_100_103 proto bird scope link metric 32
10.3.1.0/24 proto bird metric 32
    nexthop via 10.3.0.254 dev net_100_103 weight 1
    nexthop via 10.3.2.253 dev net_103_105 weight 1
10.3.2.0/24 dev net_103_105 proto kernel scope link src 10.3.2.254
10.3.2.0/24 dev net_103_105 proto bird scope link metric 32
10.3.3.0/24 dev net_103_104 proto kernel scope link src 10.3.3.254
10.3.3.0/24 dev net_103_104 proto bird scope link metric 32
10.4.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.4.1.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.11.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.12.0.0/24 via 10.3.3.253 dev net_103_104 proto bird metric 32
10.100.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.103.0.0/24 dev ix103 proto kernel scope link src 10.103.0.3
10.103.0.0/24 dev ix103 proto bird scope link metric 32
10.104.0.0/24 via 10.3.3.253 dev net_103_104 proto bird metric 32
10.105.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.150.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.151.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.152.0.0/24 via 10.3.3.253 dev net_103_104 proto bird metric 32
10.153.0.0/24 via 10.3.3.253 dev net_103_104 proto bird metric 32
10.154.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.155.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.156.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.160.0.0/24 via 10.103.0.160 dev ix103 proto bird metric 32
10.161.0.0/24 via 10.103.0.161 dev ix103 proto bird metric 32
10.162.0.0/24 via 10.103.0.162 dev ix103 proto bird metric 32
10.163.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.164.0.0/24 via 10.3.3.253 dev net_103_104 proto bird metric 32
10.170.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.171.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.190.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
root@4d6dc753a492 / #
```

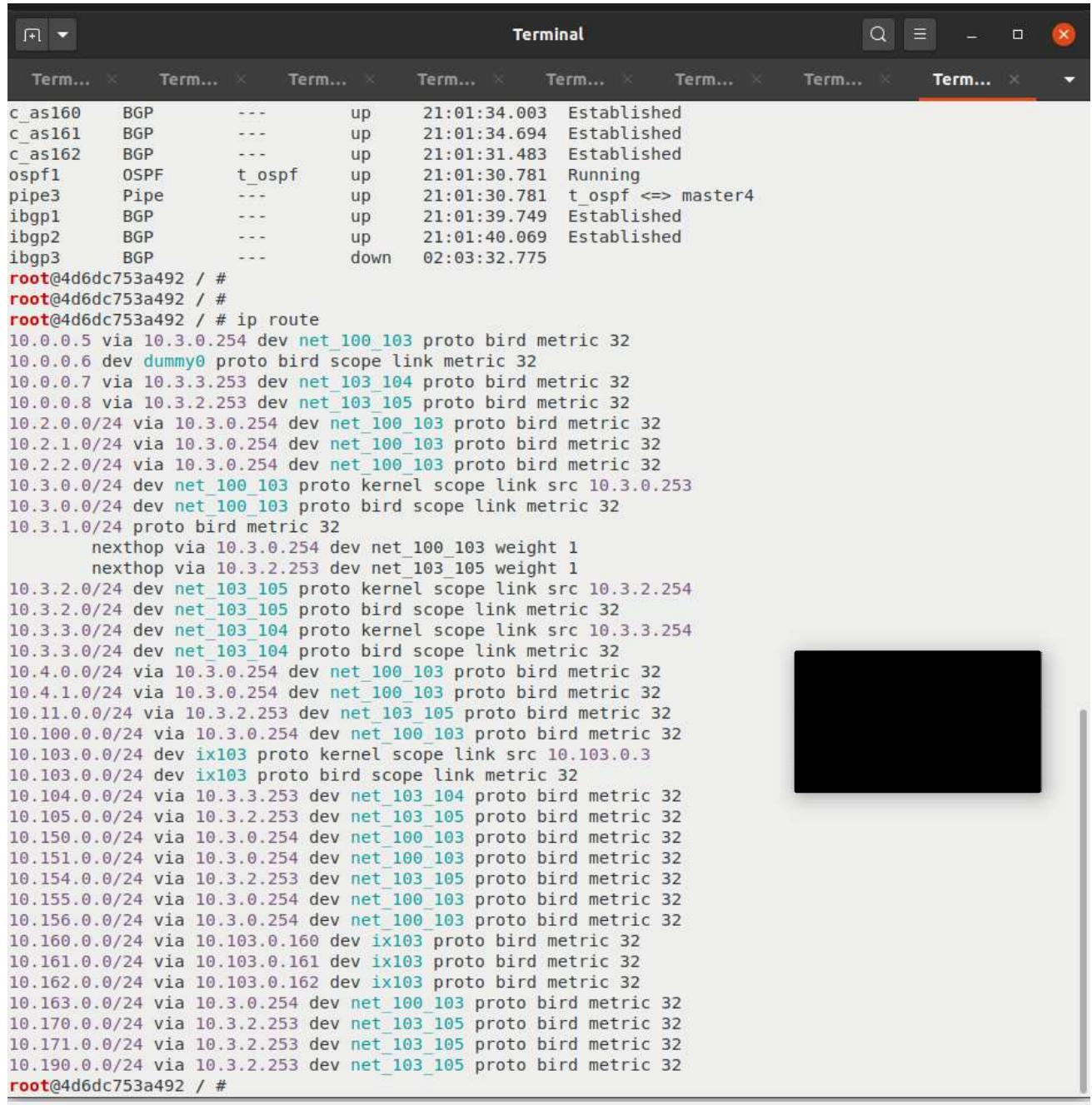
Then I disabled the IBGP.

```

root@4d6dc753a492 / # birdc show protocols
BIRD 2.0.7 ready.
Name      Proto     Table      State   Since      Info
device1   Device    ---        up      21:01:30.781
kernel1   Kernel    master4   up      21:01:30.781
local_nets Direct   ---        up      21:01:30.781
pipe1     Pipe      ---        up      21:01:30.781 t_bgp <=> master4
pipe2     Pipe      ---        up      21:01:30.781 t_direct <=> t_bgp
c_as160   BGP       ---        up      21:01:34.004 Established
c_as161   BGP       ---        up      21:01:34.695 Established
c_as162   BGP       ---        up      21:01:31.483 Established
ospf1     OSPF      t_ospf    up      21:01:30.781 Running
pipe3     Pipe      ---        up      21:01:30.781 t_ospf <=> master4
ibgp1     BGP       ---        up      21:01:39.749 Established
ibgp2     BGP       ---        up      21:01:40.069 Established
ibgp3     BGP       ---        up      21:01:38.228 Established
root@4d6dc753a492 / # birdc disable ibgp3
BIRD 2.0.7 ready.
ibgp3: disabled
root@4d6dc753a492 / # birdc show protocols
BIRD 2.0.7 ready.
Name      Proto     Table      State   Since      Info
device1   Device    ---        up      21:01:30.781
kernel1   Kernel    master4   up      21:01:30.781
local_nets Direct   ---        up      21:01:30.781
pipe1     Pipe      ---        up      21:01:30.781 t_bgp <=> master4
pipe2     Pipe      ---        up      21:01:30.781 t_direct <=> t_bgp
c_as160   BGP       ---        up      21:01:34.003 Established
c_as161   BGP       ---        up      21:01:34.694 Established
c_as162   BGP       ---        up      21:01:31.483 Established
ospf1     OSPF      t_ospf    up      21:01:30.781 Running
pipe3     Pipe      ---        up      21:01:30.781 t_ospf <=> master4
ibgp1     BGP       ---        up      21:01:39.749 Established
ibgp2     BGP       ---        up      21:01:40.069 Established
ibgp3     BGP       ---        down    02:03:32.775
root@4d6dc753a492 / #

```

Now I checked the routes again after disabling the IBGP. I noticed that the routes to several different networks were removed (namely 10.152.0.0/24, 10.153.0.0/24, and 10.164.0.0/24). This demonstrates the direct impact of IBGP3 on routing decisions within the network. Disabling this protocol changed the available routes, which could hypothetically be exploited by attackers to manipulate routing paths.



The screenshot shows a terminal window with multiple tabs open, all titled "Term...". The active tab displays the output of several commands:

- `c_as160 BGP --- up 21:01:34.003 Established`
- `c_as161 BGP --- up 21:01:34.694 Established`
- `c_as162 BGP --- up 21:01:31.483 Established`
- `ospf1 OSPF t_ospf up 21:01:30.781 Running`
- `pipe3 Pipe --- up 21:01:30.781 t_ospf <=> master4`
- `ibgp1 BGP --- up 21:01:39.749 Established`
- `ibgp2 BGP --- up 21:01:40.069 Established`
- `ibgp3 BGP --- down 02:03:32.775`

`root@4d6dc753a492 / #`

`root@4d6dc753a492 / # ip route`

```
10.0.0.5 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.0.0.6 dev dummy0 proto bird scope link metric 32
10.0.0.7 via 10.3.3.253 dev net_103_104 proto bird metric 32
10.0.0.8 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.2.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.2.1.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.2.2.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.3.0.0/24 dev net_100_103 proto kernel scope link src 10.3.0.253
10.3.0.0/24 dev net_100_103 proto bird scope link metric 32
10.3.1.0/24 proto bird metric 32
    nexthop via 10.3.0.254 dev net_100_103 weight 1
    nexthop via 10.3.2.253 dev net_103_105 weight 1
10.3.2.0/24 dev net_103_105 proto kernel scope link src 10.3.2.254
10.3.2.0/24 dev net_103_105 proto bird scope link metric 32
10.3.3.0/24 dev net_103_104 proto kernel scope link src 10.3.3.254
10.3.3.0/24 dev net_103_104 proto bird scope link metric 32
10.4.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.4.1.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.11.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.100.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.103.0.0/24 dev ix103 proto kernel scope link src 10.103.0.3
10.103.0.0/24 dev ix103 proto bird scope link metric 32
10.104.0.0/24 via 10.3.3.253 dev net_103_104 proto bird metric 32
10.105.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.150.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.151.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.154.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.155.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.156.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.160.0.0/24 via 10.103.0.160 dev ix103 proto bird metric 32
10.161.0.0/24 via 10.103.0.161 dev ix103 proto bird metric 32
10.162.0.0/24 via 10.103.0.162 dev ix103 proto bird metric 32
10.163.0.0/24 via 10.3.0.254 dev net_100_103 proto bird metric 32
10.170.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.171.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
10.190.0.0/24 via 10.3.2.253 dev net_103_105 proto bird metric 32
```

`root@4d6dc753a492 / #`

Task 2.b: Experimenting with IGP

The above screenshot from the previous task shows what the current route already looks like. As you can see, I will now disable the OSPF1 protocol as per the instructions.

```

root@4d6dc753a492 / #
root@4d6dc753a492 / #
root@4d6dc753a492 / # birdc disable ospf1
BIRD 2.0.7 ready.
ospf1: disabled
root@4d6dc753a492 / # birdc show protocols
BIRD 2.0.7 ready.
Name      Proto     Table      State   Since      Info
device1   Device    ---        up       21:01:30.781
kernel1   Kernel    master4   up       21:01:30.781
local_nets Direct   ---        up       21:01:30.781
pipe1     Pipe      ---        up       21:01:30.781  t_bgp <=> master4
pipe2     Pipe      ---        up       21:01:30.781  t_direct <=> t_bgp
c_as160   BGP       ---        up       21:01:34.004  Established
c_as161   BGP       ---        up       21:01:34.695  Established
c_as162   BGP       ---        up       21:01:31.484  Established
ospf1     OSPF      t_ospf    down    02:14:29.620
pipe3     Pipe      ---        up       21:01:30.781  t_ospf <=> master4
ibgp1    BGP       ---        up       21:01:39.750  Established
ibgp2    BGP       ---        up       21:01:40.069  Established
ibgp3    BGP       ---        down    02:03:32.776
root@4d6dc753a492 / #

```

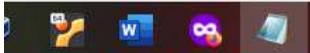


Then I checked the routes again after disabling ospf. I noticed that the routes to directly connected networks via Bird were only present. The fact that the screenshot provided lacks OSPF routes emphasizes how important Interior Gateway Protocols (IGPs), are for transit autonomous systems (ASes). Within an AS, IGPs are in charge of dynamically sharing routing data to guarantee dependable and effective packet forwarding. The network's adaptability, scalability, and fault tolerance would be severely limited if it had to rely on manual configurations in the absence of a functional IGP. The efficient operation of transit ASes, which are in charge of forwarding traffic between various networks and autonomous systems on the internet, depends on maintaining a strong IGP.

```

root@4d6dc753a492 / # ip route
unreachable 10.2.0.0/24 proto bird metric 32
unreachable 10.2.1.0/24 proto bird metric 32
unreachable 10.2.2.0/24 proto bird metric 32
10.3.0.0/24 dev net_100_103 proto kernel scope link src 10.3.0.253
10.3.0.0/24 dev net_100_103 proto bird scope link metric 32
unreachable 10.3.1.0/24 proto bird metric 32
10.3.2.0/24 dev net_103_105 proto kernel scope link src 10.3.2.254
10.3.2.0/24 dev net_103_105 proto bird scope link metric 32
10.3.3.0/24 dev net_103_104 proto kernel scope link src 10.3.3.254
10.3.3.0/24 dev net_103_104 proto bird scope link metric 32
unreachable 10.4.0.0/24 proto bird metric 32
unreachable 10.4.1.0/24 proto bird metric 32
unreachable 10.11.0.0/24 proto bird metric 32
10.103.0.0/24 dev ix103 proto kernel scope link src 10.103.0.3
unreachable 10.150.0.0/24 proto bird metric 32
unreachable 10.151.0.0/24 proto bird metric 32
unreachable 10.154.0.0/24 proto bird metric 32
unreachable 10.155.0.0/24 proto bird metric 32
unreachable 10.156.0.0/24 proto bird metric 32
10.160.0.0/24 via 10.103.0.160 dev ix103 proto bird metric 32
10.161.0.0/24 via 10.103.0.161 dev ix103 proto bird metric 32
10.162.0.0/24 via 10.103.0.162 dev ix103 proto bird metric 32
unreachable 10.163.0.0/24 proto bird metric 32
unreachable 10.170.0.0/24 proto bird metric 32
unreachable 10.171.0.0/24 proto bird metric 32
unreachable 10.190.0.0/24 proto bird metric 32
root@4d6dc753a492 / #

```



Task 3: Path Selection

Task 3.a.

Here, you can see me redirecting all the routes to a file called "all-routes". I've chosen to utilize the network prefix "10.190.0.0/24" after looking over the data because it may be reached via two different routes that were obtained from autonomous systems AS-3 and AS-2. Every route starts with an Interior Gateway Protocol (IGP) and has a unique next hop. AS-2's path is longer, with an AS path of 2 3 190, than AS-3's, which has an AS path of 3 190. Both routes have the same local preference rating of 10, notwithstanding their differences. In this situation, the shorter AS path of AS3's route makes it the recommended way to get to the destination.

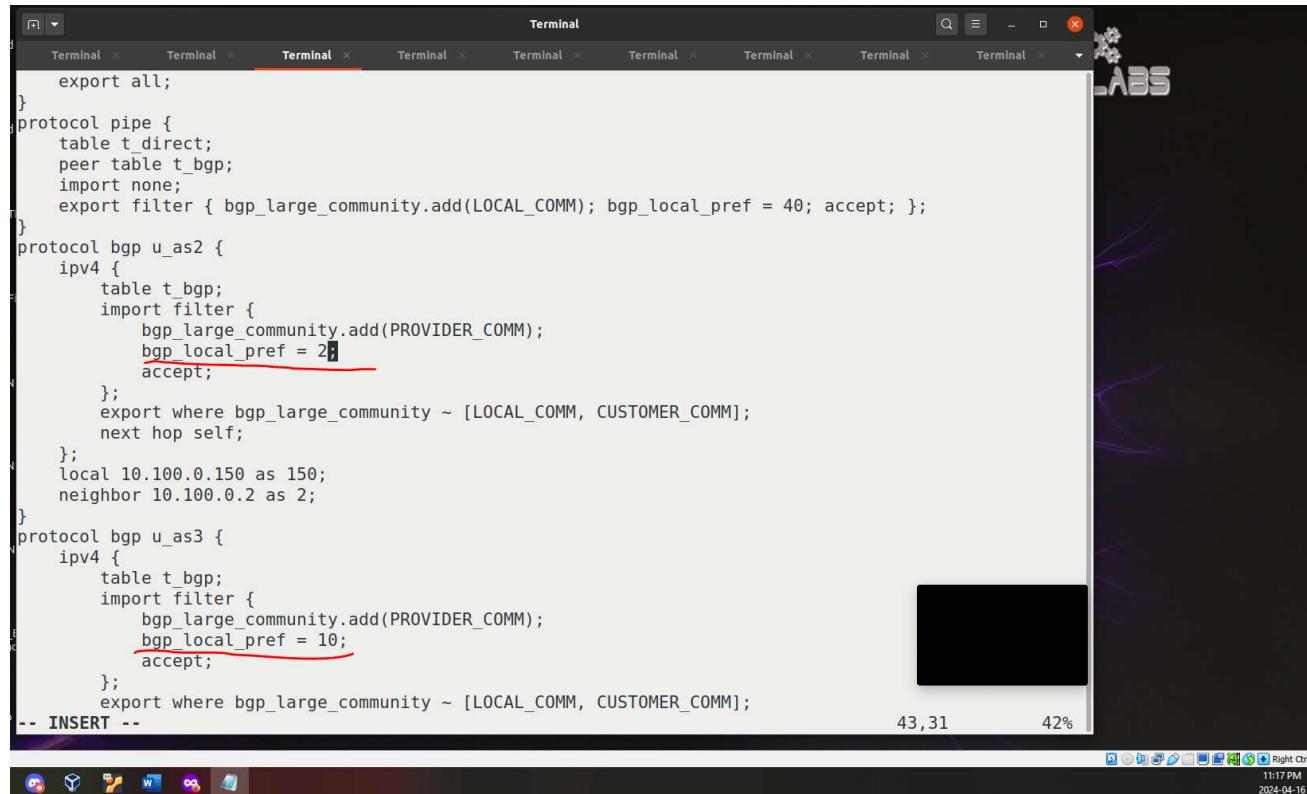
Here, you can see the routes used for my desired network prefix. You can clearly see that the route chosen by the kernel aligns with the one selected by the BGP router.



```
root@feb59914bbf1 /tmp #
root@feb59914bbf1 /tmp # ip route show 10.190.0.0/24
10.190.0.0/24 via 10.100.0.3 dev ix100 proto bird metric 32
root@feb59914bbf1 /tmp #
```

Task 3.b.

As the main upstream link, AS-3 is intended to be prioritized in the BGP configuration for AS-150. To do this, routes received from AS-3 are given a higher local preference value of 10. As the backup connection, AS-2, on the other hand, is configured with a lower local preference value of 2. With this setup, AS-3 will be the primary connection used by AS-150 for all inbound and outbound traffic, with AS-2 serving as a backup link only in case the AS-3 link fails.



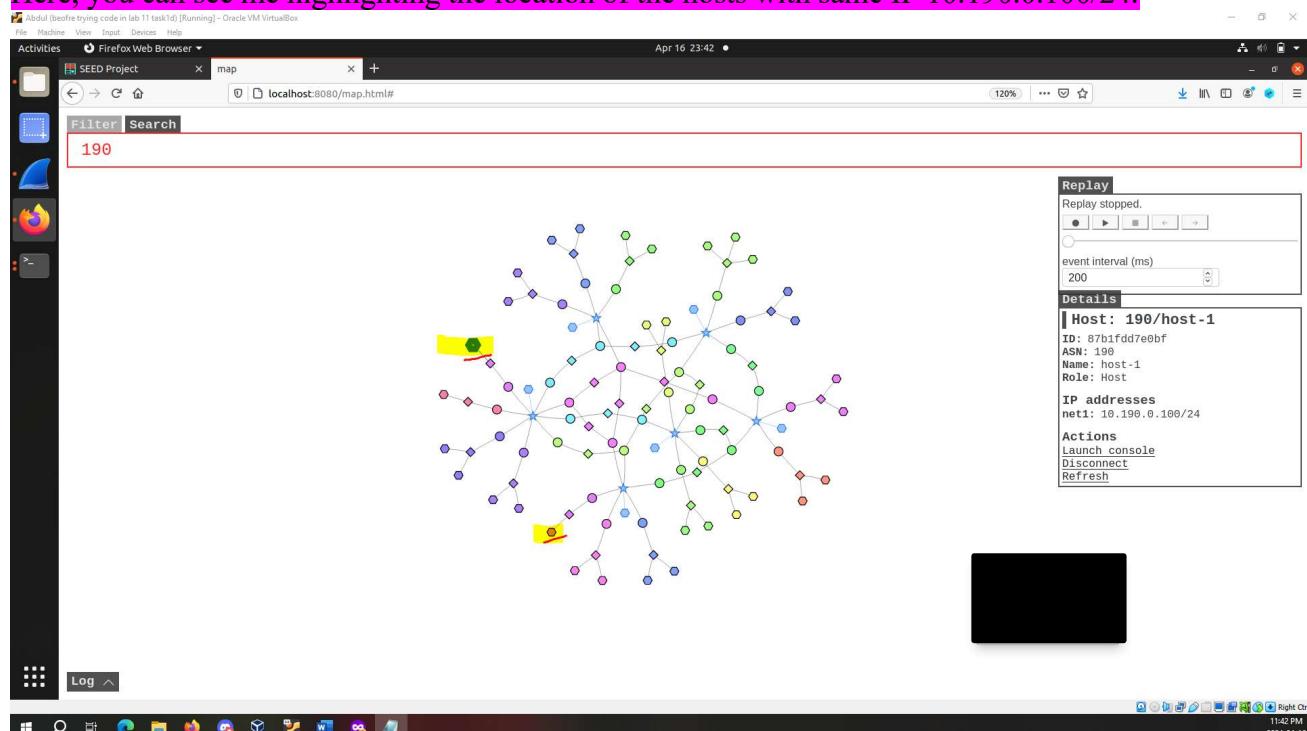
```

export all;
}
protocol pipe {
    table t_direct;
    peer table t_bgp;
    import none;
    export filter { bgp_large_community.add(LOCAL_COMM); bgp_local_pref = 40; accept; };
}
protocol bgp u_as2 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 2;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM];
        next hop self;
    };
    local 10.100.0.150 as 150;
    neighbor 10.100.0.2 as 2;
}
protocol bgp u_as3 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 10;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM];
-- INSERT --

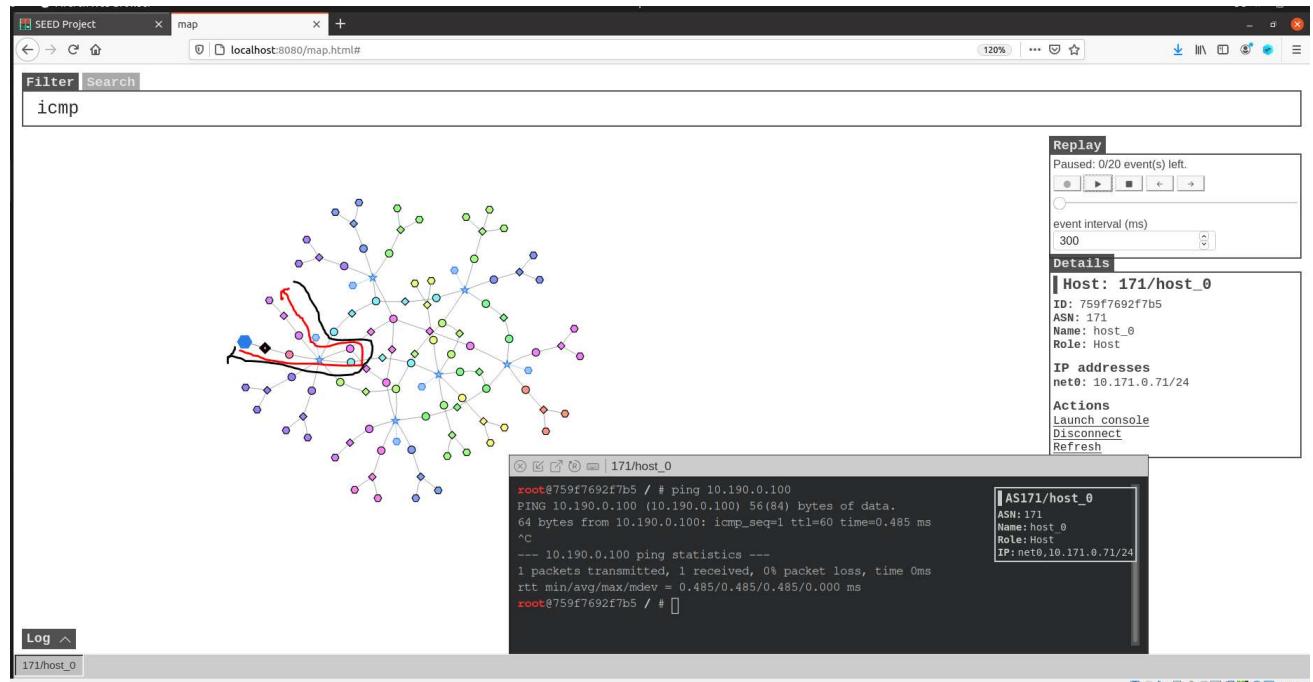
```

Task 4: IP Anycast

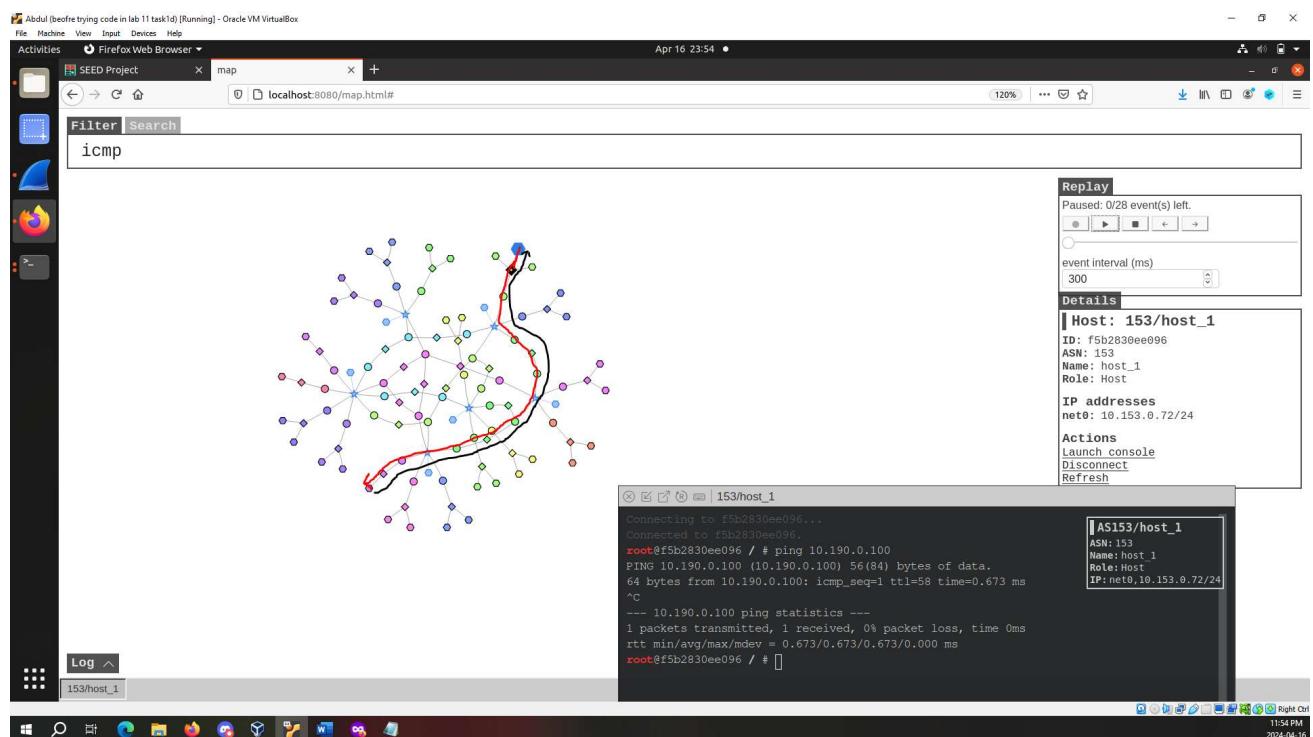
Here, you can see me highlighting the location of the hosts with same IP 10.190.0.100/24.



In order to determine the path taken, the screenshot shows my ping activity from a chosen host with the IP address "10.171.0.71/24" to the destination "10.190.0.100". The ICMP ping reply path is shown in black, while the path used by the ICMP ping request to reach "10.190.0.100" is highlighted in red.



The same thing is done in the below screenshot from a different machine. In order to determine the path taken, the screenshot shows my ping activity from a host with the IP address "10.153.0.72/24" to the destination "10.190.0.100". The ICMP ping reply path is shown in black, while the path used by the ICMP ping request to reach "10.190.0.100" is highlighted in red.

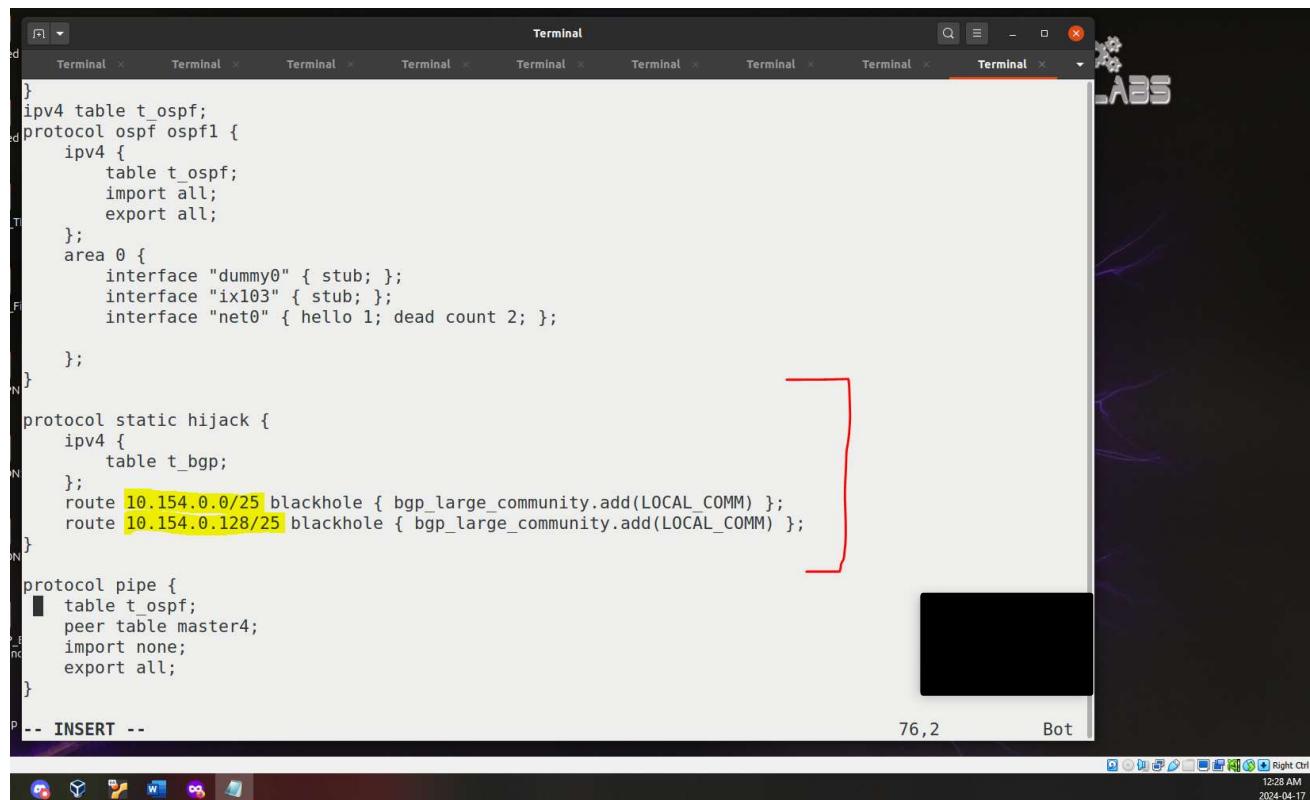


When IP anycast is used, several workstations scattered over different areas are assigned the same IP address. Using entries from the routing table, routing protocols such as BGP specify which machine is to receive incoming packets. In the scenario outlined, AS-190 oversees two disconnected networks, both connected to separate interfaces (IX-100 and IX-105) and having the same network prefix (10.190.0.0/24). 10.190.0.100 is the IP address shared by both networks. Pings from different hosts (10.171.0.71/24 and 10.153.0.72/24) to 10.190.0.100 follow different routes as a result of the routing choices made by the BGP routers. Packets with the same IP address may take different paths since each host chooses the best or shortest route to get to the destination on its own.

Task 5: BGP Prefix Attacks

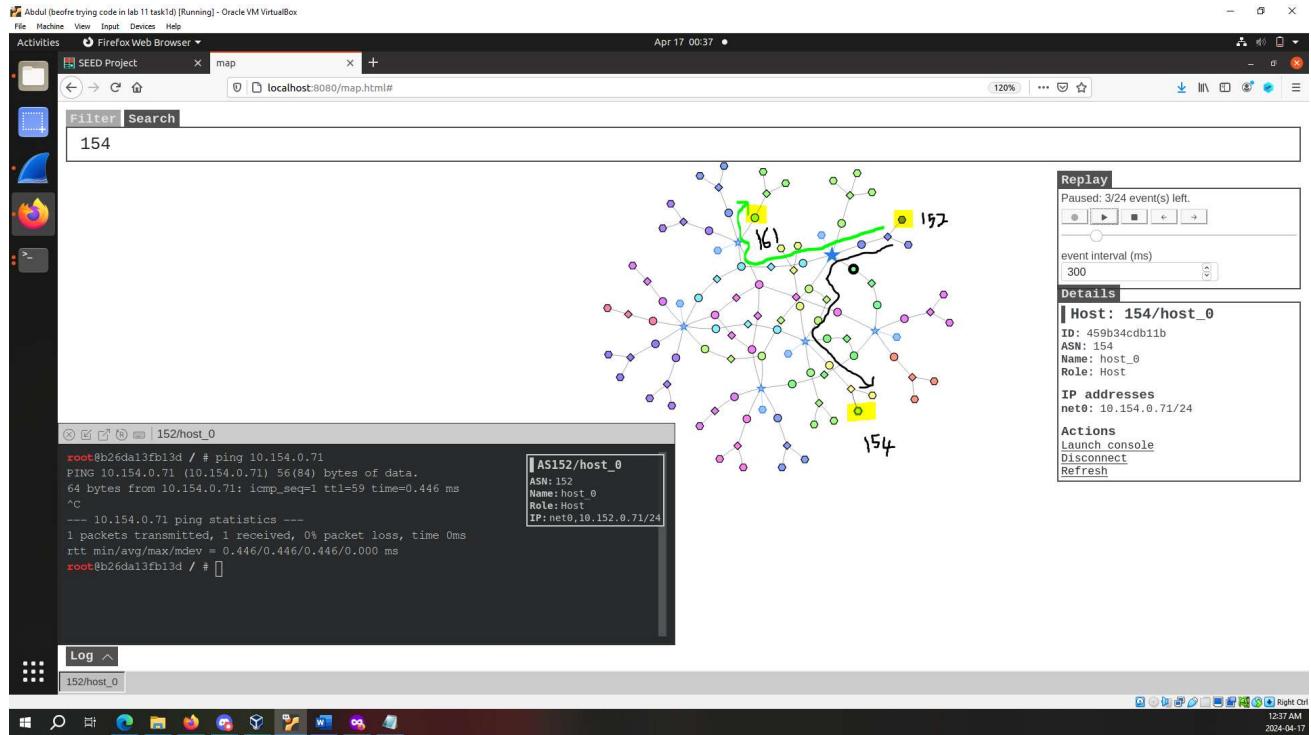
Task 5.a: Launching the Prefix Hijacking Attack from AS-161

Here, you can see me modifying the *bird.conf* file in AS-161 such that AS-161 can receive traffic from AS-154.



```
...  
    protocol static hijack {  
        ipv4 {  
            table t_bgp;  
        };  
        route 10.154.0.0/25 blackhole { bgp_large_community.add(LOCAL_COMM) };  
        route 10.154.0.128/25 blackhole { bgp_large_community.add(LOCAL_COMM) };  
    }  
...  
    protocol pipe {  
        table t_ospf;  
        peer table master4;  
        import none;  
        export all;  
    }  
-- INSERT --  
76,2 Bot  
...  
12:28 AM 2024-04-17
```

Here, you can see me attempting to ping 10.154.0.71 from host 152. The line highlighted in black shows the expected path to be taken, however the line highlighted in green is the path taken instead, towards the attacker.



Task 5.b: Fighting back from AS-154

According to the book one method of fighting back is to add more prefixes. As you can see in the below screenshot, I have attempted to do so by adding a few more prefixes in the *birds.conf* file of AS-154.

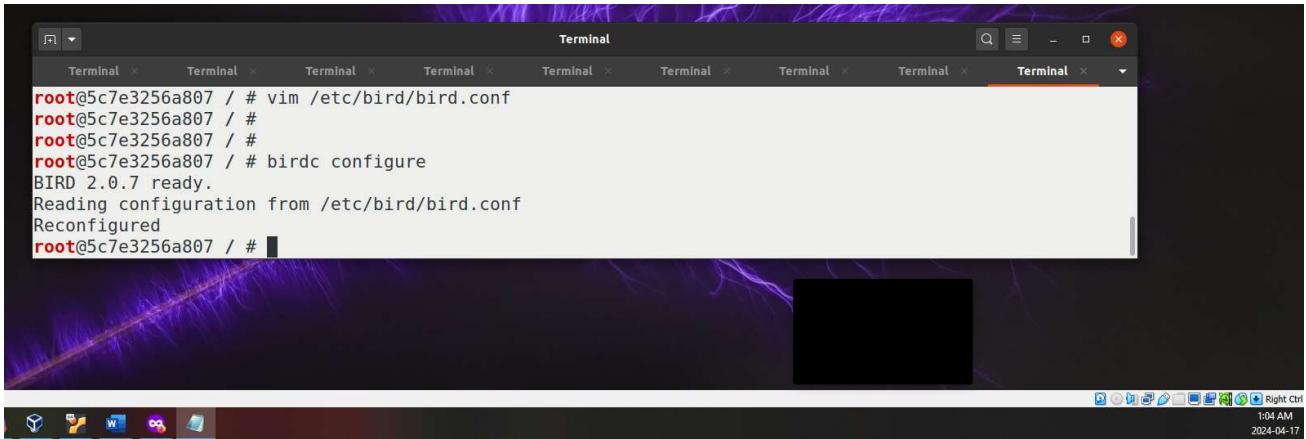
```

root@b26da13fb13d ~ % cat /etc/bird/bird.conf
# Configuration for AS 154
# OSPF configuration
protocol ospf ospf1 {
    ipv4 {
        table t_ospf;
        import all;
        export all;
    };
    area 0 {
        interface "dummy0" { stub; };
        interface "ix102" { stub; };
        interface "net0" { hello 1; dead count 2; };
    };
}
# Static routing configuration
protocol static {
    ipv4 {
        table t_bgp;
    };
    route 10.154.0.0/26 via "net0" { bgp_large_community.add(LOCAL_COMM); };
    route 10.154.0.64/26 via "net0" { bgp_large_community.add(LOCAL_COMM); };
    route 10.154.0.128/26 via "net0" { bgp_large_community.add(LOCAL_COMM); };
    route 10.154.0.192/26 via "net0" { bgp_large_community.add(LOCAL_COMM); };
}
# BGP configuration
protocol pipe {
    table t_ospf;
    peer table master4;
    import none;
    export all;
}

# BGP configuration
protocol static {
    ipv4 {
        table t_bgp;
    };
    route 10.154.0.0/26 via "net0" { bgp_large_community.add(LOCAL_COMM); };
    route 10.154.0.64/26 via "net0" { bgp_large_community.add(LOCAL_COMM); };
    route 10.154.0.128/26 via "net0" { bgp_large_community.add(LOCAL_COMM); };
    route 10.154.0.192/26 via "net0" { bgp_large_community.add(LOCAL_COMM); };
}
# End of configuration
root@b26da13fb13d ~ %

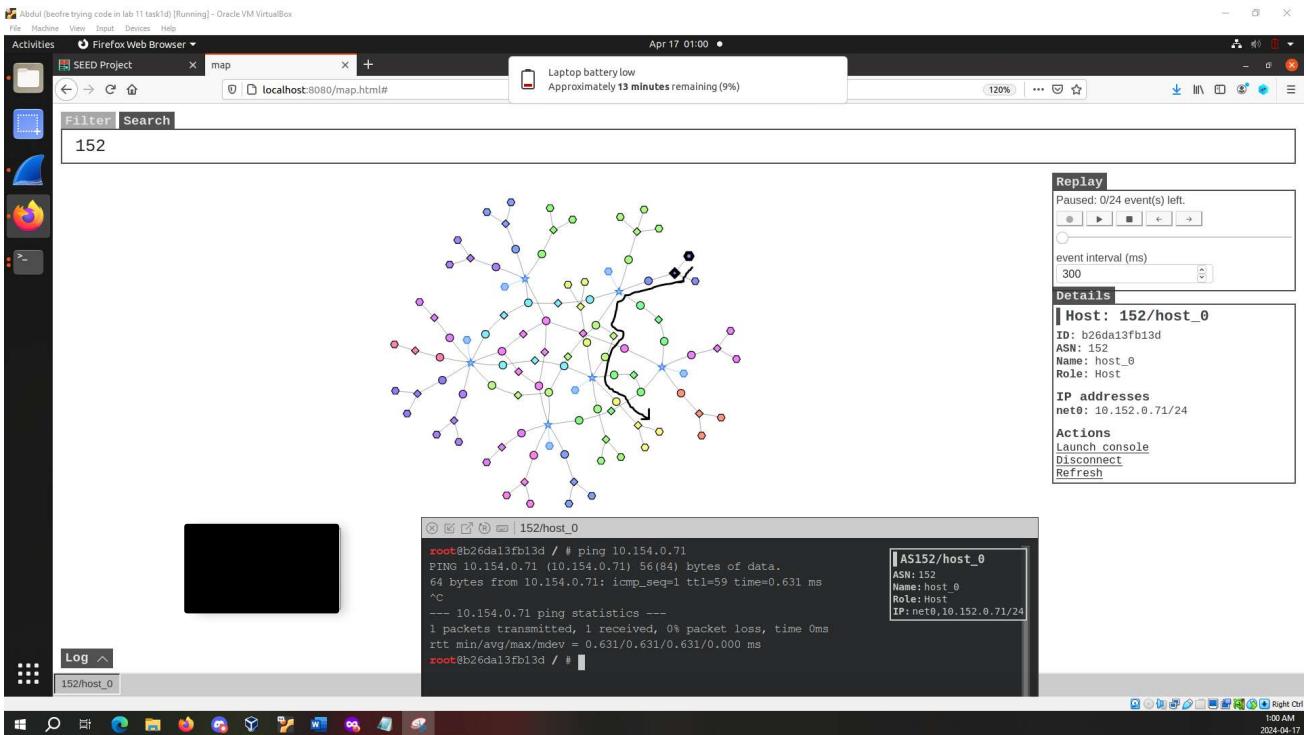
```

I ensured these changes were applied by running `birdc configure`.



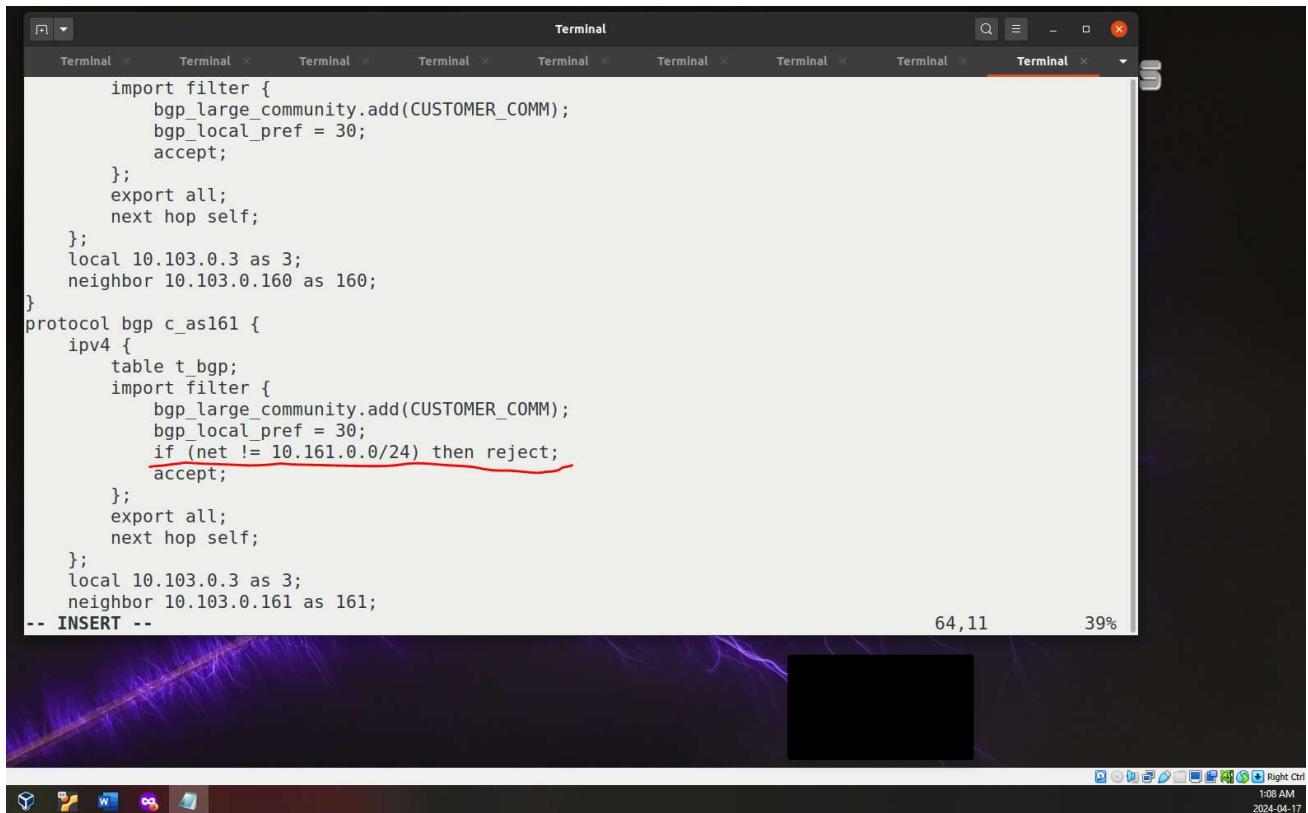
```
root@5c7e3256a807 / # vim /etc/bird/bird.conf
root@5c7e3256a807 / #
root@5c7e3256a807 / #
root@5c7e3256a807 / # birdc configure
BIRD 2.0.7 ready.
Reading configuration from /etc/bird/bird.conf
Reconfigured
root@5c7e3256a807 / #
```

After completing the above changes, and pinging once again, we can see that this time the expected path was taken (as highlighted in black).



Task 5.c: Fixing the problem at AS-3

Here, you can see me modifying the configuration by adding a single filter line in the `birds.conf` file of AS-3. This rule checks to see if the network is different from if that is the case it is rejected. This modification is intended to inhibit the spread of fake alerts, protecting the adoption of only legitimate routes. Strict filtering like this is essential to enhancing network security and stability.



```
import filter {
    bgp_large_community.add(CUSTOMER_COMM);
    bgp_local_pref = 30;
    accept;
};

export all;
next hop self;

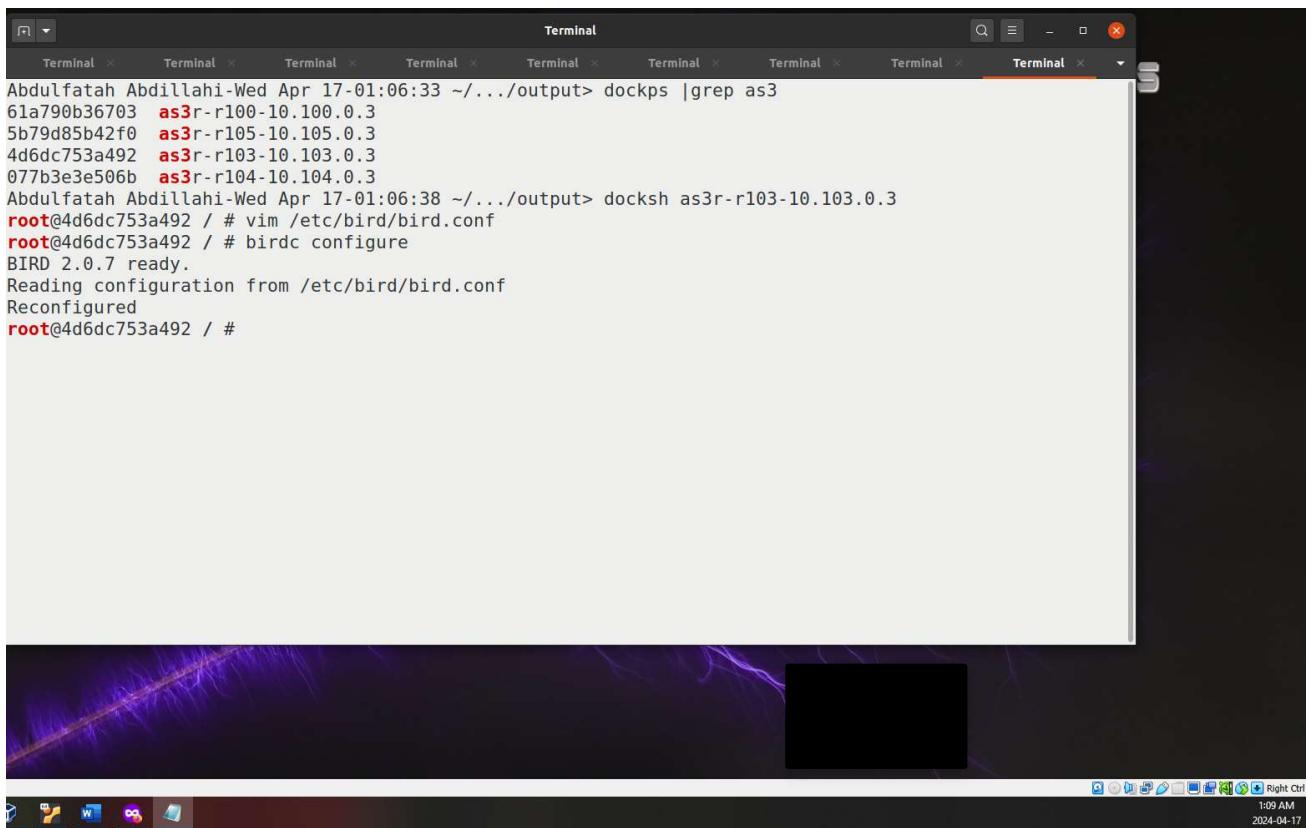
};

local 10.103.0.3 as 3;
neighbor 10.103.0.160 as 160;
}

protocol bgp c_as161 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(CUSTOMER_COMM);
            bgp_local_pref = 30;
            if (net != 10.161.0.0/24) then reject;
            accept;
        };
        export all;
        next hop self;
    };
    local 10.103.0.3 as 3;
    neighbor 10.103.0.161 as 161;
-- INSERT --

```

Finally, you can see that this modification was applied successfully.



```
Abdulfatah Abdillahi-Wed Apr 17-01:06:33 ~/.../output> dockps |grep as3
61a790b36703 as3r-r100-10.100.0.3
5b79d85b42f0 as3r-r105-10.105.0.3
4d6dc753a492 as3r-r103-10.103.0.3
077b3e3e506b as3r-r104-10.104.0.3
Abdulfatah Abdillahi-Wed Apr 17-01:06:38 ~/.../output> docksh as3r-r103-10.103.0.3
root@4d6dc753a492 / # vim /etc/bird/bird.conf
root@4d6dc753a492 / # birdc configure
BIRD 2.0.7 ready.
Reading configuration from /etc/bird/bird.conf
Reconfigured
root@4d6dc753a492 / #
```

References

https://seedsecuritylabs.org/Labs_20.04/Networking/BGP/BGP_Exploration_Attack/

https://seedsecuritylabs.org/Labs_20.04/Files/BGP_Exploration_Attack/BGP_Exploration_Attack.pdf

Du, W. (2022). Internet Security: A Hands-on Approach (Third edition). Independent.