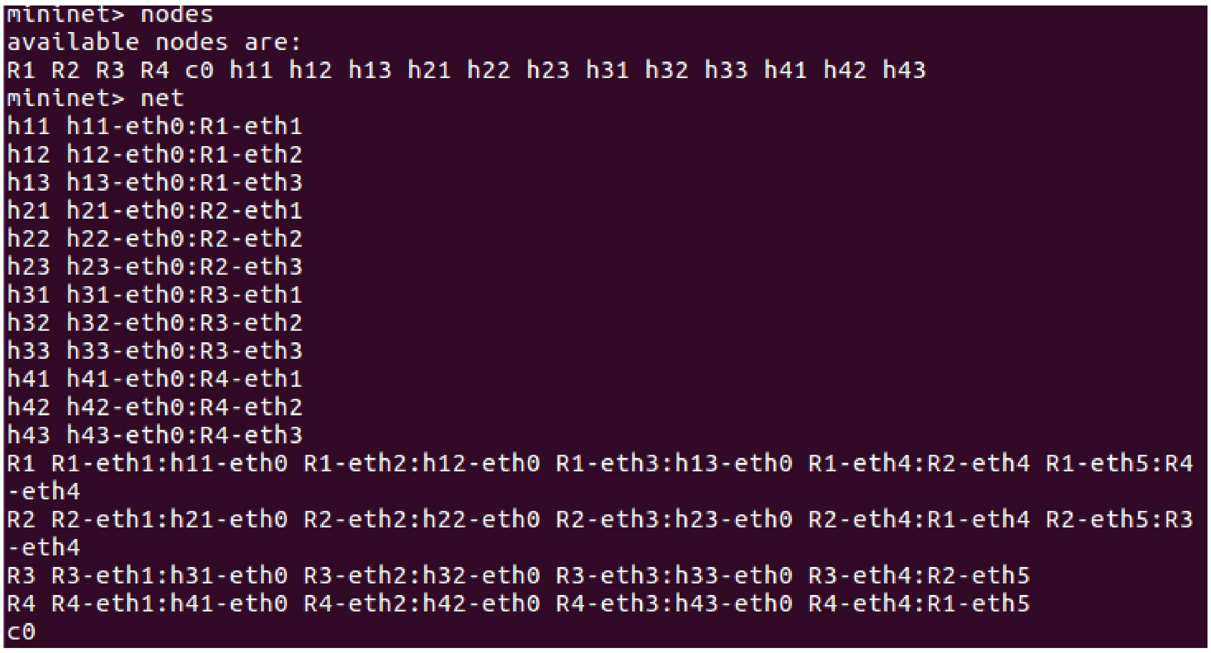
Tan Shun Yu 1001171 ISTD
Nigel Leong 1001095 ISTD

Networks Lab 5 Report

• The topology as you were able to derive it
– IP addresses of all routers
– Hosts/ IPs in the Autonomous Systems



```
mininet> nodes
available nodes are:
R1 R2 R3 R4 c0 h11 h12 h13 h21 h22 h23 h31 h32 h33 h41 h42 h43
mininet> net
h11 h11-eth0:R1-eth1
h12 h12-eth0:R1-eth2
h13 h13-eth0:R1-eth3
h21 h21-eth0:R2-eth1
h22 h22-eth0:R2-eth2
h23 h23-eth0:R2-eth3
h31 h31-eth0:R3-eth1
h32 h32-eth0:R3-eth2
h33 h33-eth0:R3-eth3
h41 h41-eth0:R4-eth1
h42 h42-eth0:R4-eth2
h43 h43-eth0:R4-eth3
R1 R1-eth1:h11-eth0 R1-eth2:h12-eth0 R1-eth3:h13-eth0 R1-eth4:R2-eth4 R1-eth5:R4
-eth4
R2 R2-eth1:h21-eth0 R2-eth2:h22-eth0 R2-eth3:h23-eth0 R2-eth4:R1-eth4 R2-eth5:R3
-eth4
R3 R3-eth1:h31-eth0 R3-eth2:h32-eth0 R3-eth3:h33-eth0 R3-eth4:R2-eth5
R4 R4-eth1:h41-eth0 R4-eth2:h42-eth0 R4-eth3:h43-eth0 R4-eth4:R1-eth5
c0
```

Figure 1: nodes and net information

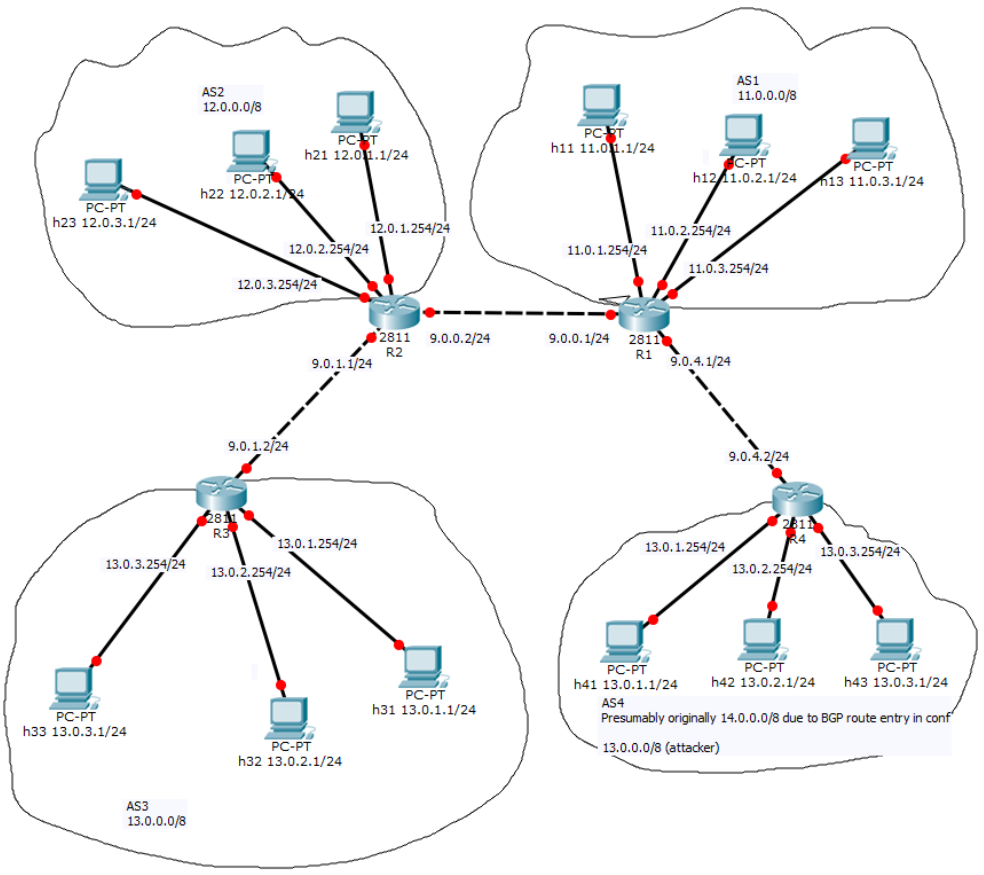


Figure 2: Topology of the Network

Tan Shun Yu 1001171 ISTD
Nigel Leong 1001095 ISTD

Using *ifconfig and show ip bgp* on each of the routers to derive the following information:

<u>AS1</u>
AS1: 11.0.0.0/8
R1-eth1(h11): 11.0.1.254/24
R1-eth2(h12): 11.0.2.254/24
R1-eth3(h13): 11.0.3.254/24
R1-eth4(R2): 9.0.0.1/24
R1-eth5(R4): 9.0.4.1/24

<u>AS2</u>
AS2: 12.0.0.0/8
R2-eth1(h21): 12.0.1.254/24
R2-eth2(h22): 12.0.2.254/24
R2-eth3(h23): 12.0.3.254/24
R2-eth4(R1): 9.0.0.2/24
R2-eth5(R3): 9.0.1.1/24

<u>AS3</u>
AS3: 13.0.0.0/8
R3-eth1(h31): 13.0.1.254/24
R3-eth2(h32): 13.0.2.254/24
R3-eth3(h33): 13.0.3.254/24
R3-eth4(R2): 9.0.1.2/24

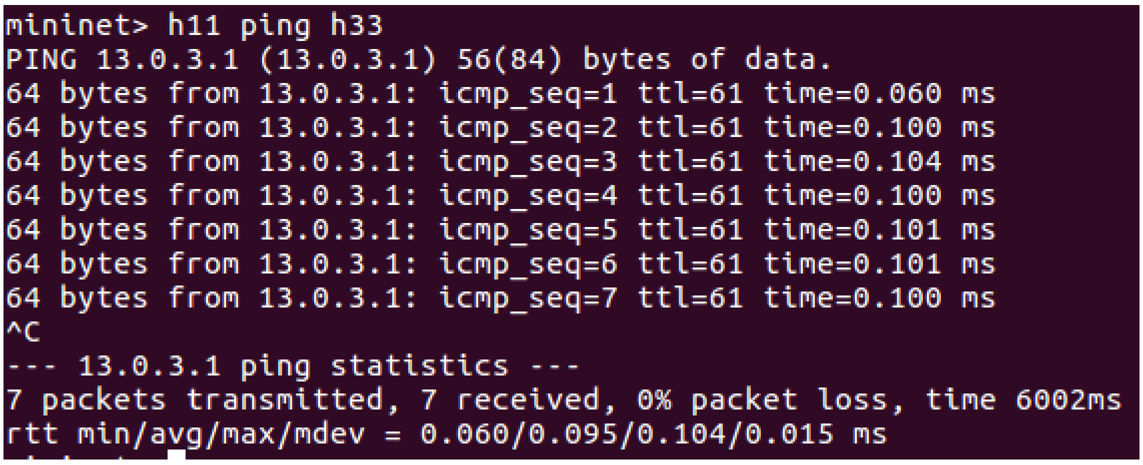
<u>AS4</u>
AS4: 14.0.0.0/8 (to be changed later to spoof as AS3 (13.0.0.0/8)
R4-eth1(h41): 13.0.1.254/24
R4-eth2(h42): 13.0.2.254/24
R4-eth3(h43): 13.0.3.254/24
R4-eth4(R1): 9.0.4.2/24

Tan Shun Yu 1001171 ISTD
Nigel Leong 1001095 ISTD

• Why was it initially not possible to reach 13.0.1.1 from AS1? How did you find out/what did you do to fix this?

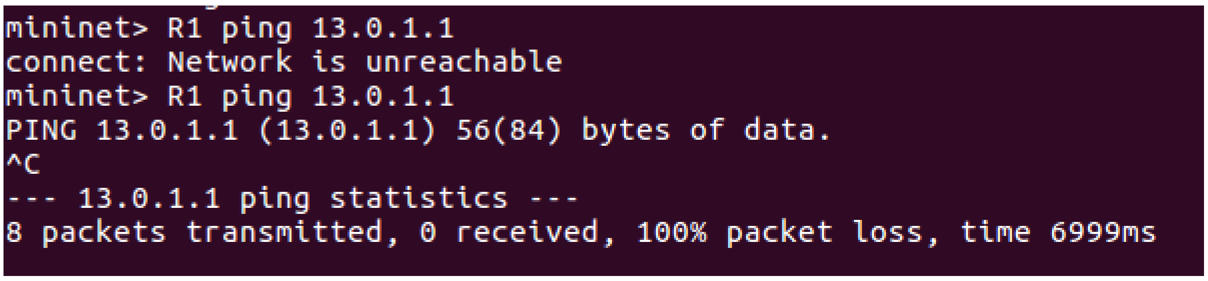
It was initially not possible to reach 13.0.1.1 from R1 AS1 because the packets sent from R1 is able to route to 13.0.1.1 but from R3, it does not know how to route back to R1 AS1 due to a missing routing table entry in R3 which is supposed to tell R3 how to route packets that have destination IP address of R1 (in this case: 9.0.0.0/24 network). There is no routing entry in R3 to route the packets to R1 but there is a routing entry in R3 to route the packets to the host h11 in AS1.

This is verified by using the command *h11 ping h33* (refer to Figure 3), which shows that h11 in AS1 is able to contact h33 in AS3 and that R3 has no problems finding the route to h11 (but not R1). Also, the command *R1 ping 13.0.1.1* (refer to Figure 4) is unable to receive any replies.



```
mininet> h11 ping h33
PING 13.0.3.1 (13.0.3.1) 56(84) bytes of data.
64 bytes from 13.0.3.1: icmp_seq=1 ttl=61 time=0.060 ms
64 bytes from 13.0.3.1: icmp_seq=2 ttl=61 time=0.100 ms
64 bytes from 13.0.3.1: icmp_seq=3 ttl=61 time=0.104 ms
64 bytes from 13.0.3.1: icmp_seq=4 ttl=61 time=0.100 ms
64 bytes from 13.0.3.1: icmp_seq=5 ttl=61 time=0.101 ms
64 bytes from 13.0.3.1: icmp_seq=6 ttl=61 time=0.101 ms
64 bytes from 13.0.3.1: icmp_seq=7 ttl=61 time=0.100 ms
^C
--- 13.0.3.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6002ms
rtt min/avg/max/mdev = 0.060/0.095/0.104/0.015 ms
```

Figure 3: h11 ping h33

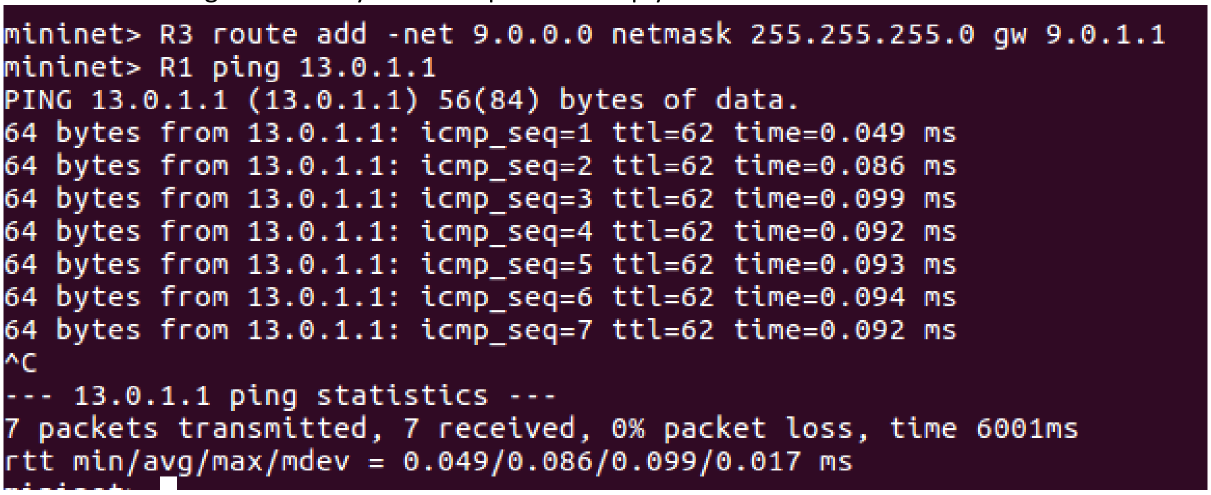


```
mininet> R1 ping 13.0.1.1
connect: Network is unreachable
mininet> R1 ping 13.0.1.1
PING 13.0.1.1 (13.0.1.1) 56(84) bytes of data.
^C
--- 13.0.1.1 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 6999ms
```

Figure 4: R1 unable to contact 13.0.1.1 at first

Tan Shun Yu 1001171 ISTD
Nigel Leong 1001095 ISTD

After configuring R3 to have a routing entry to R1 using *route* (refer to Figure 5), now we are able to successfully ping 13.0.1.1 from R1 and get a response, meaning that the ICMP packets from R1 are now able to recognise the way back and provide a reply.

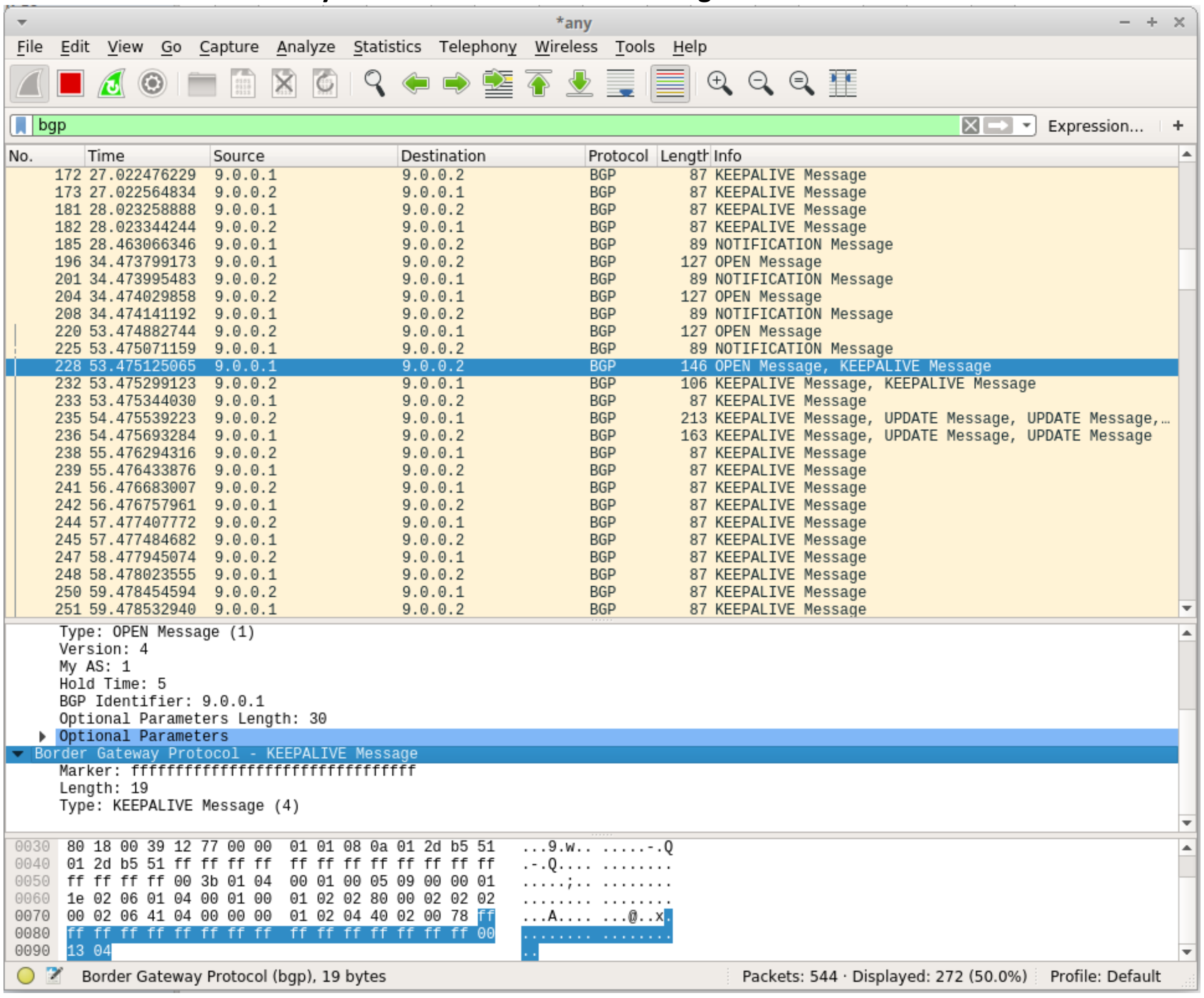


```
mininet> R3 route add -net 9.0.0.0 netmask 255.255.255.0 gw 9.0.1.1
mininet> R1 ping 13.0.1.1
PING 13.0.1.1 (13.0.1.1) 56(84) bytes of data.
64 bytes from 13.0.1.1: icmp_seq=1 ttl=62 time=0.049 ms
64 bytes from 13.0.1.1: icmp_seq=2 ttl=62 time=0.086 ms
64 bytes from 13.0.1.1: icmp_seq=3 ttl=62 time=0.099 ms
64 bytes from 13.0.1.1: icmp_seq=4 ttl=62 time=0.092 ms
64 bytes from 13.0.1.1: icmp_seq=5 ttl=62 time=0.093 ms
64 bytes from 13.0.1.1: icmp_seq=6 ttl=62 time=0.094 ms
64 bytes from 13.0.1.1: icmp_seq=7 ttl=62 time=0.092 ms
^C
--- 13.0.1.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6001ms
rtt min/avg/max/mdev = 0.049/0.086/0.099/0.017 ms
```

Figure 5: After configuring R3, now R1 is able to contact 13.0.1.1

Tan Shun Yu 1001171 ISTD
Nigel Leong 1001095 ISTD

• Describe the BGP traffic you were able to observe during re-establishment of routes.



```
No.   Time          Source    Destination  Protocol  Length Info
  172 27.022476229  9.0.0.1   9.0.0.2      BGP        87 KEEPALIVE Message
  173 27.022564834  9.0.0.2   9.0.0.1      BGP        87 KEEPALIVE Message
  181 28.023258888  9.0.0.1   9.0.0.2      BGP        87 KEEPALIVE Message
  182 28.023344244  9.0.0.2   9.0.0.1      BGP        87 KEEPALIVE Message
  185 28.463066346  9.0.0.1   9.0.0.2      BGP        89 NOTIFICATION Message
  196 34.473799173  9.0.0.1   9.0.0.2      BGP       127 OPEN Message
  201 34.473995483  9.0.0.2   9.0.0.1      BGP        89 NOTIFICATION Message
  204 34.474029858  9.0.0.2   9.0.0.1      BGP       127 OPEN Message
  208 34.474141192  9.0.0.1   9.0.0.2      BGP        89 NOTIFICATION Message
  220 53.474882744  9.0.0.2   9.0.0.1      BGP       127 OPEN Message
  225 53.475071159  9.0.0.1   9.0.0.2      BGP        89 NOTIFICATION Message
  228 53.475125065  9.0.0.1   9.0.0.2      BGP       146 OPEN Message, KEEPALIVE Message
  232 53.475299123  9.0.0.2   9.0.0.1      BGP       106 KEEPALIVE Message, KEEPALIVE Message
  233 53.475344030  9.0.0.1   9.0.0.2      BGP        87 KEEPALIVE Message
  235 54.475539223  9.0.0.2   9.0.0.1      BGP       213 KEEPALIVE Message, UPDATE Message, UPDATE Message,…
  236 54.475693284  9.0.0.1   9.0.0.2      BGP       163 KEEPALIVE Message, UPDATE Message, UPDATE Message
  238 55.476294316  9.0.0.2   9.0.0.1      BGP        87 KEEPALIVE Message
  239 55.476433876  9.0.0.1   9.0.0.2      BGP        87 KEEPALIVE Message
  241 56.476683007  9.0.0.2   9.0.0.1      BGP        87 KEEPALIVE Message
  242 56.476757961  9.0.0.1   9.0.0.2      BGP        87 KEEPALIVE Message
  244 57.477407772  9.0.0.2   9.0.0.1      BGP        87 KEEPALIVE Message
  245 57.477484682  9.0.0.1   9.0.0.2      BGP        87 KEEPALIVE Message
  247 58.477945074  9.0.0.2   9.0.0.1      BGP        87 KEEPALIVE Message
  248 58.478023555  9.0.0.1   9.0.0.2      BGP        87 KEEPALIVE Message
  250 59.478454594  9.0.0.2   9.0.0.1      BGP        87 KEEPALIVE Message
  251 59.478532940  9.0.0.1   9.0.0.2      BGP        87 KEEPALIVE Message

    Type: OPEN Message (1)
    Version: 4
    My AS: 1
    Hold Time: 5
    BGP Identifier: 9.0.0.1
    Optional Parameters Length: 30
  ▸ Optional Parameters
▾ Border Gateway Protocol - KEEPALIVE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 19
    Type: KEEPALIVE Message (4)

0030  80 18 00 39 12 77 00 00  01 01 08 0a 01 2d b5 51   ...9.w.. .....-.Q
0040  01 2d b5 51 ff ff ff ff  ff ff ff ff ff ff ff ff   .-.Q.... ........
0050  ff ff ff ff 00 3b 01 04  00 01 00 05 09 00 00 01   .....;.. ........
0060  1e 02 06 01 04 00 01 00  01 02 02 80 00 02 02 02   ........ ........
0070  00 02 06 41 04 00 00 00  01 02 04 40 02 00 78 ff   ...A.... ...@..x.
0080  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff 00   ........ ........
0090  13 04                                              ..

Border Gateway Protocol (bgp), 19 bytes          Packets: 544 · Displayed: 272 (50.0%)   Profile: Default
```

Figure 6: Wireshark packets showing the BGP messages during re-establishment

During re-establishment of routes, it can be seen that there are NOTIFICATION and OPEN messages, followed by subsequent KEEPALIVE messages after the routes have been re-established (refer to Figure 6).

Notification messages is sent by the BGP system when an error condition is detected, and after the message is sent, the BGP session and the TCP connection between the BGP systems is closed. Notification messages consist of the BGP header plus the error code and subcode, and data that describes the error.

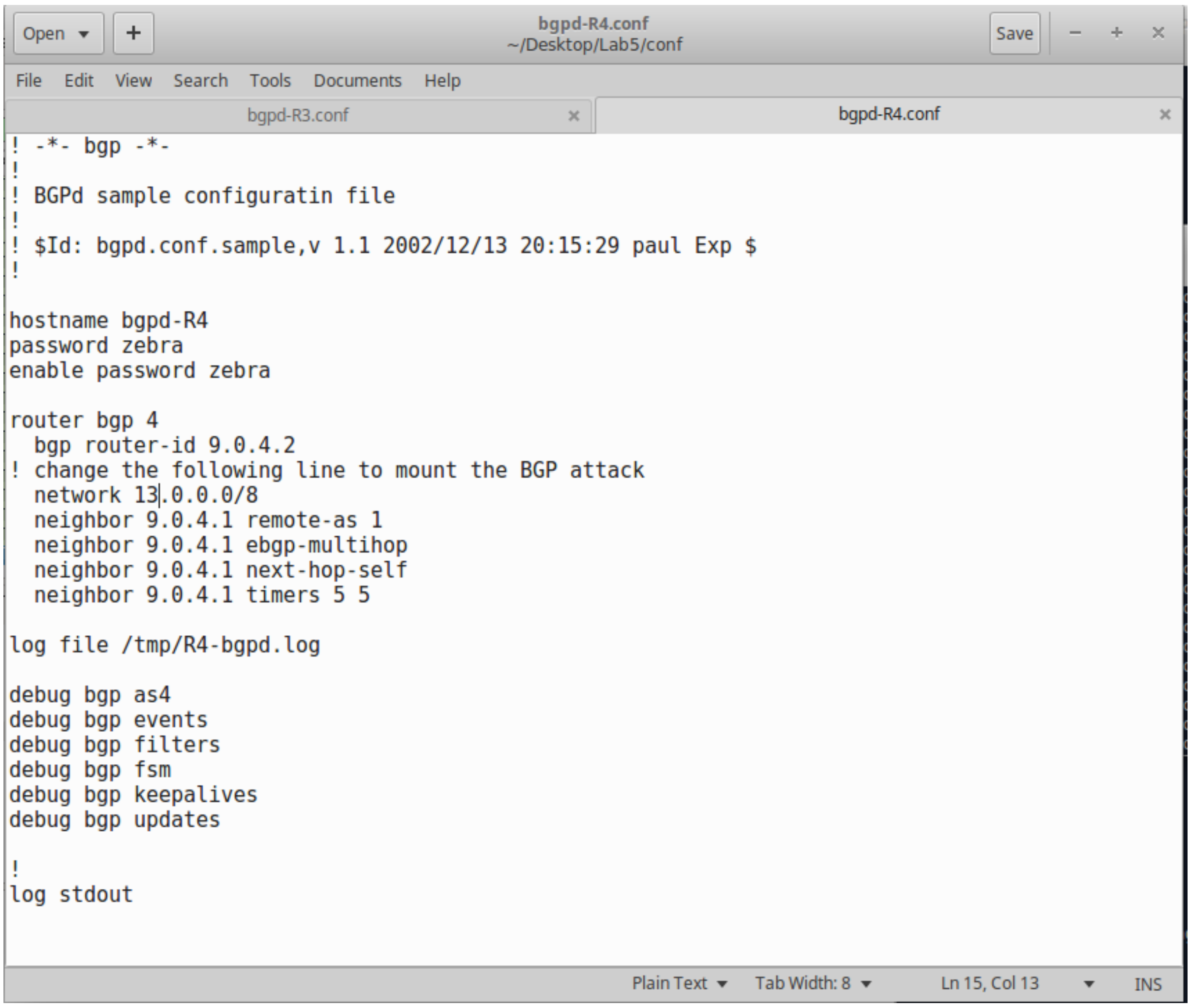
BGP open messages are exchanged after a TCP connection is established between the two BGP systems. Once the connection is established, the two systems can exchange BGP messages and data traffic.

BGP systems exchange keepalive messages to determine whether a link or host has failed or is no longer available. Keepalive messages are exchanged often enough so that the hold timer does not expire. These messages consist only of the BGP header.

Tan Shun Yu 1001171 ISTD
Nigel Leong 1001095 ISTD

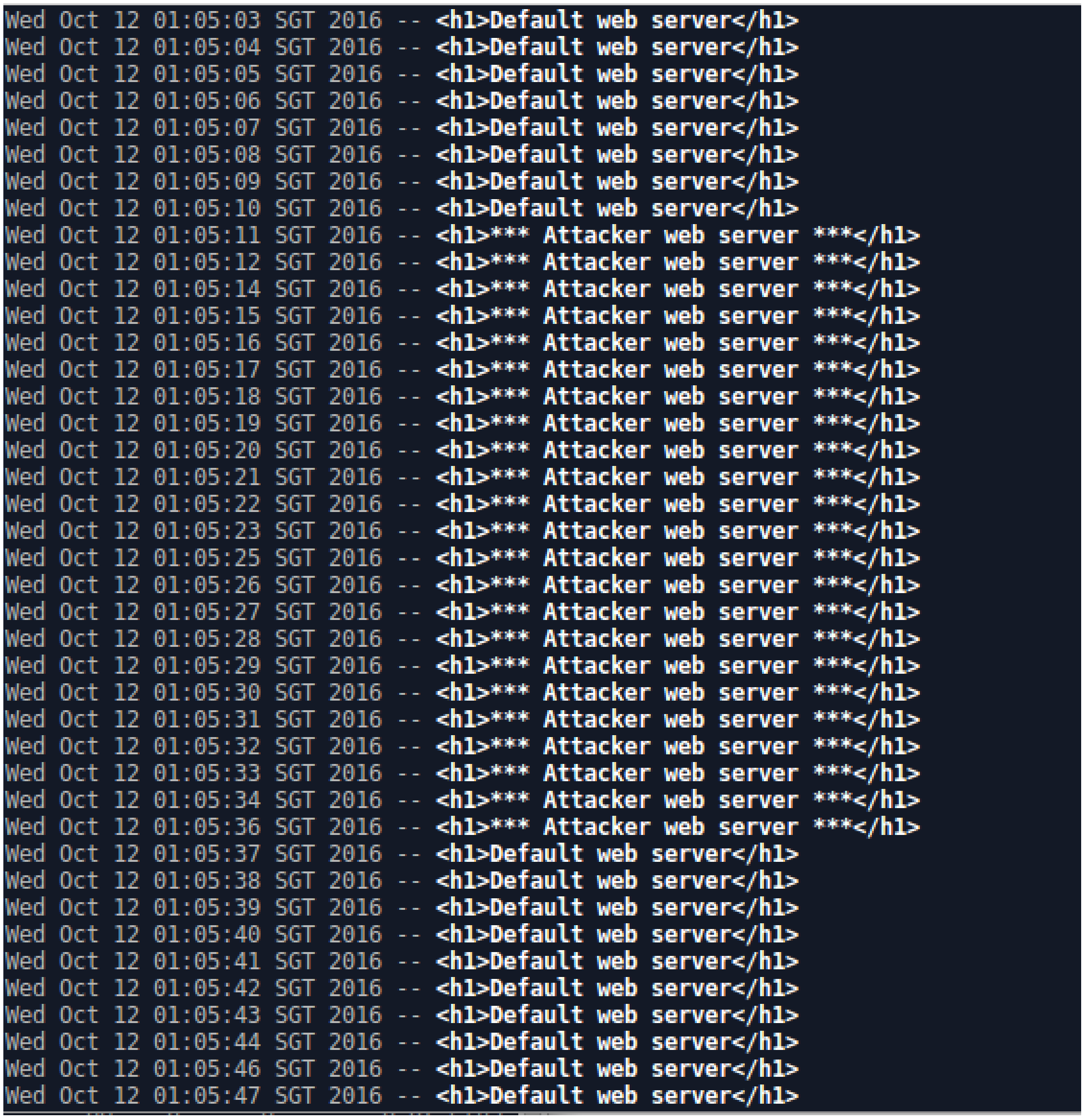
• **Describe in detail what happened when you started the attack on BGP.**

In order to start the attack, I first modified the bgpd-R4.conf file to change its network parameter from 14.0.0.0/8 to 13.0.0.0/8 (refer to Figure 7 below), which the latter is the webserver we want to disguise as.



```
! -*- bgp -*-
!
! BGPd sample configuratin file
!
! $Id: bgpd.conf.sample,v 1.1 2002/12/13 20:15:29 paul Exp $
!

hostname bgpd-R4
password zebra
enable password zebra

router bgp 4
  bgp router-id 9.0.4.2
! change the following line to mount the BGP attack
  network 13.0.0.0/8
  neighbor 9.0.4.1 remote-as 1
  neighbor 9.0.4.1 ebgp-multihop
  neighbor 9.0.4.1 next-hop-self
  neighbor 9.0.4.1 timers 5 5

log file /tmp/R4-bgpd.log

debug bgp as4
debug bgp events
debug bgp filters
debug bgp fsm
debug bgp keepalives
debug bgp updates

!
log stdout
```

Figure 7: Modified bgpd-R4.conf file

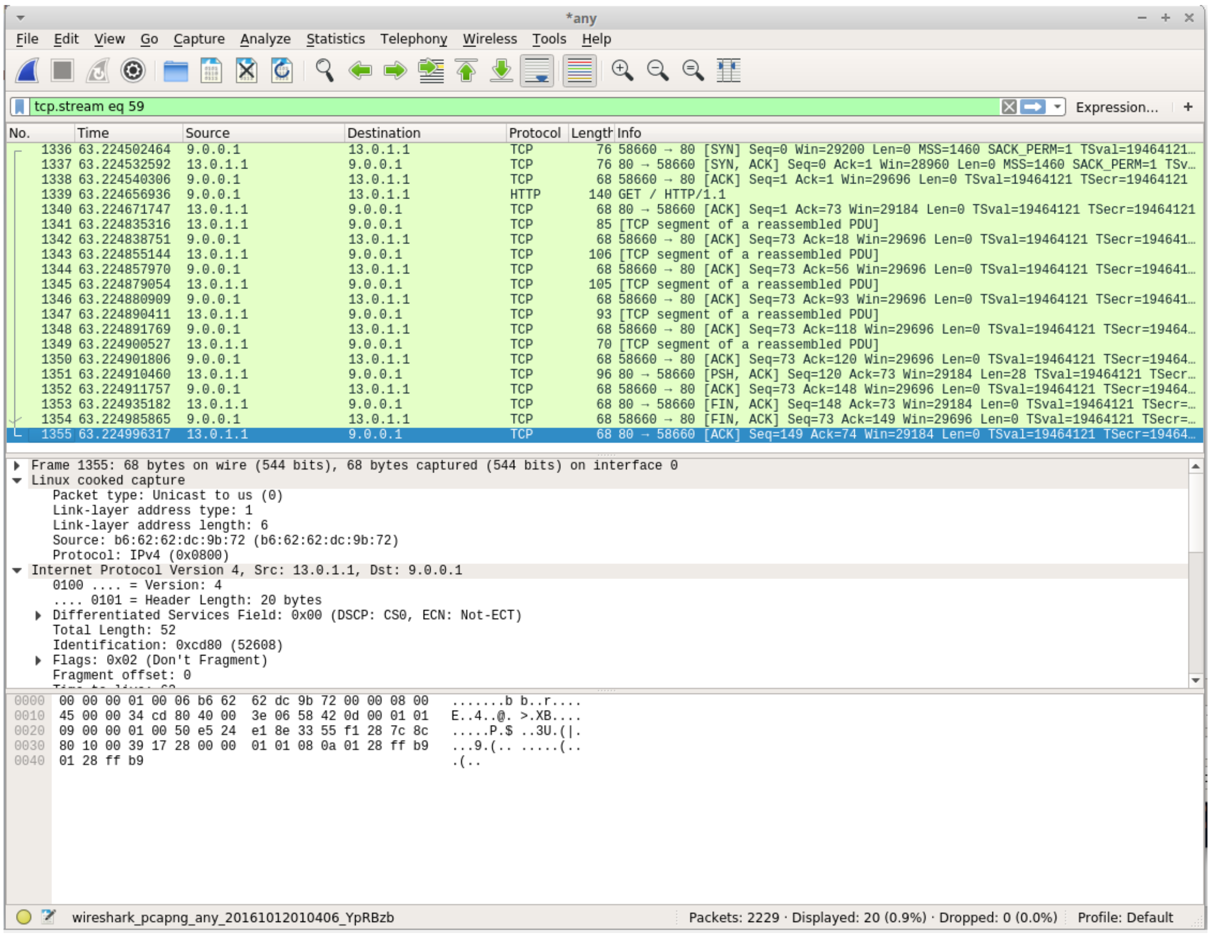


```
Wed Oct 12 01:05:03 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:04 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:05 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:06 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:07 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:08 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:09 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:10 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:11 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:12 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:14 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:15 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:16 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:17 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:18 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:19 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:20 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:21 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:22 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:23 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:25 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:26 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:27 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:28 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:29 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:30 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:31 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:32 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:33 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:34 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:36 SGT 2016 -- <h1>*** Attacker web server ***</h1>
Wed Oct 12 01:05:37 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:38 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:39 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:40 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:41 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:42 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:43 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:44 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:46 SGT 2016 -- <h1>Default web server</h1>
Wed Oct 12 01:05:47 SGT 2016 -- <h1>Default web server</h1>
```

Figure 8: website.sh script running

I then ran the provided website script using *./website.sh R1*, which left it continuously contacting a webserver on 13.0.1.1 from R1. It started to return ‘Default web server’, which is expected since I haven’t started the attack. Then, I ran *./start_rogue.sh* to launch the attack. The website results now return ‘*** Attacker web server ***’ (coming from R4) instead of ‘Default web server’ (coming from R3). Refer to Figure 8 above.

Tan Shun Yu 1001171 ISTD
Nigel Leong 1001095 ISTD


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

tcp.stream eq 59

```
No.  Time         Source     Destination  Protocol Length Info
1336 63.224502464 9.0.0.1    13.0.1.1     TCP       76 58660 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=19464121…
1337 63.224532592 13.0.1.1   9.0.0.1      TCP       76 80 → 58660 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSv…
1338 63.224540306 9.0.0.1    13.0.1.1     TCP       68 58660 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=19464121 TSecr=19464121
1339 63.224656936 9.0.0.1    13.0.1.1     HTTP     140 GET / HTTP/1.1
1340 63.224671747 13.0.1.1   9.0.0.1      TCP       68 80 → 58660 [ACK] Seq=1 Ack=73 Win=29184 Len=0 TSval=19464121 TSecr=19464121
1341 63.224835316 13.0.1.1   9.0.0.1      TCP       85 [TCP segment of a reassembled PDU]
1342 63.224838751 9.0.0.1    13.0.1.1     TCP       68 58660 → 80 [ACK] Seq=73 Ack=18 Win=29696 Len=0 TSval=19464121 TSecr=194641…
1343 63.224855144 13.0.1.1   9.0.0.1      TCP      106 [TCP segment of a reassembled PDU]
1344 63.224857970 9.0.0.1    13.0.1.1     TCP       68 58660 → 80 [ACK] Seq=73 Ack=56 Win=29696 Len=0 TSval=19464121 TSecr=194641…
1345 63.224879054 13.0.1.1   9.0.0.1      TCP      105 [TCP segment of a reassembled PDU]
1346 63.224880909 9.0.0.1    13.0.1.1     TCP       68 58660 → 80 [ACK] Seq=73 Ack=93 Win=29696 Len=0 TSval=19464121 TSecr=194641…
1347 63.224890411 13.0.1.1   9.0.0.1      TCP       93 [TCP segment of a reassembled PDU]
1348 63.224891769 9.0.0.1    13.0.1.1     TCP       68 58660 → 80 [ACK] Seq=73 Ack=118 Win=29696 Len=0 TSval=19464121 TSecr=19464…
1349 63.224900527 13.0.1.1   9.0.0.1      TCP       70 [TCP segment of a reassembled PDU]
1350 63.224901806 9.0.0.1    13.0.1.1     TCP       68 58660 → 80 [ACK] Seq=73 Ack=120 Win=29696 Len=0 TSval=19464121 TSecr=19464…
1351 63.224910460 13.0.1.1   9.0.0.1      TCP       96 80 → 58660 [PSH, ACK] Seq=120 Ack=73 Win=29184 Len=28 TSval=19464121 TSecr…
1352 63.224911757 9.0.0.1    13.0.1.1     TCP       68 58660 → 80 [ACK] Seq=73 Ack=148 Win=29696 Len=0 TSval=19464121 TSecr=19464…
1353 63.224935182 13.0.1.1   9.0.0.1      TCP       68 80 → 58660 [FIN, ACK] Seq=148 Ack=73 Win=29184 Len=0 TSval=19464121 TSecr=…
1354 63.224985865 9.0.0.1    13.0.1.1     TCP       68 58660 → 80 [FIN, ACK] Seq=73 Ack=149 Win=29696 Len=0 TSval=19464121 TSecr=…
1355 63.224996317 13.0.1.1   9.0.0.1      TCP       68 80 → 58660 [ACK] Seq=149 Ack=74 Win=29184 Len=0 TSval=19464121 TSecr=19464…
```

```
▸ Frame 1355: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▾ Linux cooked capture
    Packet type: Unicast to us (0)
    Link-layer address type: 1
    Link-layer address length: 6
    Source: b6:62:62:dc:9b:72 (b6:62:62:dc:9b:72)
    Protocol: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 13.0.1.1, Dst: 9.0.0.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0xcd80 (52608)
  ▸ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
```

```
0000  00 00 00 01 00 06 b6 62  62 dc 9b 72 00 00 08 00   .......b b..r....
0010  45 00 00 34 cd 80 40 00  3e 06 58 42 0d 00 01 01   E..4..@. >.XB....
0020  09 00 00 01 00 50 e5 24  e1 8e 33 55 f1 28 7c 8c   .....P.$ ..3U.(|.
0030  80 10 00 39 17 28 00 00  01 01 08 0a 01 28 ff b9   ...9.(.. .....(..
0040  01 28 ff b9                                        .(..
```

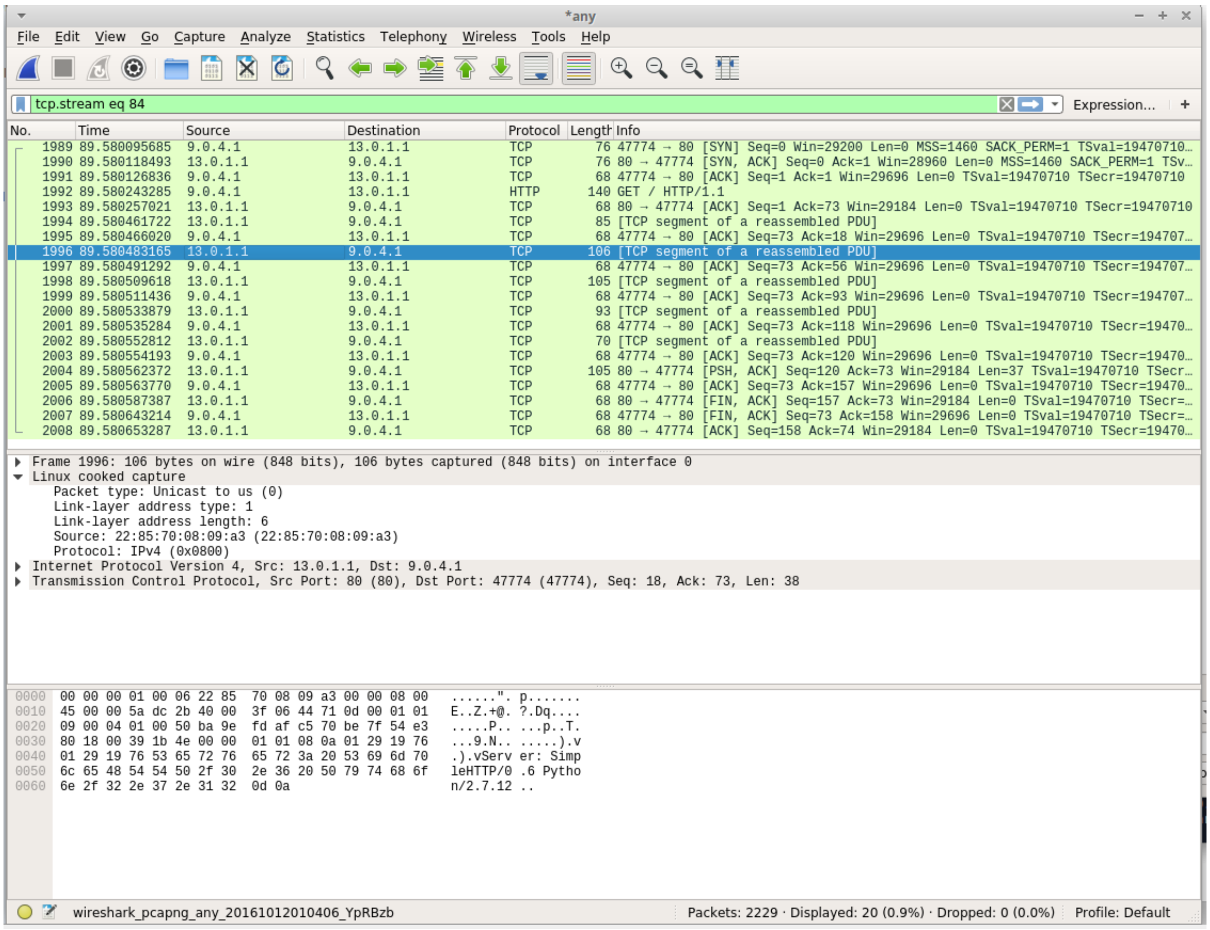
wireshark_pcapng_any_20161012010406_YpRBzb   Packets: 2229 · Displayed: 20 (0.9%) · Dropped: 0 (0.0%)   Profile: Default

Figure 9: Wireshark packets showing default web server packets

By running Wireshark alongside the attack, I was able to capture and observe the network traffic.

Before I launched the attack, I followed the TCP stream and found out that the network traffic was as desired by the user, with the packets being forwarded from 9.0.0.1 at R1 (refer to Figure 2) to the real destination 13.0.1.1 at AS3. It can also be observed that the source MAC address is b6:62:62:dc:9b:72. Also, I can see from the TCP stream that the message being sent back to R1 was Default web server. Refer to Figure 9 above.

Tan Shun Yu 1001171 ISTD
Nigel Leong 1001095 ISTD


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

```
tcp.stream eq 84

No.  Time          Source      Destination  Protocol Length Info
1989 89.580095685  9.0.4.1     13.0.1.1     TCP       76 47774 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=19470710…
1990 89.580118493  13.0.1.1    9.0.4.1      TCP       76 80 → 47774 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSv…
1991 89.580126836  9.0.4.1     13.0.1.1     TCP       68 47774 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=19470710 TSecr=19470710
1992 89.580243285  9.0.4.1     13.0.1.1     HTTP     140 GET / HTTP/1.1
1993 89.580257021  13.0.1.1    9.0.4.1      TCP       68 80 → 47774 [ACK] Seq=1 Ack=73 Win=29184 Len=0 TSval=19470710 TSecr=19470710
1994 89.580461722  13.0.1.1    9.0.4.1      TCP       85 [TCP segment of a reassembled PDU]
1995 89.580466020  9.0.4.1     13.0.1.1     TCP       68 47774 → 80 [ACK] Seq=73 Ack=18 Win=29696 Len=0 TSval=19470710 TSecr=194707…
1996 89.580483165  13.0.1.1    9.0.4.1      TCP      106 [TCP segment of a reassembled PDU]
1997 89.580491292  9.0.4.1     13.0.1.1     TCP       68 47774 → 80 [ACK] Seq=73 Ack=56 Win=29696 Len=0 TSval=19470710 TSecr=194707…
1998 89.580509618  13.0.1.1    9.0.4.1      TCP      105 [TCP segment of a reassembled PDU]
1999 89.580511436  9.0.4.1     13.0.1.1     TCP       68 47774 → 80 [ACK] Seq=73 Ack=93 Win=29696 Len=0 TSval=19470710 TSecr=194707…
2000 89.580533879  13.0.1.1    9.0.4.1      TCP       93 [TCP segment of a reassembled PDU]
2001 89.580535284  9.0.4.1     13.0.1.1     TCP       68 47774 → 80 [ACK] Seq=73 Ack=118 Win=29696 Len=0 TSval=19470710 TSecr=19470…
2002 89.580552812  13.0.1.1    9.0.4.1      TCP       70 [TCP segment of a reassembled PDU]
2003 89.580554193  9.0.4.1     13.0.1.1     TCP       68 47774 → 80 [ACK] Seq=73 Ack=120 Win=29696 Len=0 TSval=19470710 TSecr=19470…
2004 89.580562372  13.0.1.1    9.0.4.1      TCP      105 80 → 47774 [PSH, ACK] Seq=120 Ack=73 Win=29184 Len=37 TSval=19470710 TSecr…
2005 89.580563770  9.0.4.1     13.0.1.1     TCP       68 47774 → 80 [ACK] Seq=73 Ack=157 Win=29696 Len=0 TSval=19470710 TSecr=19470…
2006 89.580587387  13.0.1.1    9.0.4.1      TCP       68 80 → 47774 [FIN, ACK] Seq=157 Ack=73 Win=29184 Len=0 TSval=19470710 TSecr=…
2007 89.580643214  9.0.4.1     13.0.1.1     TCP       68 47774 → 80 [FIN, ACK] Seq=73 Ack=158 Win=29696 Len=0 TSval=19470710 TSecr=…
2008 89.580653287  13.0.1.1    9.0.4.1      TCP       68 80 → 47774 [ACK] Seq=158 Ack=74 Win=29184 Len=0 TSval=19470710 TSecr=19470…

▸ Frame 1996: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
▾ Linux cooked capture
    Packet type: Unicast to us (0)
    Link-layer address type: 1
    Link-layer address length: 6
    Source: 22:85:70:08:09:a3 (22:85:70:08:09:a3)
    Protocol: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 13.0.1.1, Dst: 9.0.4.1
▸ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 47774 (47774), Seq: 18, Ack: 73, Len: 38

0000  00 00 00 01 00 06 22 85  70 08 09 a3 00 00 08 00   ......". p.......
0010  45 00 00 5a dc 2b 40 00  3f 06 44 71 0d 00 01 01   E..Z.+@. ?.Dq....
0020  09 00 04 01 00 50 ba 9e  fd af c5 70 be 7f 54 e3   .....P.. ...p..T.
0030  80 18 00 39 1b 4e 00 00  01 01 08 0a 01 29 19 76   ...9.N.. .....).v
0040  01 29 19 76 53 65 72 76  65 72 3a 20 53 69 6d 70   .).vServ er: Simp
0050  6c 65 48 54 54 50 2f 30  2e 36 20 50 79 74 68 6f   leHTTP/0 .6 Pytho
0060  6e 2f 32 2e 37 2e 31 32  0d 0a                     n/2.7.12 ..

wireshark_pcapng_any_20161012010406_YpRBzb    Packets: 2229 · Displayed: 20 (0.9%) · Dropped: 0 (0.0%)    Profile: Default
```

Figure 10: Wireshark packets showing attacker web server packets

After launching the attack, I followed the TCP stream and realised a few differences that confirmed the success of my attack.

The packets are now being forwarded from 9.0.4.1 (refer to Figure 2), which the direction is headed for the attacker AS4 to the spoofed 13.0.1.1 destination. Also, we can observe the source MAC address is now 22:85:70:08:09:a3, which is different from the previous legitimate MAC address of b6:62:62:dc:9b:72. The TCP Stream shows that the message being sent to R1 is now *** Attacker web server***. Refer to Figure 10 above.