

The topology as you were able to derive it

IP addresses of all routers

Host/IPs in the As

Using if config and show ip bgp on each router to derive the following information

```

Terminal - laura@desktop: ~/Desktop/lab4
File Edit View Terminal Tabs Help
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
mininet>

```

Figure 1: nodes and net information

AS1

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

AS2

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

R1-eth4(R1): 9.0.0.2/24

R1-eth5(R3): 9.0.1.1/24

AS3

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

R1-eth4(R2): 9.0.1.2/24

AS4

AS1: 14.0.0.0/8 (spoofed 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

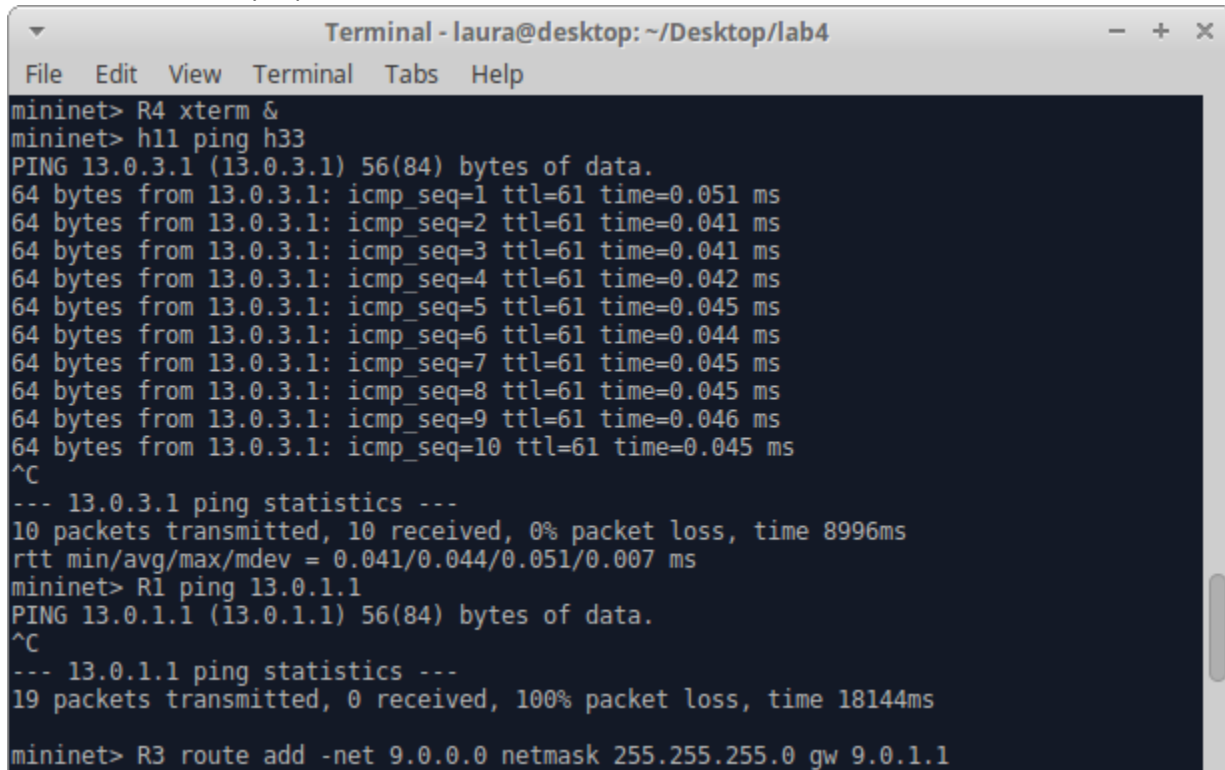
R4-eth4(R2): 9.0.4.2/24

What 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 (9.0.0.0/24)
- 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

Verify using h11 (Figure 2) 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 (also Figure 2) is

unable to receive any replies.

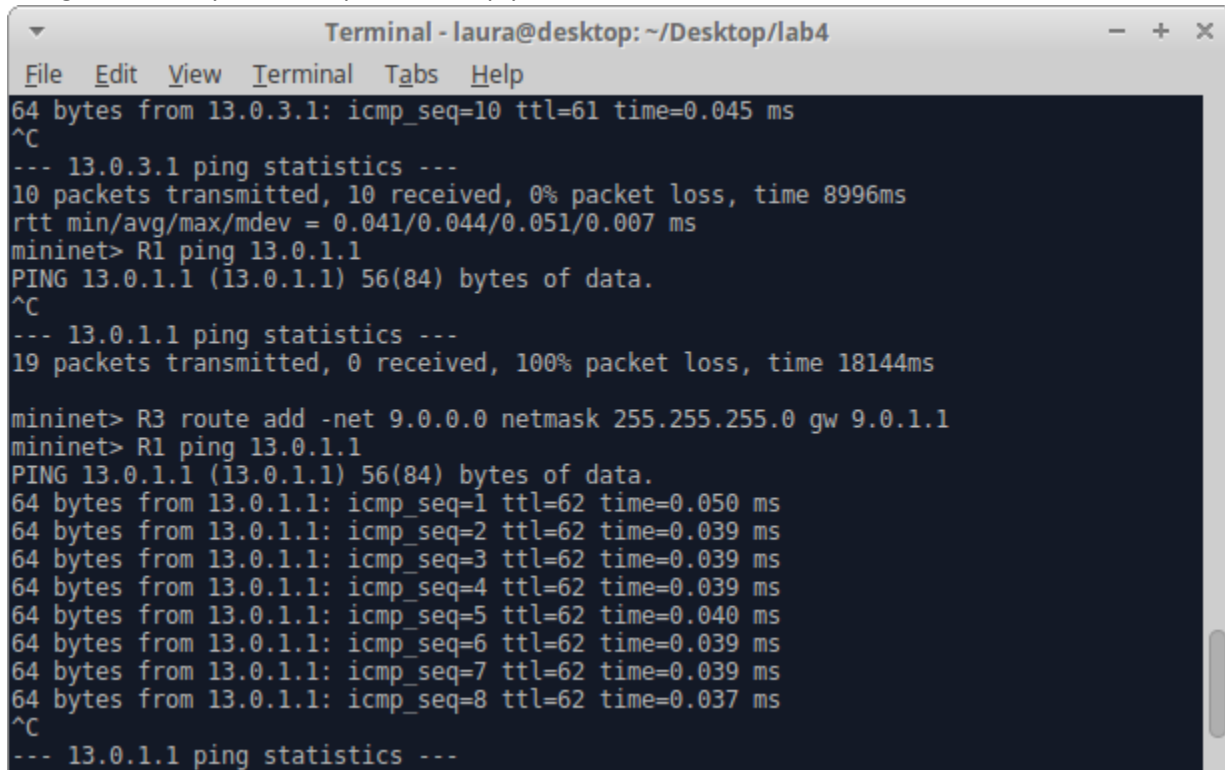
A terminal window titled "Terminal - laura@desktop: ~/Desktop/lab4" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
mininet> R4 xterm &
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=0.051 ms
64 bytes from 13.0.3.1: icmp_seq=2 ttl=61 time=0.041 ms
64 bytes from 13.0.3.1: icmp_seq=3 ttl=61 time=0.041 ms
64 bytes from 13.0.3.1: icmp_seq=4 ttl=61 time=0.042 ms
64 bytes from 13.0.3.1: icmp_seq=5 ttl=61 time=0.045 ms
64 bytes from 13.0.3.1: icmp_seq=6 ttl=61 time=0.044 ms
64 bytes from 13.0.3.1: icmp_seq=7 ttl=61 time=0.045 ms
64 bytes from 13.0.3.1: icmp_seq=8 ttl=61 time=0.045 ms
64 bytes from 13.0.3.1: icmp_seq=9 ttl=61 time=0.046 ms
64 bytes from 13.0.3.1: icmp_seq=10 ttl=61 time=0.045 ms
^C
--- 13.0.3.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8996ms
rtt min/avg/max/mdev = 0.041/0.044/0.051/0.007 ms
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 ---
19 packets transmitted, 0 received, 100% packet loss, time 18144ms
mininet> R3 route add -net 9.0.0.0 netmask 255.255.255.0 gw 9.0.1.1
```

Figure 2: h11 ping h33, R1 unable to contact 13.0.1.1 at first

After configuring R3 to have a routing entry to R1 using *route* (See Figure 3), 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

recognize the way back and provide a reply

A terminal window titled "Terminal - laura@desktop: ~/Desktop/lab4" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal output shows a sequence of network commands and their results. It starts with a ping from 13.0.3.1 to 13.0.1.1, followed by a statistics summary for 13.0.3.1. Then, a ping from 13.0.1.1 to 13.0.1.1 is attempted, resulting in a 100% packet loss. After configuring R3 with a route, a second ping from 13.0.1.1 to 13.0.1.1 is successful, showing eight successful replies with varying times. The window ends with a statistics summary for 13.0.1.1.

```
Terminal - laura@desktop: ~/Desktop/lab4
File Edit View Terminal Tabs Help
64 bytes from 13.0.3.1: icmp_seq=10 ttl=61 time=0.045 ms
^C
--- 13.0.3.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8996ms
rtt min/avg/max/mdev = 0.041/0.044/0.051/0.007 ms
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 ---
19 packets transmitted, 0 received, 100% packet loss, time 18144ms

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=0.050 ms
64 bytes from 13.0.1.1: icmp_seq=2 ttl=62 time=0.039 ms
64 bytes from 13.0.1.1: icmp_seq=3 ttl=62 time=0.039 ms
64 bytes from 13.0.1.1: icmp_seq=4 ttl=62 time=0.039 ms
64 bytes from 13.0.1.1: icmp_seq=5 ttl=62 time=0.040 ms
64 bytes from 13.0.1.1: icmp_seq=6 ttl=62 time=0.039 ms
64 bytes from 13.0.1.1: icmp_seq=7 ttl=62 time=0.039 ms
64 bytes from 13.0.1.1: icmp_seq=8 ttl=62 time=0.037 ms
^C
--- 13.0.1.1 ping statistics ---
```

Figure 3: After configuring R3, now R1 is able to contact 13.0.1.1

Describe the BGP traffic you were able to observe during the reestablishment 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	54.475392223	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	Version	My AS	Hold Time	BGP Identifier	Optional Parameters Length
OPEN Message (1)	4	1	5	9.0.0.1	30

Marker	Length	Type
ffffffffffffffffffffffffffff	19	KEEPALIVE Message (4)

Hex	ASCII
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	-.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	...e... ..
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 00
0090 13 04	..

Figure 4: 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 Keep Alive messages after the routes have been re-established

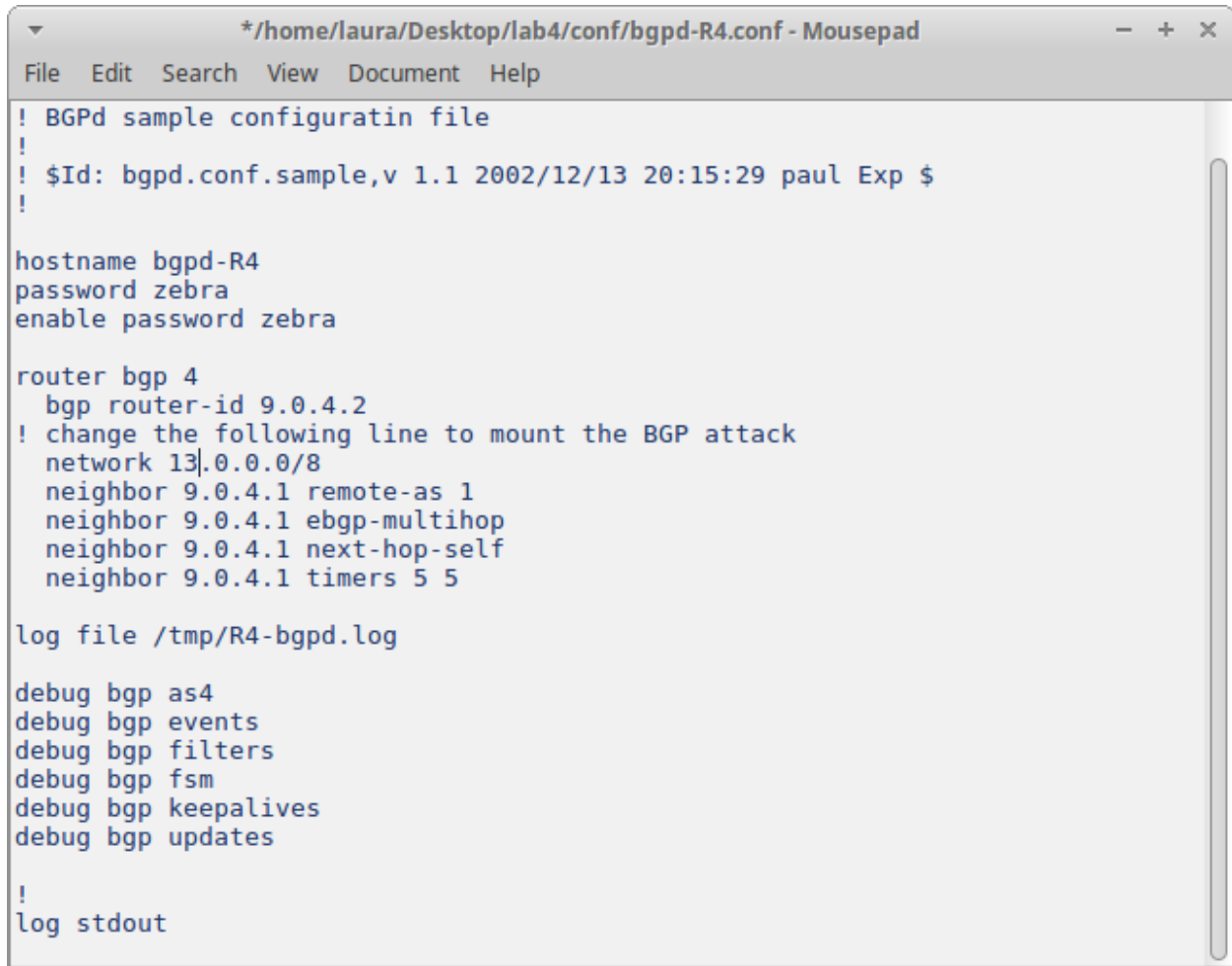
Notification messages are sent by the BGP system when an error condition is detected, and after the message is sent, the BGP session and the TCP connection between the BGP systems is closed. Notification messages consist of the BGP header plus the error code and subcode, and data that describes the error.

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. Keep alive messages are exchanged often enough so that the hold timer does not expire. These messages consist only of the BGP header.

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 5) which the latter is the webserver we want to disguise as.



```
! 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

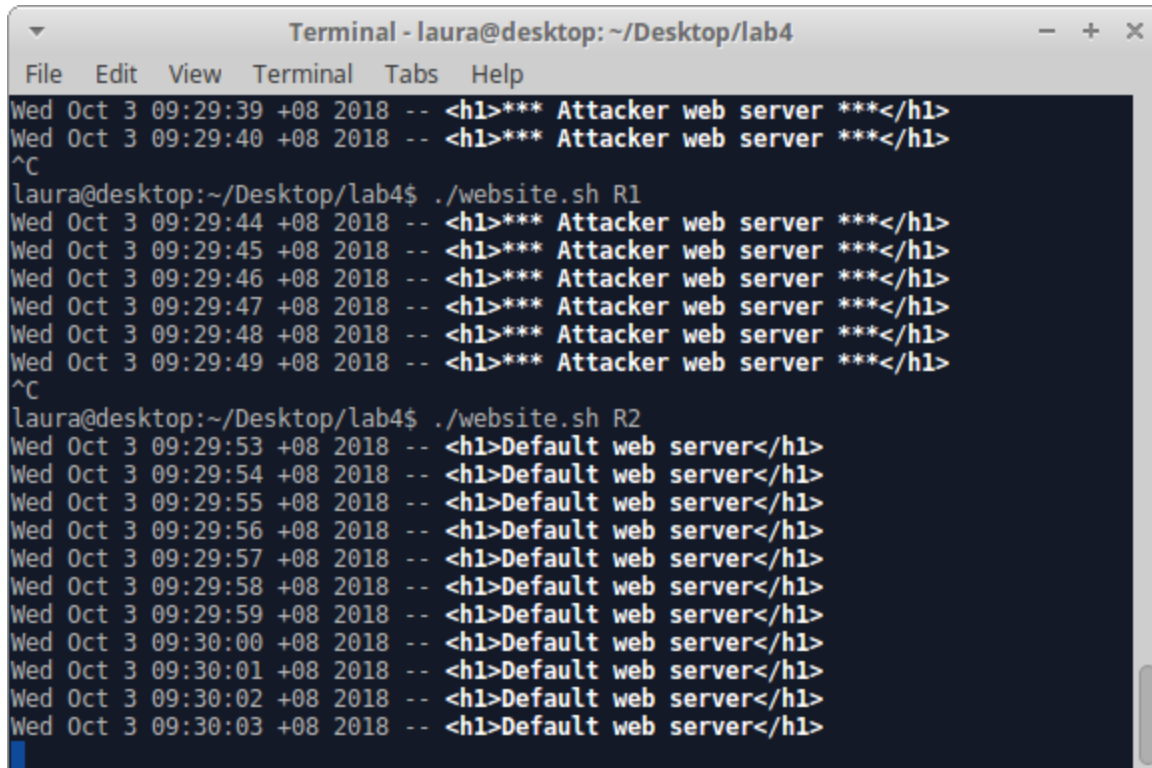
router bgp 4
  bgp router-id 9.0.4.2
! change the following line to mount the BGP attack
  network 13.0.0.0/8
  neighbor 9.0.4.1 remote-as 1
  neighbor 9.0.4.1 ebgp-multihop
  neighbor 9.0.4.1 next-hop-self
  neighbor 9.0.4.1 timers 5 5

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 5: Modified bgpd-R4.conf file



```
Terminal - laura@desktop: ~/Desktop/lab4
File Edit View Terminal Tabs Help
Wed Oct 3 09:29:39 +08 2018 -- <h1>*** Attacker web server ***</h1>
Wed Oct 3 09:29:40 +08 2018 -- <h1>*** Attacker web server ***</h1>
^C
laura@desktop:~/Desktop/lab4$ ./website.sh R1
Wed Oct 3 09:29:44 +08 2018 -- <h1>*** Attacker web server ***</h1>
Wed Oct 3 09:29:45 +08 2018 -- <h1>*** Attacker web server ***</h1>
Wed Oct 3 09:29:46 +08 2018 -- <h1>*** Attacker web server ***</h1>
Wed Oct 3 09:29:47 +08 2018 -- <h1>*** Attacker web server ***</h1>
Wed Oct 3 09:29:48 +08 2018 -- <h1>*** Attacker web server ***</h1>
Wed Oct 3 09:29:49 +08 2018 -- <h1>*** Attacker web server ***</h1>
^C
laura@desktop:~/Desktop/lab4$ ./website.sh R2
Wed Oct 3 09:29:53 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:29:54 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:29:55 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:29:56 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:29:57 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:29:58 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:29:59 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:30:00 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:30:01 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:30:02 +08 2018 -- <h1>Default web server</h1>
Wed Oct 3 09:30:03 +08 2018 -- <h1>Default web server</h1>
```

Figure 6: 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 returned 'Default web server' which is expected since I haven't started the attack yet. Then, I ran `./start_rogue.sh` to launch the attack. The website results now return '***Attacker web server***' (from R4) instead of Default web server (R3) See Figure 6.

*R1-eth5

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
58	18.005206916	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
59	18.005216499	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
60	19.005424769	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
61	19.006047863	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
62	19.006055649	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
63	20.005718399	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
64	20.006134341	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
65	20.006141896	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
66	21.005947293	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
67	21.006193365	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
68	21.006199497	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
69	22.006212984	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
70	22.006270527	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
71	22.006277420	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
72	23.006514027	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
73	23.006601288	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
74	23.006609459	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
75	24.006812637	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
76	24.006901477	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
77	24.006909831	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
78	25.007110821	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
79	25.007202811	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
80	25.007211429	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
81	26.007432837	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
82	26.007512547	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
83	26.007519690	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
84	27.007694802	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
85	27.007776817	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
86	27.007784173	9.0.4.1	9.0.4.2	TCP	66	179 → 38028 [ACK]...
87	28.007964207	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message

▶ Frame 59: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

▼ Ethernet II, Src: 4a:68:ab:be:73:1a (4a:68:ab:be:73:1a), Dst: f2:d0:35:56:0b:ce (f2:d0:35:56:0b:ce)

▶ Destination: f2:d0:35:56:0b:ce (f2:d0:35:56:0b:ce)

▶ Source: 4a:68:ab:be:73:1a (4a:68:ab:be:73:1a)

Type: IPv4 (0x0800)

▶ Internet Protocol Version 4, Src: 9.0.4.1, Dst: 9.0.4.2

▼ Transmission Control Protocol, Src Port: 179, Dst Port: 38028, Seq: 362, Ack: 362, Len: 0

Source Port: 179

Destination Port: 38028

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 362 (relative sequence number)

Acknowledgment number: 362 (relative ack number)

Header Length: 32 bytes

▶ Flags: 0x010 (ACK)

Window size value: 57

[Calculated window size: 57]

[Window size scaling factor: -1 (unknown)]

Checksum: 0x1a29 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

0000 f2 d0 35 56 0b ce 4a 68 ab be 73 1a 08 00 45 c0 ..5V..Jh..s...E.

0010 00 34 91 fc 40 00 ff 06 cf 04 09 00 04 01 09 00 .4..@...

0020 04 02 00 b3 94 8c 53 10 79 cc d9 97 f3 58 80 10S. y....X..

0030 00 39 1a 29 00 00 01 01 08 0a 00 11 08 bd 00 11 .9.)....

0040 08 bd ..

Source Hardware Address (eth.src), 6 bytes Packets: 633 · Displayed: 633 (100.0%) Profile: Default

Figure 7: 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 that the network traffic was as desired by the user, with the packets being forwarded from 9.0.0.1 at R1 to the real destination 13.0.1.1 at AS3. It can also be observed that the source MAC address is 4a:68:ab:be:73:1a. Also, I can see from the TCP stream that the message being sent back to R1 was default web server.

*R1-eth5

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
2579	112.093041437	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
2580	112.093051942	9.0.4.2	9.0.4.1	TCP	66	37220 → 179 [ACK]...
2581	112.093060506	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
2582	112.093064856	9.0.4.1	9.0.4.2	TCP	66	179 → 37220 [ACK]...
2583	112.189354212	9.0.4.1	13.0.1.1	TCP	74	56564 → 80 [SYN] ...
2584	112.189375254	13.0.1.1	9.0.4.1	TCP	74	80 → 56564 [SYN] ...
2585	112.189382734	9.0.4.1	13.0.1.1	TCP	66	56564 → 80 [ACK] ...
2586	112.189415686	9.0.4.1	13.0.1.1	HTTP	138	GET / HTTP/1.1
2587	112.189420234	13.0.1.1	9.0.4.1	TCP	66	80 → 56564 [ACK] ...
2588	112.189589406	13.0.1.1	9.0.4.1	TCP	83	[TCP segment of a...
2589	112.189592000	9.0.4.1	13.0.1.1	TCP	66	56564 → 80 [ACK] ...
2590	112.189617914	13.0.1.1	9.0.4.1	TCP	104	[TCP segment of a...
2591	112.189618950	9.0.4.1	13.0.1.1	TCP	66	56564 → 80 [ACK] ...
2592	112.189634799	13.0.1.1	9.0.4.1	TCP	103	[TCP segment of a...
2593	112.189635606	9.0.4.1	13.0.1.1	TCP	66	56564 → 80 [ACK] ...
2594	112.189653306	13.0.1.1	9.0.4.1	TCP	91	[TCP segment of a...
2595	112.189654907	9.0.4.1	13.0.1.1	TCP	66	56564 → 80 [ACK] ...
2596	112.189672033	13.0.1.1	9.0.4.1	TCP	68	[TCP segment of a...
2597	112.189675358	9.0.4.1	13.0.1.1	TCP	66	56564 → 80 [ACK] ...
2598	112.189683998	13.0.1.1	9.0.4.1	HTTP	103	Continuation
2599	112.189685049	9.0.4.1	13.0.1.1	TCP	66	56564 → 80 [ACK] ...
2600	112.189693176	13.0.1.1	9.0.4.1	TCP	66	80 → 56564 [FIN, ...
2601	112.189700274	9.0.4.1	13.0.1.1	TCP	66	56564 → 80 [FIN, ...
2602	112.189706798	13.0.1.1	9.0.4.1	TCP	66	80 → 56564 [ACK]...
2603	113.094170869	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
2604	113.094173096	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
2605	113.094181262	9.0.4.1	9.0.4.2	TCP	66	179 → 37220 [ACK]...
2606	113.131912108	9.0.4.2	9.0.4.1	TCP	66	37220 → 179 [ACK]...
2607	113.222802750	9.0.4.1	13.0.1.1	TCP	74	56566 → 80 [SYN] ...
2608	113.222824549	13.0.1.1	9.0.4.1	TCP	74	80 → 56566 [SYN, ...
2609	113.222824549	9.0.4.1	13.0.1.1	TCP	66	56566 → 80 [ACK]...

▶ Frame 2602: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

▼ Ethernet II, Src: f2:d0:35:56:0b:ce (f2:d0:35:56:0b:ce), Dst: 4a:68:ab:be:73:1a (4a:68:ab:be:73:1a)

▶ Destination: 4a:68:ab:be:73:1a (4a:68:ab:be:73:1a)

▶ Source: f2:d0:35:56:0b:ce (f2:d0:35:56:0b:ce)

Type: IPv4 (0x0800)

▶ Internet Protocol Version 4, Src: 13.0.1.1, Dst: 9.0.4.1

▼ Transmission Control Protocol, Src Port: 80, Dst Port: 56564, Seq: 158, Ack: 74, Len: 0

Source Port: 80

Destination Port: 56564

[Stream index: 109]

[TCP Segment Len: 0]

Sequence number: 158 (relative sequence number)

Acknowledgment number: 74 (relative ack number)

Header Length: 32 bytes

▶ Flags: 0x010 (ACK)

Window size value: 57

[Calculated window size: 29184]

[Window size scaling factor: 512]

Checksum: 0x1b28 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

0000 4a 68 ab be 73 1a f2 d0 35 56 0b ce 08 00 45 00 Jh..s...5V...E.

0010 00 34 0f 71 40 00 3f 06 11 52 0d 00 01 01 09 00 .4.q@.?.R.....

0020 04 01 00 50 dc f4 a8 9d 10 46 aa 49 af 63 80 10 ...P....F.I.C..

0030 00 39 1b 28 00 00 01 01 08 0a 00 10 00 c0 00 10 .9.(.....

0040 00 c0 ..

Source Hardware Address (eth.src), 6 bytes Packets: 3860 · Displayed: 3860 (100.0%) Profile: Default

Figure 8: Wireshark packets showing attacker web server packets

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

The packets are now being forwarded from 9.0.4.1, 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 f2:d0:35:56:0b:ce, which is different from the previous legitimate MAC address of 4a:68:ab:be:73:1a. The TCP stream shows that the message being sent to R1 is now ***Attacker web server***