Person-in-the-Middle via ARP Spoofing

Kaeden Berg

a.

Kali MAC

08:00:27:3a:73:53

b.

Kali Eth IP

10.0.2.15/24

c.

Meta MAC

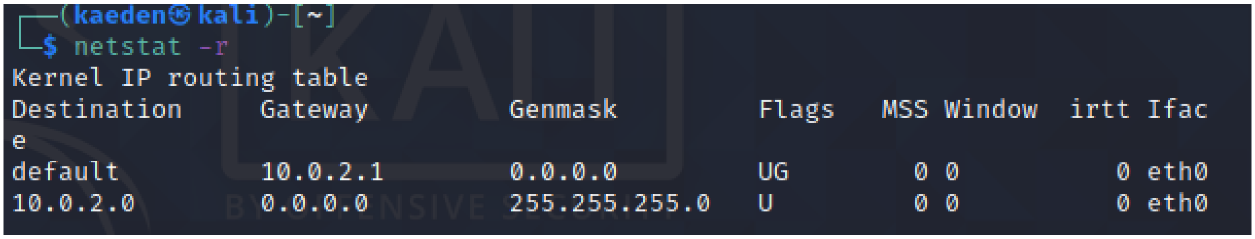
08:00:27:e5:1a:5b

d.

Meta Eth IP

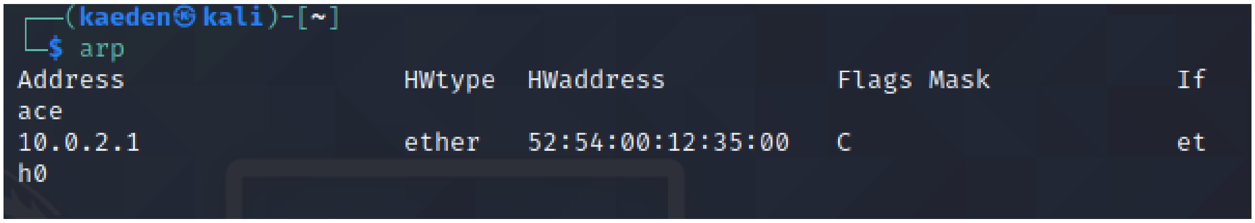
10.0.2.4/24

e.



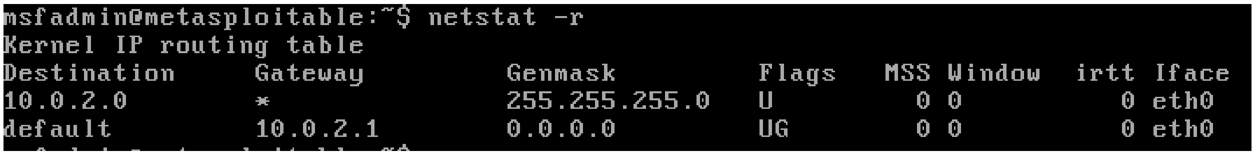
```
┌──(kaeden㉿kali)-[~]
└─$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Ifac
e
default         10.0.2.1        0.0.0.0         UG        0 0          0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
```

f.



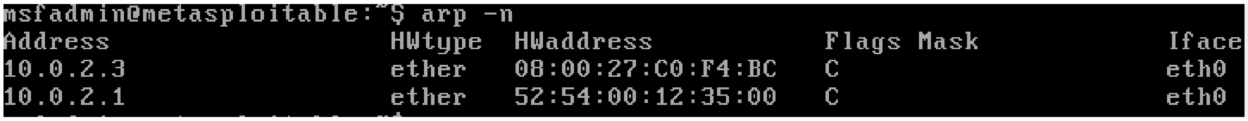
```
┌──(kaeden㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            If
ace
10.0.2.1                 ether   52:54:00:12:35:00   C                     et
h0
```

g.



```
msfadmin@metasploitable:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.2.0        *               255.255.255.0   U         0 0          0 eth0
default         10.0.2.1        0.0.0.0         UG        0 0          0 eth0
```

h.



```
msfadmin@metasploitable:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.3                 ether   08:00:27:C0:F4:BC   C                     eth0
10.0.2.1                 ether   52:54:00:12:35:00   C                     eth0
```

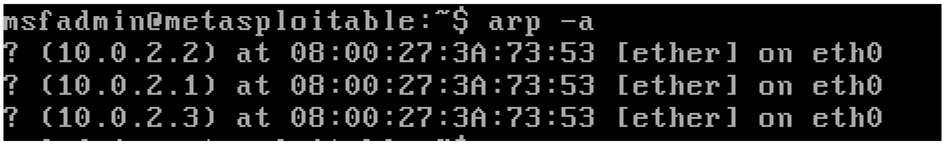
i.

52.54.00.12.35.00, this appears to be the gateway to the default route, which is used to forward packets who have a destination address not in the routing table.

j.

I did not see any captured packets on wireshark, I did get an http response on Metasploitable.

l.



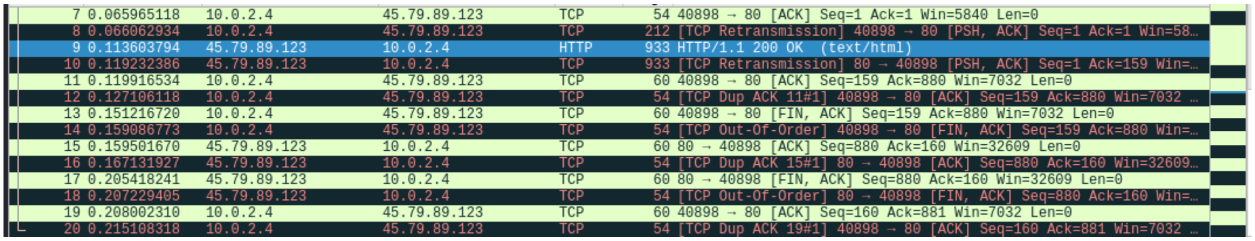
```
msfadmin@metasploitable:~$ arp -a
? (10.0.2.2) at 08:00:27:3A:73:53 [ether] on eth0
? (10.0.2.1) at 08:00:27:3A:73:53 [ether] on eth0
? (10.0.2.3) at 08:00:27:3A:73:53 [ether] on eth0
```

Metasploitable's ARP cache states that every IP address in the network, (10.0.2.1-10.0.2.3) can be reached through Kali's MAC address.

m.

Metasploitable will send the packet to 08:00:27:3a:73:53, which is Kali's MAC address. It will send the packet to this address because Kali is advertising that every IP address in the network can be reached through Kali.

n.



```
 7 0.065965118  10.0.2.4      45.79.89.123  TCP   54 40898 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0
 8 0.066062934  10.0.2.4      45.79.89.123  TCP  212 [TCP Retransmission] 40898 → 80 [PSH, ACK] Seq=1 Ack=1 Win=58…
 9 0.113603794  45.79.89.123  10.0.2.4      HTTP 933 HTTP/1.1 200 OK  (text/html)
10 0.119232386  45.79.89.123  10.0.2.4      TCP  933 [TCP Retransmission] 80 → 40898 [PSH, ACK] Seq=1 Ack=159 Win=…
11 0.119916534  10.0.2.4      45.79.89.123  TCP   60 40898 → 80 [ACK] Seq=159 Ack=880 Win=7032 Len=0
12 0.127106118  10.0.2.4      45.79.89.123  TCP   54 [TCP Dup ACK 11#1] 40898 → 80 [ACK] Seq=159 Ack=880 Win=7032 …
13 0.151216720  10.0.2.4      45.79.89.123  TCP   60 40898 → 80 [FIN, ACK] Seq=159 Ack=880 Win=7032 Len=0
14 0.159086773  10.0.2.4      45.79.89.123  TCP   54 [TCP Out-Of-Order] 40898 → 80 [FIN, ACK] Seq=159 Ack=880 Win=…
15 0.159501670  45.79.89.123  10.0.2.4      TCP   60 80 → 40898 [ACK] Seq=880 Ack=160 Win=32609 Len=0
16 0.167131927  45.79.89.123  10.0.2.4      TCP   54 [TCP Dup ACK 15#1] 80 → 40898 [ACK] Seq=880 Ack=160 Win=32609…
17 0.205418241  45.79.89.123  10.0.2.4      TCP   60 80 → 40898 [FIN, ACK] Seq=880 Ack=160 Win=32609 Len=0
18 0.207229405  45.79.89.123  10.0.2.4      TCP   54 [TCP Out-Of-Order] 80 → 40898 [FIN, ACK] Seq=880 Ack=160 Win=…
19 0.208002310  10.0.2.4      45.79.89.123  TCP   60 40898 → 80 [ACK] Seq=160 Ack=881 Win=7032 Len=0
20 0.215108318  10.0.2.4      45.79.89.123  TCP   54 [TCP Dup ACK 19#1] 40898 → 80 [ACK] Seq=160 Ack=881 Win=7032 …
```

I do see an HTTP response on Metasploitable, and I am able to see the entire conversation on Kali, http responses and all. However, there appears to be a lot of retransmissions and out of order packets. This happens if I retry the capture, I'm a little curious why this is.

p.

So Kali used ettercap to generate false ARP announcements to send to Metasploitable, which say that any IP addresses on the network can be reached at Kali. It also announces to the other devices on the network that Metasploitable can be reached at Kali as well. Because ARP doesn't authenticate these announcements, Metasploitable (and the other devices) believed these announcements. Metasploitable updated its ARP table, and any outgoing traffic is sent to Kali. Likewise, any traffic going to Metasploitable is also routed through Kali.

q.

To detect ARP spoofing, we would have to authenticate the announcements. Ideally we have each device transmit its MAC and IP to a trusted server, that would save these pairs. This would let us check malicious ARP announcements, as if they don't match our authoritative server, then we know ARP spoofing is happening. IP addresses can be dynamic, and can change, so we would have to have some way to let these changes happen. Another way to detect it would be to look for multiple IP addresses associated with a single MAC address. We wouldn't want to cause a false positive, so if the company wanted to associate those IP addresses with the same device, then our detecting software should allow for a specific device to be ignored in this detection.