COMP3010 Lab

The challenge: Honeypot - Answers

6. Stage 1 – Discovering network architecture

1. What are the IP addresses of the hosts sitting on the honeynet's network segment? What can you tell about them by their MAC addresses?

   ARP traffic is local to the network segment. Analyze arp traffic using tcpdump with the arp bpf filter. You will only find two systems answering to arp queries:

   arp reply 192.168.1.254 is-at O:50:56:cO:O:2 arp
   reply 192.168.1.3 is-at O:50:56:6b:O:76

   The MAC address prefix of both arp messages is 00:50:56, a prefix that has been assigned to VMware virtual network interface cards. Unless the system owners have purposely changed the MAC addresses to disguise the real underlying platform, we can say that both systems are running on top of VMware virtual machines, in which case we would be in the presence of a Virtual Honeynet.

2. What is the IP address of the honeypot?

   If you analyze the traffic, you will see that most of the traffic is directed to the 192.168.1.3 address, so this must be the honeypot. The 192.168.1.254 address only receives syslog traffic, hence it must be the syslog server.

3. What is the IP address of the local gateway(s)?

   You can identify the gateway by taking a look at the traffic with non-local IP addresses. These packets will have the MAC address of the gateway. You can isolate that particular traffic running tcpdump with the following filter:

   tcpdump –nn -e -r challenge.pcap ip and not dst net 192.168.0.0/16 and not dst host 255.255.255.255

   Note that we used the -e flag in order to be able to see layer 2 (link layer) information, specifically MAC addresses.

   The destination MAC address is 0:50:56:cO:0:2. Now you can then use the arp traffic to find out which IP address corresponds to that particular MAC address with the following filter: tcpdump -nn -e -r challenge.pcap arp and ether src 0:50:56:cO:0:2

   The output is 192.168.1.254

4. Can you identify any router(s) in addition to the local gateway? Can you estimate how many hops away it is?

If you take a look at typical router icmp error messages (such us destination unreachable- network unreachable or destination unreachable-port unreachable), and exclude those coming from the gateway:

tcpdump -nn -r challenge.pcap "not host 192.168.200.254 and icmp[0] = 3 and (icmp[1] = 0 or icmp[1] = 1)"

You will see that one message stands out:
09:25:26.400249 00:50:56:cO:00:02 > 00:50:56:6b:00:76, etherertype IPv4 (Ox0800), length 70: 195.221.127.214> 192.168.1.3: ICMP: host 145.238.110.68 unreachable, length 36

The IP address 195.221.127.214 is most likely a router. You can estimate the distance to the router by taking a look at the TTL with which that particular packet has arrived to the network. Remember that you can use the -v flag in order to tell tcpdump to show the TTL and IP ID information:

09:25:26.400249 00:50:56:cO:00:02 > 00:50:56:6b:00:76, etherertype IPv4 (Ox0800), length 70: (tos 0x0, ttl 245, id 54727, offset 0, flags [DF], proto: ICMP (1), length 56) 195.221.127.214> 192.168.1.3: ICMP: host 145.238.110.68 unreachable, length 36

As you can see, the arriving TTL is 245. Initial TTLs are typically 64, 128 and 255. So this particular router is probably 9 hops away.

5. What are the IP addresses of the DNS servers used by the honeypot?

   You can run tcpdump with the following filter:

   Tcpdump –nn –r challenge.pcap “src host 192.168.1.3 and udp and dst port 53” The

   result is 62.151.2.8

6. Can you identify NTP traffic? If so, identify the IP address(es) of the NTP server(s)?

   The network Time Protocol has been assigned port 123, both tcp and udp. You can use the following filter to identify ntp traffic:

   Tcpdump –nn –r challenge.pcap “src host 192.168.1.3 and dst port 123” The IP

   addresses of NTP servers are:
   134.214.100.6
   145.238.110.68
   193.49.205.19

7. Stage 2 – Profiling traffic

1. Is there fragmentation activity in the capture?

   The tcpdump filter for this would be:

tcpdump -nn -r challenge.pcap "ip[6] & 0x20 != 0 or ip[6:2] & 0x1fff!= 0" You will find no fragmentation in the dump.

2. What protocols can you find in the traffic (layer 2 & 3 only)?

   If we progressively exclude the standard protocols, arp, icmp, tcp or udp, we will see that there is no traffic left:

   tcpdump -nn -r challenge.pcap not tcp and not udp and not icmp and not arp

   The only protocols in the dump are arp at the link level, ip and icmp at the network level, and tcp and udp at the transport level.

3. What types of ICMP messages can you find in the traffic? Find a way to do it with tcpdump and with snort.

   There is no easy mechanism for extracting the types of ICMP traffic. It would require to iterate several times on the traffic, discarding each time the ICMP messages discovered in the previous iteration:
   tcpdump -nn -r challenge.pcap icmp
   tcpdump -nn -r challenge.pcap "icmp and not (icmp[0]=0 and icmp[1]=0)"
   …

   However, there is an easier way to do it if you used Snort. We can use snort in "log to file" mode (the default), with a bpf filter that extracts icmp traffic only (you will have to change directory in the bootcamp_handson):

   snort -r challenge.pcap -l snort_log icmp

   You will see the classic directory structure by IP address, and inside each of the directories the name of the file will reveal the type of ICMP traffic, so you can identify now which messages are present.

   In summary, there are four types of ICMP messages in the dataset:
   - host unreachable (type 3, code 1)
   - port unreachable (type 3, code 3)
   - echo request (type 8, code 0)
   - echo reply (type 0, code 0).

4. Which ports were open in the honeypot? Which were closed? Can you recognize which well known services are supposedly running on the ones that were open?

   You can easily extract the probed ports using tcpdump with the below filter. tcpdump -nn -r challenge.pcap "dst host 192.168.1.3 and tcp[13] & 0x02 != 0”

5. Can you find evidence of possible backscatter activity (someone spoofing your IP address in TCP connections to third party hosts)? What other explanations could you give for this traffic? What is the most likely one?

   Using the following tcpdump filter will discover the SYN/ACKs coming into our network:

tcpdump -r challenge.pcap -nn "(tcp[13] = 0x12) and not src host 192.168.1.3"

There is one packet that is interesting, as it is completely unrelated to any activity from our part:

07:27:39.025647 IP 200.77.121.251.9775 > 192.168.1.3.27021: S
3285035247:3285035247(0) ack 1 win 0 <mss 1380>

This may be backscatter activity, but it might also be reconnaissance (trying to determine if our host is alive). In this particular case, if this were to be backscatter the third party system would have been probed on port 9775, which does not correspond to any well-known service or backdoor. Additionally, the IP address 200.77.121.251 seems to belong to a home user (customer-COB-121-251.megared.net.mx), which does not seem in principle a juicy target. Both facts seem to support the theory of some kind of scanning activity instead of backscatter traffic.

8. Stage 3 – Identifying attacks

1. Run the traffic through snort to identify attacks. How many different types of attacks did snort catch?

Snort identified the worm activity, but not any system compromises. It did flag error responses from the honeypot though.

2. Can you identify worm activity?

There are a number of alerts that are probably related to worm activity. The most obvious ones are:

From VM:
MS-SQL Worm overflow attempt
MS-SQL Worm propagation attempt
MISC OpenSSL Worm traffic

These packets relate to worm Slammer, propagating over 1434/udp From

3. Can you identify any sessions that correspond to system compromises?

Apart from worm activity, the honeypot got compromised. Although snort did not generate alerts for this activity, it is still possible to pick up the offending traffic through error messages, and analysis from wireshark. For example, you could view Conversations in Wireshark to identify the following suspicious sessions:

200.184.43.197 – 80/tcp, 443/tcp
61.61.123.123 - 80/tcp, 443,tcp

By Following TCP Stream, or explorting objects in wireshark, you can identify and save the files that were transferred in these sessions.