**Part 1:**

VM A: 10.0.2.6

VM B(Zeek): 10.0.2.4

VM C: 10.0.2.5

VM D(being treated as outside attacker): 10.0.2.7

I couldn’t seem to get my personal pc to detect hosts being up on the subnet of my VMs, so to show the steps of the lab out outlined in the lab introduction I will be simulating this outside attacker as VM D but still located on the same subnet.

Outside VM: 192.168.56.1

nmap -sT 10.0.2.0/24 and Zeek VM running tcpdump

A screenshot of a computer

Description automatically generated

Command for SYN Flood Attack on 10.0.2.6 using port 80 which is open according to the nmap scan

sudo netwox 76 -i 10.0.2.6 -p 80 -s raw

The conn.log file provides some really good information about the nmap scan on the subnet, in the image below it shows that the nmap scan was coming from VM D(10.0.2.7) and checking the ports on the 10.0.2.0/24 subnet so each connection attempt is shown in this log for each of the port scans. The highlighted rows show that on 10.0.2.5 port 53 that the response logged is RSTO which means that tcp port is open. Also 10.0.2.6 port 22 has the RSTO response instead of being rejected(REJ) which also shows in the log what ports were found to be open that the attacker now knows about.

A picture containing table

Description automatically generated

The image below is important regarding the nmap scan in relation to the following SYN Flood attack. This is because the scan information logged shows that the attack was able to find on 10.0.2.6 that port 80 was open and potentially vulnerable to a SYN Flood attack on that machine. This is the information that an attack could use to carry out that attack on port 80 or any of the other open tcp ports that the nmap scan found, which the conn.log file shows what IP address the scans were originating from and what machines were found with open ports.

A picture containing calendar

Description automatically generated

The conn.log file also shows the SYN Flood attack taking place as the spoofed IP addresses are different and targeting port 80 to cause a denial of service. If the attacker was not careful, this would be a chance for the Zeek logs to provide some helpful information that a machine with the IP address of 10.0.2.7 was scanning for open ports on the subnet. Then shortly after, a SYN Flood attack started from random IP addresses, this could then potentially be associated with an attack coming from the IP address that was scanning the ports if it was not spoofed. The conn.log contains information of the SYN packets coming through port 80 in mass amounts from the random IP addresses to the machine being affected, 10.0.2.6 which is on the same subnet as the Zeek VM.

Table

Description automatically generated with medium confidence

Looking through the other logs created by Zeek there is not anything additional that stands out in terms of indicating that there was an attack other than in the conn.log. The weird.log, image below, shows that the attacking VM 10.0.2.7 accessed port 80 and the message of bad\_tcp\_checksum could show that there was a potential attack taking place in relation to the attack on port 80, but there is no additional information in terms of the machine that was suffering from the attack.

Graphical user interface, application

Description automatically generated

My thoughts on how effectively Zeek was able to detect any unusual activity is that it did not explicitly point out any information in the logs that I saw. I would’ve expected some information on the SYN Flood Attack to be located in the weird.log file but did not find much in that log file, the other log files also did not contain any information that clearly pointed out an attack from 10.0.2.7 on 10.0.2.6 through port 80 other than the conn.log file. It was very evident with the nmap and the SYN Flood attack in the conn.log file what was happening, but it also required the user to scroll through large amounts of data since every piece of information was logged into that file. A human reader could clearly look at the conn.log file to see that there was an attack taking place on the 10.0.2.6 VM, but I did not see any information automatically separated in the log files by Zeek detecting any anomalies on it’s own. Zeek is very useful in it’s default script that all of the information is logged and a human user could analyze that to see patterns or indications of an attack. However, it was not apparent that Zeek on it’s own was able to differentiate using the default script that an attack was taking place or even that there was anything abnormal taking place that could be considered an anomaly. From this part of the lab my current takeaway is that the default script that Zeek runs when logging all of the information is really useful when it will be sorted through by a human user, but that Zeek on its own as a detection method for anomalous activity did not show that it would log something like a SYN Flood attack as a separate detection from normal traffic.

The only information that I think could also be attributed to the attack is in the capture\_loss.log where the line highlighted shows that there was a high number of acks and gaps at a certain timestamp, but this does not directly associate any other information with an attack.

Shape, rectangle

Description automatically generated