CSE 406

Computer Security Sessional

Final Report on Project No. 1

ARP Cache Poisoning with Man in the Middle Attack

Submitted by-

Raihan Rasheed

Student ID. 1605062

Lab Group. B1

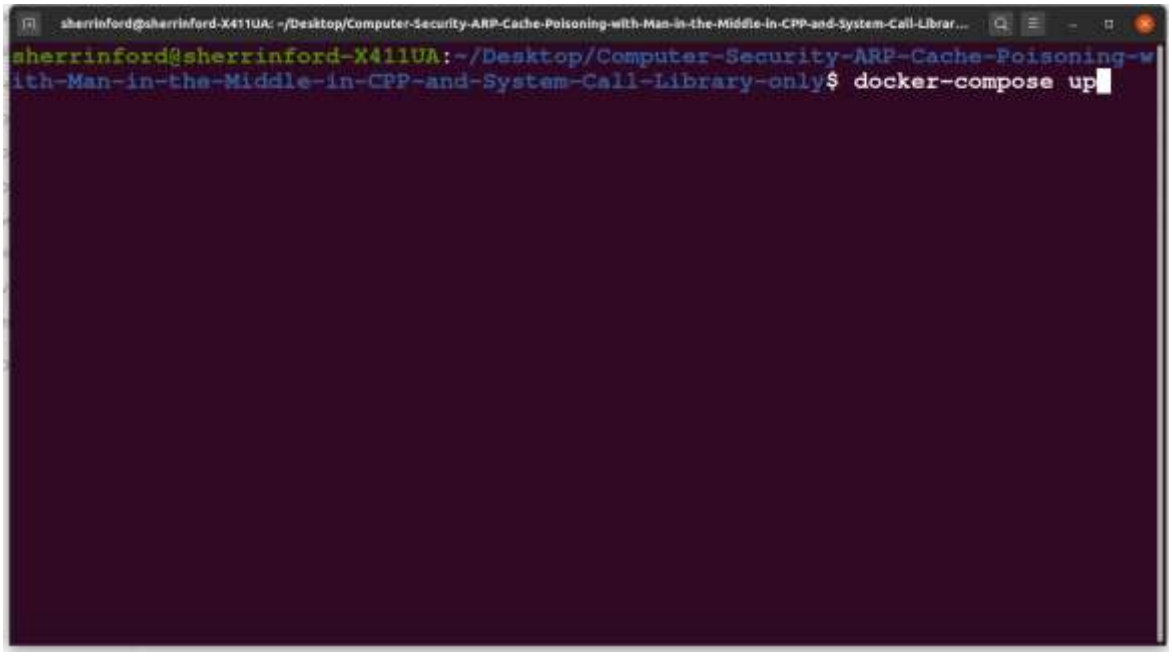
Project Group No. 2

Level 4 Term 1

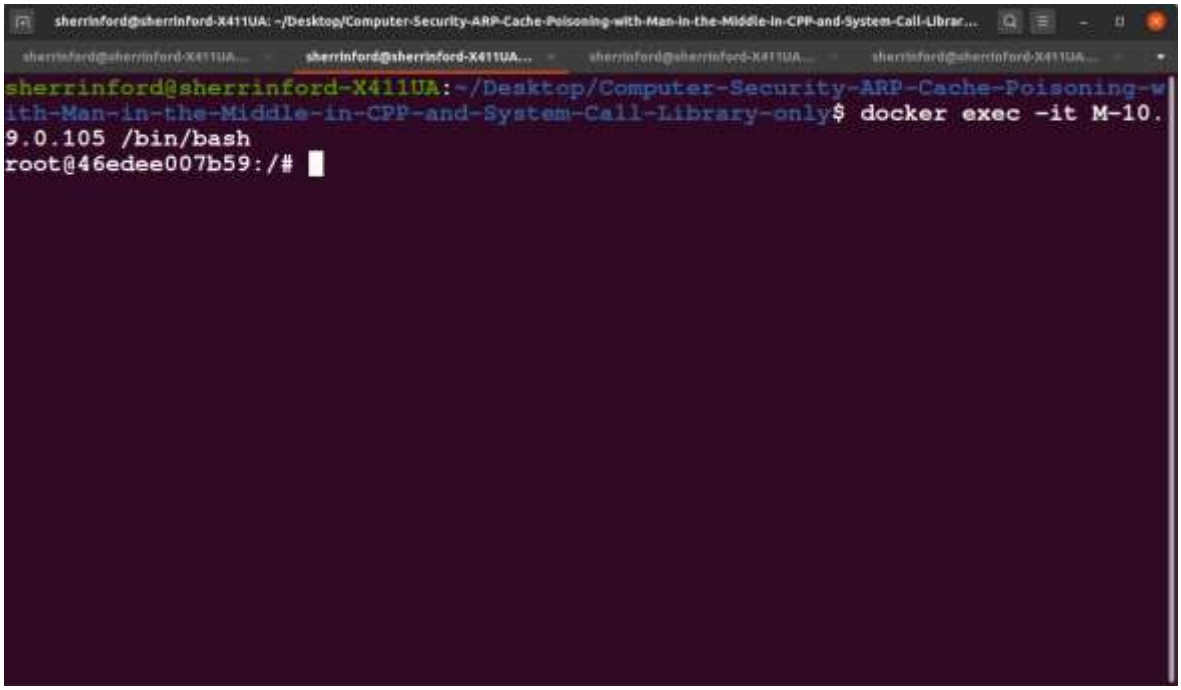
Department of CSE, BUET

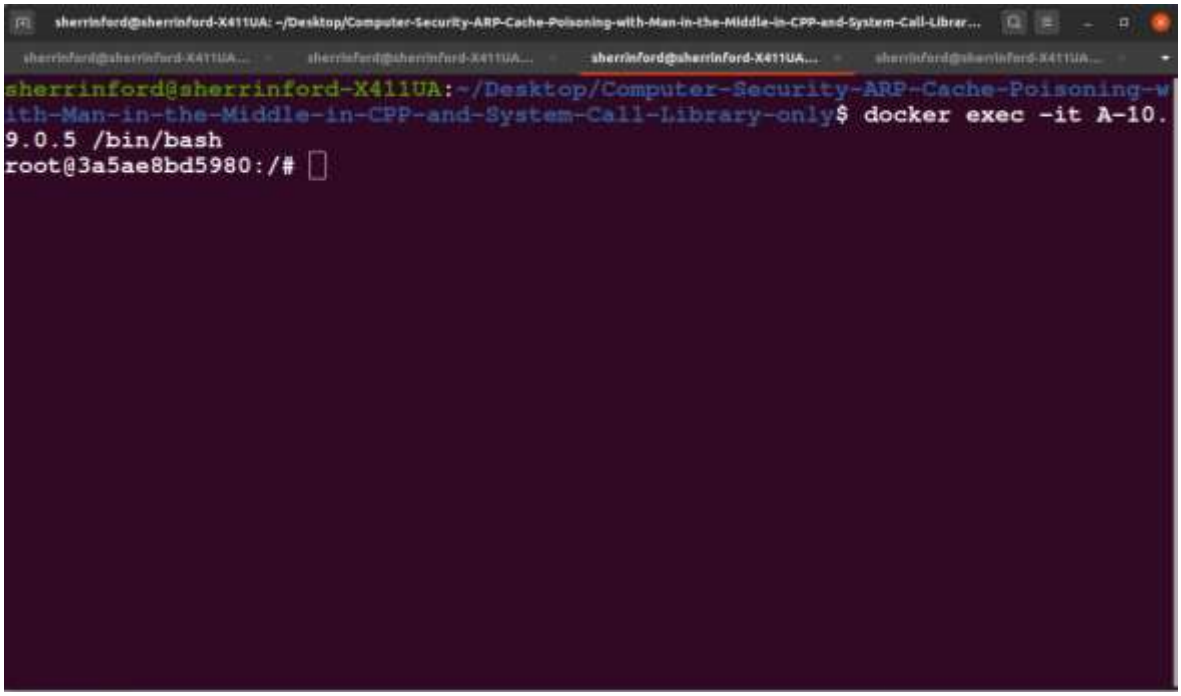
Attack Steps

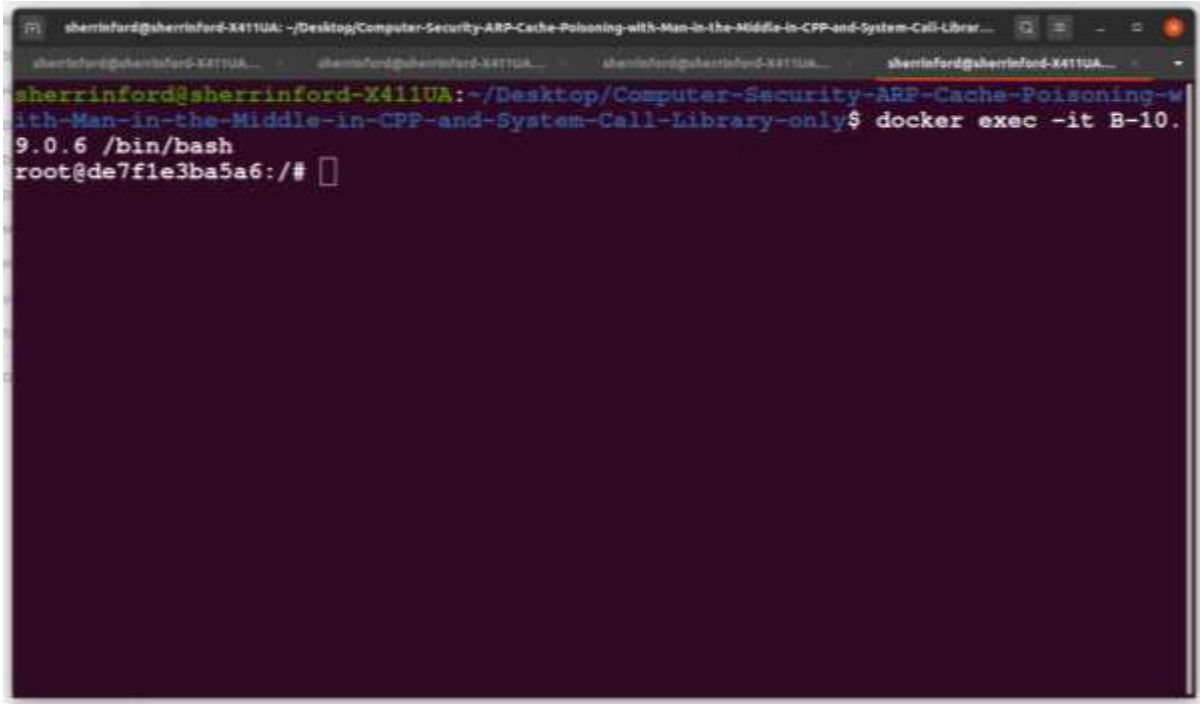
1. Go to project directory and open a terminal from linux and type following command in the picture.



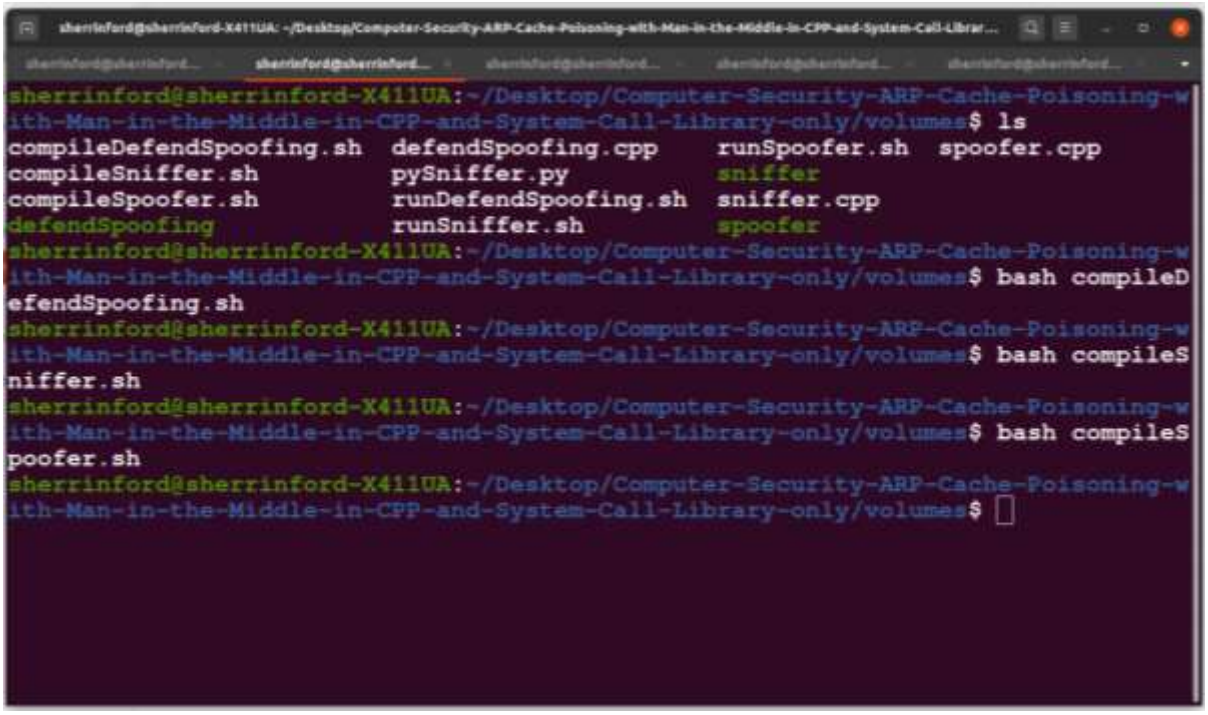
2. Open three other tabs to spawn attacker, host-A, host-B containers respectively. Same process will be applicable if we want additional terminals for respective containers.







3. From host operating system compile all attack codes inside volumes folder. Like below-



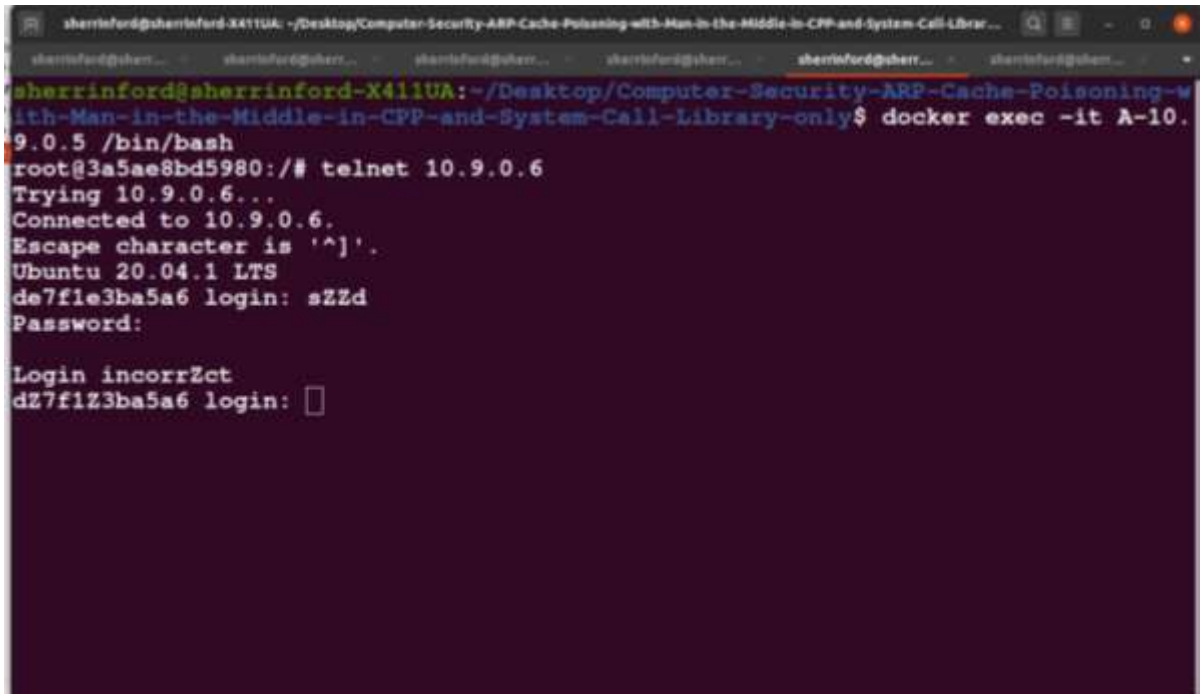
4. Go to an attacker terminal and run arp spoofing code. That will poison host-A and host-B's ARP cache.

```
sherrinford@sherrinford-X411UA: ~/Desktop/Computer-Security-ARP-Cache-Poisoning-with-Man-in-the-Middle-in-CPP-and-System-Call-Librar...
sherrinford@sherrinford...  sherrinford@sherrinford...  sherrinford@sherrinford...  sherrinford@sherrinford...  sherrinford@sherrinford...
sherrinford@sherrinford-X411UA:~/Desktop/Computer-Security-ARP-Cache-Poisoning-w
ith-Man-in-the-Middle-in-CPP-and-System-Call-Library-only$ docker exec -it M-10.
9.0.105 /bin/bash
root@46edee007b59:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr  volumes
root@46edee007b59:/# cd volumes/
root@46edee007b59:/volumes# ls
compileDefendSpoofing.sh  defendSpoofing.cpp    runSpoofer.sh  spoofer.cpp
compileSniffer.sh         pySniffer.py          sniffer
compileSpoofer.sh         runDefendSpoofing.sh  sniffer.cpp
defendSpoofing            runSniffer.sh         spoofer
root@46edee007b59:/volumes# bash runSpoofer.sh
inside bind socket
calling bind
exiting bind
Socket opened successfully
inside craftARPFrame
ARP frame: 02 42 0a 09 00 05 02 42 0a 09 00 69 08 06 00 01 08 00 06 04 00 02 02
42 0a 09 00 69 0a 09 00 06 02 42 0a 09 00 05 0a 09 00 05
exiting crafting
ARP NO. 0
total sent: 42
sendVictim1: 0
```

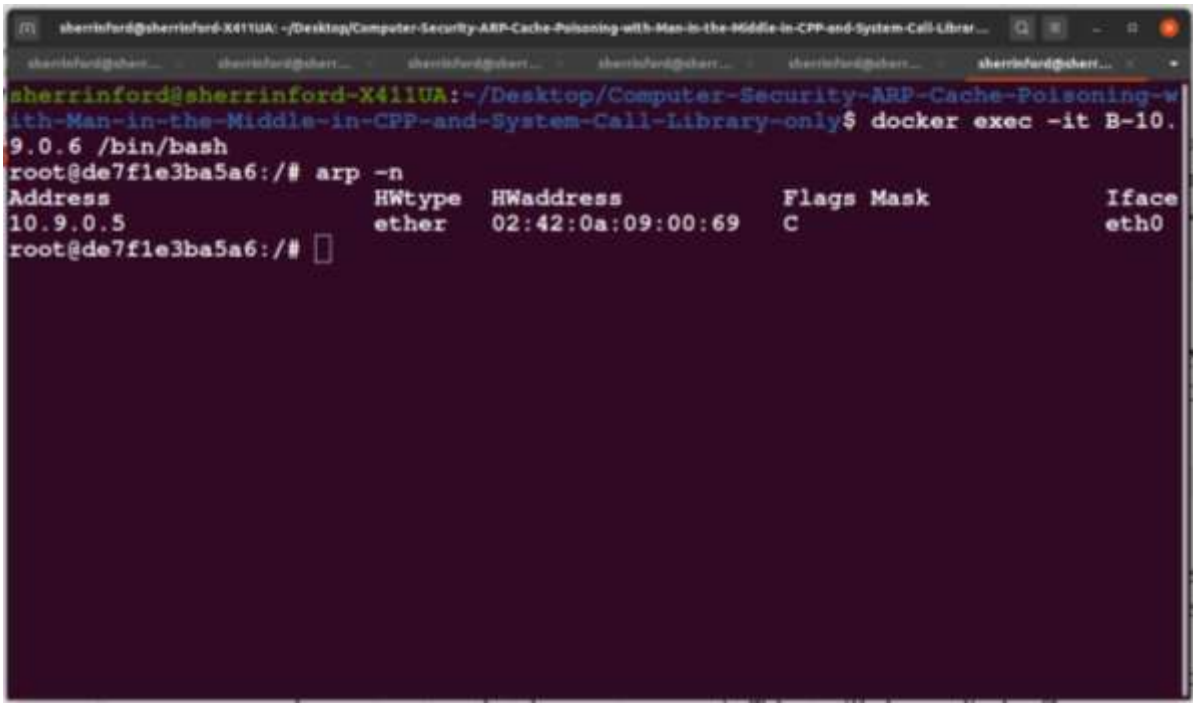
5. Open another attacker terminal. At first stop ip forwarding and then we will run sniffing code like below.

```
sherrinford@sherrinford-X411UA:~/Desktop/Computer-Security-ARP-Cache-Poisoning-w
ith-Man-in-the-Middle-in-CPP-and-System-Call-Library-only$ docker exec -it M-10.
9.0.105 /bin/bash
root@46edee007b59:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr  volumes
root@46edee007b59:/# cd volumes/
root@46edee007b59:/volumes# ls
compileDefendSpoofing.sh  defendSpoofing.cpp    runSpoofer.sh  spoofer.cpp
compileSniffer.sh         pySniffer.py          sniffer
compileSpoofer.sh         runDefendSpoofing.sh  sniffer.cpp
defendSpoofing            runSniffer.sh         spoofer
root@46edee007b59:/volumes# bash runSniffer.sh
inside bind socket
calling bind
exiting bind
Socket opened successfully
1 42
2 42
```

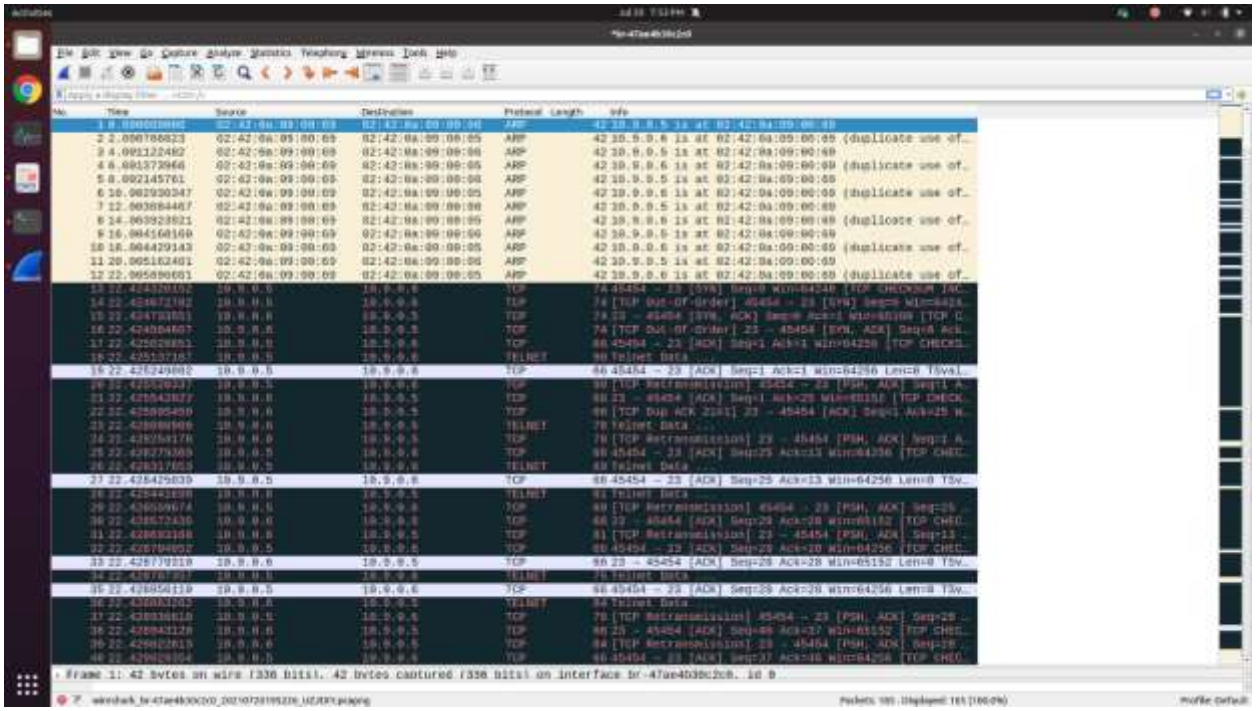
6. To demonstrate MITM in telnet application. We first run telnet command from host-A to host-B's IP address. As we can see when we enter username seed out attack code replace occuerance of letter 'e' with 'Z' hence even though we type seed on our keyboard it shows sZZd on the screen.



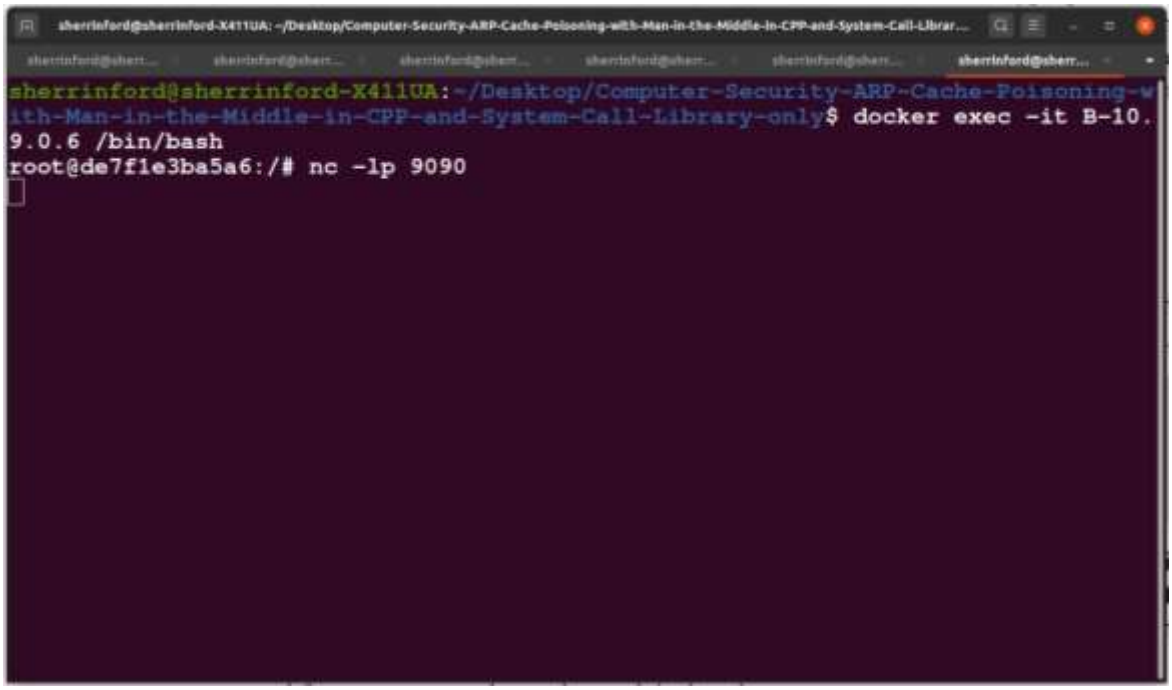
7. To verify that indeed we have intercepted the message we can lookup arp table of host-B. It shows host-A's IP address is bind with hardware address of attacker like expected.



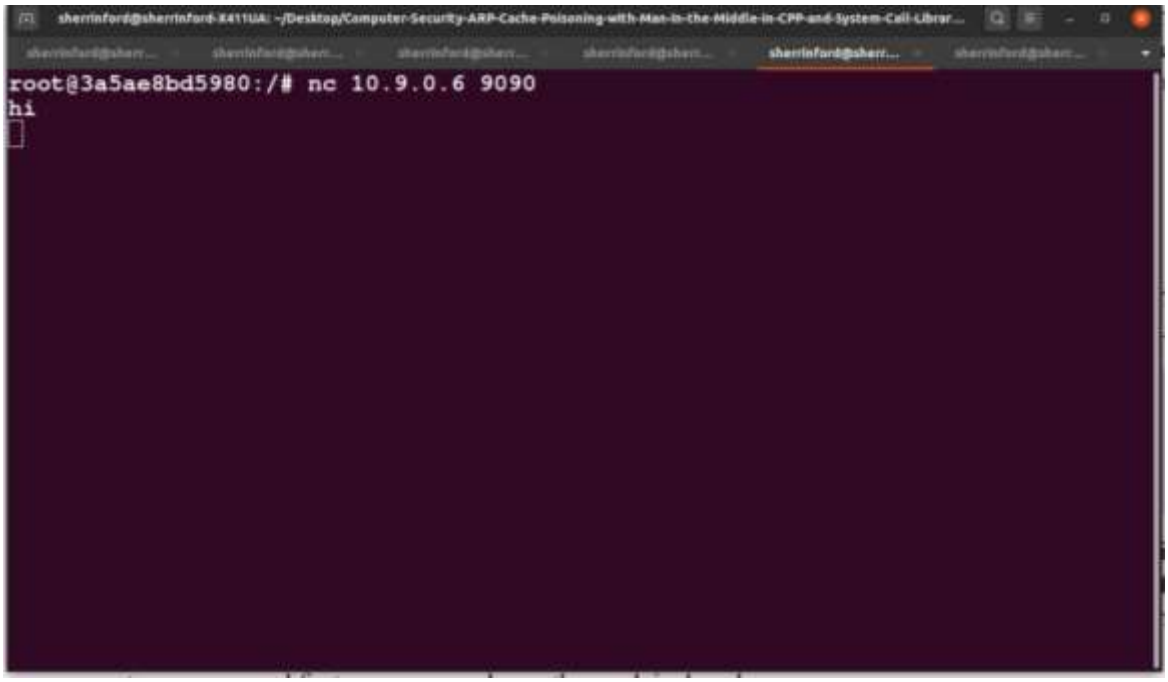
8. We can verify the attack again using wireshark. We have taken a screenshot of docker network interface. As we can see from packet 1 to 12 are spoofed ARP reply. We can then see in 13 and 14 same packet is being passed from host-A to attacker machine and then attacker machine to host-B. For every packet sent from host-A to host-B and vice-versa attacker machine at first intercept it and then forward it with some modification. Here attacker just change any occurrence of letter ‘e’ with ‘Z’.

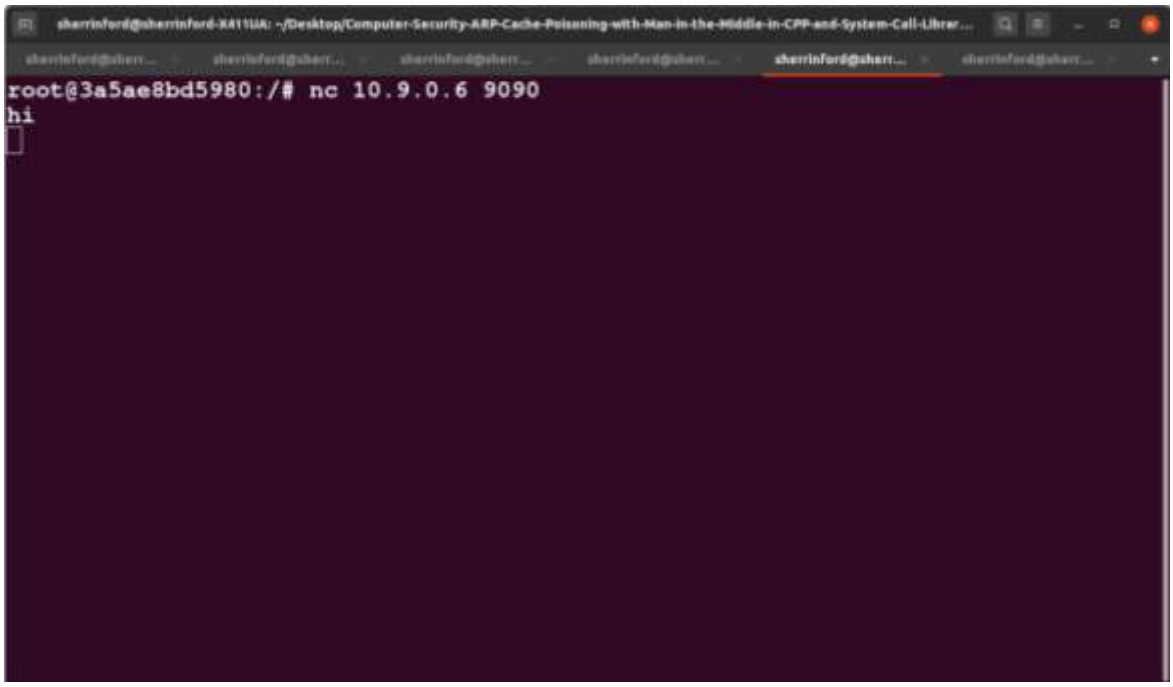


9. Now lets demonstrate netcat attack. We don't need to change or restart anything to demonstrate attack over netcat after we have shown telnet attack. In a host-B terminal run netcat on port 9090 like below-

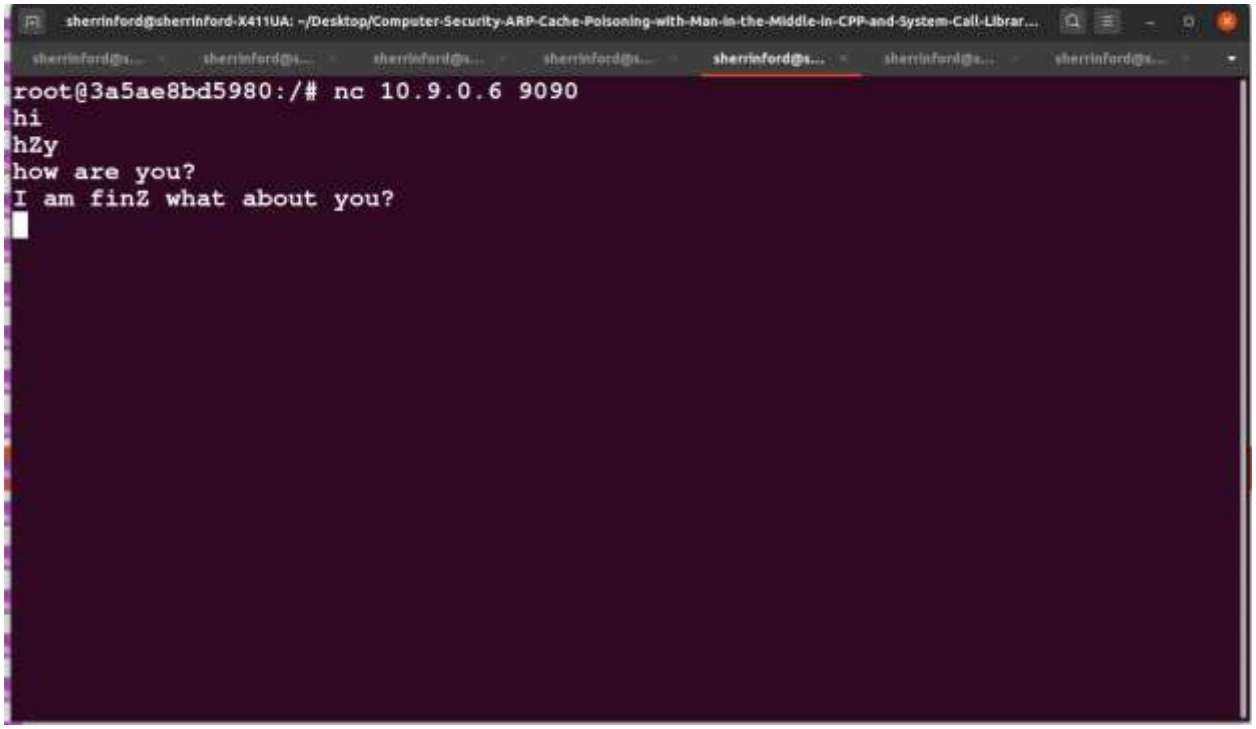


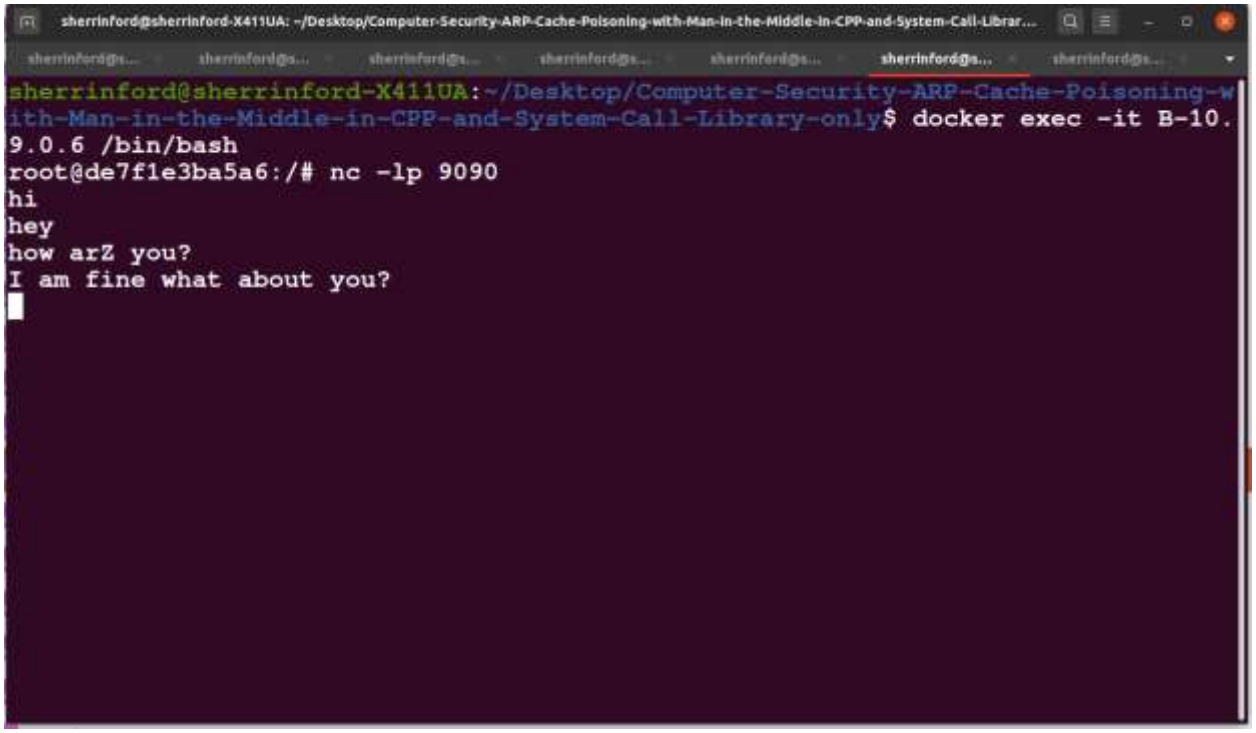
10. Now in host-A try to connect with port 9090 of host-B and send message “hi”. This message will also showup on host-B. This message was transferred unchanged as this message doesn’t have any occurrence of ‘e’ in it. Snapshots will look like below-



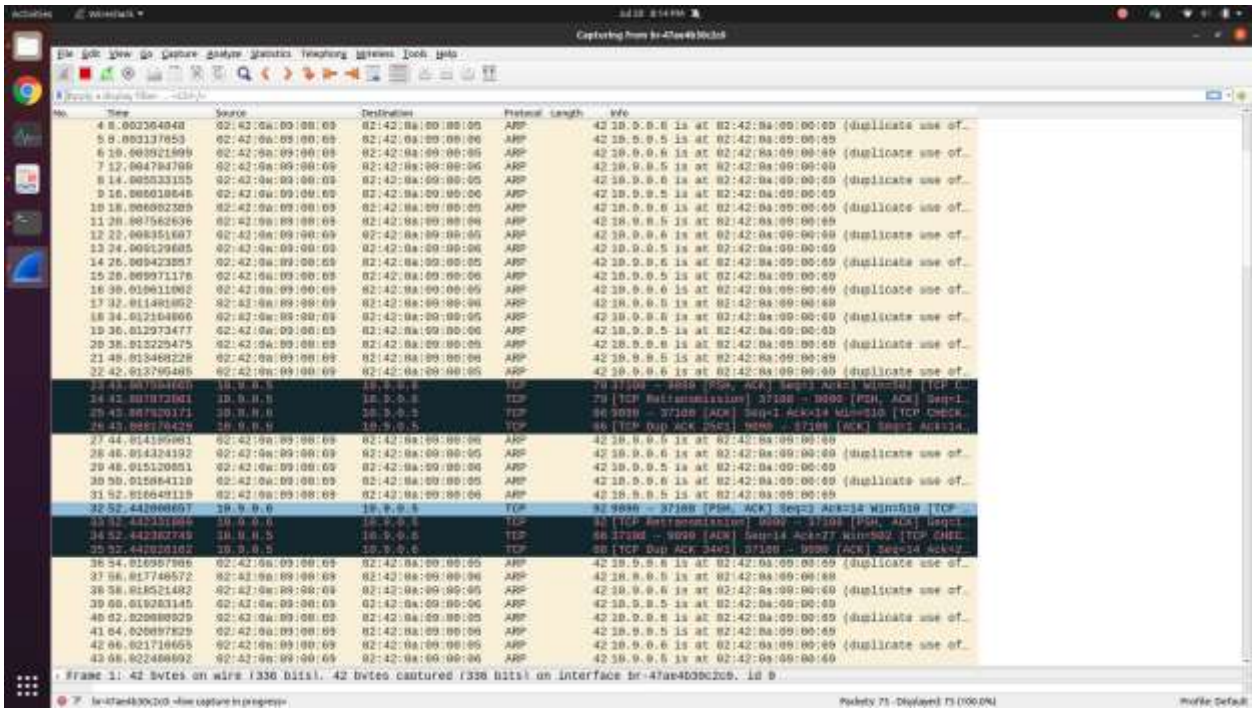


11. Now just sending some random message so that we can understand the changed done in MITM attack.

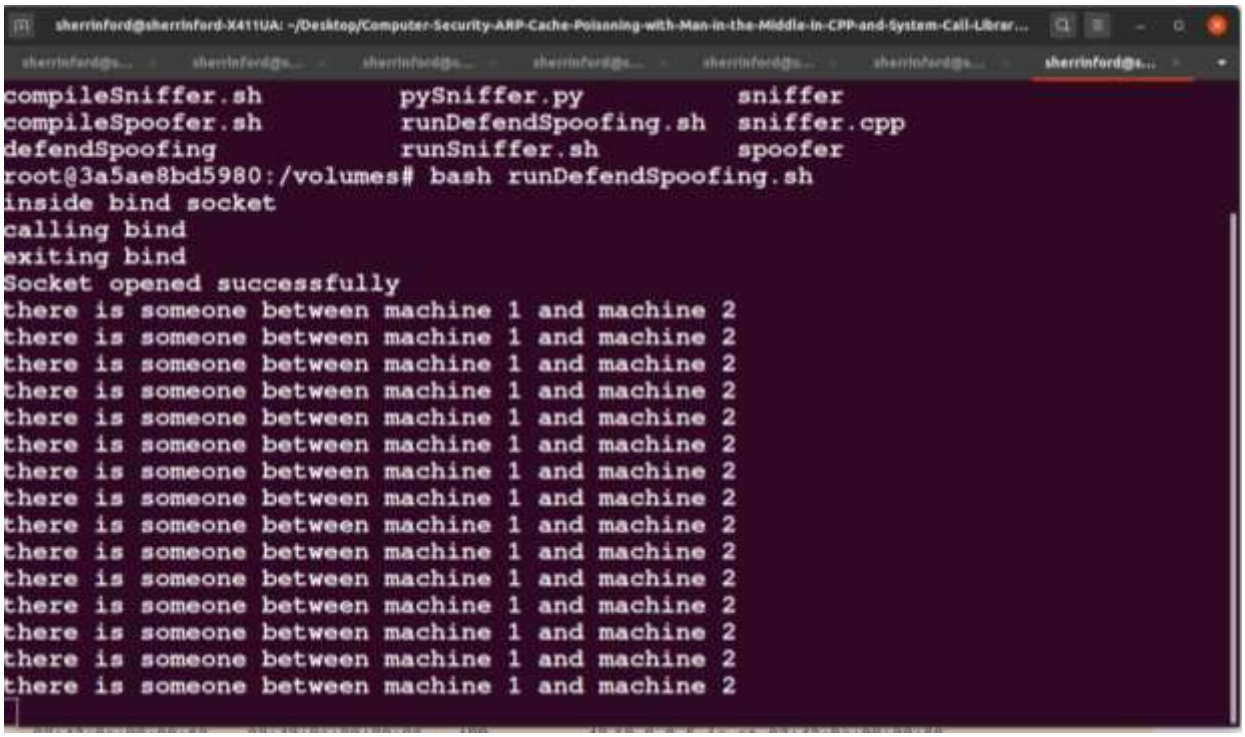




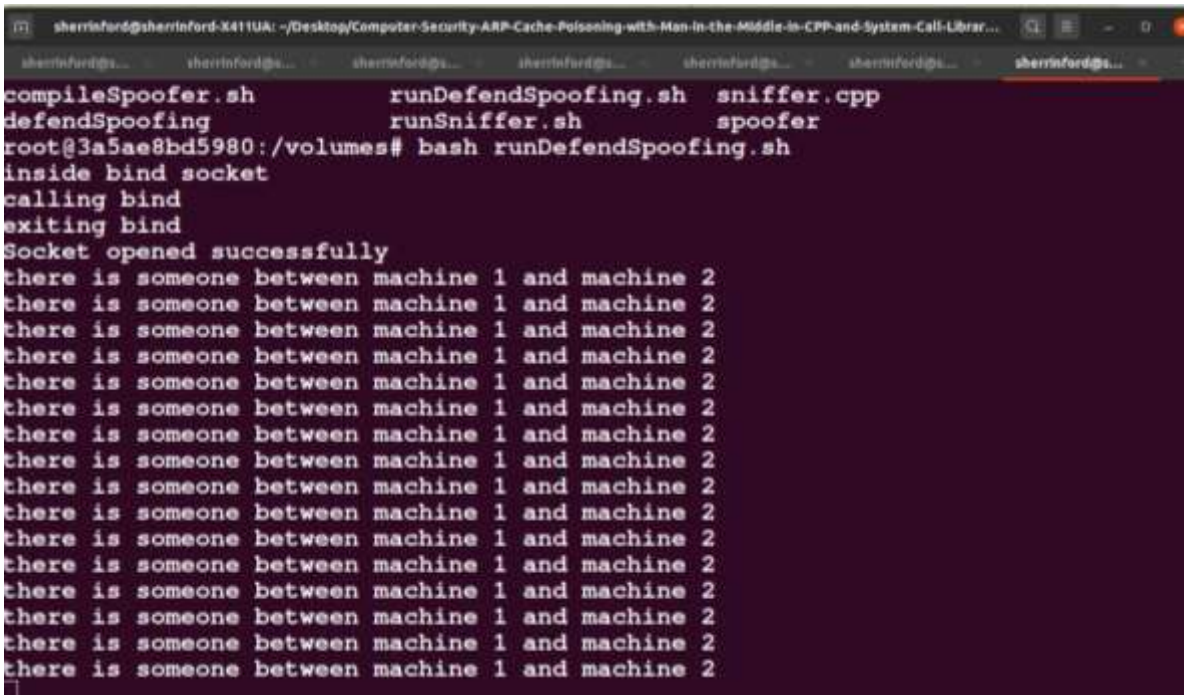
12.Wireshark generates same sort of output like it did for telnet.



13. We have a counter measure tool that can detect if someone is spoofing host-B or not. This tool should be run from host-A like below. As output suggest it catches spoofed ARP reply from attacker immediately and reports back.



14. If we now close the spoofer in attacker that generates false ARP reply then our defense tool stop generating new message informing that no one is currently spoofing host-B.



Discussion

Our attack was completely successful as we can not only sniff frame transferred from host-A to host-B we could also modify any part of the frame if we wanted to. Output shown in step 6, 7, 8 for telnet and output shown in step 11, 12, 13 proves that our attack was successful indeed for netcat application.

We have also written a counter measure code. That will check whether any machine is between the communication of host-A to host-B. It sniffs all ARP packets and check whether anyone is spoofing host-B or not. In steps 14, 15 we have demonstrated its function.