**Assignment 02**

NAME: AMARTEY EZEKIEL ELVIS

INDEX NUMBER: 3022320

**Secure Network Systems Lab: Network Traffic Sniffing and ARP Spoofing Attack - Exercise 2 Report**

**Introduction**

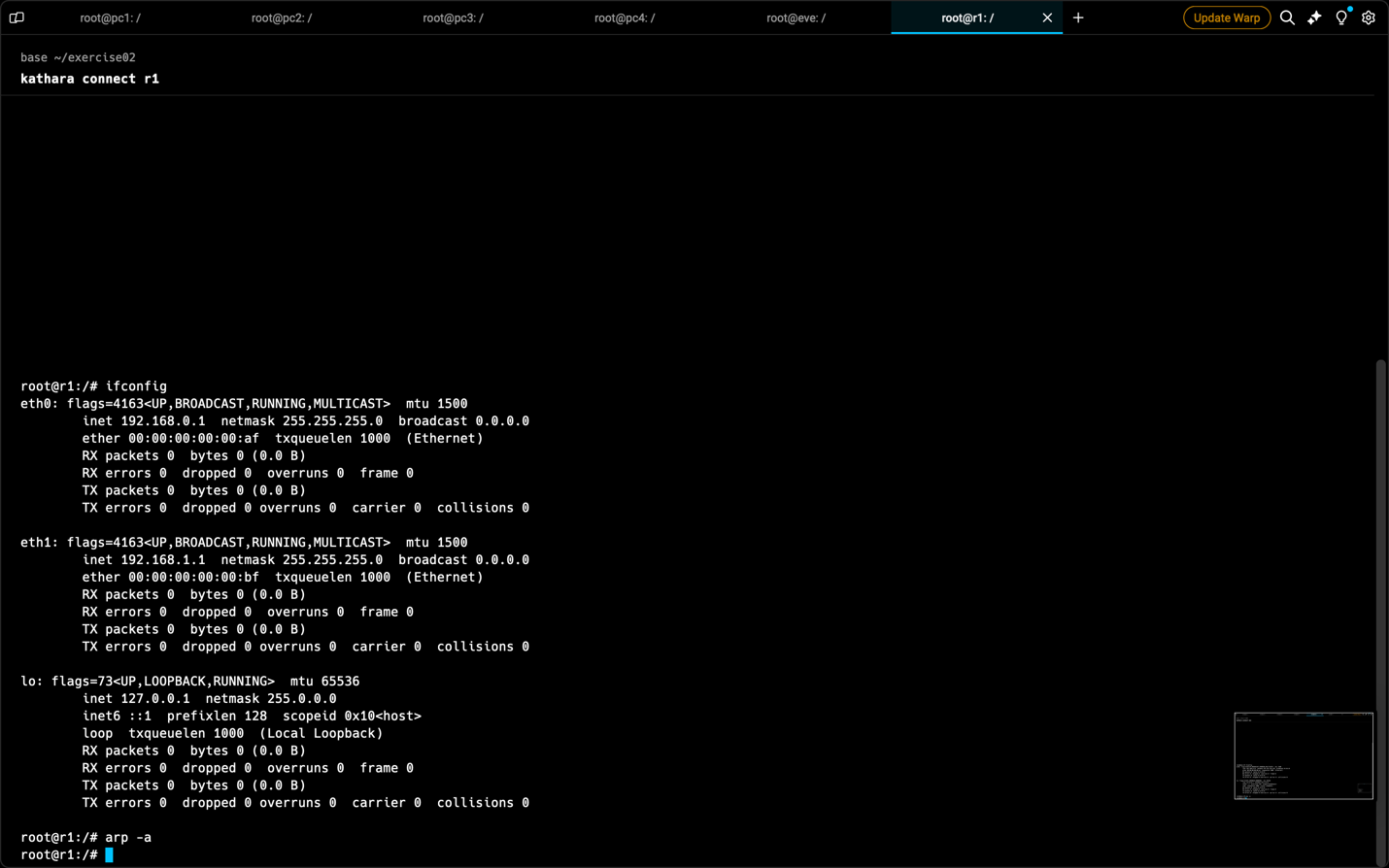
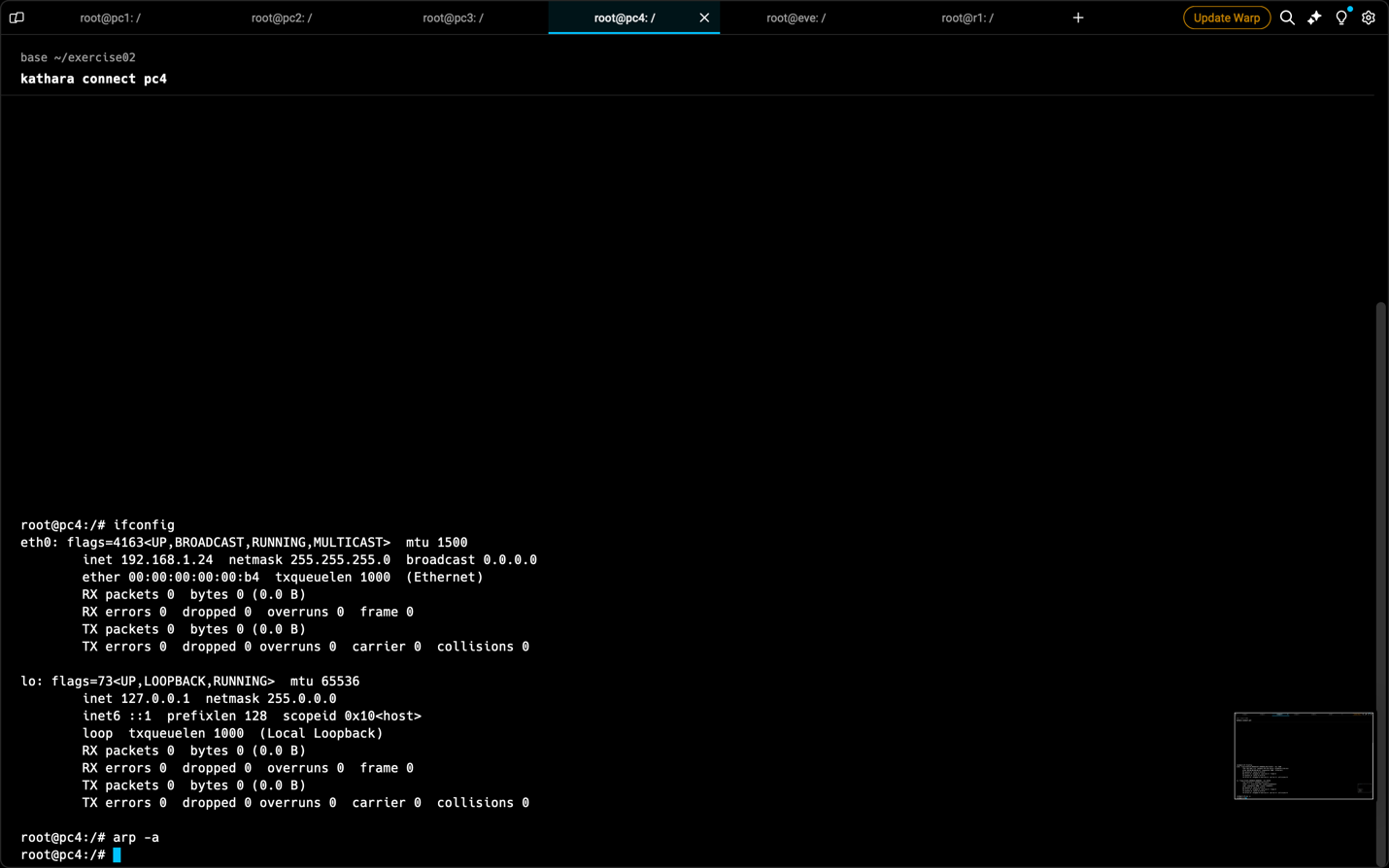
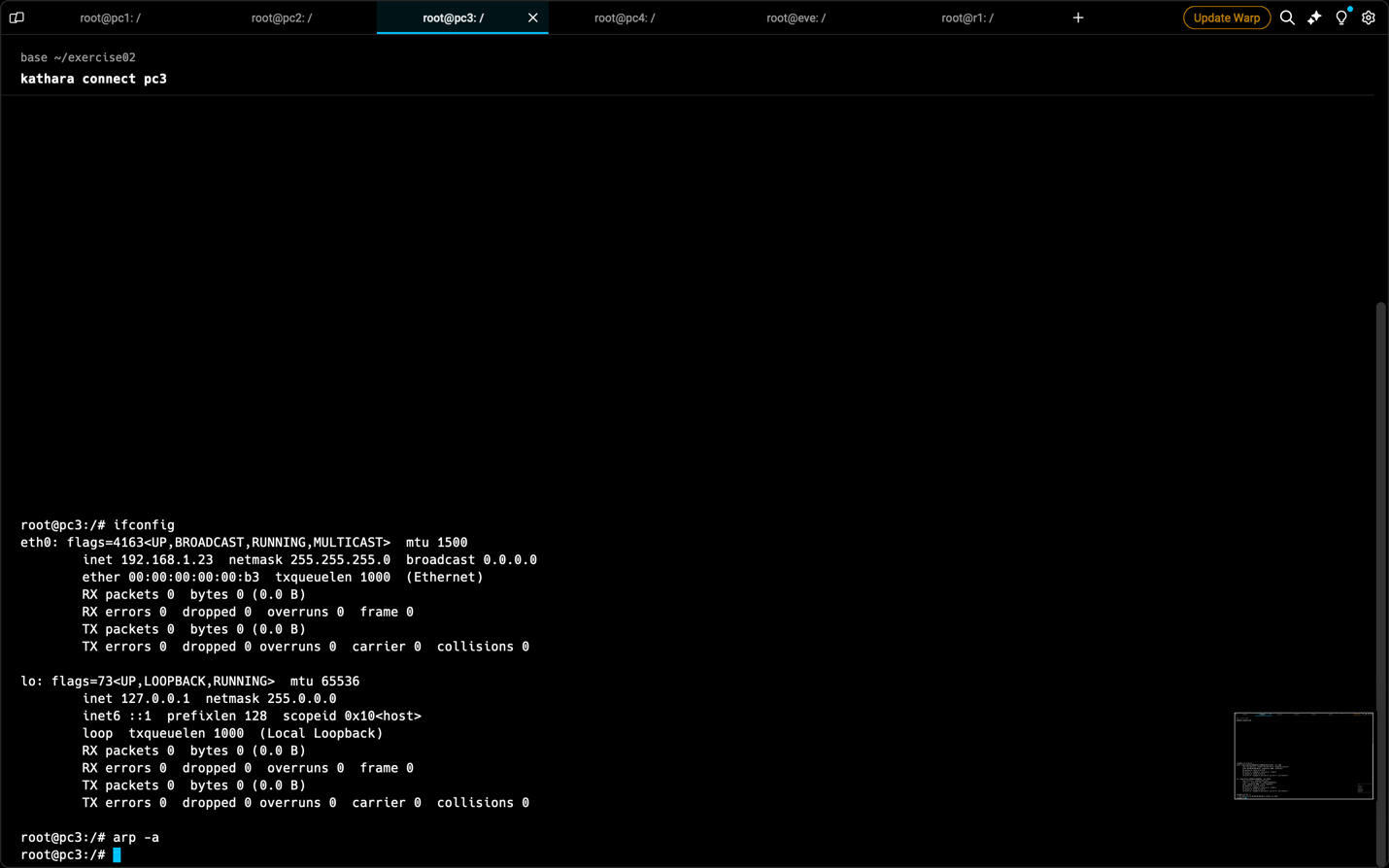
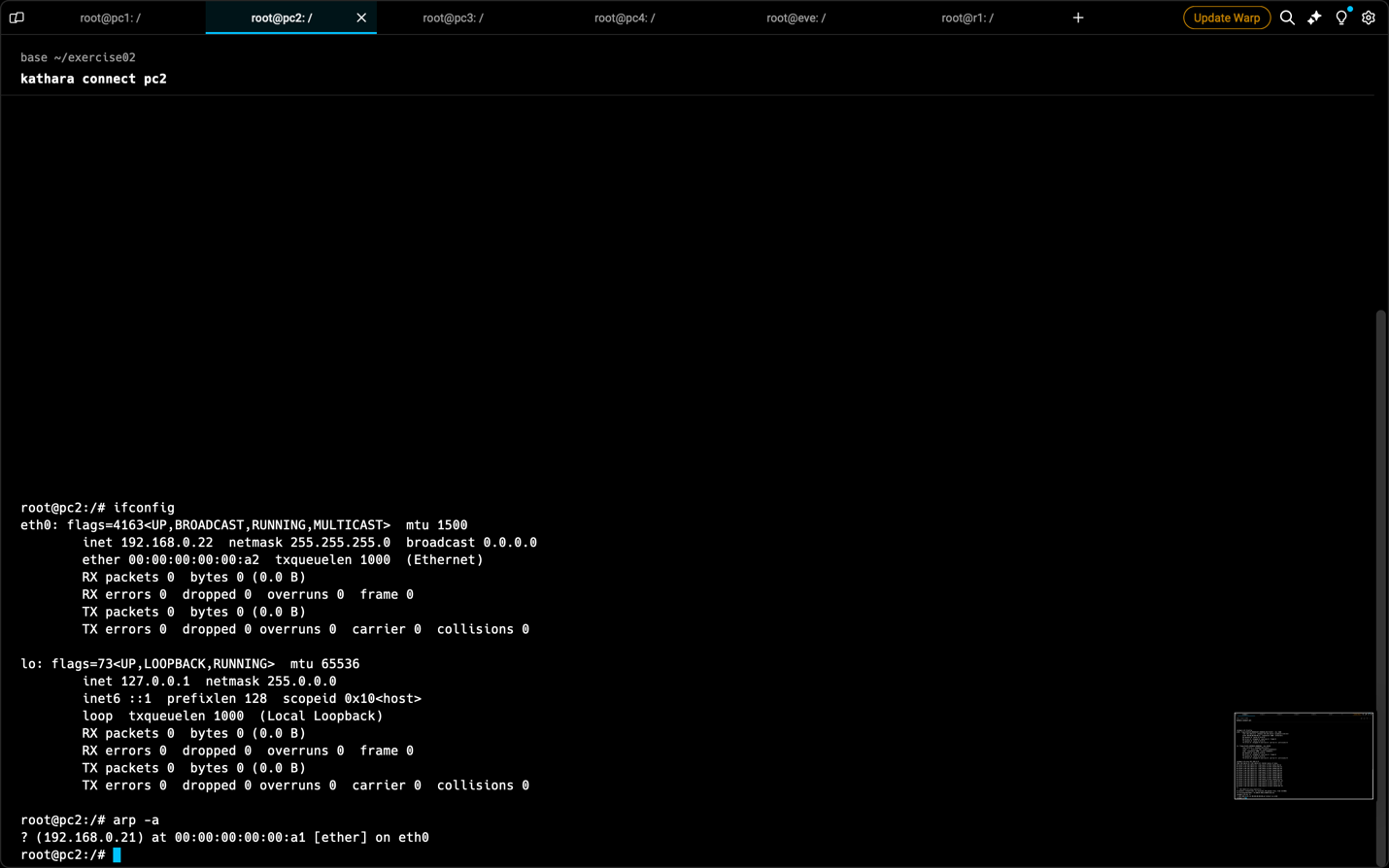
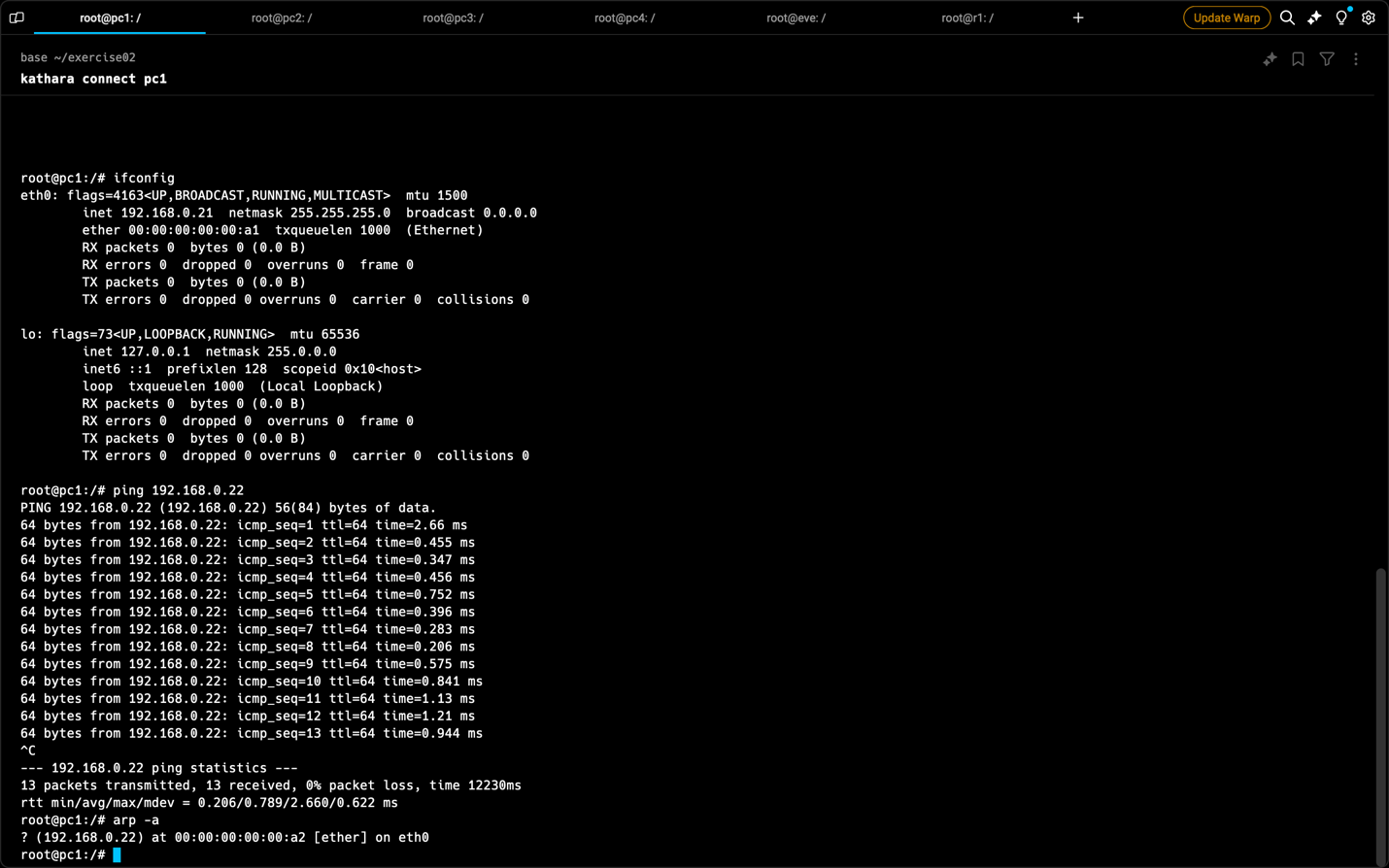
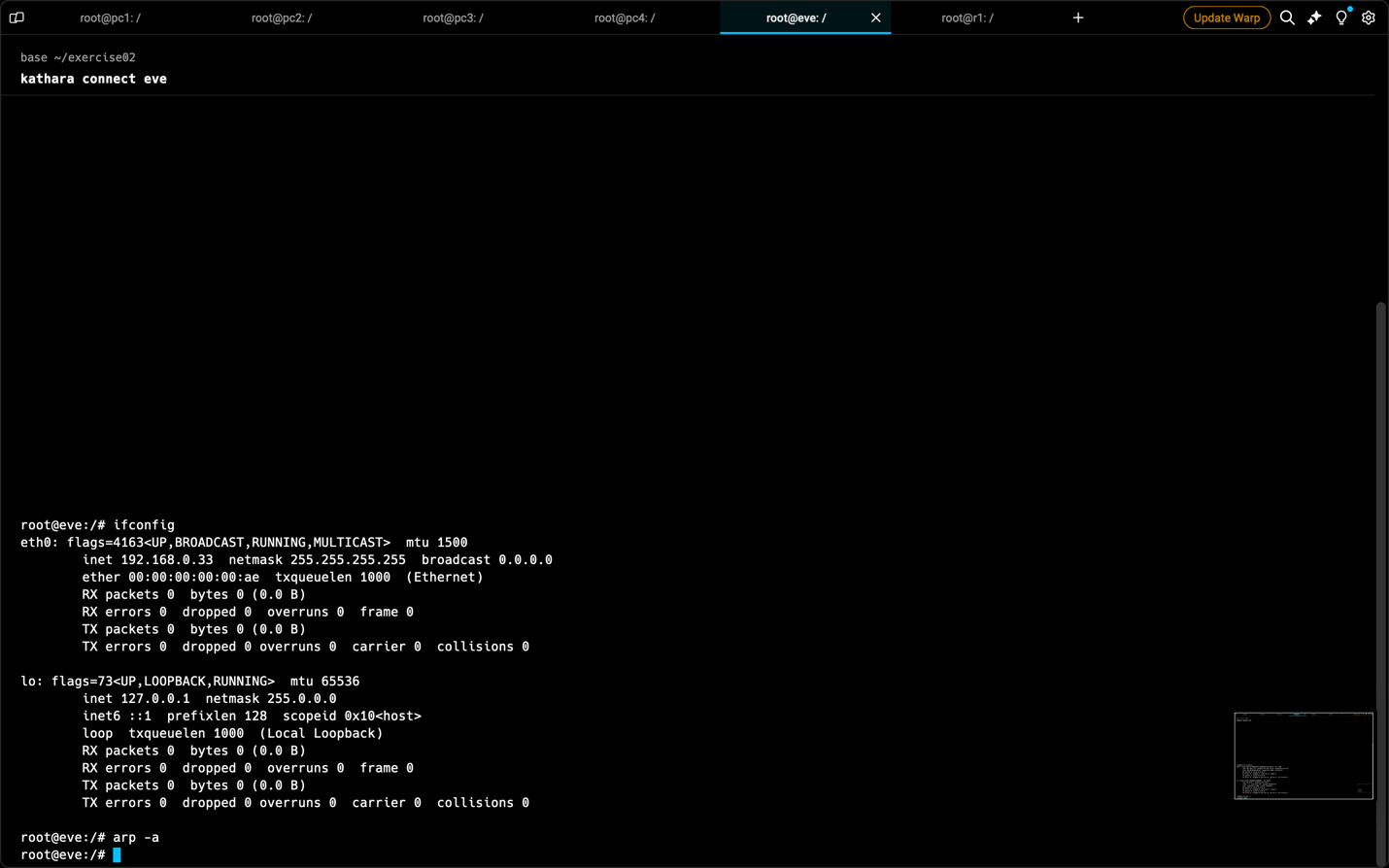
This report documents the findings of Exercise 2 in the Secure Network Systems Lab, focusing on network traffic sniffing and ARP spoofing attacks. The lab utilized the Kathara framework to configure a simulated network environment.

**Lab Setup**

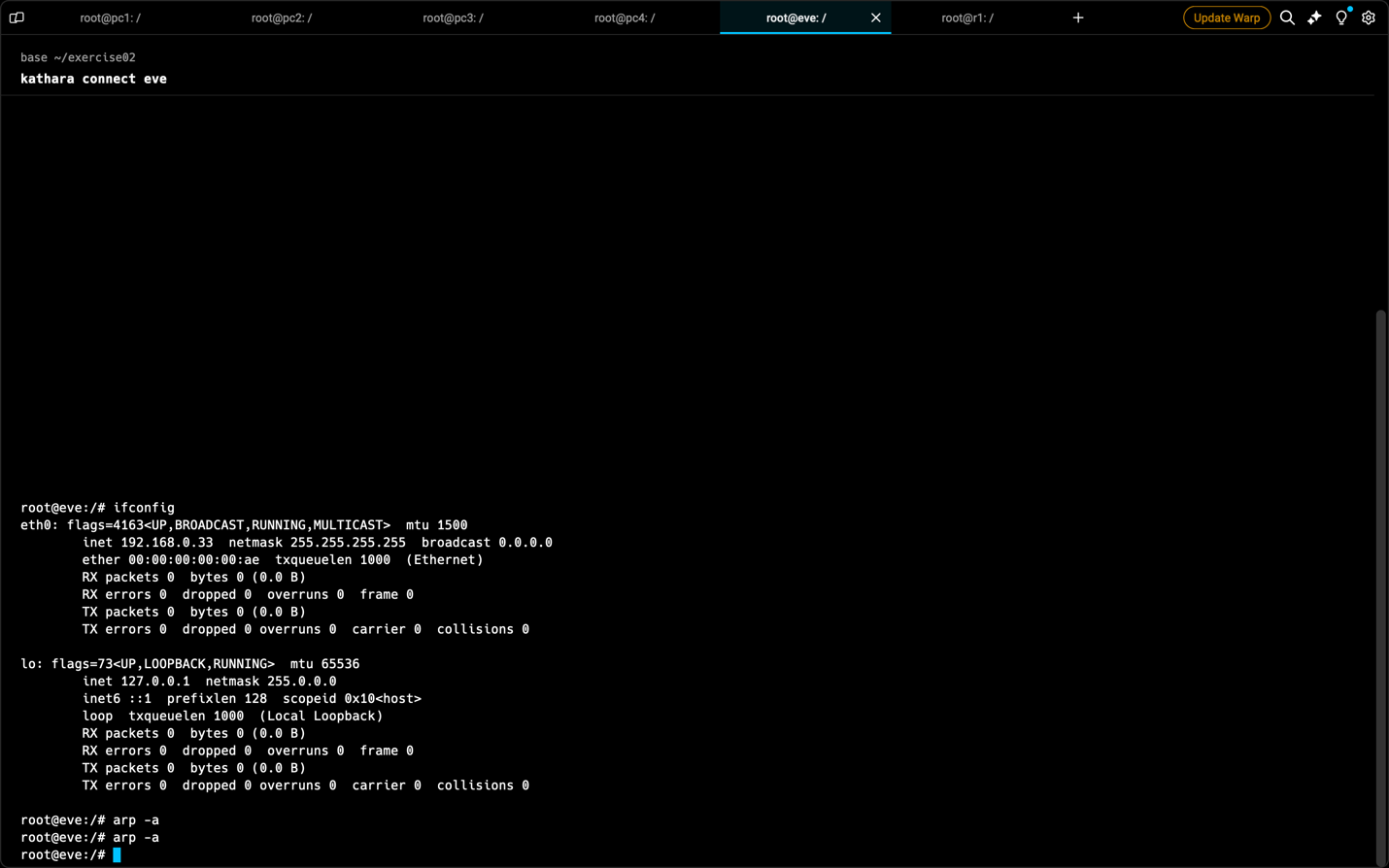
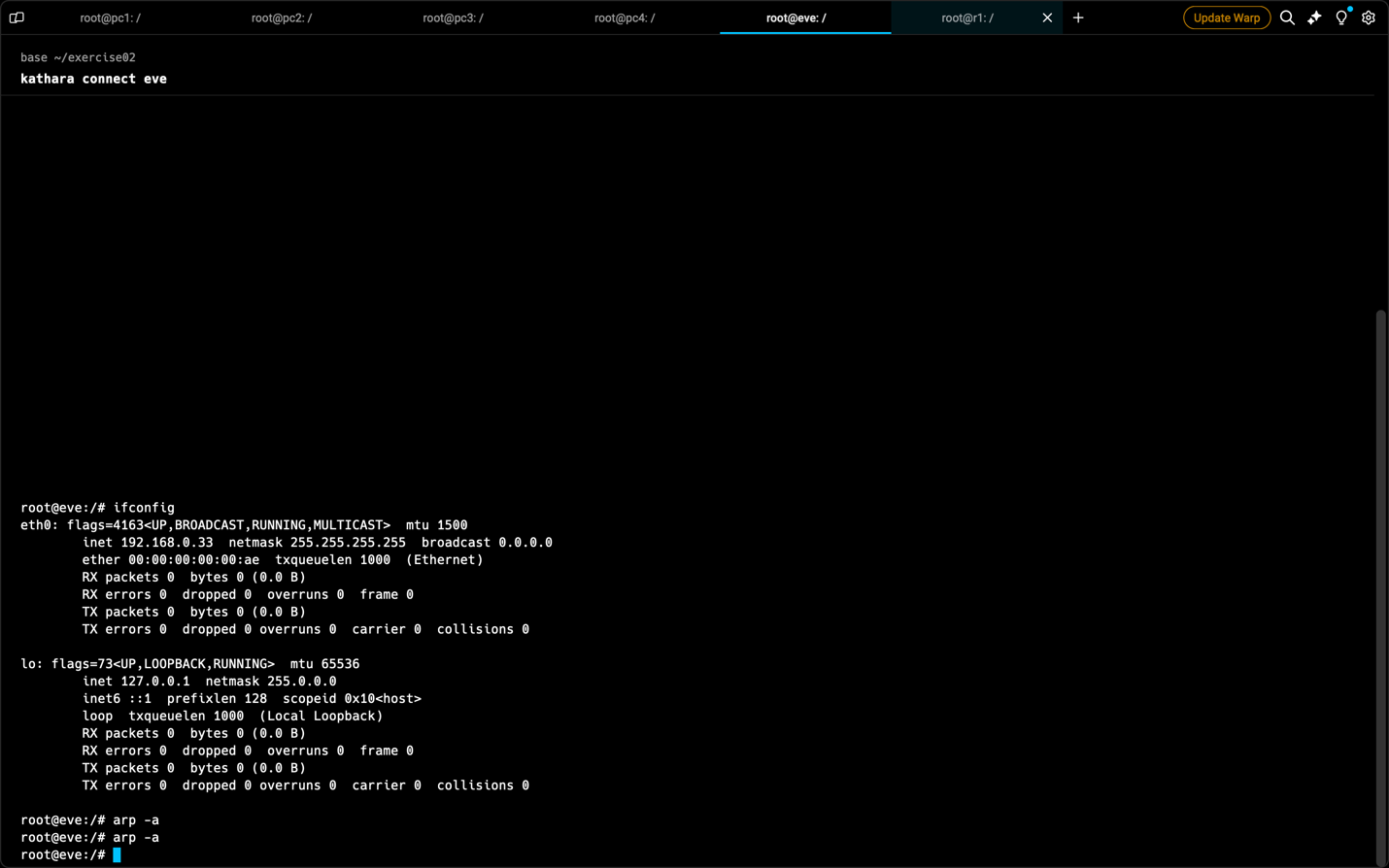
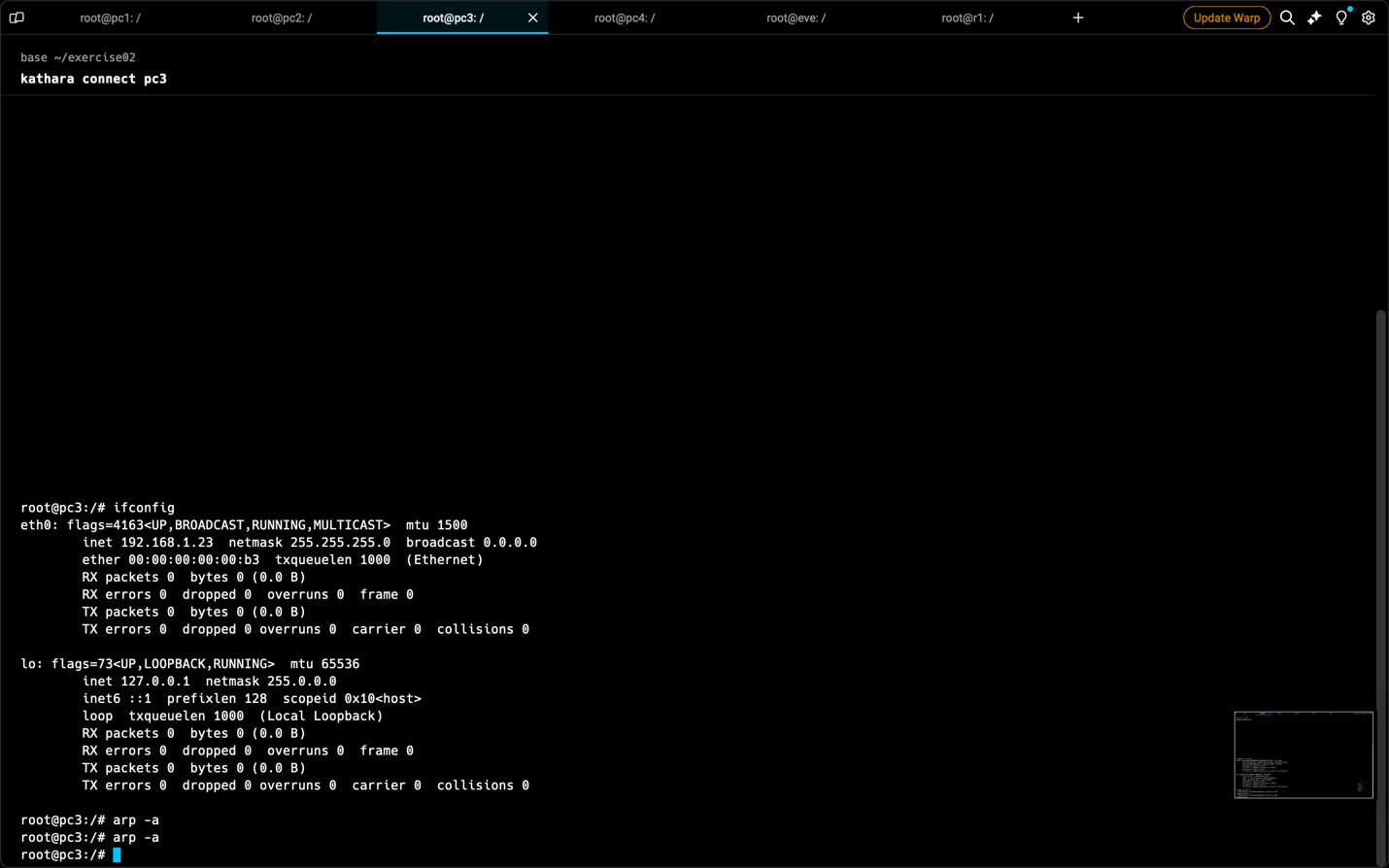
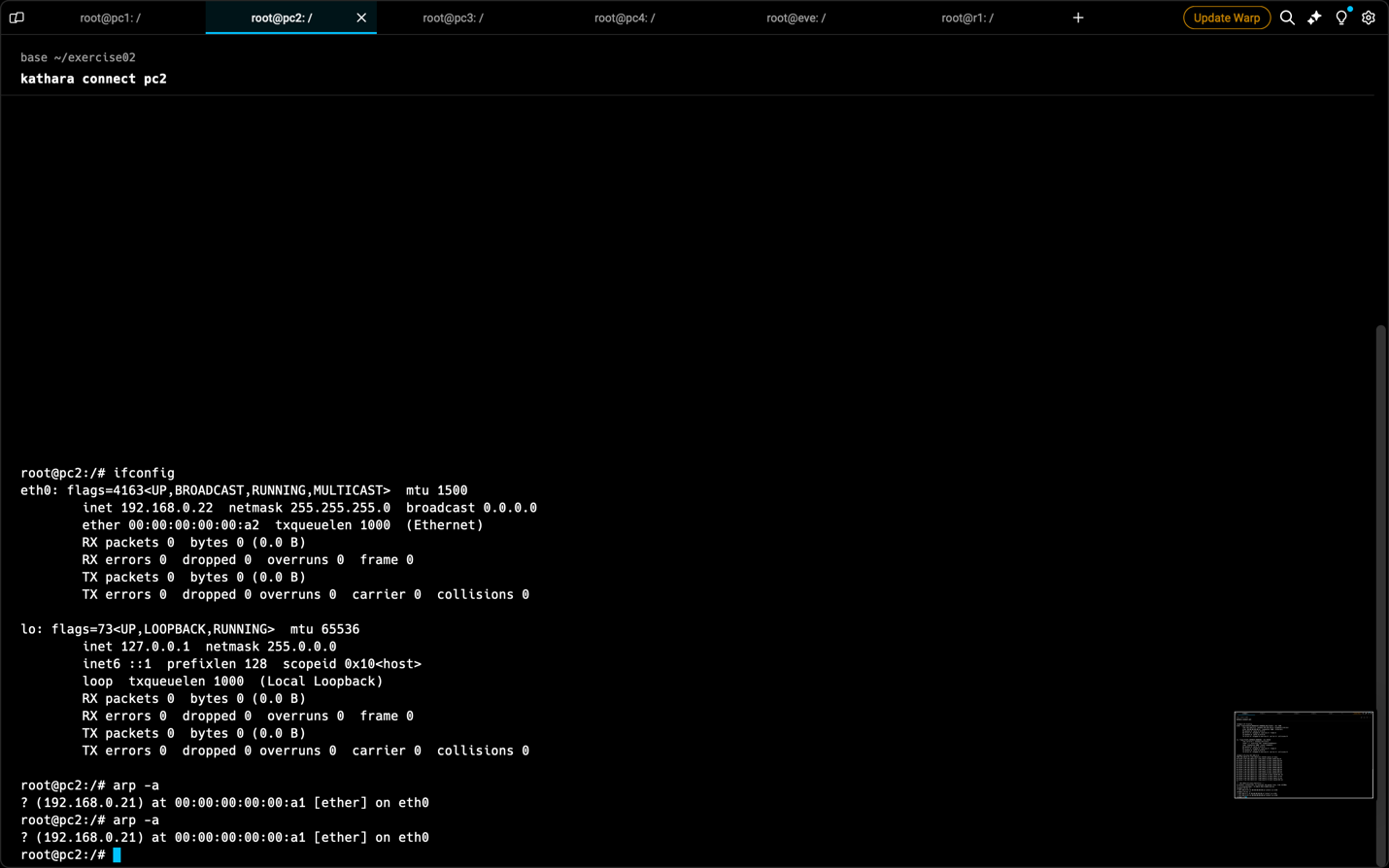
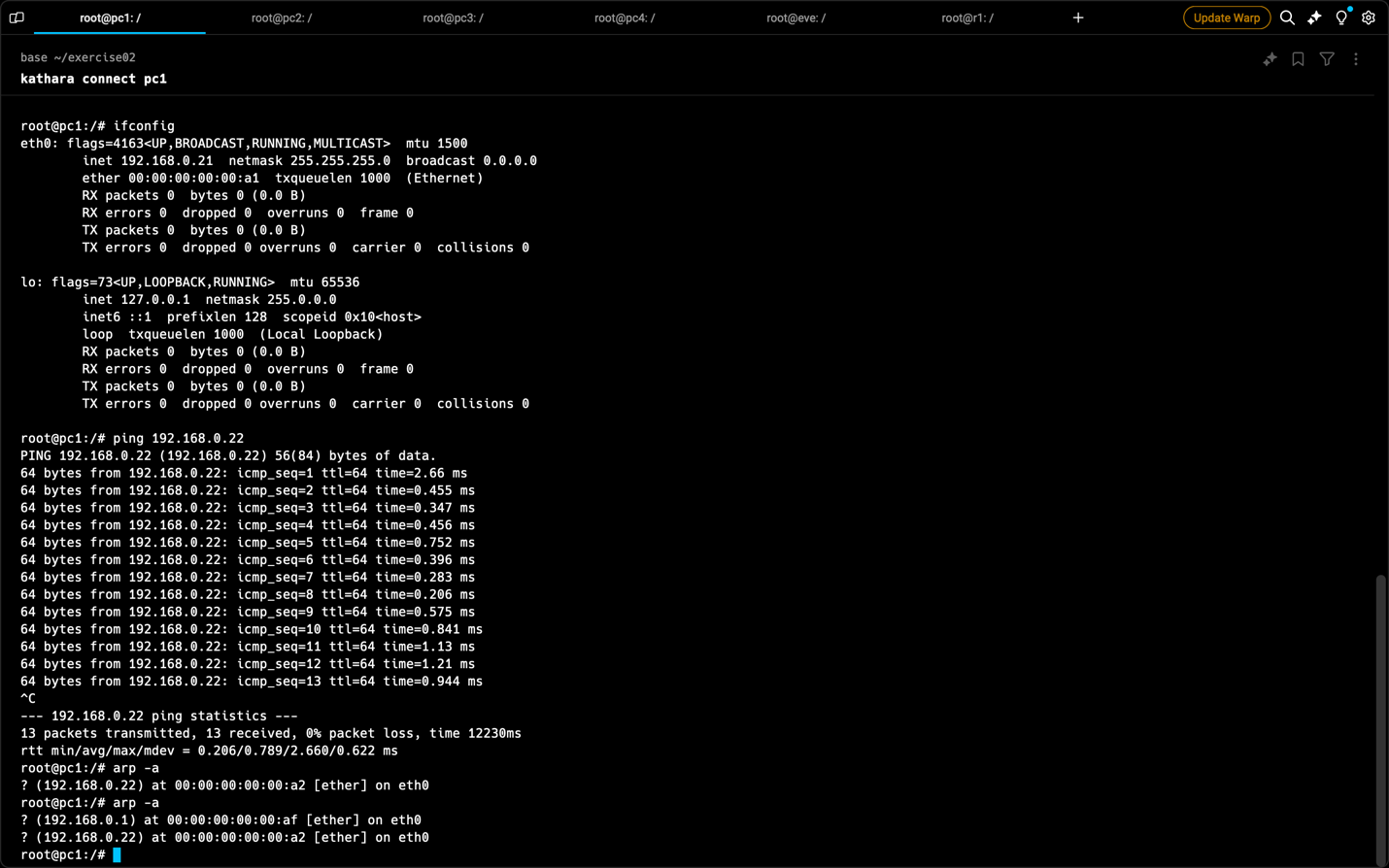
1. **Directory and File Creation:**
   * A directory named exercise02 was created to store all lab files.
   * Inside exercise02, another directory named shared was created with read/write permissions for all users.
2. **lab.conf Configuration:**
   * A lab.conf file was created defining the network topology with device configurations.
   * All devices, except "eve," used the "kathara/base" image.
   * The "eve" device used a custom-built image with the nemesis tool for ARP spoofing.
   * IPv6 networking was disabled for all devices.
3. **Building the Custom Image:**
   * The provided Dockerfile was used to build a custom image named "kathara\_nemesis" containing the nemesis tool.
4. **Wireshark Integration:**
   * A device named "wireshark" was added to the lab.conf using the "lscr.io/linuxserver/wireshark" image.
   * The "wireshark" device was attached to network A for traffic sniffing.

**Sniffing Initial ARP Traffic**

1. **Ping Tests:**
   * pc1 pinged pc2, and the ARP cache entries were captured on all devices using arp -a. Screenshots were named <device\_name>\_arp1.png.
   * The process was repeated with pc4 pinging pc1, and screenshots were named <device\_name>\_arp2.png.
   * Here are some images from this test.



The above images are the tests from pinging pc2 from pc1



The above images are a result of pinging pc1 from pc4

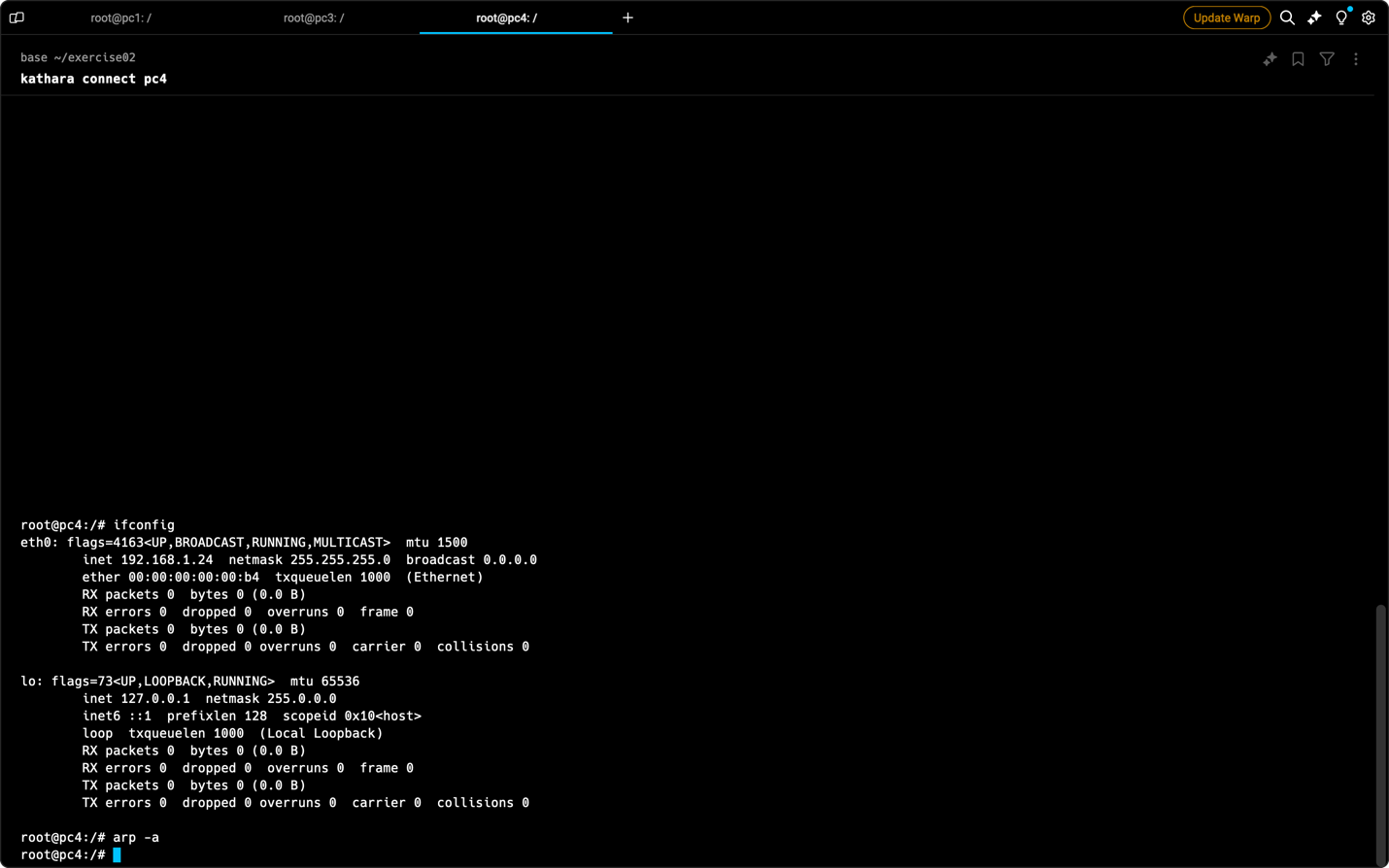
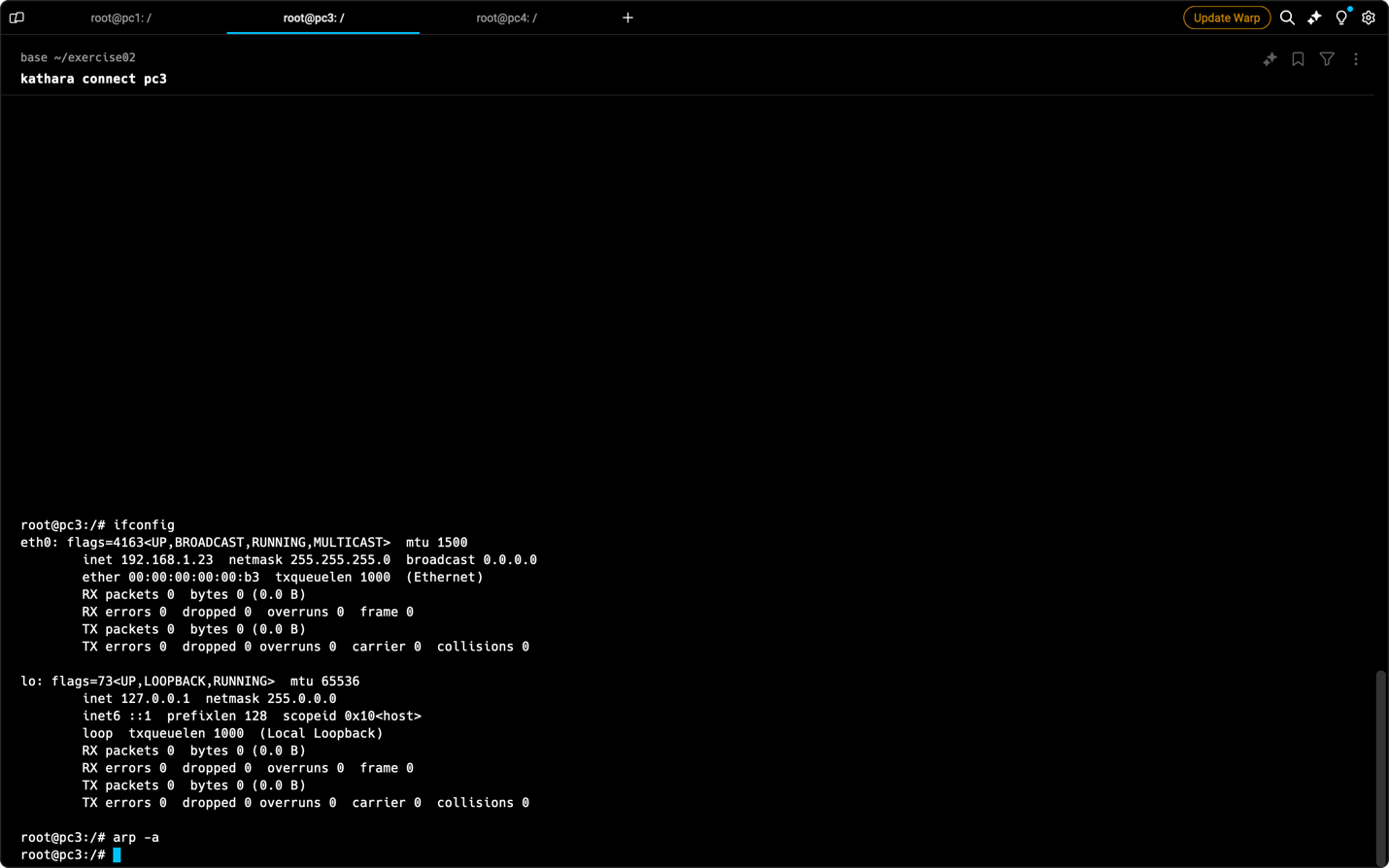
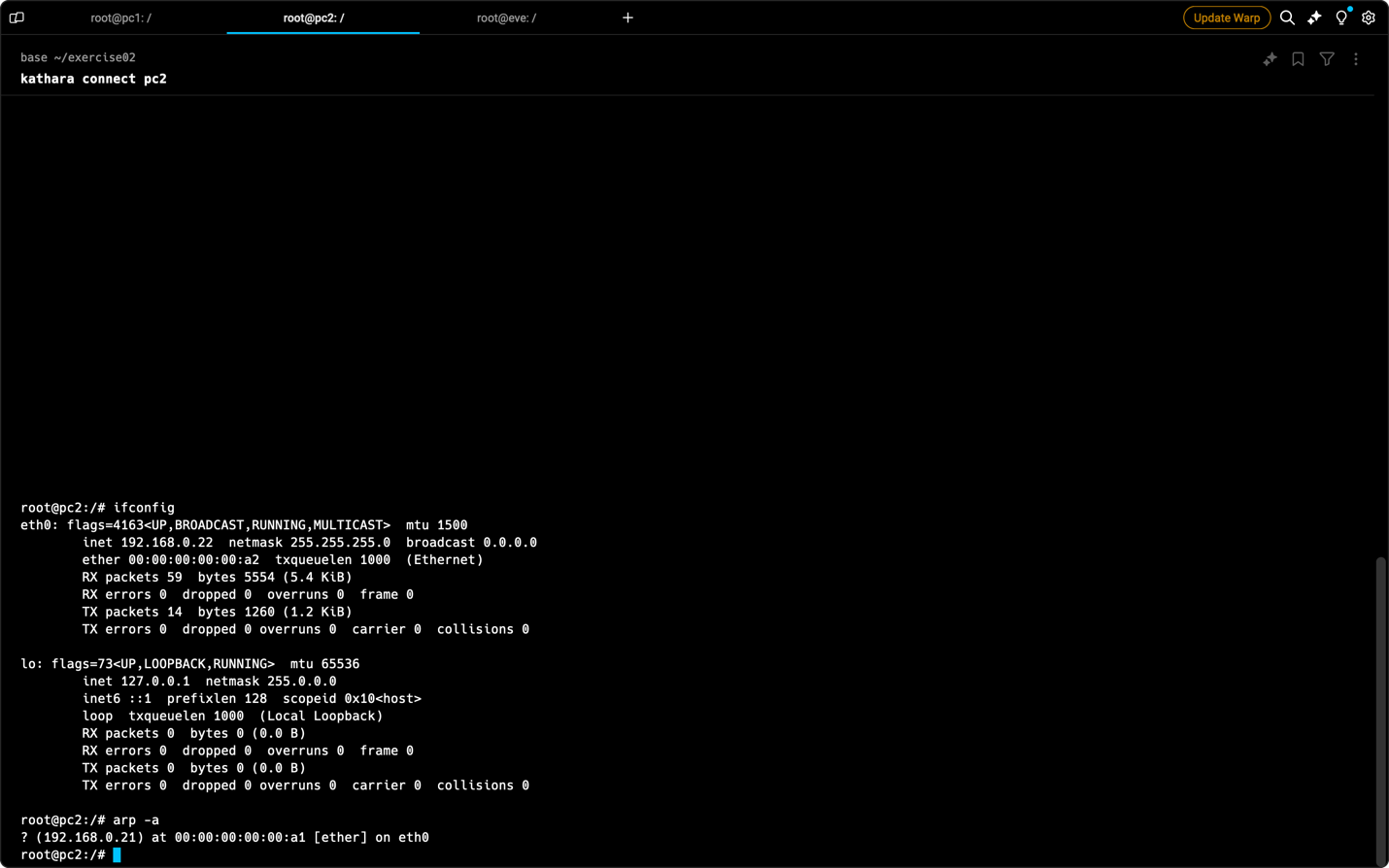
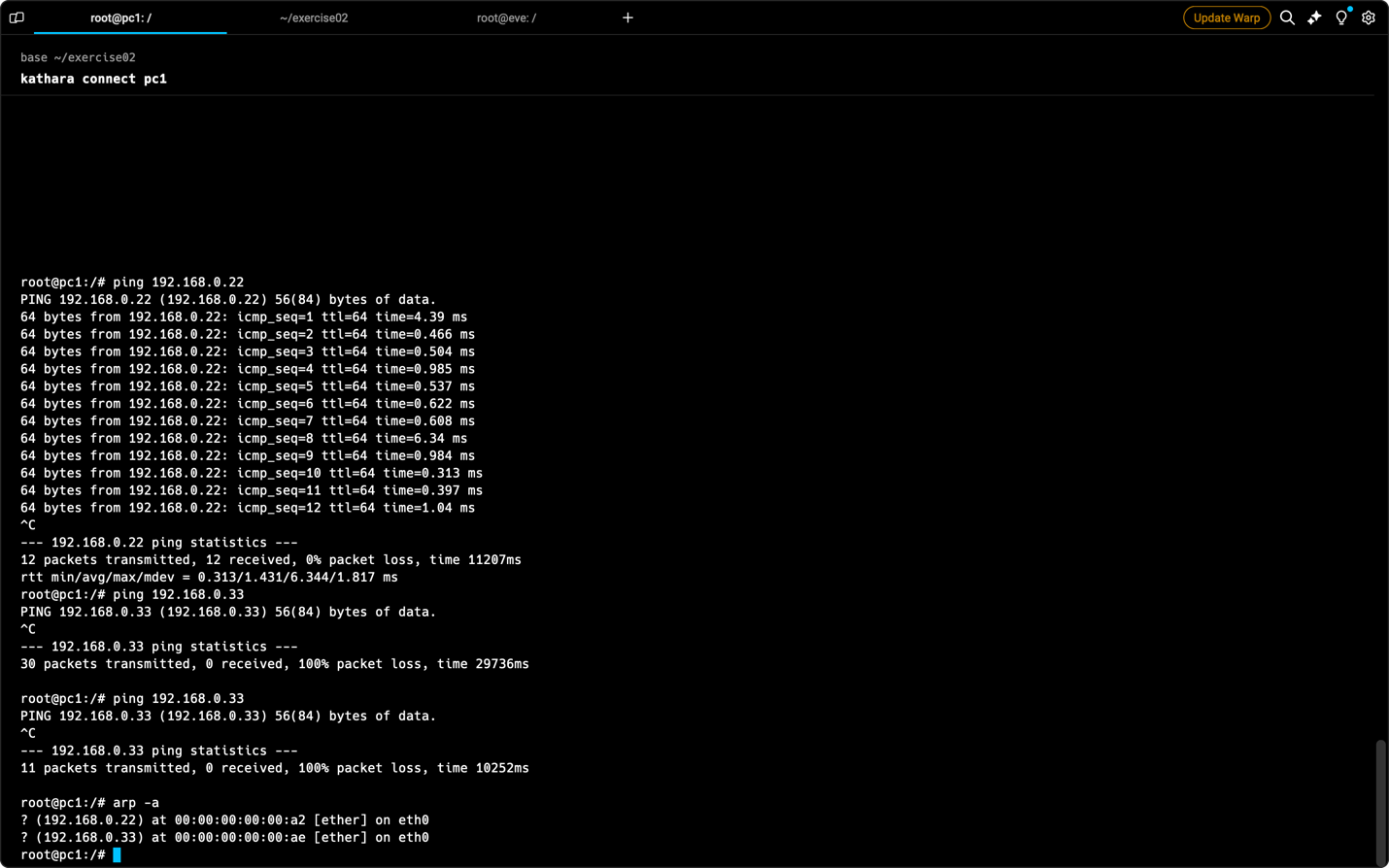
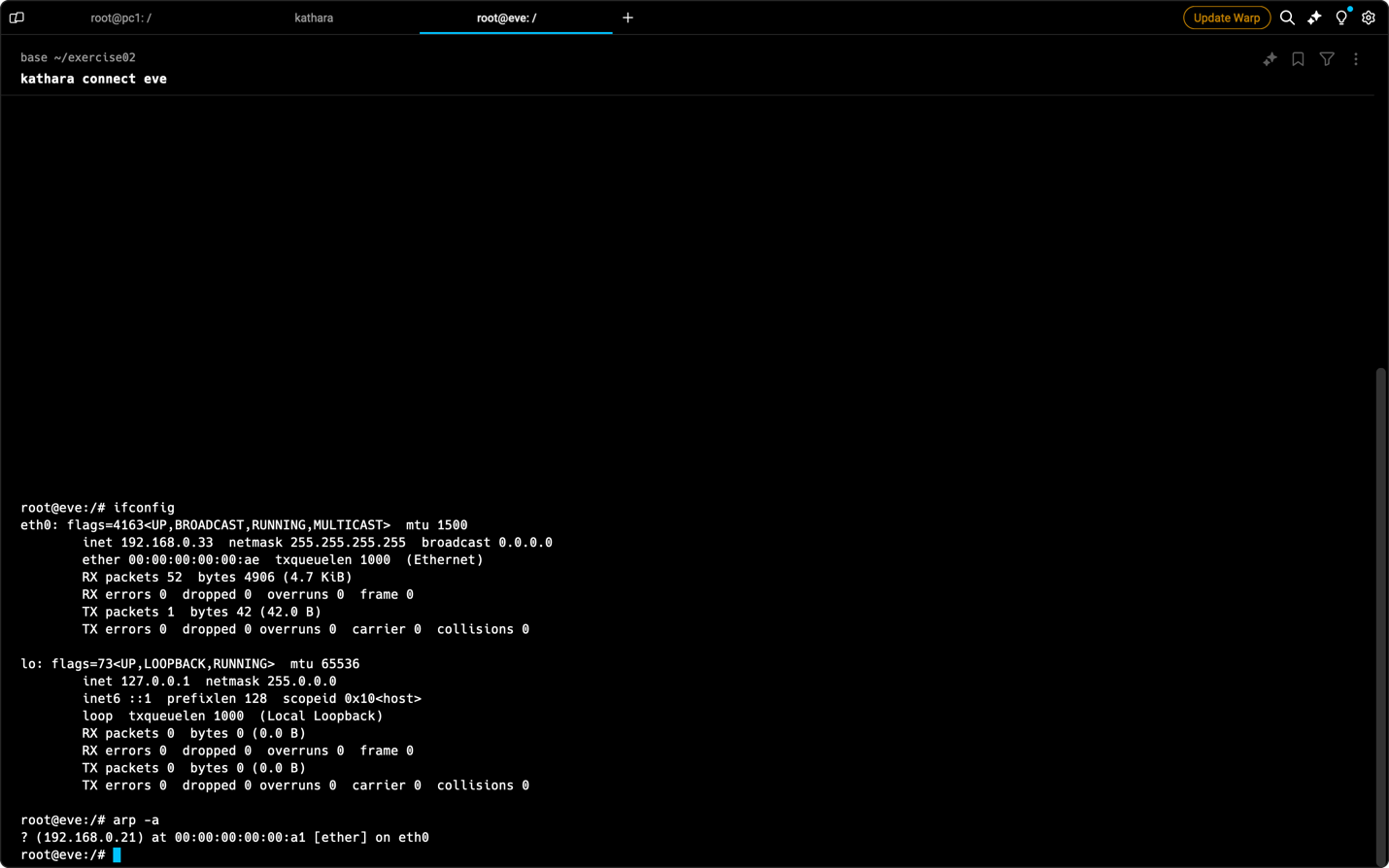
**Wireshark Capture:**

* + Wireshark captured network traffic on network A during the ping tests, storing the data as capture1.pcapng in the /shared directory.

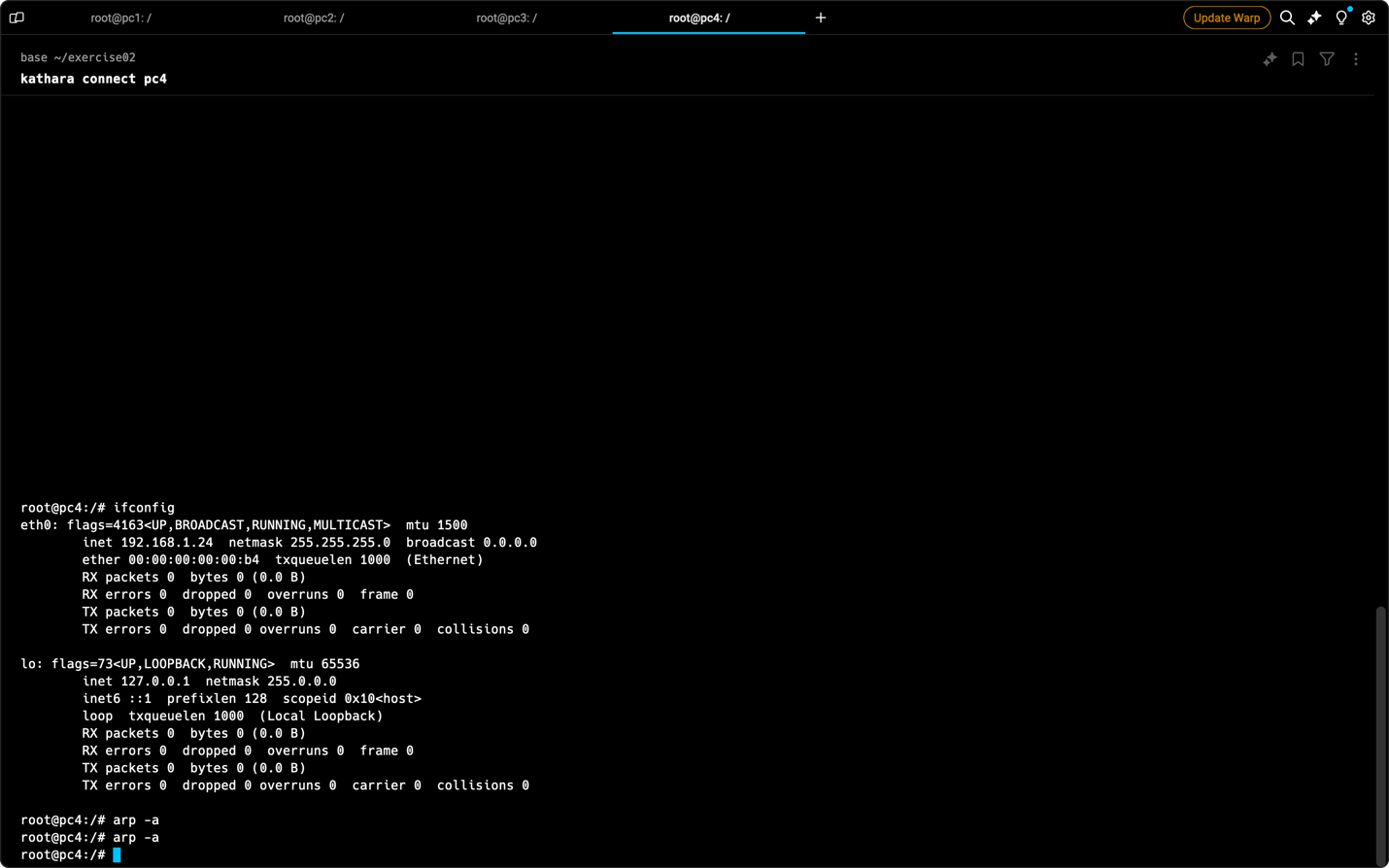
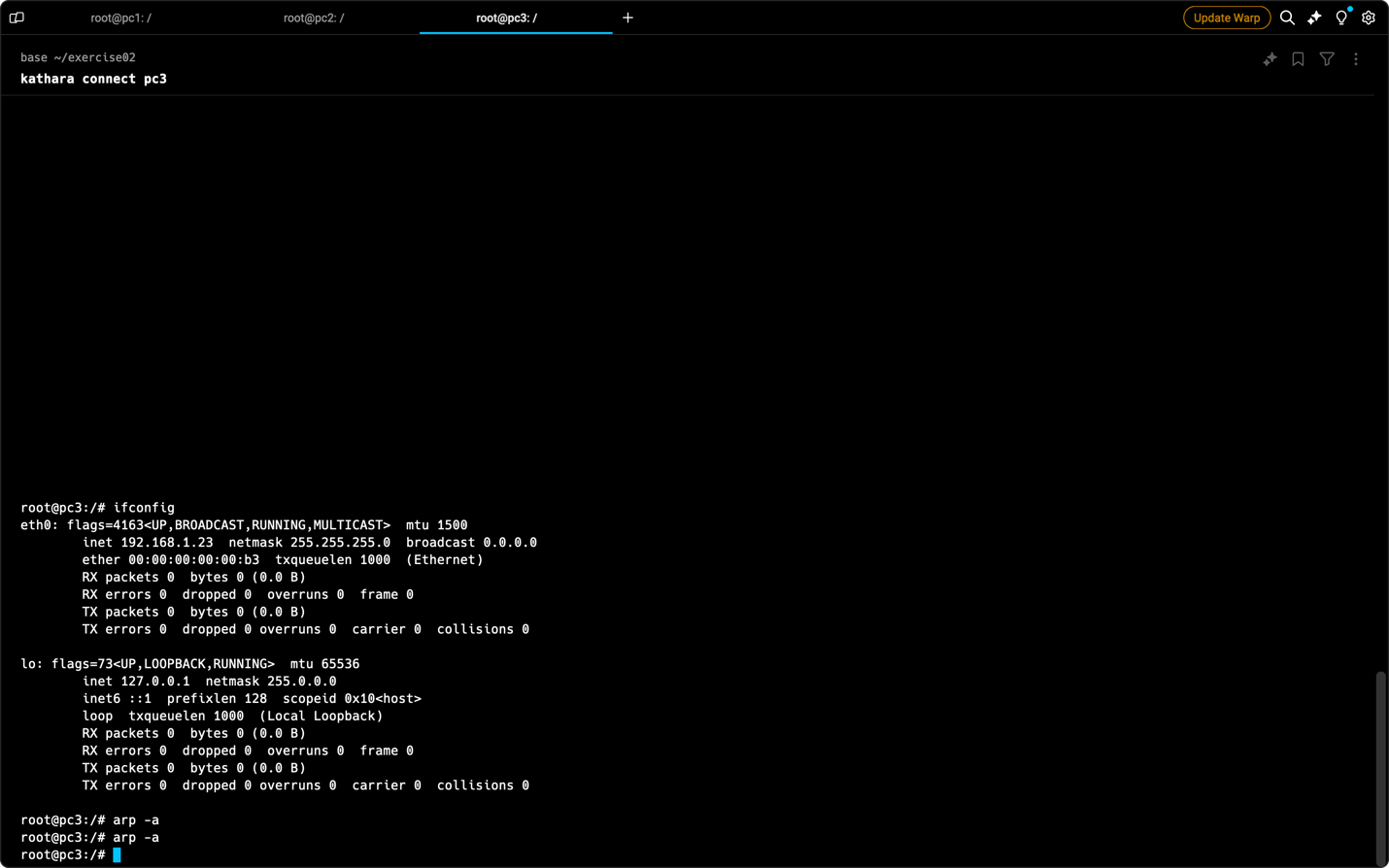
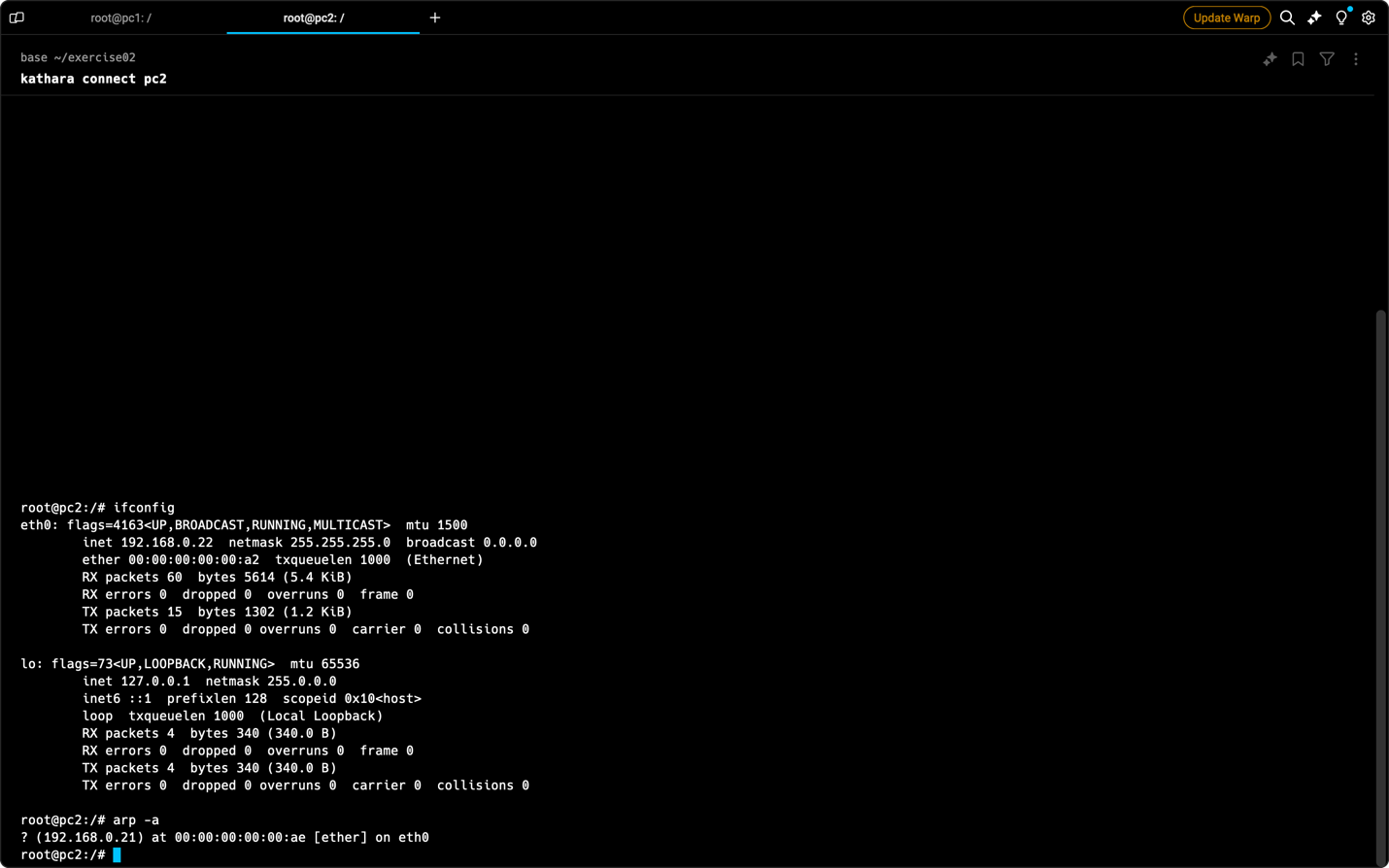
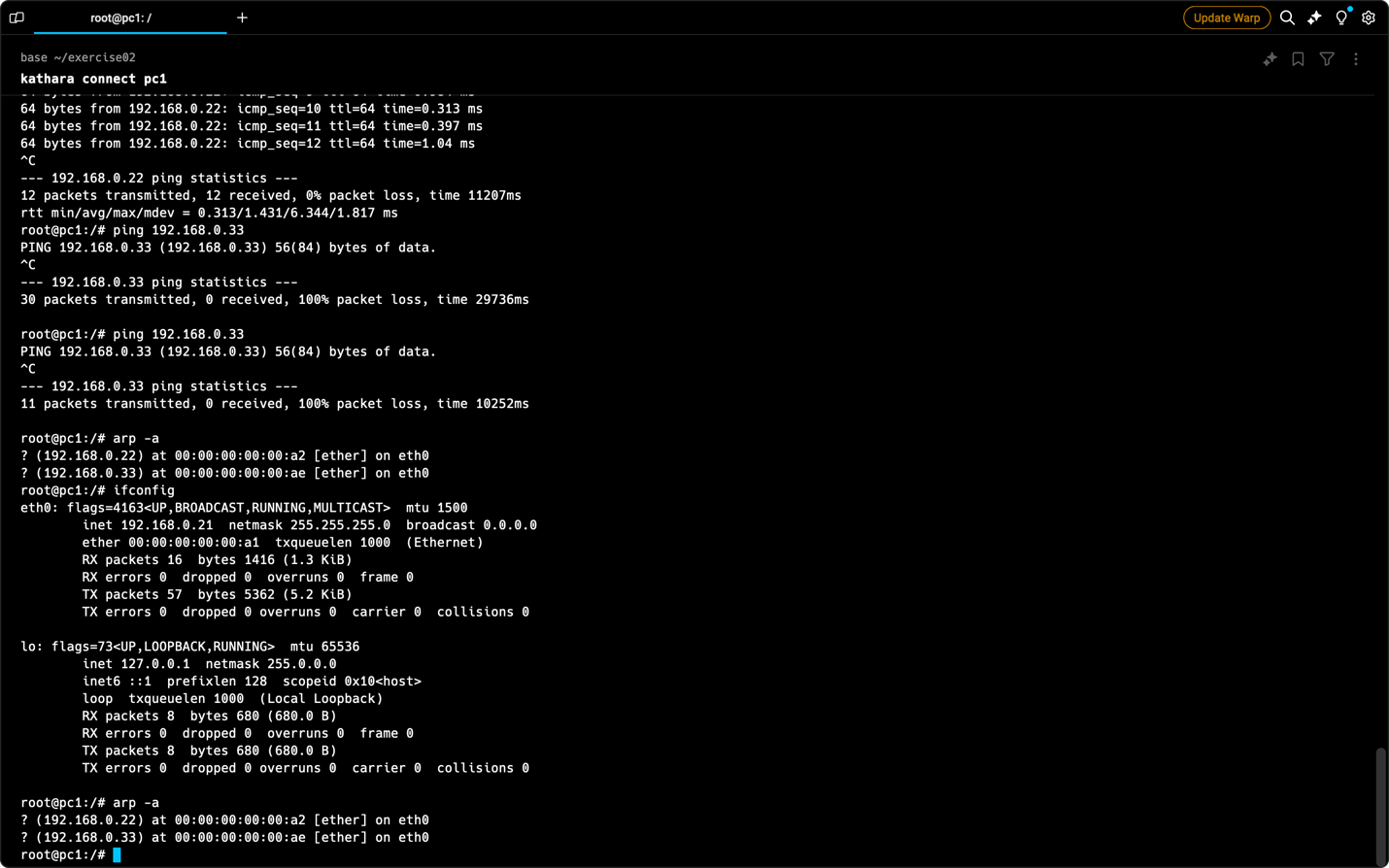
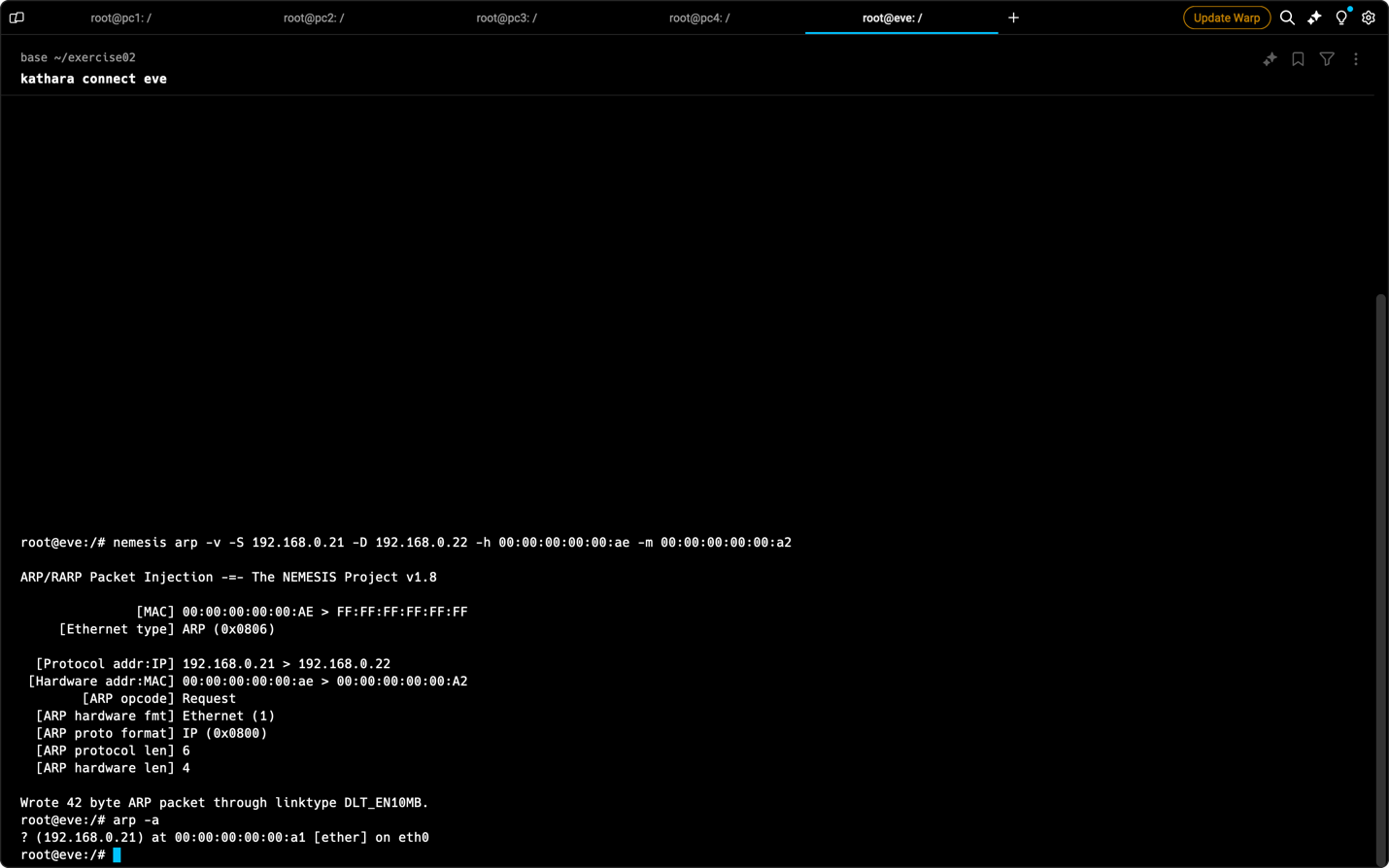
1. **Lab Restart and Sniffing:**
   * The lab was stopped and restarted using kathara lclean and kathara lstart.
   * Wireshark was again attached to network A for traffic sniffing.

**ARP Spoofing Attack**

1. **Ping Tests and ARP Cache Inspection:**
   * pc1 pinged pc2 and eve. Eve also pinged pc2. ARP cache entries were captured on all devices and saved as <device\_name>\_arp3.png.
   * Below are the captured results.



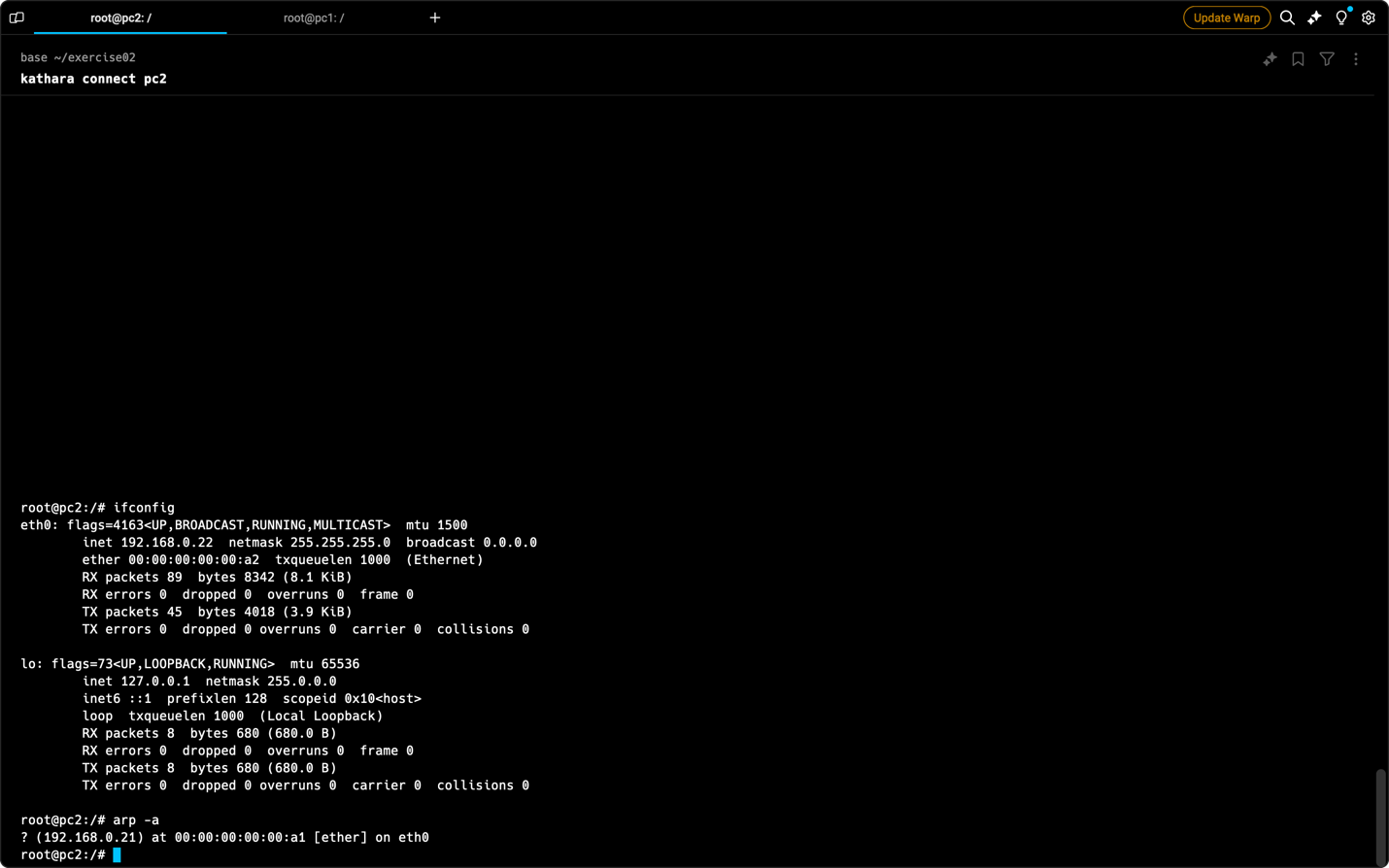
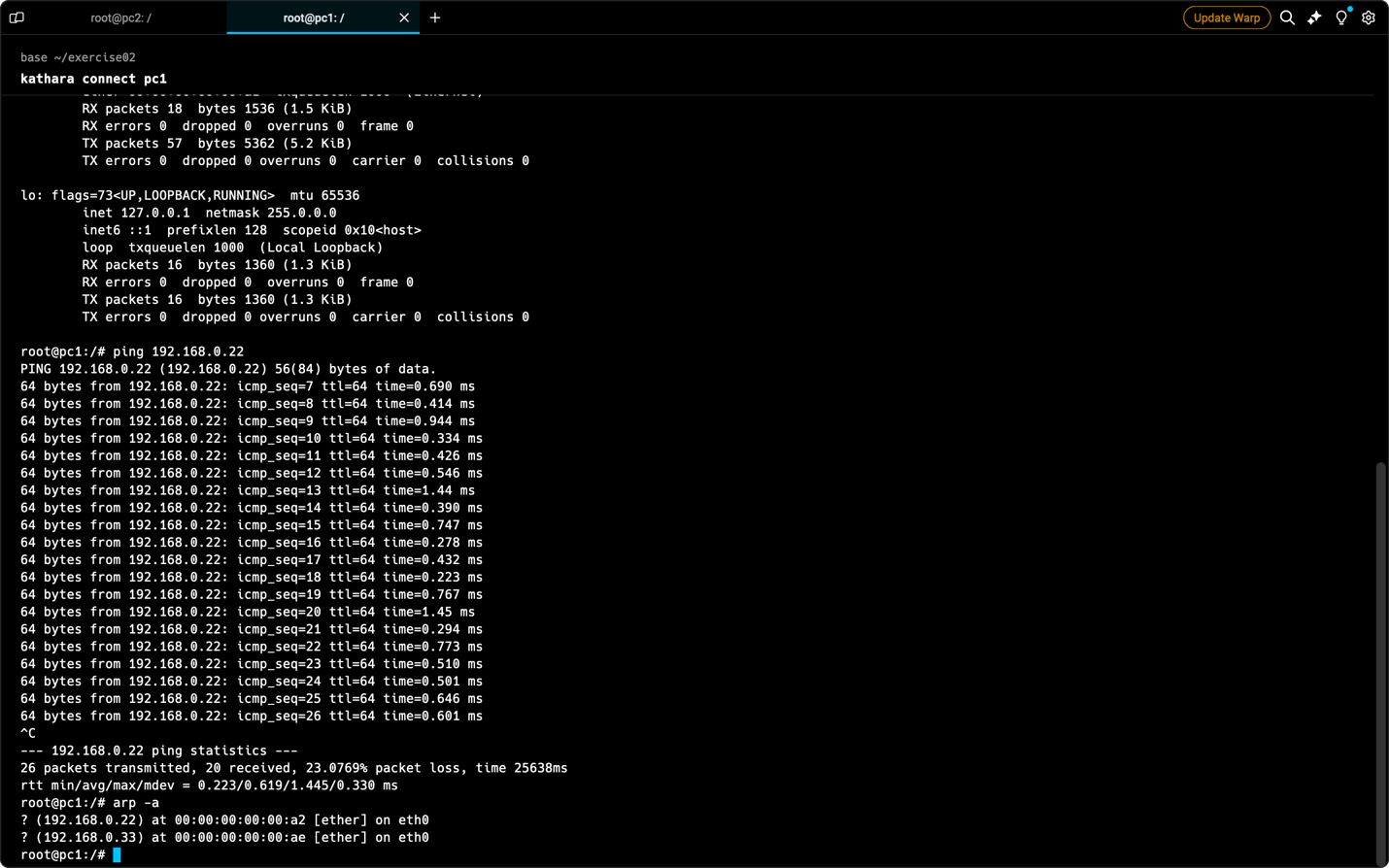
1. **ARP Spoofing with Nemesis:**
   * From eve, the nemesis tool was used to perform an ARP spoofing attack with the following command:
   * nemesis arp -v -S <pc1 IP> -D <pc2 IP> -h <MAC of eve> -m <MAC of pc2>
2. **Post-Spoofing ARP Cache Inspection:**
   * ARP cache entries were captured on pc1, pc2, and eve after the attack. Screenshots were saved as <device\_name>\_arp4.png.
   * Below are the screenshots.



1. **Wireshark Capture:**
   * Wireshark captured network traffic during the spoofing attack, storing the data as capture2.pcapng in the /shared directory.

**Observations**

* **Initial ARP Cache (Steps 6 & 7):** The initial ARP cache entries for each device reflected the learned MAC addresses of other devices on the same network segment based on the ping tests.
* **Post-Spoofing ARP Cache (Step 12):** After the ARP spoofing attack from eve, pc1's ARP cache incorrectly mapped pc2's IP address to eve's MAC address. This indicates a successful spoofing attack, where pc1 believes eve is the legitimate path to reach pc2. The ARP cache entries on pc2 and eve remained unchanged.
* **Captured Traffic (Steps 13 & Analysis):** Analyzing the captured traffic in Wireshark (capture1.pcapng and capture2.pcapng) can reveal details of the communication between devices before and after the ARP spoofing attack. This might include the attacker's manipulation of ARP packets and potential data interception. Below are the screenshots taken this analysis



**Conclusion**

This lab exercise demonstrated network traffic sniffing and ARP spoofing attacks in a simulated environment. By capturing ARP cache entries and analyzing network traffic, we observed how an attacker can manipulate communication paths and potentially intercept sensitive information. This highlights the importance of network security measures like network segmentation, port security, and encrypted communication protocols to mitigate such attacks.