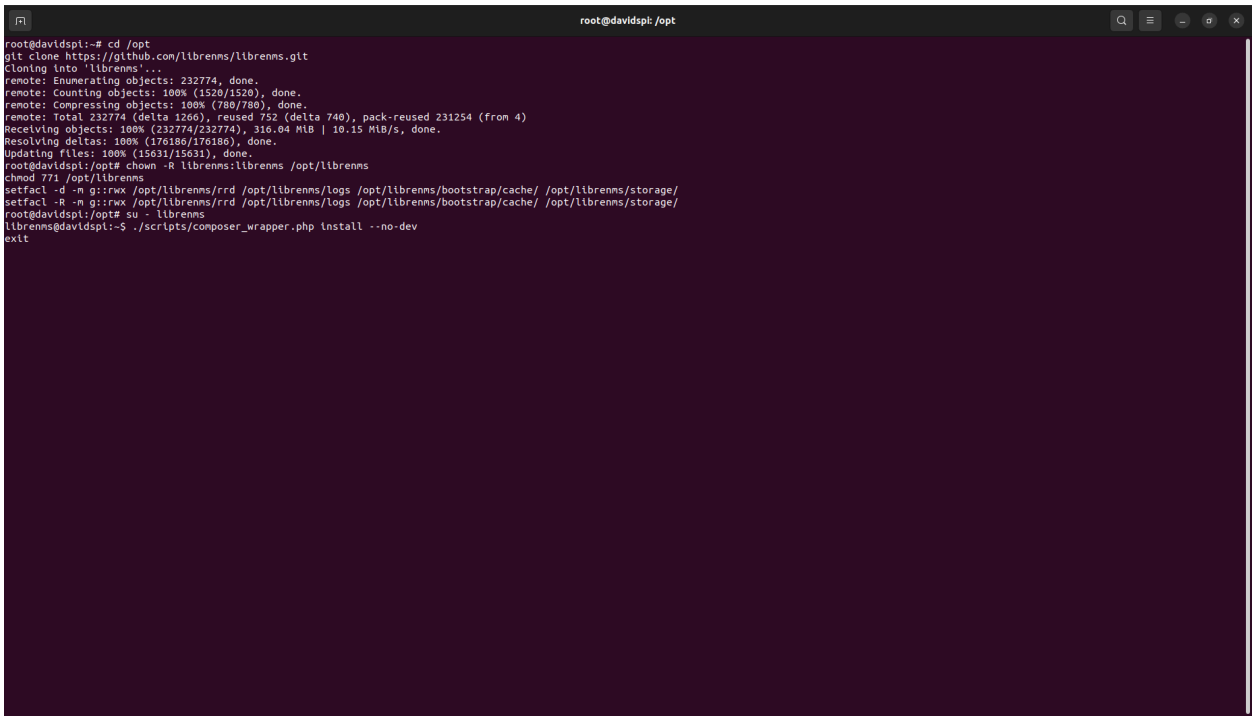
CDP Flood Attack

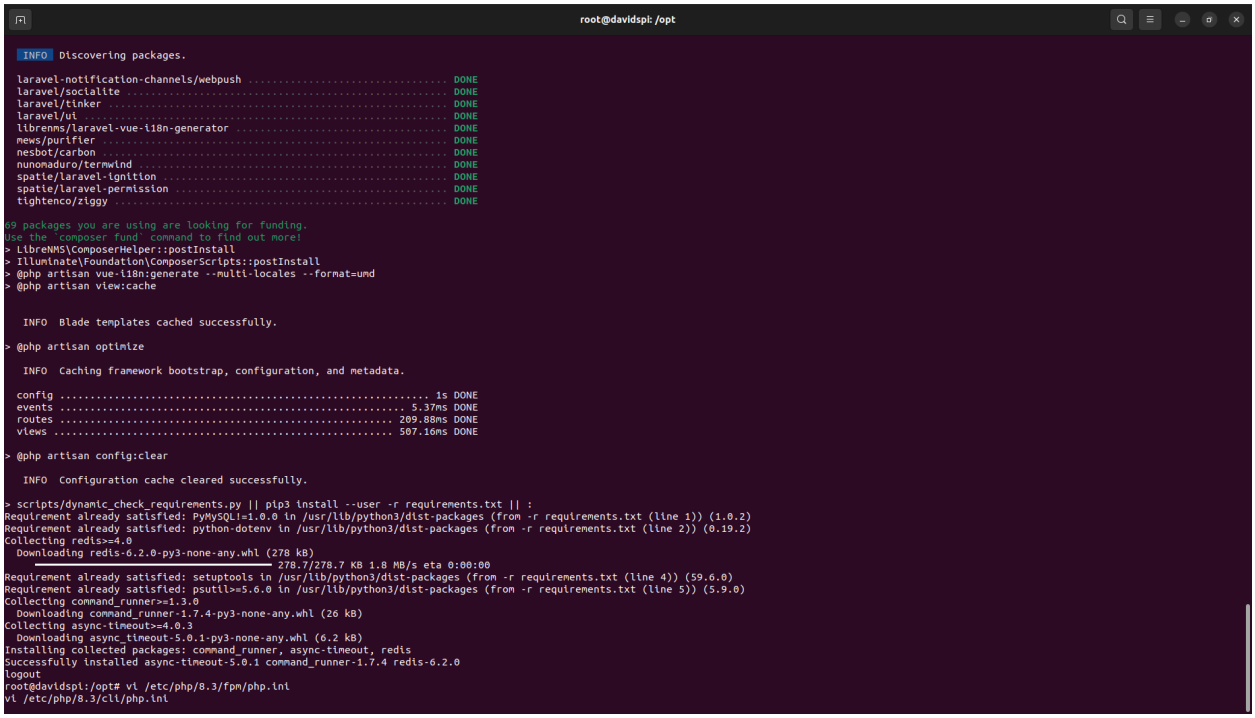
Objective: In this Kali Linux lab I used Yersinia to launch a CDP flood attack which filled the CDP neighbor table with fake entries. I set up a SPAN port to mirror traffic and used Wireshark to capture and analyze the packets, allowing me to see the flood in real time. I used LibreNMS to monitor the system performance and noticed a spike in CPU usage during the attack. To stop the issue, I used SSH from LibreNMS to access the affected device and shut down the interface receiving the CDP packets. This lab shows how different tools can be used together to detect, analyze, and respond to network issues.

Equipment: (2) Raspberry Pi, Yersinia, Wireshark, Cisco 3560 Switch, and LibreNMS

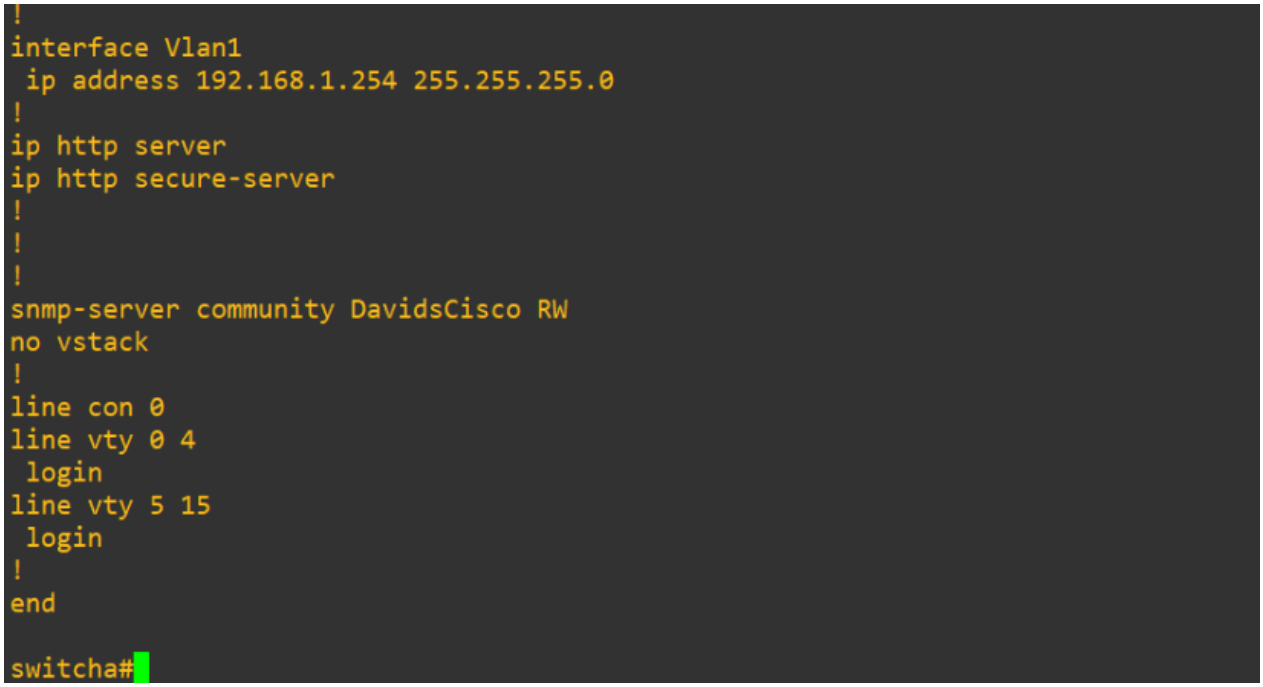
Steps:

1. I ran the command **sudo apt update && upgrade -y** on my Ubuntu Raspberry Pi to ensure it's up to date
2. I ran the command **git clone https://github.com/librenms/librenms.git** to install LibreNMS and all the necessary dependencies



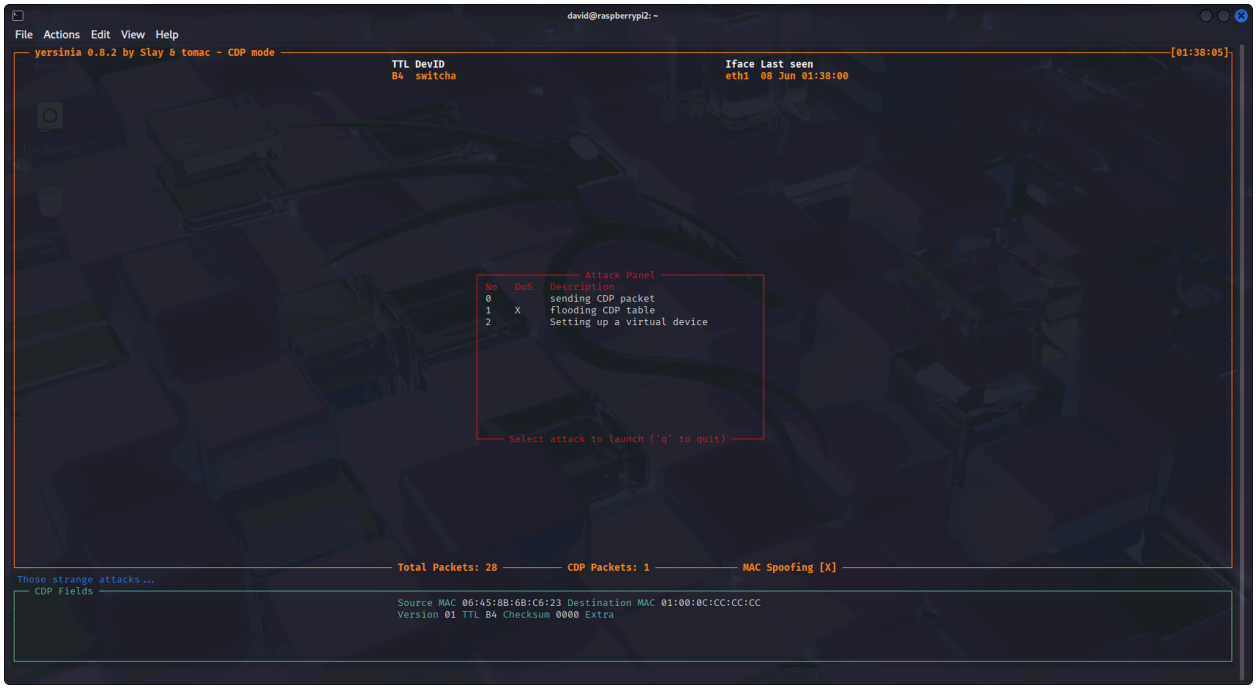


3. **Configure the SNMP community and Interface VLAN 1** on my 3560 Switch



4. Set up the attack **Yersinia on my Kali Linux Raspberry Pi** (making sure my eth1 port picks up the switch)

5. **Yersinia detects the Cisco 3560 Switch successfully** and I prepare to flood the **CDP neighbor Table**

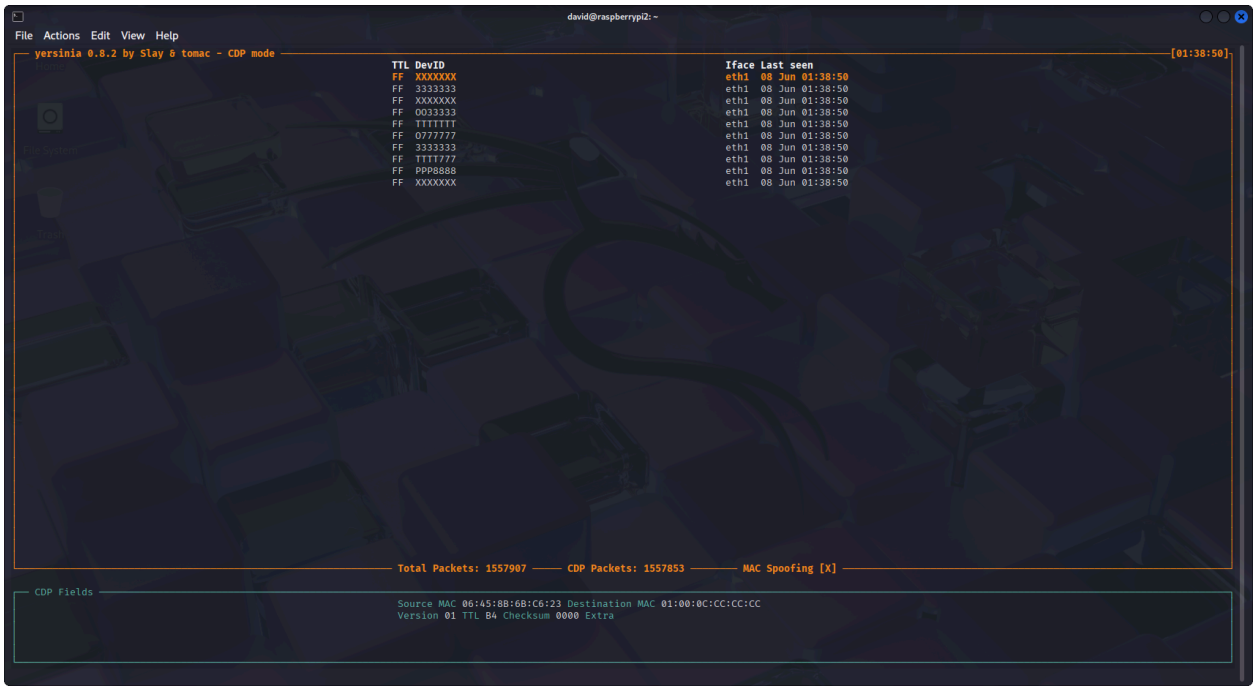


6. Before I do the attack I **configure basic SPAN** to simulate how a **TAP** would work in the Network Engineering field

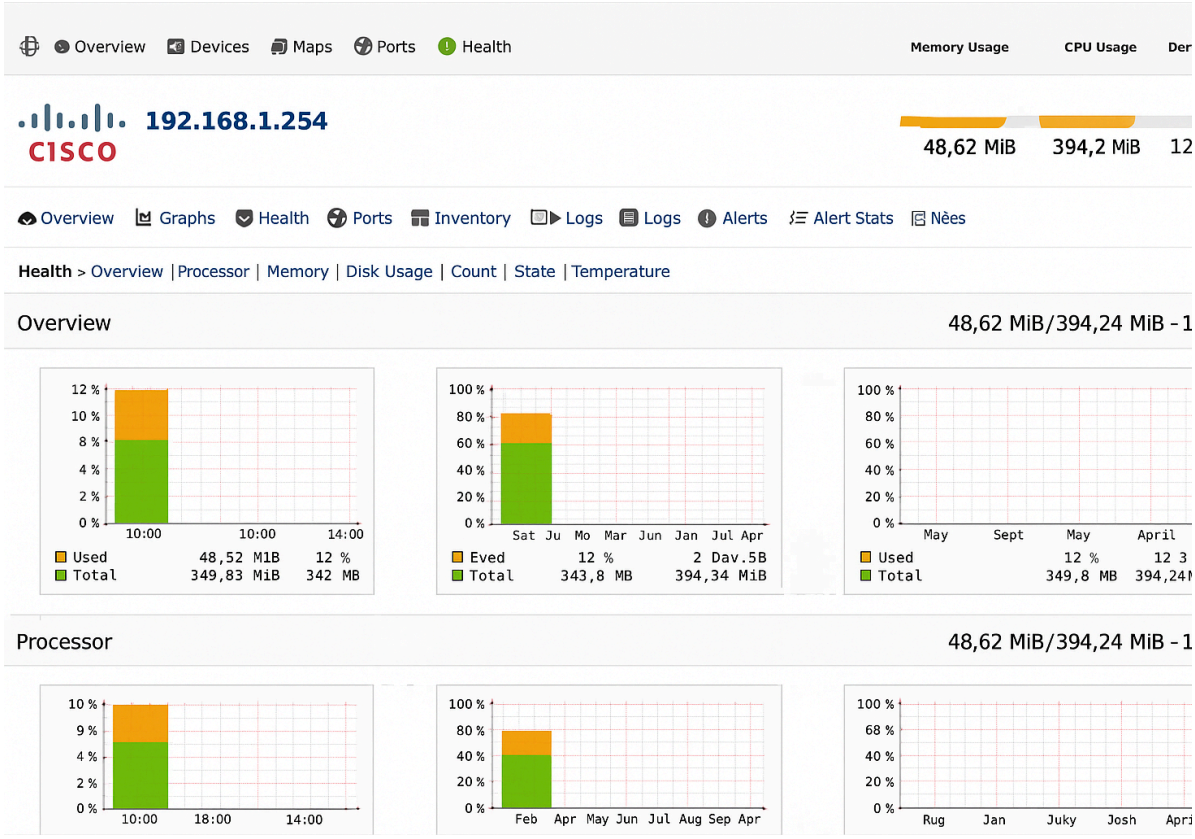
```
switcha(config)#monitor session 1 source interface gigabitEthernet 1/0/2
switcha(config)#monitor session 1 destination interface gigabitEthernet 1/0/3
switcha(config)#do show monitor session 1
Session 1
---------
Type                      : Local Session
Source Ports              :
    Both                  : Gi1/0/2
Destination Ports       : Gi1/0/3
    Encapsulation       : Native
          Ingress       : Disabled


switcha(config)#
```

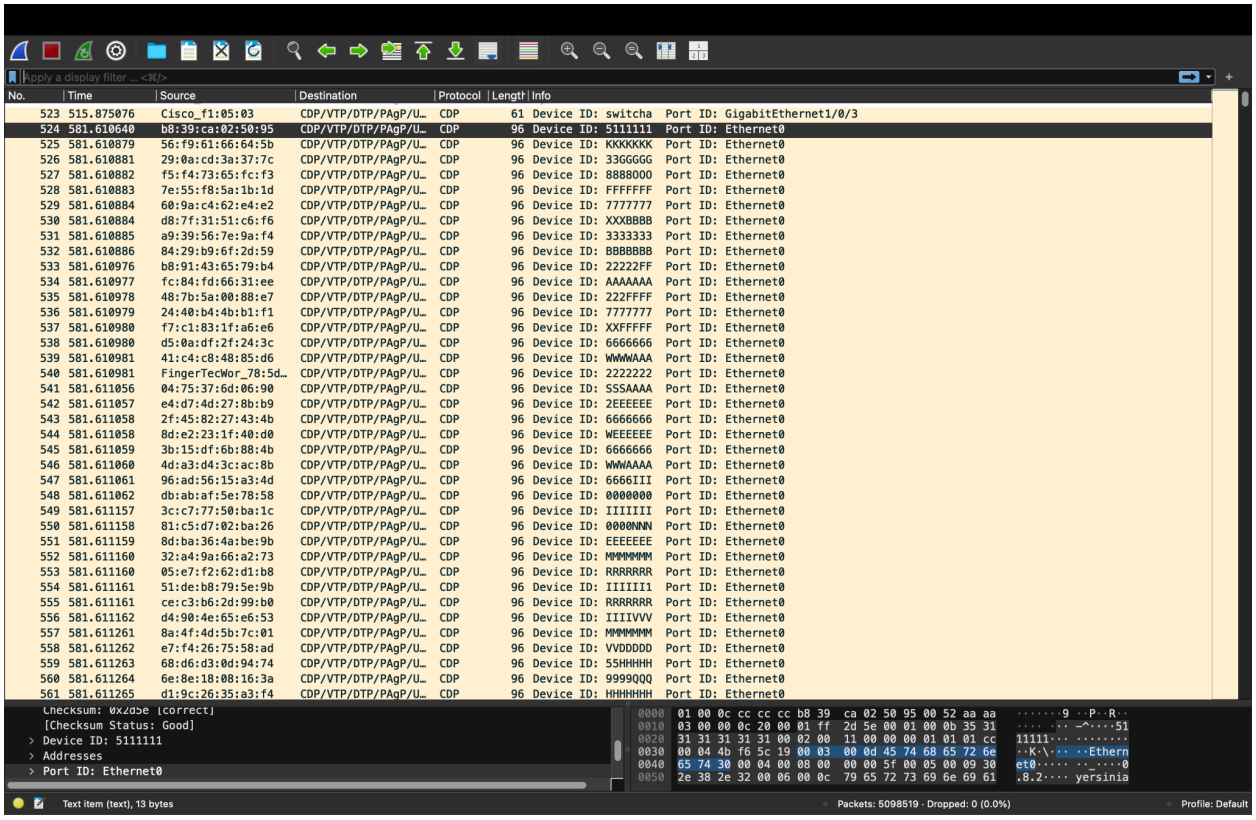
7. **I start the attack**



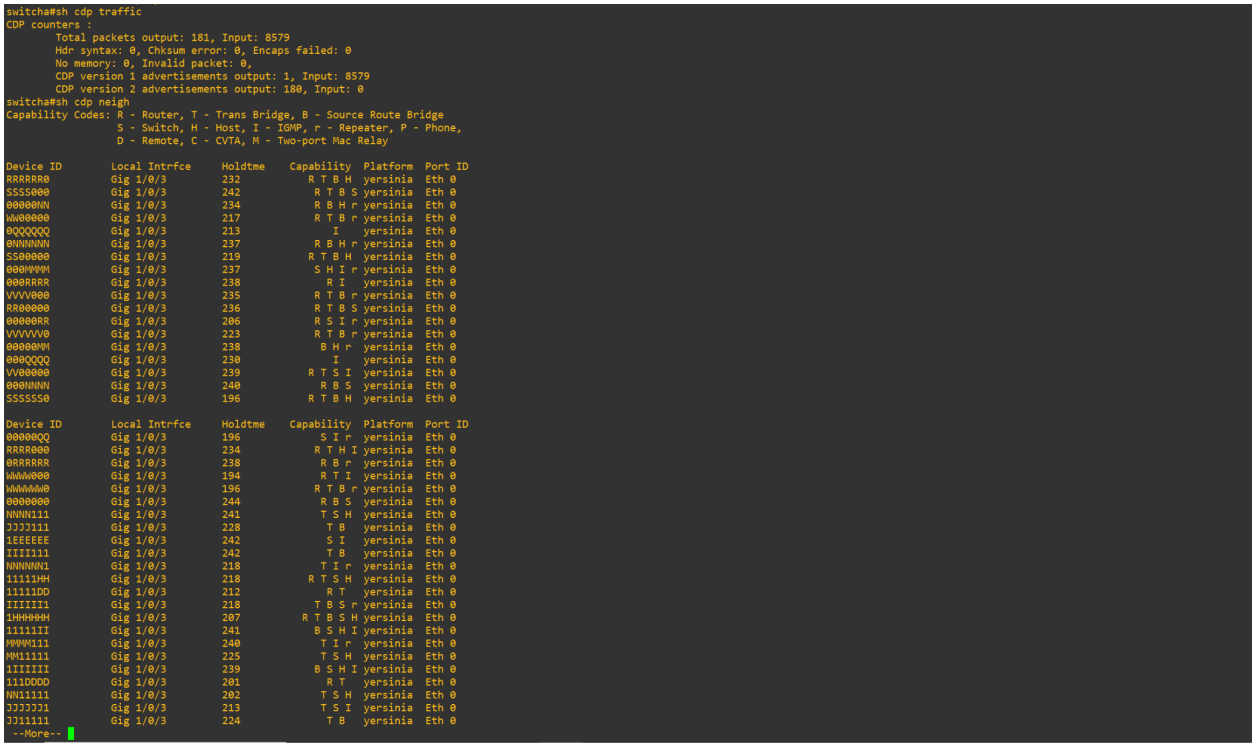
8. I go to my **LibreNMS graph and see that there is unusual spike**



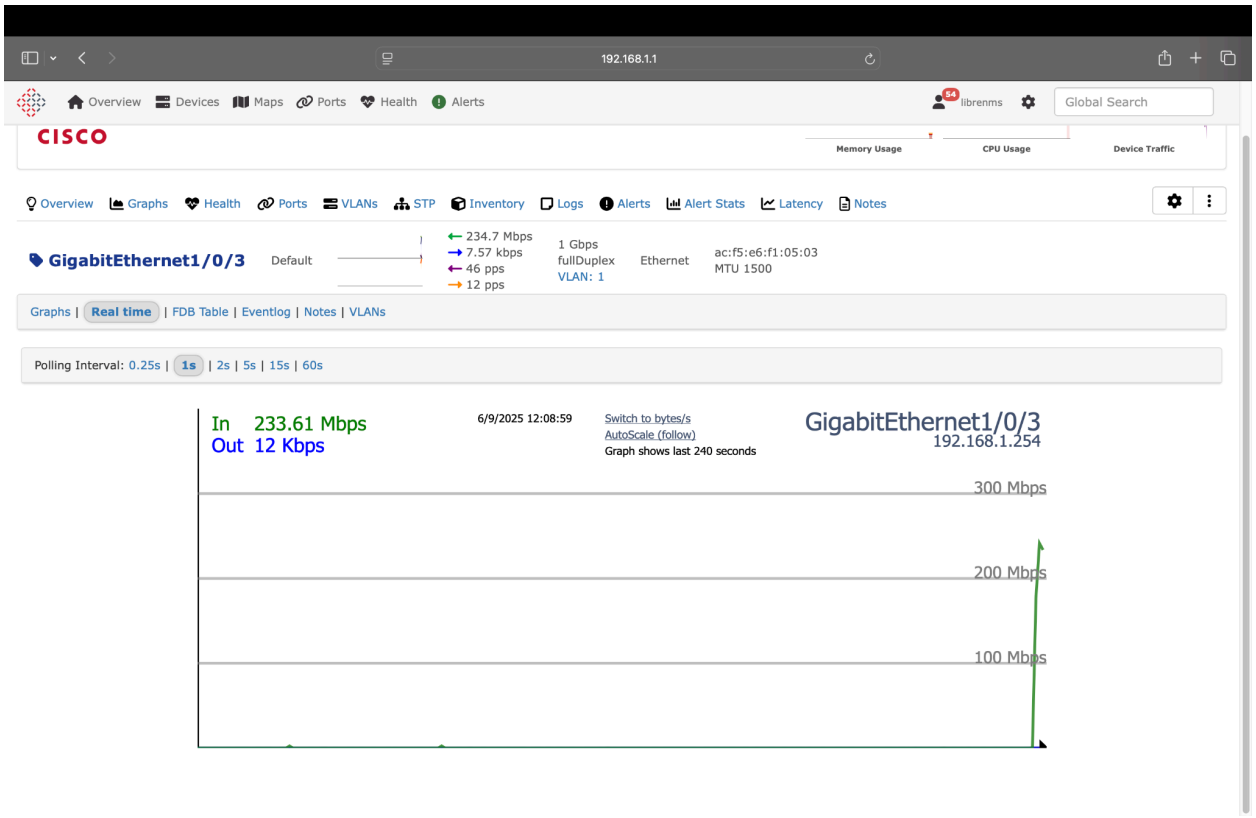
9. My **SNAP is working with Wireshark** to help me identify what I'm getting attacked with



10. I **use my switch to see if my CDP table is getting flooded**



11. I use **LibreNMS to identify what interface can possibly be causing this**, then I see my G1/0/3 interface has an unusual bandwidth input



12. I use **LibreNMS built in SSH function to quickly shutdown the interface** (as you can see from the background, there is no data flowing anymore)