Unifi 8-port switch

Port 1 : 4001-T
Boct 2: 4002-T
Port 3.4001-U, 4002-T
Ports 4-16 : 10- T, 20 -T, 4001-T, 4002-T, 1-U
T- tagged VLANS
U- untagged VLANS

Port Mirroring Configuration (on Unifi 8-port switch)

Unifi Controller (web app or Cloud Key) or SSH access to switch

Define port mirroring settings

1. Login to Unifi Controller

2. Go to Settings → Networks → Advanced Features → Port Mirroring
3. Set "Source Ports": All the ports connected to devices/WAN you want to monitor (e.g., port 4-15).
4. Set "Destination Port": The port where your laptop parser is connected (e.g., port 16).
5. Apply and save.

The switch is configured to mirror traffic from selected ports to the parser device

Capture and Analyze Traffic

- Connect your parser device running HEAT application to the mirrored port on Unifi 8-port switch (port 16)
- Connect the devices you want to monitor to the ports 4-15 on Unifi 8-port switch via USB Ethernet adaptor and generate some traffic
- Start the parser application to capture network traffic. This will observe all flows, including DHCP, DNS, NAT, and routing protocol packets.
- The parser will analyze the traffic and generate a VyOS configuration file that matches your current network setup

Testing VyOs configuration

- Port 1: connected to switch for LAN (Port 4 on switch)
- Port 2: connected to switch for WAN (Port 3 on switch)
- Port 3 & Port 4: Can be used for additional LAN segments, DMZ, or other network needs.
- Console Port: Connect a serial cable from your computer to the console port for initial setup
- USB Ports: Use for installing VyOS from a USB stick or for transferring files.

Install VyOS on the Lanner Box (if needed)

- Download the VyOS ISO image and write it to a USB stick.
- Boot the Lanner box from the USB and install VyOS.

Label the interfaces appropriately

Create a Bootable USB Drive

Load and Apply the Generated Configuration

- Connect the lanner box to your network

Copy the generated VyOS configuration file to the Lanner box

After generating your VyOS configuration file (typically named `config.boot`) on your computer, copy it to a FAT32-formatted USB stick.

IInsert the USB stick into a USB port on the Lanner box (either USB Port 1 or 2).

- Log in to VyOS using the console, SSH, or directly with a keyboard/monitor.
- Mount the USB stick. VyOS usually mounts USB drives under `/media/`, but you may need to check with `lsblk` or `sudo fdisk -l` to find the correct device (e.g., `/dev/sdb1`).

```
sudo mkdir /mnt/usb

sudo mount /dev/sdb1 /mnt/usb
```

Copy your configuration file from the USB stick to the VyOS config directory:

```
sudo cp /mnt/usb/config.boot /config/config.boot
```

Unmount the USB stick:

```
sudo umount /mnt/usb
```

Load the Configuration in VyOS:

Enter configuration mode:

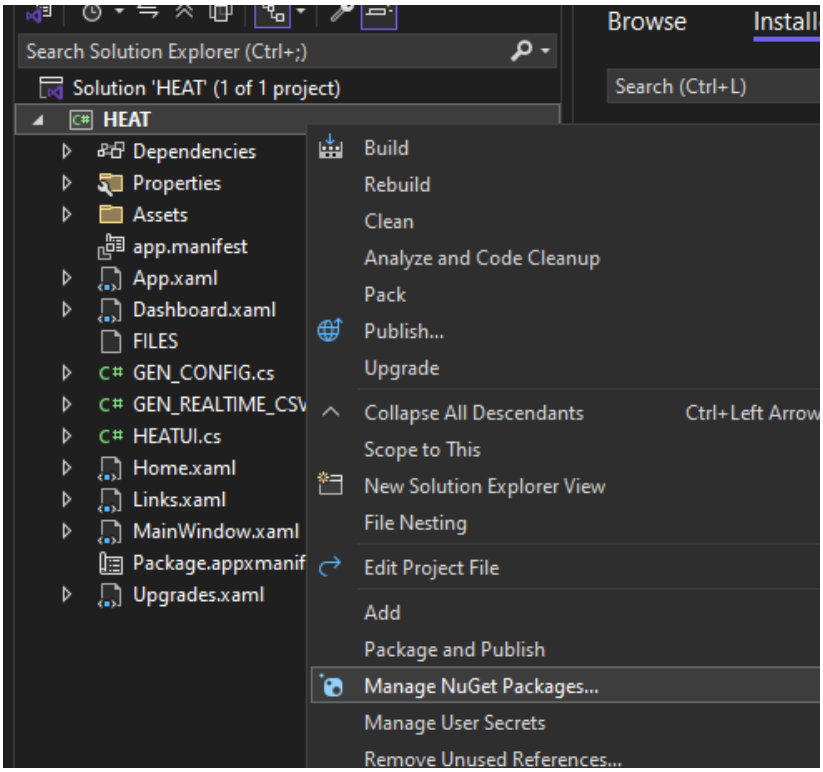
```
configure
load /config/config.boot
commit
Save
exit
```

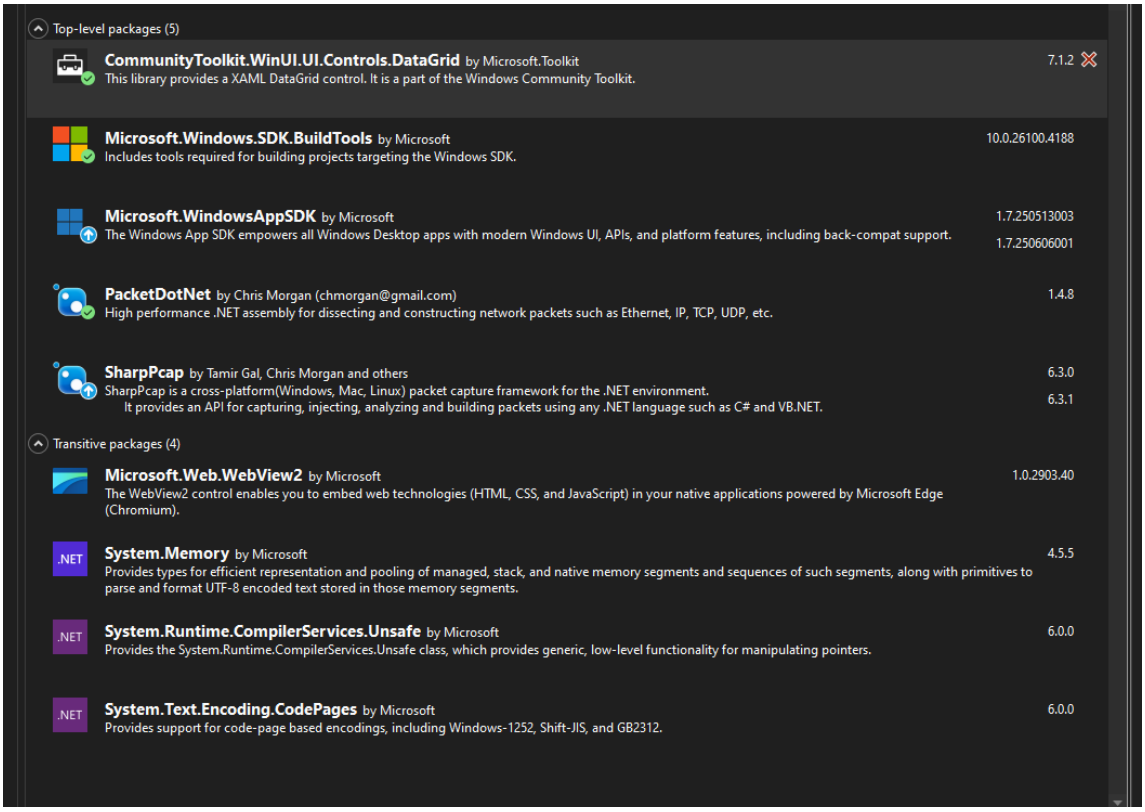
This will load it onto the Lanner box

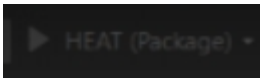
Lanner Box with VyOS (for testing VyOs)

The Lanner box serves as the physical router/firewall running VyOS. The Lanner box is connected to the network in the similarly to the previous firewall:

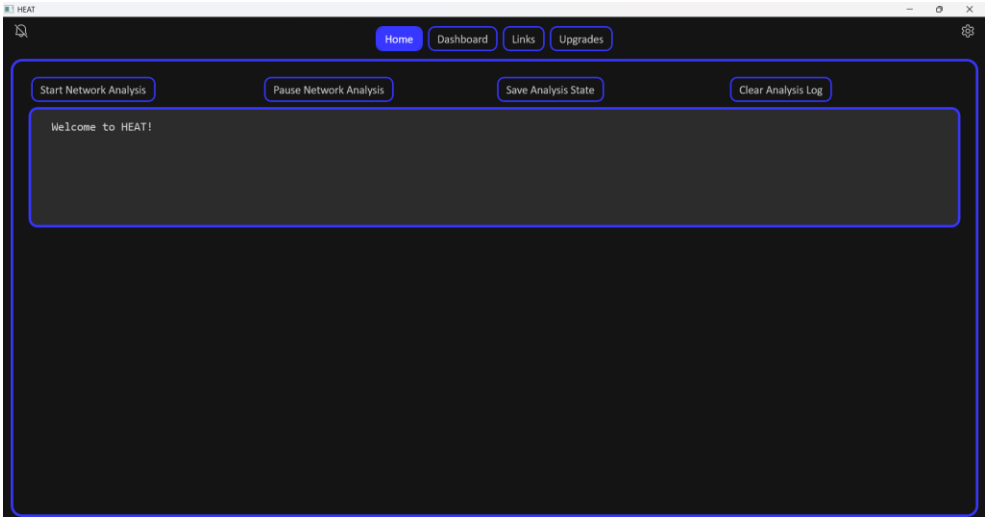
- Disconnect your pfSense firewall and replace it with the VyOS-powered Lanner box.
- Ensure all network devices are connected as before.
- Use your parser device to continue monitoring traffic and verify that the new setup matches your previous configuration and that all services are working as expected.

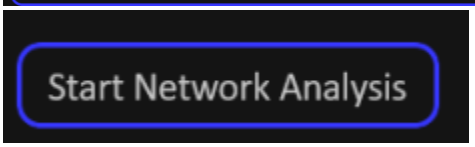
1. Connect the tap port ethernet out to your computer or ethernet to usb c adapter for mobile.
2. Download the file from github and extract it.
3. Open visual studio as an administrator.
4. Start a new project in visual studio making the APP folder the root folder and the solution HEAT within.
5. Right click the HEAT solution and click manage NuGet Packages
6. 

7. Make sure you have these packages


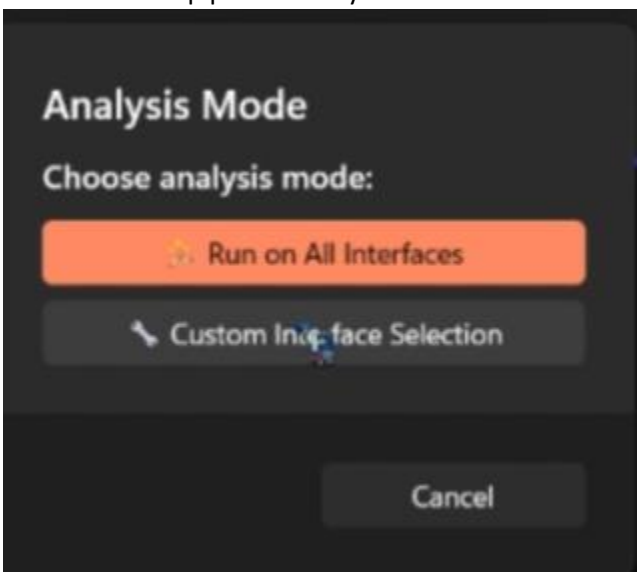
8. Click Run as a package 

9. You will now see this

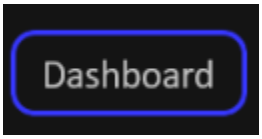


10. 

11. Run it on the tap port and any combination of networks



12. While the traffic analysis is happening connect your virtual devices and generate traffic (see next page for detail).
13. 

when you're done with the traffic generation.
14. 

15. 

16. Your files will now be on the desktop

Virtual Machine Testing + Functionality Guide

A: Vlan Tagging

1: Download and install Lubuntu: https://lubuntu.me/downloads/

2: Set up the virtual machines through whatever virtualization software that you use

3: Set up a virtual Ethernet port with a VLAN tagging. Do this by clicking on virtual box global tools, then host network manager. Then under Nat networks, click Create, and name it whatever you'd like. Then, under the setwork settings, attach the virtual machine to the new network.

4: Find a network interface, type in **ip link** in your terminal. Pick any one besides LO and take note of it. Then type these commands into your terminal to ensure your VM has what's required for the future steps:

sudo apt update

sudo apt install vlan -y

sudo modprobe 8021q

5: In the Lubuntu terminal, type in this command: ls /etc/netplan/. There should be a yaml file there. Once you have the name of that file (in my case 01-network-manager-all.yaml), type in sudo nano /etc/netplan/name-of-.yaml-file-here.

6: Once the file is open, update your network file so that it looks like this:

```
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp0s3: {}  (Instead of enp0s3 replace it and all instances of it with the earlier network interface)
  vlans:
    enp0s3.100: (your network interface.100)
      id: 100
      link: enp0s3  (your network interface)
```

addresses: [192.168.100.11/24]

Ensure proper indentation, then save the file by clicking ctrl + O. Then type **sudo netplan apply** into your terminal. This sets up the VLAN tagging on your network interface. Repeat this same process on your second Lubuntu virtual machine, except change 192.168.100.11/24 to 192.168.100.10/24.

7: Then once this is set up, restart your virtual machines. Once restarted, on one VM, ping the other with this command: **ping 192.168.100.10**. If your VM receives a response, then your VM's have their static IP's set up.

8: In order to send VLAN Tagged packets, run this command to install python: **sudo apt install python3-scapy -y**. Once that's installed, run this multi-line command script:

sudo python3 - <<EOF

from scapy.all import *

sendp(Ether()/Dot1Q(vlan=100)/IP(src="192.168.100.10", dst="192.168.100.11")/ICMP(), iface="enp0s3")

EOF

This sends packets to your other VM, ensure that the DST and SRC Ip's are for the correct virtual machine Ip's.

9: In order to receive the VLAN tagged packets, you need to run this command line on your other VM: **sudo tcpdump -i enp0s3 -nn -e vlan -w vlan_test.pcap**. This will generate a .pcap file, which can then be analyzed by wireshark. Or you can run this command to analyze it on the VM itself: **sudo tcpdump -nn -e -r vlan_test.pcap vlan**

10 (Optional): Set up shared folders. For Oracle Virtual box ensure your VM's are closed, then navigate to the VirtualBox manager and click on your virtual machine's settings. Then navigate to shared folders, and click the plus to add a new one. Then navigate to a folder you would like to be shared, and click on make permanent, and auto mount, then select ok. Then start your VM, and in the top list of actions, click on Devices, then Insert CD image. Then run any installers it asks you to, and run these commands:

sudo apt update

sudo apt install build-essential dkms linux-headers-$(uname -r)

sudo mkdir /media/cdrom

sudo mount /dev/cdrom /media/cdrom

sudo /media/cdrom/VBoxLinuxAdditions.run

Then Reboot your VM. Then run this final command to give yourself access to the folder:

sudo usermod -aG vboxsf $(whoami)

Once again, reboot your VM. Then the folder will be right there under your devices next to your shared drive.

B: VyOS DHCP Configuration

1: In all of your virtual machines, in network settings, set adapter 2 (or 3 for the VyOS ones) to internal network.

2: In your VyOS virtual machine type in **configure,** to enter configuration mode

3: Then type in the command line interface: **set interfaces ethernet eth1 address 192.168.100.1/24**. Then just press enter.

4: Save and commit your changes by typing in: **Save** then **Commit** then **Exit**. Restart your VM

5: Enter Configuration mode again with **configure,** then type in these commands.

set service dhcp-server shared-network-name LABNET authoritative

set service dhcp-server shared-network-name LABNET subnet 192.168.100.0/24 default-router 192.168.100.1

set service dhcp-server shared-network-name LABNET subnet 192.168.100.0/24 name-server 192.168.100.1

set service dhcp-server shared-network-name LABNET subnet 192.168.100.0/24 lease 86400

set service dhcp-server shared-network-name LABNET subnet 192.168.100.0/24 range 0 start 192.168.100.100

set service dhcp-server shared-network-name LABNET subnet 192.168.100.0/24 range 0 stop 192.168.100.200

Then once again, **commit, save, exit.**

6: Restart your VyOS virtual machine, then run this command: **show interfaces**. Note your eth1 Ip address, it should be 192.168.100.1/24. If not repeat the previous steps.

7: Load up your lubuntu virtual machines, ensure they have a network adapter set to internal network. Then type these commands into your terminal:

sudo dhcpcd -k enp0s8

sudo dhcpcd enp0s8

This sends a request to the DHCP server to assign the enp0s8 interface with an IP in the range of 192.168.100.100-200.

Test this with the command: **ip addr show enp0s8**

Ensure it has an IP in that range.

8: Now, run this command on one of your virtual machines: **ping 192.168.100.1**. As long as your VyOS virtual machine is running, you should be able to ping it.

9: Load up your second Lubuntu machine. Then run these commands:

sudo dhcpcd -k enp0s8

sudo dhcpcd enp0s8

Now, both of your Lubuntu machines will have a VyOS DHCP assigned, take note of them for our next step.

10: Try to ping the other virtual machines IP with **ping <Other VM's Address Here>**. If they go through, then perfect, if not you may need to troubleshoot.

11: Running the previously mentioned commands to assign your enp0s8 interface to a new IP will cause traffic to be generated that can then be captured by your program. (When the program is run on Linux).