

Network Scanner:

Tech Stack:Python

Threading (divided into threads not to interfere with the main thread i.e used for GUI)

Linux Distribution:Daemon(MacOs)

Libraries:subprocess,threading,platform,tkinter

Code:

```
import subprocess
#import platform
import threading
import tkinter as tk #(GUI)
from tkinter import scrolledtext

# single IP to check if it's active
def ping_ip(ip):
    try:
        param = '-c'
        output = subprocess.check_output(['ping', param, '1', '-W', '2', ip],
        stderr=subprocess.STDOUT)
        return True
    except subprocess.CalledProcessError:
        return False
    except Exception as e:
        print(f"Error pinging {ip}: {e}")
        return False

# Scan a subnet for active devices
def network_scan(subnet):
    active_ips = []
    for i in range(1, 255): # Scan IPs from 254
        ip = f"{subnet}.{i}"
        if ping_ip(ip):
            print(f"Active: {ip}")
            active_ips.append(ip)
    return active_ips

# Here the network scan is kept in a thread to keep GUI responsive(to prevent from the
# interference as it was affecting the sacnning process)
def start_scan():
    subnet = subnet_entry.get().strip()
    if not subnet or len(subnet.split('.'))!=3:
        results_box.delete(1.0, tk.END)
        results_box.insert(tk.END, "Invalid subnet format! Use format: 192.168.1\n")
        return

    results_box.delete(1.0, tk.END)
    results_box.insert(tk.END, "Scanning...\n")

def threaded_scan():
    active_devices = network_scan(subnet)
    root.after(0, update_results, active_devices)

    threading.Thread(target=threaded_scan, daemon=True).start()

# Updating the results box with scan results
def update_results(active_devices):
    results_box.delete(1.0, tk.END)
```

```

if active_devices:
    results_box.insert(tk.END, "Active devices found:\n")
    results_box.insert(tk.END, "\n".join(active_devices) + "\n")
else:
    results_box.insert(tk.END, "No active devices found.\n")

# GUI setup
root = tk.Tk()
root.title("Network Scanner")

tk.Label(root, text="Enter Subnet (e.g., 192.168.1)").pack() #example of subnet: 192.168.1(Here used 192.168.29 as subnet)
subnet_entry = tk.Entry(root)
subnet_entry.pack()

scan_button = tk.Button(root, text="Scan", command=start_scan)
scan_button.pack()

results_box = scrolledtext.ScrolledText(root, width=50, height=20)
results_box.pack()

root.mainloop()

```

Test:

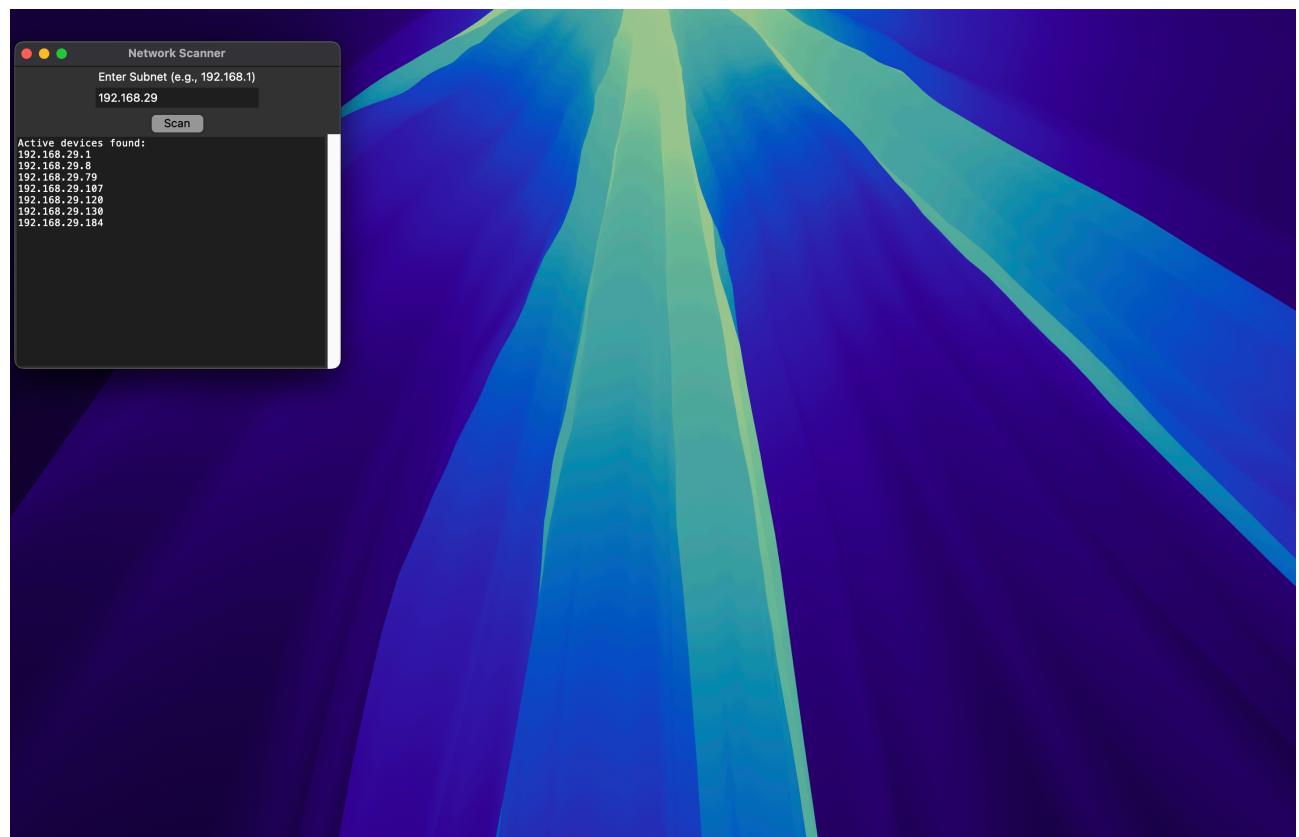


Fig:Test Window