Network Sniffer Project - Complete Guide

CodeAlpha Cybersecurity Internship - Task 1

📌 What is This Project?

This is a **Network Packet Sniffer** - a tool that captures and analyzes internet traffic moving through your computer network. Think of it like a "network microscope" that lets you see all the invisible data packets traveling between your computer and the internet.

🔧 What You Need to Run It

1. Software Requirements:

- **Python 3.x** (Download from python.org)
- **Scapy library** (Network packet tool)
- **Npcap** (Windows packet capture driver)

2. Installation Steps:

Step 1: Install Python

- Go to https://python.org
- Download Python 3.x
- During installation, CHECK "Add Python to PATH"

Step 2: Install Required Tools (Run as Administrator):

- pip install scapy

Step 3: Install Npcap

- Download from: https://npcap.com
- Install with **WinPcap compatibility mode**

- **RESTART** computer after installation

🚀 How to Use the Sniffer

Basic Commands:

1. List available network interfaces:

- python network_sniffer.py -l

2. Capture 10 packets (auto-select interface):

- python network_sniffer.py -c 10

3. Capture with specific interface:

- python network_sniffer.py -i "Wi-Fi" -c 15

4. Generate test traffic:

- python test_traffic.py

📊 Understanding the Output

When you run the sniffer, you'll see information like:

text

[📦] Packet #1

Time: 19:19:40

Summary: Ether / ARP who has 192.168.1.1 says 192.168.1.4

ARP: 192.168.1.4 → 192.168.1.1

Size: 42 bytes

What This Means:

- 📦 **Packet #1**: First captured packet
- **Time**: When it was captured
- **Summary**: Brief description

- **From → To**: Source and destination IP addresses
- **Protocol**: Type of network communication
- **Size**: How big the packet is

🔍 What Types of Packets You'll See

1. ARP Packets

- **Purpose**: Find devices on your local network
- **Example**: "Who has 192.168.1.1?"
- **Meaning**: Your computer looking for the router

2. ICMP Packets

- **Purpose**: Ping/troubleshooting packets
- **Example**: "192.168.1.4 → 8.8.8.8"
- **Meaning**: Testing connection to Google DNS

3. TCP Packets

- **Purpose**: Web browsing, emails, file transfers
- **Example**: "443 → 55335"
- **Meaning**: HTTPS website traffic
- **Common Ports**:
    - 80 = HTTP (normal websites)
    - 443 = HTTPS (secure websites)
    - 25 = Email
    - 53 = DNS (website names to IP addresses)

4. UDP Packets

- **Purpose**: Video calls, online games, DNS
- **Example**: "DNS query to 8.8.8.8:53"
- **Meaning**: Looking up a website address

⚠️ Important Notes

Must Run as Administrator!

- On Windows, **right-click Command Prompt**
- Select **"Run as administrator"**
- Otherwise, packet capture won't work

Two Windows Method:

1. **Window 1** (Admin): Run the sniffer
- python network_sniffer.py -c 20
2. **Window 2** (Normal): Generate traffic
- python test_traffic.py

Troubleshooting:

- **No packets?** Open a website or run ping google.com
- **"Permission denied"?** Run as Administrator
- **"No interfaces found"?** Install Npcap and restart

🎯 What This Project Teaches

Cybersecurity Concepts Learned:

1. **Packet Analysis**: Reading network traffic
2. **Protocol Understanding**: TCP, UDP, ICMP, ARP
3. **Network Security**: How data travels online
4. **Tool Usage**: Command-line security tools
5. **Traffic Filtering**: Capturing specific types of data

Technical Skills Gained:

- Python programming
- Network protocol knowledge

- Command-line interface usage
- Problem-solving skills
- Network troubleshooting

✅ Project Success Checklist

- Python installed correctly
- Scapy library installed
- Npcap installed with WinPcap mode
- Running Command Prompt as Administrator
- Can list network interfaces (-l flag)
- Can capture packets
- Can identify different protocols
- Can understand packet information
- Can generate test traffic

🔒 Privacy & Legal Notice

Important: Use Responsibly!

1. **Only capture your own traffic** on your own network
2. **Don't capture others' data** without permission
3. **Educational use only** - for learning purposes
4. **Respect privacy laws** in your country
5. **Use on your home network**, not public Wi-Fi

What You CAN Do:

- Monitor your own computer's traffic
- Learn how networks work
- Test your own applications

- Debug network problems

What You CAN'T Do:

- Capture others' data without consent
- Use on networks you don't own
- Violate terms of service
- Break any laws

📚 Learning Resources

For Beginners:

- Wireshark (visual network analyzer)
- TryHackMe (free cybersecurity courses)
- NetworkChuck YouTube channel

Next Steps:

1. Try filtering specific traffic (only HTTP, only DNS)
2. Save captures to a file
3. Analyze packet contents
4. Build a simple GUI interface
5. Learn about encryption and HTTPS

💡 Pro Tips

1. **Start small**: Capture 5-10 packets first
2. **Use filters**: -f "tcp port 80" for only web traffic
3. **Save output**: Redirect to file: python network_sniffer.py > capture.txt
4. **Experiment**: Open different websites while sniffing
5. **Compare**: Run ping in another window while sniffing

🎓 Conclusion

You've successfully built a **working network sniffer** that can:

- Capture real network traffic
- Identify different protocols
- Show source and destination
- Disp'lay packet timing and size
- Work with Windows and Npcap

This completes CodeAlpha Task 1: Basic Network Sniffer!