🚀 NetSnoop Complete Setup Guide

"Born to Track"

From Windows to Web Dashboard in 15 Minutes

📋 What You'll Achieve

By the end of this guide, you'll have:

- ✅ **WSL Linux environment** running on Windows
- ✅ **NetSnoop installed** and configured
- ✅ **Real-time monitoring** active in terminal
- ✅ **Web dashboard** accessible in your browser
- ✅ **Live system alerts** showing CPU, memory, and process activity

Total Time: ~15 minutes | **Skill Level:** Beginner-friendly

1 Install WSL (Windows Users)

1.1 Choose Your WSL Version

Option A: Quick WSL Setup (Recommended for most users)

```
# Open PowerShell as Administrator
# Press Win + X, select "Windows PowerShell (Admin)"

# Install WSL with Ubuntu in one command (Windows 10 version 2004+)
wsl --install

# This automatically installs WSL2 with Ubuntu
```

Option B: Manual WSL1 Setup (If you prefer WSL1)

```
# Open PowerShell as Administrator
# Enable WSL feature only
dism.exe /online /enable-feature /featurename:Microsoft-Windows-
Subsystem-Linux /all /norestart

# Restart your computer
# Then install Ubuntu from Microsoft Store (continues with WSL1)
```

Option C: Already have WSL1? Keep it!

```
# Check your current WSL version
wsl --list --verbose

# If it shows Version 1 and NetSnoop works fine, no need to upgrade!
```

1.2 Restart Your Computer

⚠️ **Important:** Restart now - required for WSL to work properly

2 Install Ubuntu Linux

2.1 Install Ubuntu from Microsoft Store

1. **Open Microsoft Store** (search "Microsoft Store" in Start Menu)
2. **Search for "Ubuntu"**
3. **Install "Ubuntu 22.04 LTS"** (recommended)
4. **Click "Get" or "Install"**

2.2 Launch Ubuntu for First Time

1. **Search "Ubuntu" in Start Menu** and click it
2. **Wait for installation** (this takes 2-3 minutes)
3. **Create username** (lowercase, no spaces)

```
Enter new UNIX username: yourusername
```

4. **Set password** (you won't see characters as you type)

```
New password: [your_secure_password]
Retype new password: [your_secure_password]
```

2.3 Install Essential Python Tools

```
# Update package lists first
sudo apt update

# Install Python and pip (REQUIRED before creating virtual environments)
sudo apt install python3 python3-pip python3-venv -y

# Upgrade system packages
sudo apt upgrade -y

# Install additional useful tools
sudo apt install curl wget git -y
```

🔧 **Why install pip in base system?**

- Virtual environments copy pip from the base system
- Without base pip, virtual environments can't install packages
- This is a one-time setup requirement

3 Verify Your Environment

3.1 Check Your WSL Setup

```
# Check which WSL version you're using
wsl --list --verbose

# Check Python version (should be 3.7+)
python3 --version

# Check pip version
pip3 --version
```

```
Expected Output:
NAME STATE VERSION
* Ubuntu Running 1 <-- WSL1 (perfectly fine!)
or
* Ubuntu Running 2 <-- WSL2 (also great!)

Python 3.10.12
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

💡 **Note:** NetSnoop works great on both WSL1 and WSL2! No need to upgrade if WSL1 is working for you.

3.2 Create Project Directory

```
# Create and navigate to project folder
mkdir ~/netsnoop-project
cd ~/netsnoop-project

# Verify you're in the right place
pwd
```

```
Expected Output:
/home/yourusername/netsnoop-project
```

4 Install NetSnoop

4.1 Create Virtual Environment (Recommended)

```
# Create virtual environment
python3 -m venv netsnoop-env

# Activate virtual environment
source netsnoop-env/bin/activate
```

```
# Your prompt should now show (netsnoop-env)
```

Your terminal prompt should look like:
`(netsnoop-env) yourusername@computername:~/netsnoop-project$`

4.2 Install NetSnoop from PyPI

```
# Install specific NetSnoop version 0.1.4
pip install netsnoop==0.1.4

# Verify correct version installation
pip show netsnoop
```

```
Expected Output:
Name: netsnoop
Version: 0.1.4
Summary: Real-time Linux system monitoring tool
...
```

🎯 **Why version 0.1.4?**

- This is the stable, tested version used in development
- Ensures compatibility with this setup guide
- Avoids potential issues with newer untested versions

4.3 Initialize NetSnoop

```
# Run initialization script
netsnoop-init
```

```
Expected Output:
🚀 NetSnoop v0.1.4 Initialization Started
✅ Created anomalies.csv
✅ Created netsnoop_persistent.txt
✅ Added sample data
✅ Setup complete! Ready to monitor.
```

4.4 Verify Files Created

```
# Check created files
ls -la

# You should see:
# anomalies.csv
# netsnoop_persistent.txt
```

5 Start System Monitoring

5.1 Open First Terminal for Monitoring

```
# Make sure you're in the project directory with virtual environment
active
cd ~/netsnoop-project
source netsnoop-env/bin/activate

# Start the monitoring engine
python3 -m netsnoop.acm_monitor
```

```
Expected Output:
[INFO] NetSnoop Monitor Started - v0.1.4
[INFO] Monitoring CPU, Memory, and Process activity
[INFO] Thresholds: CPU=80%, Memory=85%
[INFO] Scanning system every 5 seconds...
[INFO] ✅ System stable - CPU: 15%, Memory: 45%
[DETECTED] 🟡 Process burst: 8 new processes detected
[INFO] ✅ System stable - CPU: 22%, Memory: 48%
```

🎉 **Success!** Your system is now being monitored in real-time.

6 Launch Web Dashboard

6.1 Open Second Terminal Window

Keep the first terminal running! Open a new terminal:

Method 1 - Ubuntu App:

- Click on Ubuntu in Start Menu again (opens new window)

Method 2 - Windows Terminal:

- Press `Ctrl + Shift + T` if using Windows Terminal

6.2 Navigate and Activate Environment

```
# Navigate to project directory
cd ~/netsnoop-project

# Activate virtual environment
source netsnoop-env/bin/activate
```

6.3 Start the Web Dashboard

```
# Launch Streamlit dashboard
streamlit run $(python3 -c "import netsnoop; print(netsnoop.__path__[0] +
'/dashboard.py')")
```

```
Expected Output:
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.x.x:8501
```

6.4 Open Dashboard in Browser

1. **Copy the Local URL:** `http://localhost:8501`
2. **Open your web browser** (Chrome, Edge, Firefox)
3. **Paste the URL** in the address bar
4. **Press Enter**

7 Explore Your Dashboard

7.1 Dashboard Overview

You should now see the NetSnoop web interface with:

📊 Real-time System Metrics

- Current CPU usage percentage
- Memory consumption levels
- Active process counts

📈 Live Charts

- CPU usage timeline
- Memory usage trends
- Process activity graphs

🚨 Alert Feed

- Recent anomalies detected
- Color-coded severity levels
- Timestamp information

7.2 Dashboard Features

- 🔄 **Auto-refresh:** Updates every 2 seconds
- 📅 **Time filters:** View different time ranges
- 🔝 **Severity filters:** Filter by alert type

- 📋 **Export:** Download CSV reports

8 Test NetSnoop with Prototype Tests

8.1 Download Test Scripts

```
# Make sure you're in your project directory
cd ~/netsnoop-project

# Download the test scripts from GitHub
curl -O
https://raw.githubusercontent.com/ChitviJoshi/NetSnoop/main/tests/burst_test.py
curl -O
https://raw.githubusercontent.com/ChitviJoshi/NetSnoop/main/tests/memory_test.py
curl -O
https://raw.githubusercontent.com/ChitviJoshi/NetSnoop/main/tests/cpu_test.py

# Make scripts executable
chmod +x *.py

# Verify download
ls -la *.py
```

8.2 Test CPU Monitoring

Keep your NetSnoop monitor running in Terminal 1!

```
# In Terminal 2 (or a new terminal), activate environment
cd ~/netsnoop-project
source netsnoop-env/bin/activate

# Run CPU stress test
python3 cpu_test.py
```

What to expect:

- **Terminal 1 (Monitor):** Should show CPU spike alerts
- **Dashboard:** CPU usage chart should spike
- **Test Duration:** ~30 seconds of high CPU activity

```
Expected Monitor Output:
[DETECTED] 🔴 CRITICAL: High CPU usage: 89.2%
[DETECTED] 🔴 CRITICAL: High CPU usage: 91.5%
[DETECTED] 🔴 CRITICAL: High CPU usage: 87.8%
[INFO] ✅ CPU returned to normal: 23.1%
```

8.3 Test Memory Monitoring

```
# Run memory stress test
python3 memory_test.py
```

What to expect:

- **Terminal 1 (Monitor):** Should show memory usage alerts
- **Dashboard:** Memory usage chart should increase
- **Test Duration:** ~30 seconds of high memory usage

```
Expected Monitor Output:
[DETECTED] 🟠 WARNING: High memory usage: 78.3%
[DETECTED] 🔴 CRITICAL: High memory usage: 87.9%
[DETECTED] 🔴 CRITICAL: High memory usage: 91.2%
[INFO] ✅ Memory returned to normal: 45.6%
```

8.4 Test Process Burst Detection

```
# Run process burst test
python3 burst_test.py
```

What to expect:

- **Terminal 1 (Monitor):** Should show process burst alerts
- **Dashboard:** Process count should spike
- **Test Duration:** ~20 seconds of rapid process creation

```
Expected Monitor Output:
[DETECTED] 🟠 WARNING: Process burst detected: 15 new processes
[DETECTED] 🟠 WARNING: Process burst detected: 22 new processes
[DETECTED] 🟠 WARNING: Process burst detected: 18 new processes
[INFO] ✅ Process activity returned to normal
```

8.5 Run All Tests Together

```
# Run comprehensive test suite
echo "🧪 Starting NetSnoop Test Suite..."
python3 cpu_test.py &
sleep 10
python3 memory_test.py &
sleep 10
python3 burst_test.py &

# Wait for all tests to complete
wait
echo "✅ All tests completed!"
```

What to expect:

- **Multiple simultaneous alerts** in Terminal 1

- **Dashboard showing activity** across all metrics
- **CSV log file growth** with test events

✅ Final Verification Checklist

Make sure everything is working after running tests:

- ☐ **WSL Ubuntu running** - `wsl --list --verbose` shows Ubuntu
- ☐ **Python 3.7+ installed** - `python3 --version` shows correct version
- ☐ **NetSnoop 0.1.4 installed** - `pip show netsnoop` shows version 0.1.4
- ☐ **Files created** - `ls` shows `anomalies.csv` and `netsnoop_persistent.txt`
- ☐ **Monitor running** - Terminal 1 shows continuous monitoring output
- ☐ **Dashboard accessible** - Browser shows http://localhost:8501
- ☐ **Real-time updates** - Dashboard refreshes with new data
- ☐ **Test scripts work** - All three prototype tests run successfully
- ☐ **Alerts functioning** - Tests generate visible alerts in monitor and dashboard
- ☐ **CSV logging active** - `anomalies.csv` grows with test events

🐛 Troubleshooting

Issue	Solution
`netsnoop-init` not found	`pip install --upgrade netsnoop==0.1.4`
Dashboard shows no data	Check if monitor is running: `ps aux | grep netsnoop`
Permission errors	`chmod +x ~/.local/bin/netsnoop-init`
WSL issues	`wsl --update` then restart

NetSnoop Installation Fails

```
# If pip isn't found, install it in base system first
sudo apt install python3-pip

# Update pip to latest version
pip3 install --upgrade pip

# Install specific NetSnoop version
pip install netsnoop==0.1.4

# If version conflicts occur
pip uninstall netsnoop
pip install netsnoop==0.1.4

# Clear pip cache if needed
pip3 cache purge
```

Test Scripts Not Working

```
# If curl fails to download test scripts
wget
https://raw.githubusercontent.com/ChitviJoshi/NetSnoop/main/tests/burst_test.py
wget
https://raw.githubusercontent.com/ChitviJoshi/NetSnoop/main/tests/memory_test.py
wget
https://raw.githubusercontent.com/ChitviJoshi/NetSnoop/main/tests/cpu_test.py

# If tests don't generate alerts, check thresholds
# Tests should push CPU >80% and Memory >85%

# If process burst test fails
ps aux | grep python # Check if multiple python processes exist
```

🛣️ Next Steps

🎉 Congratulations!

You now have NetSnoop running with:

- ✅ Real-time system monitoring
- ✅ Interactive web dashboard
- ✅ Automated anomaly detection
- ✅ Persistent data logging
- ✅ Validated test results

Explore NetSnoop Features

- **Customize thresholds** - Modify CPU/Memory alert levels
- **Export data** - Download CSV reports for analysis
- **Monitor long-term** - Leave NetSnoop running for continuous monitoring
- **Integrate with workflows** - Use CSV data in other tools

Get Support

- 📧 **Email:** chitvijoshi2646@gmail.com
- 🐛 **Issues:** GitHub Issues
- 📦 **PyPI:** netsnoop package
- 📚 **GitHub:** ChitviJoshi/NetSnoop

💡 Pro Tips

- **Keep both terminals open** - Monitor in one, dashboard in the other

- **Bookmark the dashboard** - http://localhost:8501 for quick access
- **Check CSV logs regularly** - Historical data helps identify patterns
- **Run tests periodically** - Ensure NetSnoop is working correctly
- **Monitor system patterns** - Look for recurring anomalies
- **Adjust thresholds** - Customize based on your system's normal behavior

---

🌟 NetSnoop - "Born to Track"

Built with ❤️ for developers and system administrators

Version 0.1.4 | **Setup Guide v1.0** | **Support:** chitvijoshi2646@gmail.com

NetSnoop is an open-source project. For the latest updates, visit our GitHub repository.
This guide was created to help users get started with NetSnoop quickly and efficiently.