Netsniff-ng- Packet Sniffer

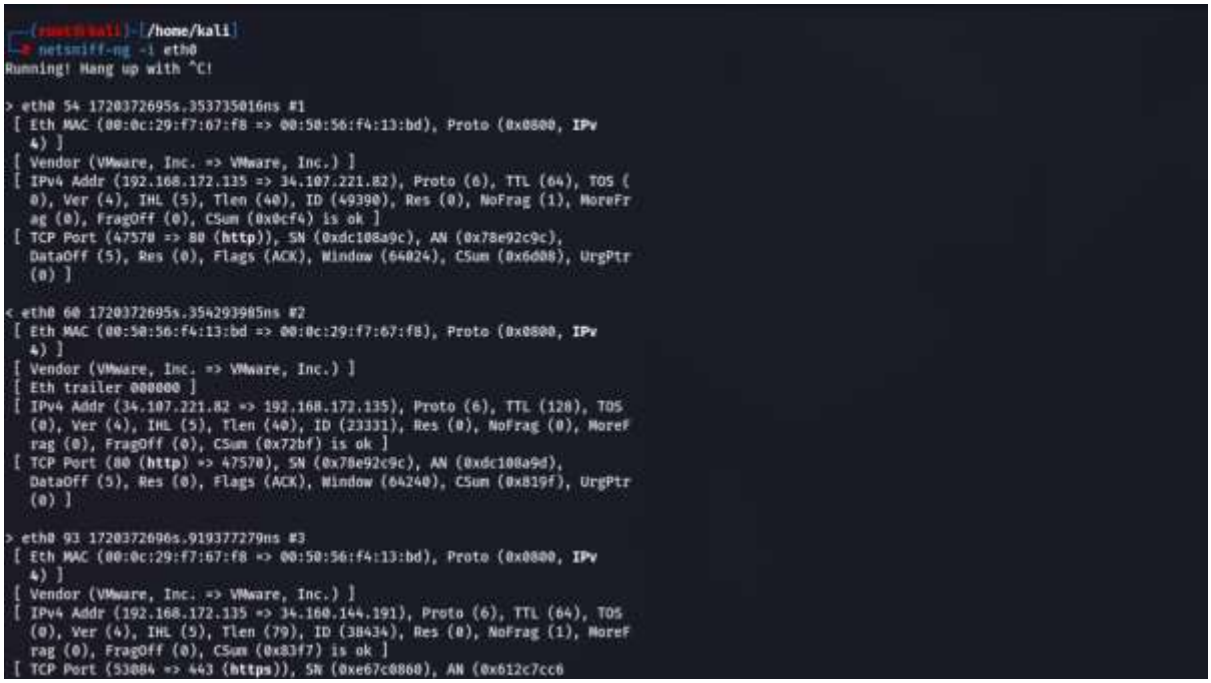
Packet sniffing is the process of capturing and analyzing network packets as they travel across a network. It involves intercepting data packets, which contain information such as source and destination addresses, protocols, and payload data. This technique is essential for network management, troubleshooting, and security monitoring, as it allows administrators to identify and resolve issues, monitor network performance, and detect malicious activities. Advanced packet sniffing tools, like Netsniff-ng, leverage high-efficiency capture methods, support for various network interfaces, and real-time analysis capabilities. These tools can handle high-speed network traffic with minimal performance impact, ensuring comprehensive and accurate data capture. Additionally, advanced features such as zero-copy capture, hardware-based timestamping, and packet injection and replay enhance their functionality, making them invaluable for sophisticated network diagnostics and security operations.

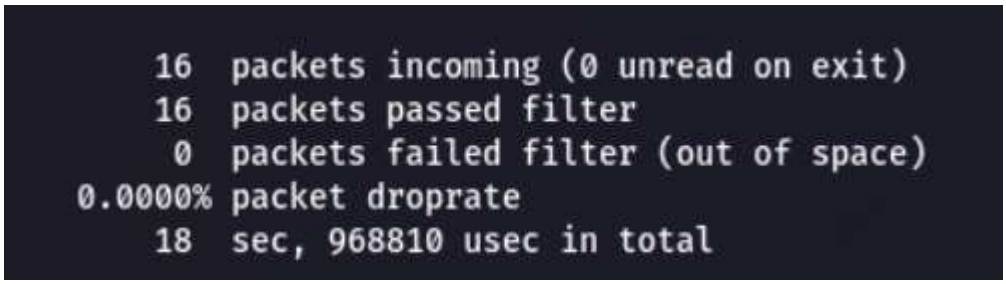
Netsniff-ng is a robust and efficient network packet sniffing and analysis tool for Linux, designed for high-speed network environments. As an open-source tool, it offers a suite of utilities for capturing, replaying, and analyzing network traffic. The core component, `netsniff-ng`, allows for high-performance packet capture, minimizing packet loss and ensuring accurate data collection. The toolkit includes `trafgen` for traffic generation, `mausezahn` for crafting and sending custom packets, `ifpps` for interface performance

statistics, `flowtop` for live traffic flow monitoring, and `bpfc` for compiling Berkeley Packet Filter rules. Netsniff-ng operates via a command-line interface, making it ideal for scripting and automation. It supports advanced features such as zero-copy packet capture, hardware-based timestamping, and multi-interface support. This versatility and efficiency make Netsniff-ng a valuable tool for network administrators and security professionals.

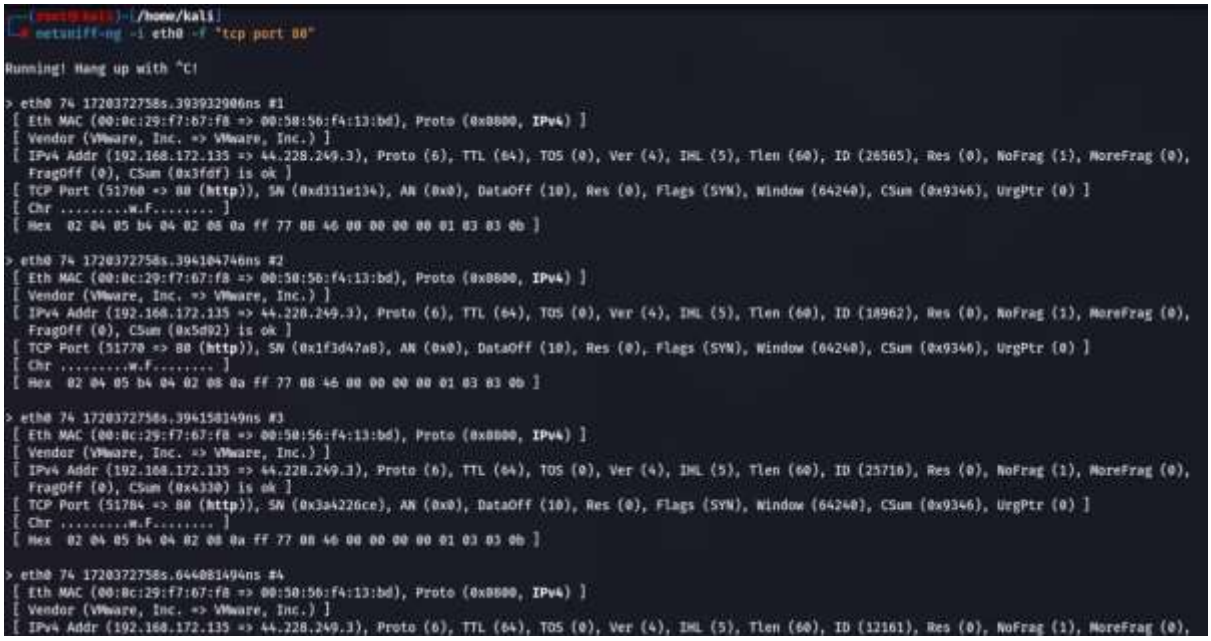
Example: -

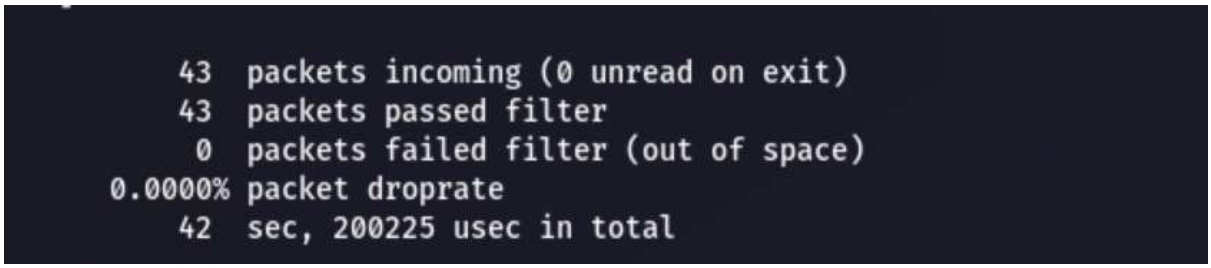
1. Firstly, we use flag -i to define our interface. In our case netsniff will capture packets from eth0 interface.





2. Now we used -f to filter a specific port. Here we capture TCP 80 port.





3. In this example, we use -o flag to store the output in a pcap file