Packet Sniffing Cheatsheet

A concise cheat sheet covering packet sniffing techniques, tools, and essential commands for network analysis and troubleshooting. This guide provides a quick reference for capturing and analyzing network traffic.

Introduction to Packet Sniffing

What is Packet Sniffing?

Packet sniffing is the process of capturing and logging network traffic. It allows you to inspect the data packets that are transmitted over a network.

It is used for network troubleshooting, security analysis, and monitoring network performance.

Ethical use requires explicit permission from network administrators.

Key Concepts

Promiscuous Mode	Network interface card (NIC) captures all packets on the network, not just those addressed to it.
Packet Analyzer	Software or hardware used to capture and analyze network packets.
Capture Filter	Defines which packets to capture based on criteria like IP address, port, or protocol.

Common Tools

- **tcpdump**: Command-line packet analyzer.
- **Wireshark**: Graphical network protocol analyzer.
- **tshark**: Command-line version of Wireshark.

Using tcpdump

Basic tcpdump Usage

Capture all packets on the default interface:

```
sudo tcpdump
```

Capture packets on a specific interface:

```
sudo tcpdump -i eth0
```

Capture a specific number of packets:

```
sudo tcpdump -c 10
```

Filtering with tcpdump

Capture packets from a specific host:	`sudo tcpdump src host 192.168.1.100`
Capture packets to a specific host:	`sudo tcpdump dst host 192.168.1.100`
Capture packets on a specific port:	`sudo tcpdump port 80`
Capture TCP packets:	`sudo tcpdump tcp`
Capture UDP packets:	`sudo tcpdump udp`

Saving captured packets

Save captured packets to a file:

```
sudo tcpdump -w capture.pcap
```

Read packets from a capture file:

```
sudo tcpdump -r capture.pcap
```

Using Wireshark

Wireshark Interface

Wireshark provides a graphical user interface for capturing and analyzing packets.
Key components include:

- **Capture Filter**: Specifies which packets to capture.
- **Display Filter**: Specifies which packets to display.
- **Packet List Pane**: Displays captured packets.
- **Packet Details Pane**: Displays detailed information about a selected packet.
- **Packet Bytes Pane**: Displays the raw data of a selected packet.

Basic Wireshark Usage

1. **Select Interface**: Choose the network interface to capture from.
2. **Start Capture**: Click the 'Start' button (or press Ctrl+E) to begin capturing packets.
3. **Stop Capture**: Click the 'Stop' button (or press Ctrl+E) to stop capturing packets.
4. **Apply Filters**: Use display filters to narrow down the packets displayed.

Wireshark Display Filters

Filter by IP Address:	`ip.addr == 192.168.1.100`
Filter by Source IP Address:	`ip.src == 192.168.1.100`
Filter by Destination IP Address:	`ip.dst == 192.168.1.100`
Filter by Port:	`tcp.port == 80`
Filter by Protocol:	`http`

Advanced Techniques

Following TCP Streams

Wireshark allows you to follow TCP streams to view the entire conversation between two endpoints.

- Right-click on a TCP packet.
- Select 'Follow' -> 'TCP Stream'.

This displays the entire TCP conversation in a new window, making it easier to analyze the data exchanged.

Analyzing HTTP Traffic

Filter HTTP traffic:	`http`
View HTTP request headers:	Expand the 'Hypertext Transfer Protocol' section in the Packet Details pane.
View HTTP response data:	Look for the 'HTTP Data' section in the Packet Details pane.

Detecting Anomalies

Packet sniffing can be used to detect network anomalies such as:

- Unusual traffic patterns.
- Suspicious connections.
- Unauthorized access attempts.

Analyze packet sizes, protocols, and communication patterns to identify potential security threats.