What is Telnet?

Telnet = **TELecommunication NETwork**

- It's one of the **oldest network protocols**, created in the **1960s**.
- It allows a **remote computer (client)** to connect to another computer (server) and **interact with it like a terminal** (i.e., a command-line interface).
- Think of it like **"remote keyboard access"** to another system.

⚙️ **How Does Telnet Work?**

➤ **Client-Server Model**

- **Client** = your machine
- **Server** = target machine
- Default **port: 23 (TCP)**

➤ **Steps:**

1. Your machine connects to the target using port 23:

bash

CopyEdit

telnet <target-ip>

2. Once connected, both communicate using a **Network Virtual Terminal (NVT)** — a standard way of sending and receiving characters.
3. You type commands → Telnet sends them to the server.
4. Server processes the command → sends output back to your screen.

🔑 **Important Terms (Keywords):**

Term	Meaning
NVT	Network Virtual Terminal – standard data format so different systems can communicate.
Client-Server	Your system (client) talks to another (server) to control it.
Port 23	Default port for Telnet communication.
Plaintext	Data sent without encryption (anyone can read it if they sniff).
Echo Mode	Controls whether typed characters are shown on the screen.

Term	Meaning
Option Negotiation	Client & server agree on terminal settings (size, type, etc.).

💻 What Can You Do with Telnet?

🔧 1. Remote Login & Admin

Log into systems like routers, switches, or servers.

🧪 2. Testing Network Services

Example:

bash

CopyEdit

telnet mail.example.com 25

Manually test if a mail server is working (SMTP).

🧰 3. Debugging Apps

Connect to services like HTTP, SMTP, IRC, and test manually.

👴 4. Legacy System Access

Old printers, IoT, routers, etc., still use Telnet sometimes.

🛡️ Why is Telnet Dangerous?

Reason	Explanation
No Encryption	Everything is sent in plaintext.
Sniffable	Anyone can intercept and read data using Wireshark.
Weak Auth	Uses only username and password (easy to brute-force).
No Integrity Check	No way to know if communication is being tampered.
Easy RCE	If logged in, you can run dangerous commands.
Poor Logging	Most Telnet servers don’t log properly, making it hard to detect attacks.

🔐 How Ethical Hackers Exploit Telnet

📍 Step-by-Step Workflow:

✅ **1. Reconnaissance**

Check if port 23 is open

bash

CopyEdit

nmap -p 23 <target-ip>

nmap -sV <target-ip>

Banner Grabbing (Get info about the service)

bash

CopyEdit

nc -nv <target-ip> 23

✅ **2. Brute-Force Credentials**

Hydra Example:

bash

CopyEdit

hydra -L users.txt -P passwords.txt <target-ip> telnet

Metasploit Module:

bash

CopyEdit

msfconsole

use auxiliary/scanner/telnet/telnet_login

set RHOSTS <target-ip>

set USER_FILE users.txt

set PASS_FILE passwords.txt

exploit

✅ **3. Packet Sniffing**

If you're inside the same network:

bash

CopyEdit

wireshark -Y telnet

- You'll see everything — usernames, passwords, commands — in **plaintext**.

✅ 4. Check for Default Logins

Try:

- admin:admin
- root:root
- guest:guest
- user:user

Many routers, IoT devices, and printers still use defaults.

✅ 5. Post-Exploitation

If login is successful:

- You get shell access!
- Run:

bash

CopyEdit

whoami

uname -a

cat /etc/passwd

Then:

- Enumerate users
- Read config files
- Find sensitive data
- Use for **pivoting** to other systems

✅ 6. Search for Vulnerable Telnet Versions

Find Telnet server version (from banner or Nmap), then:

- Search Exploit-DB
- Search CVEs (e.g., "telnet CVE-2024...")

Example:

CVE-2024-40891 — Telnet command injection on Zyxel devices.

🚧 Common Problems

Problem	Solution
Telnet not found	It’s rare externally, try **internal scanning** once inside a network.
Blocked by firewall	Try internal networks or pivoting from another system.
Too old system	Still exploitable — even better for learning.
Honeypot	Be careful — could be a trap set by defenders.

Indepth Understanding

Telnet Protocol for Ethical Hacking Purposes

Telnet (TELecommunication NETwork) is one of the oldest network protocols, dating back to the late 1960s. It was designed to provide a generic, bidirectional, eight-bit byte oriented communication facility, primarily for interfacing terminal devices and terminal-oriented processes.

How Telnet Works (The Basics):

Telnet operates on a client-server model and uses TCP port 23 by default. Here's a simplified breakdown:

1. **Connection Establishment:** A Telnet client (your computer) initiates a TCP connection to a Telnet server (the remote system) on port 23.
2. **Network Virtual Terminal (NVT):** Once connected, both the client and server operate as if they are communicating with a "Network Virtual Terminal" (NVT). This is an imaginary, standardized terminal that allows for communication compatibility between various types of real terminals and operating systems. The client translates your local keystrokes into NVT characters, sends them to the server, and the server translates NVT characters back into a format understandable by its own system.
3. **Command and Data Exchange:** All commands you type on your client are sent to the remote server, and the server's responses (like command output or prompts) are sent back to your client, essentially creating a text-based, interactive command-line session.
4. **Option Negotiation:** Telnet includes a mechanism for negotiating "options" between the client and server. These options can cover things like terminal type (e.g., VT100), window size, echo mode (whether the client or server echoes characters back), and more. This allows for adaptability between different systems.

Services Provided by Telnet:

Telnet itself is a *protocol* for remote terminal access. The "services" it provides are primarily related to enabling remote interaction with a system's command-line interface. These include:

- **Remote Login/Administration:** The primary use of Telnet was to log into a remote computer and execute commands as if you were physically sitting in front of it. This allowed system administrators to manage servers, routers, switches, and other network devices remotely.
- **Network Diagnostics:** Telnet can be used to test connectivity to specific ports on a remote device. For example, you could telnet mail.example.com 25 to see if an SMTP server is listening and then manually issue SMTP commands.
- **Testing and Debugging Applications:** Developers could use Telnet to directly interact with network services (like HTTP, SMTP, IRC) to send commands and examine responses, aiding in debugging.
- **Accessing Legacy Systems/BBSs:** Older systems or Bulletin Board Systems (BBSs) might still expose Telnet interfaces.

How Services are Provided:

The remote login and command execution services are provided directly through the established TCP connection. Your keystrokes are sent as data, and the server's responses are sent back as data, all within the Telnet protocol's structure. The protocol defines special "control characters" (like "Interpret As Command" - IAC) that distinguish commands from regular data, allowing for negotiation of options and out-of-band signaling.

Telnet in the "Latest" Context:

In modern IT environments, **Telnet is overwhelmingly considered an obsolete and insecure protocol for remote administration and should be avoided at all costs for anything sensitive.** Its primary use today for ethical hacking is to identify misconfigured or forgotten legacy systems.

Why Telnet is Insecure:

The fundamental flaw of Telnet, and why it's considered so dangerous, is its **lack of encryption**.

- **Cleartext Credentials:** All usernames and passwords, along with every command and every piece of output, are transmitted in plaintext over the network.
- **Eavesdropping/Sniffing:** Anyone with access to the network path between the client and server can easily intercept and read all communication using tools like Wireshark. This includes login credentials, configuration changes, sensitive data, etc.
- **Man-in-the-Middle (MitM) Attacks:** An attacker can position themselves between the client and server, intercepting, reading, and even modifying the traffic. Since there's no encryption or integrity checking, the client and server have no way of knowing the communication has been tampered with.
- **Lack of Authentication Strength:** Telnet typically relies on simple username/password authentication, which is vulnerable to brute-force and dictionary attacks.
- **Vulnerability to Malware:** If an attacker gains access via Telnet, they can easily execute arbitrary commands, install malware, or compromise the system further.

- **Inadequate Logging:** Telnet servers often have minimal logging capabilities, making it difficult to detect or investigate security incidents.

Ethical Hacking with Telnet: How to Make Use of it Effectively

As an ethical hacker, Telnet isn't a protocol you "break" cryptographically. Instead, you exploit its inherent insecurity and common misconfigurations.

Understanding telnet Client Usage:

It's crucial to distinguish between the Telnet *protocol* and the telnet *client program*. The telnet client is a versatile, raw TCP client that can connect to any open TCP port, not just port 23 (where the Telnet *service* typically runs).

Summary: Commands You Can Use with the telnet Client

Scenario	Available Commands	Notes
telnet to Port 23 (Telnet service/shell)	✅ All Linux commands	This is the traditional use: connecting to a server running a Telnet service. If configured for a shell, it functions like an SSH shell, allowing execution of any command the user account has permissions for (e.g., ls, cd, cat). Not all Telnet services grant a full shell; some might be restricted.
telnet to other Service Ports (21, 25, 80)	⚠️ Protocol-specific commands only	Here, the telnet client is used as a generic TCP client to establish a raw connection to *any other TCP service* (e.g., **FTP** on 21, **SMTP** on 25, **HTTP** on 80). You are **not** speaking the Telnet protocol; you are speaking the language of the service on that port. Examples: USER, PASS (for FTP); HELO, MAIL FROM (for SMTP); GET / HTTP/1.0 (for HTTP). Linux shell commands will not work.
No Telnet service or misconfigured	❌ No commands work	If no service is listening on the targeted port or the service is misconfigured, your input will likely be ignored, or the connection will be immediately closed. The telnet client might hang or the connection might terminate.

Export to Sheets

The telnet client is used for these other ports due to its simplicity, utility for debugging and testing by manually sending raw protocol commands, and its historical ubiquity on operating systems.

Ethical Hacking Steps:

1. **Reconnaissance and Enumeration:**
    - **Port Scanning (Nmap):** Identify if Telnet is running.
        - nmap -p 23 <target-ip>
        - nmap -sV <target-ip> (to detect service version)

- **Banner Grabbing (Netcat):** Connect directly to port 23 to grab the server banner, revealing OS, Telnet server software, and version, which can hint at known vulnerabilities.
    - nc -nv <target-ip> 23
- **Nmap Telnet Scripts:** Use Nmap's dedicated scripts:
    - telnet-encryption: Checks for any encryption support.
    - telnet-ntlm-info: For Microsoft Telnet services, attempts to extract NTLM authentication details.
    - telnet-brute: Brute-forces Telnet login credentials.
- **OSINT/Google Dorking:** Search for public mentions of Telnet access, default credentials for specific devices (e.g., old router models), or leaked configurations.

2. **Vulnerability Exploitation (Focus on Cleartext and Weak Authentication):**
    - **Sniffing (Wireshark):** This is the most straightforward and effective attack. If you have network access to the segment, you can capture all Telnet traffic.
        - Launch Wireshark, select the correct network interface, and set a display filter for telnet. All communication, including login attempts and credentials, will be visible in plain text. This is a critical finding.
        - wireshark -Y telnet (or set filter in GUI)
    - **Weak/Default Credentials:** Many legacy devices or poorly configured systems use default credentials (e.g., admin:admin, root:password). Attempt these first.
    - **Brute-Force/Dictionary Attacks:** Use tools like Hydra or Metasploit's telnet_login module to rapidly try common username/password combinations.
        - hydra -L users.txt -P passwords.txt <target-ip> telnet
        - In msfconsole: use auxiliary/scanner/telnet/telnet_login set RHOSTS <target-ip> set USER_FILE /path/to/user.txt set PASS_FILE /path/to/pass.txt exploit
    - **Vulnerable Telnet Server Software:** Older Telnet daemons (like telnetd or proprietary implementations) might have specific known vulnerabilities (e.g., buffer overflows, command injection flaws) that could lead to remote code execution.
        - Identify the version via banner grabbing or Nmap.
        - Search CVE databases (CVE, Exploit-DB) for exploits specific to that version (e.g., recent Zyxel vulnerabilities like CVE-2024-40891 exploiting Telnet for unauthenticated command injection).
    - **Passwordless Authentication:** Some very old or specific Telnet setups might allow passwordless login for guest or public access. Attempting to log in with a username and no password can reveal this.
3. **Post-Exploitation (if access is gained):**

- **Command Execution:** The goal is typically to gain a shell on the remote system to execute commands, enumerate further, and potentially elevate privileges.
- **Information Gathering:** Collect system information, user lists, configuration files, and any sensitive data.
- **Lateral Movement:** Use the compromised Telnet server as a pivot point to access other systems on the network.

Problems Faced by Ethical Hackers:

1. **Limited Usage in Modern Networks:** Most organizations have migrated away from Telnet due to its glaring security flaws. Finding an internet-facing Telnet server (beyond honeypots or legacy IoT devices) is becoming less common.
2. **Firewalls:** Properly configured firewalls block inbound connections to port 23 by default, preventing external access.
3. **Intrusion Detection/Prevention Systems (IDS/IPS):** Brute-force attempts against Telnet will likely trigger alerts on IDS/IPS, potentially leading to IP blocking or administrator notification.
4. **Network Segmentation:** Even if Telnet is found internally, network segmentation might prevent direct access from the ethical hacker's initial point of entry.
5. **Honeypots:** Many publicly exposed Telnet services are actually honeypots designed to trap and analyze attacker behavior. Connecting to one will likely alert the defenders and provide no real advantage.
6. **No Interesting Data:** Even if a Telnet service is found, it might be on a device with no sensitive data or access to critical systems.

How to Overcome These Problems:

For ethical hackers, dealing with Telnet means adapting their methodology:

1. **Internal Network Focus:** Telnet is much more likely to be found on internal networks, especially on older embedded devices, industrial control systems (ICS/SCADA), or misconfigured network appliances. Ethical hackers often look for Telnet *after* gaining initial access to an internal network via other vulnerabilities (e.g., web application flaws, phishing).
2. **Persistent Reconnaissance:** Continue scanning and identifying services, even on seemingly insignificant devices. A forgotten Telnet service on an old printer or IoT device could be a stepping stone.
3. **Payload Delivery:** If authentication is bypassed (e.g., default credentials), the focus quickly shifts to delivering payloads (e.g., reverse shells) to gain a more stable and powerful connection than the basic Telnet shell.
4. **Demonstrate Impact:** Since Telnet's vulnerability is well-known, the ethical hacker's value comes from *demonstrating the impact* of its presence. Show how easy it is to sniff credentials, gain a shell, or extract sensitive information. This provides concrete evidence for remediation.
5. **Recommend Strong Alternatives:** The primary recommendation will always be to disable Telnet entirely and replace it with SSH (Secure Shell) for remote management. If Telnet *must*

be used (e.g., for very specific legacy equipment), strong compensating controls like IP whitelisting, firewalls, and running it only over a VPN or dedicated management network are crucial. Even then, the ethical hacker should emphasize the risks.

6. **Leverage Network Sniffing (if internal access):** This is still the "killer" attack for Telnet. If you can get on the same network segment, Telnet is a goldmine for credentials. Tools like ARP spoofing (for internal MitM) can help facilitate this.