**Sniffing in a Controlled Environment**

**Assignment Objective**

**Goal:**  
Showcase the hidden dangers of transmitting data without encryption by capturing login credentials with a packet sniffer in a safe, isolated environment. This hands-on exercise highlights just how easy it is for attackers to harvest sensitive information if security best practices are neglected.

**Lab Setup**

| **System** | **Role** | **Details** |
| --- | --- | --- |
| Kali Linux | Attacker | Wireshark & vsftpd FTP server |
| Windows 10 | Victim | Runs FTP client (CMD) |
| Network | Host-Only vSwitch | Both VMs on same private subnet |

**Tools & Their Impact**

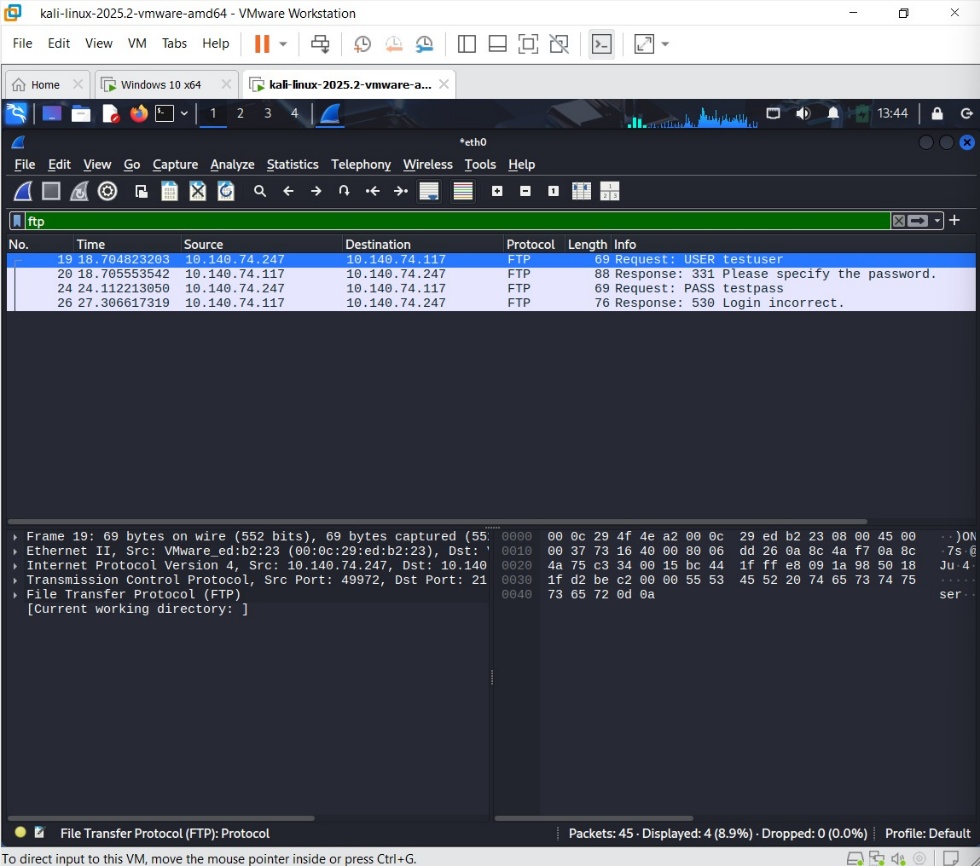
* **Wireshark:**  
  Our window into the network’s soul; captures and reveals all passing data packets, for better or worse.
* **FTP Client (Command Prompt):**  
  Simulates a typical user logging in—unknowingly exposing credentials over an unprotected channel.
* **vsftpd FTP Server:**  
  Provides the playground for FTP authentications; allows us to witness how unencrypted protocols jeopardize security.

**Step-by-Step Experience**

1. **Set up the FTP Server:**  
   Installed and configured vsftpd on Kali Linux to accept logins.
2. **Check Connection:**  
   Used ping to ensure Kali and Windows could communicate over their private network.
3. **Start Traffic Capture:**  
   Launched Wireshark on Kali and selected the correct interface (eth0). Watching raw packets flow in real time is eye-opening!
4. **Simulate Login:**
   * On Windows, opened the command prompt and connected via:  
     ftp <Kali\_IP>
   * Entered test credentials:  
     Username: testuser  
     Password: testpass
5. **Analyze the Packets:**
   * Filtered Wireshark view to show only FTP protocol:

ftp

* + Scrolled through captured packets and clearly found:
    - USER testuser
    - PASS testpass

1. **Save Evidence:**  
   Exported the session as a .pcapng file for documentation.
2. **Capture a Screenshot:**  
   

**Why This Matters: Real Risks**

* **Eavesdropping Made Easy:**  
  Anyone with basic tools can see what should be private—usernames, passwords, even full conversations!
* **Credential Theft:**  
  Attackers can steal logins from unencrypted protocols like FTP, Telnet, and HTTP with shocking ease.
* **Session Hijacking:**  
  Sometimes, just capturing session tokens is enough to impersonate a user, no need for passwords.

**How to Stay Safe: Actionable Solutions**

**Upgrade to Encrypted Protocols:**

* Use **SFTP** or **FTPS** instead of FTP
* Switch all web traffic to **HTTPS**
* Enforce **TLS/SSL** for all applications

**Isolate and Secure Networks:**

* Create VLANs to separate sensitive systems
* Apply strict **port security** on switches
* Disable **promiscuous mode** unless absolutely necessary

**Encrypt Everything:**

* Deploy **VPNs** for remote or internal connections
* Educate users and staff about phishing and insecure logins

**Key Takeaways**

This experiment isn’t just academic—it’s *alarmingly real*. Transmitting sensitive information in plaintext turns routine actions into major security incidents, sometimes within seconds. Even a beginner can intercept and read credentials on an open network.