Name: Vanshika Deswal
Roll Number: 23BCE1610
Course / Department: B.Tech CSE
Project Title: Man-in-the-Middle Attack using DNS Spoofing and Credential Harvesting

---

Executive Summary

This project demonstrates a Man-in-the-Middle (MITM) attack using DNS spoofing to redirect a victim to a fake login page. The attack was carried out in a virtual lab environment using Kali Linux as the attacker machine and Ubuntu as the victim. The attacker used Bettercap to perform DNS spoofing, hosted a fake HTTP login server, and successfully captured credentials from the victim. Wireshark was used to analyze the captured HTTP traffic, confirming the effectiveness of the attack.

Project Overview

Problem Statement:

Demonstrate how a man-in-the-middle attacker can use DNS spoofing to intercept and manipulate user requests, ultimately leading to credential theft.

Objectives:

- Perform DNS spoofing in a controlled environment.
- Redirect victim to a fake login page.
- Capture login credentials submitted by the victim.
- Analyze network traffic using Wireshark.

Scope of Work:

Included:

- DNS spoofing using Bettercap
- Hosting and deploying a fake login page
- Capturing HTTP credentials via a custom Python server
- Traffic analysis with Wireshark

Excluded:

- HTTPS spoofing or SSL stripping
- Use of real-world domains beyond lab scope

Tools & Lab Setup

Primary Tools Used:

- Kali Linux
- Ubuntu
- Bettercap
- Python HTTP Server
- Wireshark

Environment Details:

- **Virtual Machine Setup:** UTM on macOS
- **Target VM:** Ubuntu Desktop 22.04 LTS
- **Network Mode:** Host-only network (manual IP assignment)

Tool Configuration & Commands:

Bettercap Commands:

```
sudo bettercap -iface eth0
```

Inside Bettercap shell:

```
set dns.spoof.domains veryrealloginpage.com
set dns.spoof.address 192.168.64.2
dns.spoof on
```

Python Server (server.py):

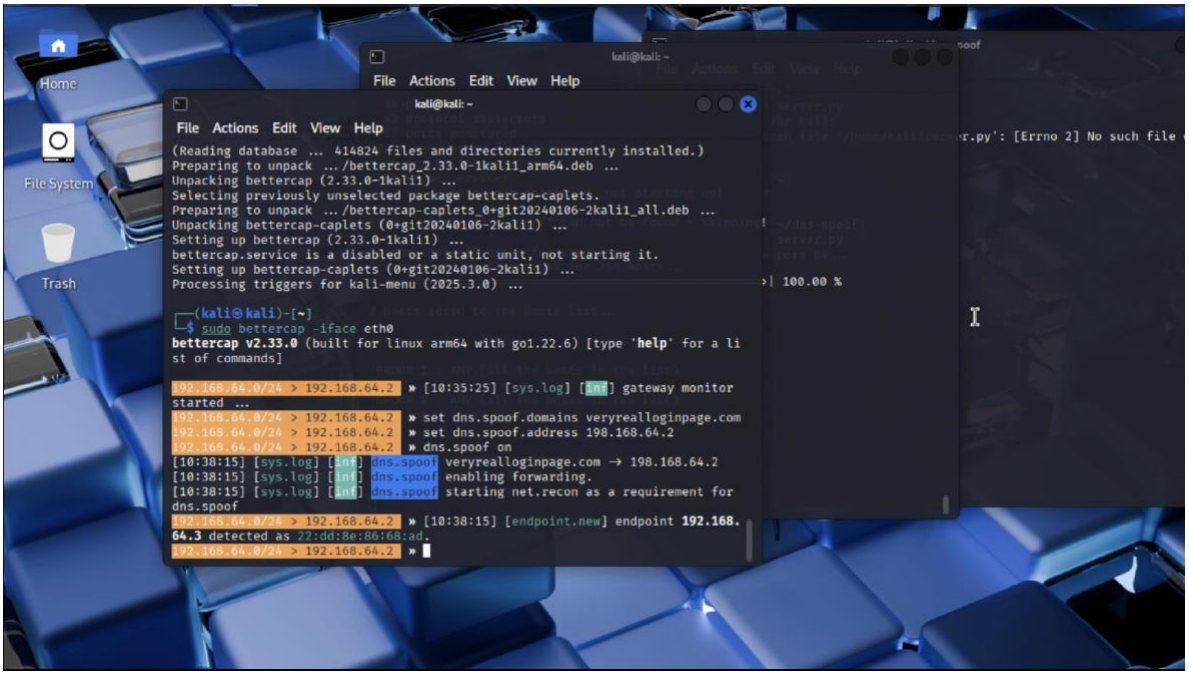
- Hosted on attacker (Kali)
- Handles GET and POST requests for /login
- Logs received credentials to terminal

Implementation & Execution Summary

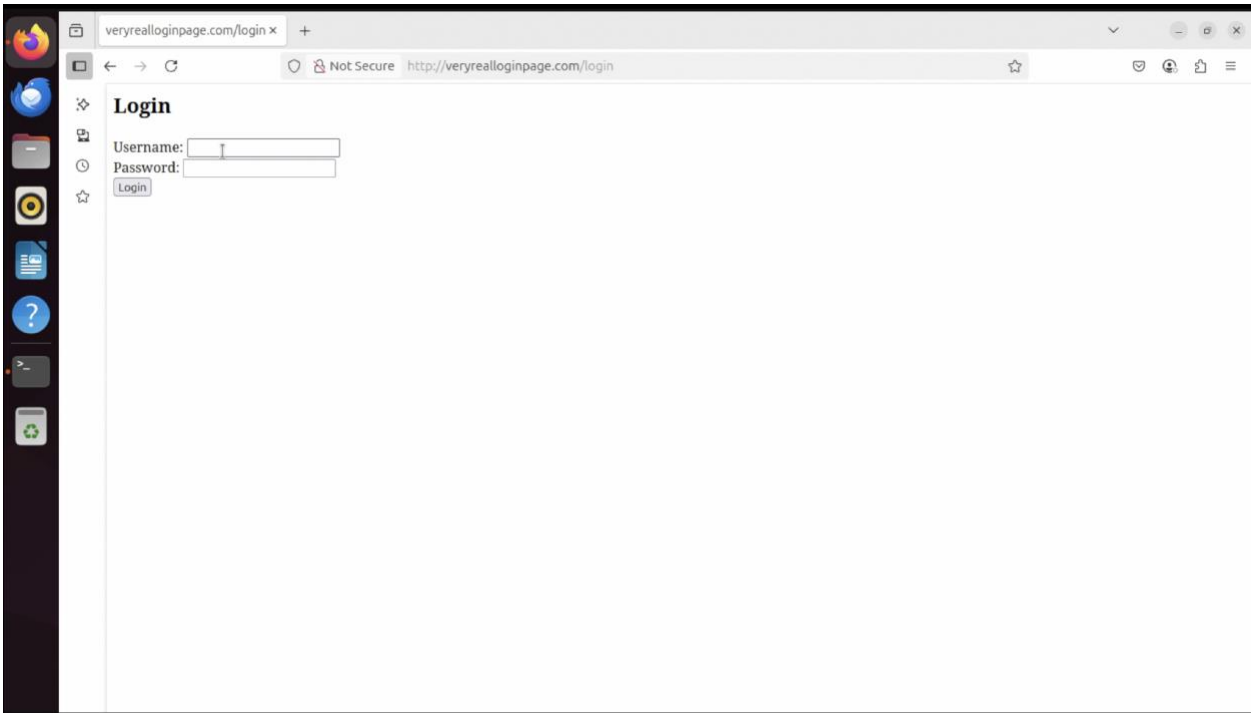
- Created `login.html` page to simulate a realistic login interface.
- Wrote a Python HTTP server to serve the fake page and capture POST data.
- Configured Bettercap to spoof the domain `veryrealloginpage.com` to point to the attacker IP.
- Victim accessed the spoofed domain via browser and submitted login credentials.
- Attacker terminal logged the stolen credentials.
- Wireshark was used to analyze and verify the POST request and captured data.

Screenshots:

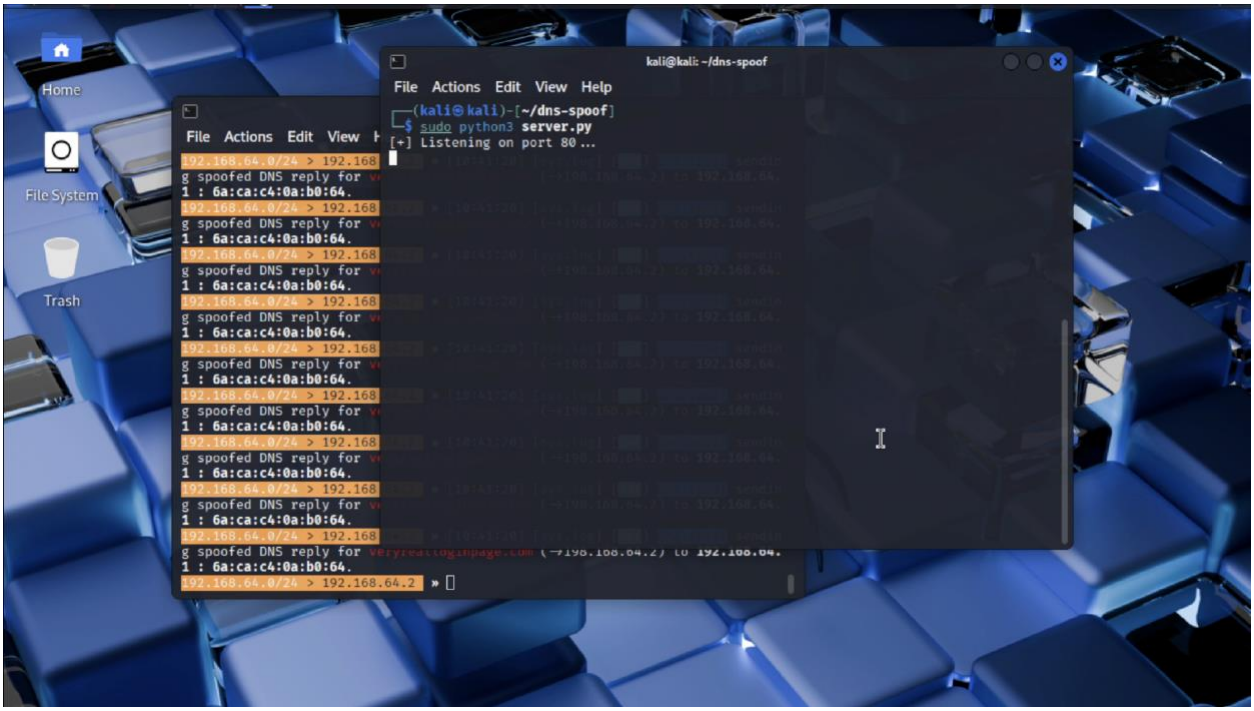
1. Bettercap terminal running DNS spoofing



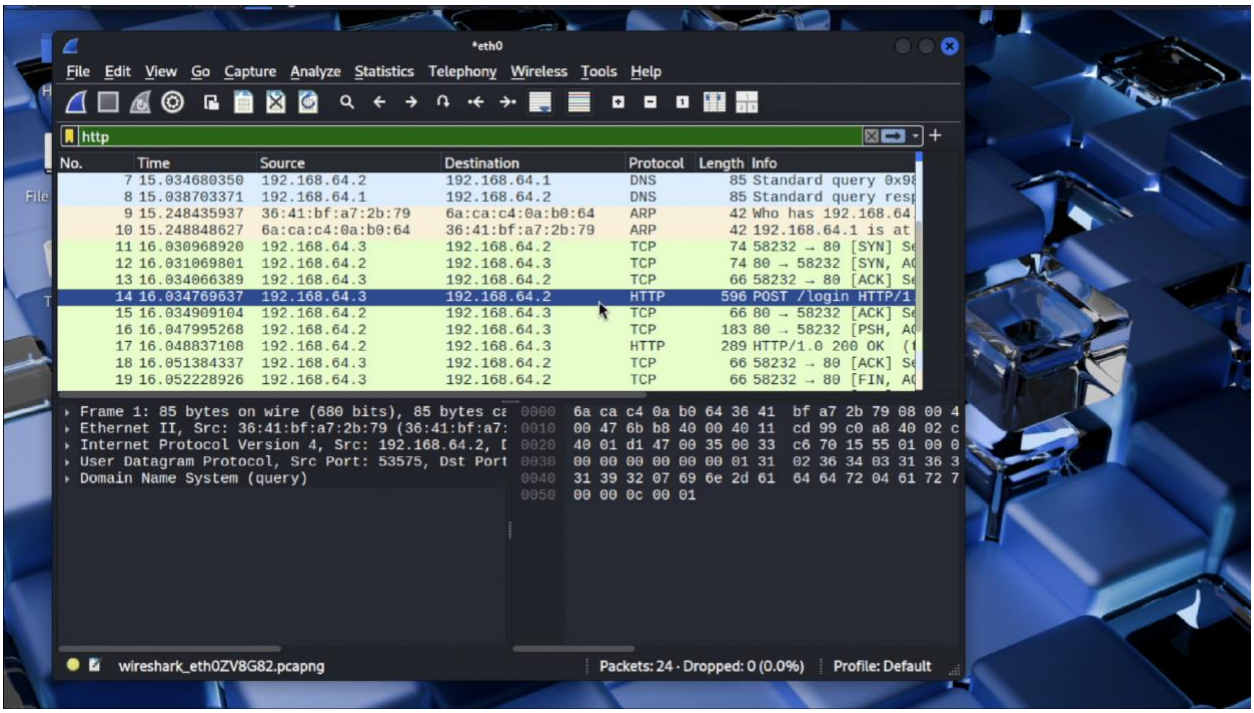
2. Fake login page in victim browser



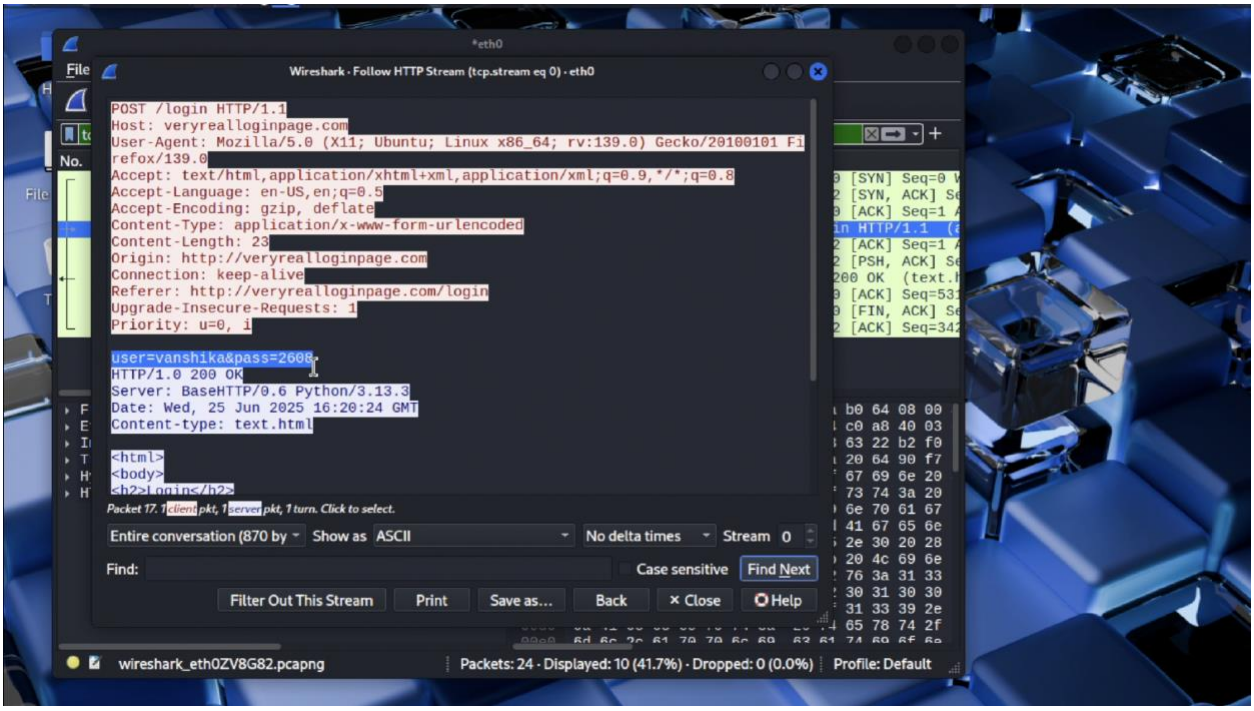
3. Kali terminal starting a fake server



4. Wireshark showing HTTP POST request



5. HTTP stream showing extracted credentials



Challenges Faced

- Ettercap DNS spoofing plugin was missing; had to pivot to Bettercap.
- Faced HTTPS redirection issues with real domains; used custom domain `veryrealloginpage.com`.
- Errors in Python server (e.g., variable `content-length`) were debugged and corrected.
- Network configuration between VMs took time to align properly.

Findings & Analysis

- Successfully captured HTTP POST data containing login credentials in plain text.
- DNS spoofing via Bettercap worked reliably in a closed lab setup.

- Wireshark confirmed data was sent over HTTP and exposed to MITM attackers.
- Realized importance of HTTPS and secure DNS (DNSSEC) to prevent such attacks.

Learning Outcomes

- Gained practical experience with MITM attacks and DNS spoofing.
- Understood Bettercap command-line interface and scripting.
- Learned to build and debug simple web servers in Python.
- Strengthened ability to work with packet analyzers like Wireshark.
- Improved understanding of attacker methodology and victim behavior.

Future Scope

- Implement SSL stripping to extend attack to HTTPS sites.
- Automate entire spoofing process using Bettercap modules and scripts.
- Integrate credential storage with database backend.
- Add phishing page themes for different brands to increase realism.
- Test effectiveness with browser HSTS and DNSSEC enabled.