PYTHON FOR PENETRATION TESTING COURSE

- **Actual Price**: ₹3,000
- **Special Offer**: ₹2,000 for the first 10 students🎉🎉
- **Schedule**: Classes will be from 6:00 PM to 7:00 PM IST, Monday to Friday
- **Demo Class**: 8th November 2024 (Friday)
- **Course Start Date**: 11th November 2024 (Monday)
- **Batch Size**: 20 students (Live)
- **Language**: Hindi
- **Registration Fee**: ₹50

- After paying the registration fee, students will be added to a **WhatsApp group** where all course details will be shared.
- After the **demo class**, students need to pay the full course fee of ₹2,000 (for the first 10 students) to continue with the course. The price for the rest will be ₹3,000.

- **Live classes** on Google Meet with practical, hands-on sessions.
- **Doubt clearing sessions** available during every class.
- Focus on building real-world Python tools for **penetration testing**.
- Course taught **in Hindi** for better understanding.

To register, type "Python" on WhatsApp at +91 9627797555 and secure your spot in the batch!

COURSE SYLLABUS

Week 1: Python Basics for Penetration Testing

Day 1: Setup Python, Pip, and Libraries

- **Task**: Install Python, pip, and requests, scapy, socket.

Day 2: Python Basics (Variables, Loops, Functions)

- **Task**: Write a script that calculates IP ranges using basic math.

Day 3: Networking with Python

- **Task**: Write a TCP/UDP client-server connection program.

Day 4: File Handling

- **Task**: Create a script to read a list of IPs and check if they are reachable.

Day 5: Libraries for Penetration Testing (socket, scapy)

- **Task**: Create a simple banner-grabbing script.

Day 6: Network Scanning Basics

- **Task**: Write a Python port scanner like Nmap.

Day 7: Week 1 Review & Challenge

- **Task**: Combine concepts into a simple tool to scan and grab banners.

Week 2: Network and Web Scanning

Day 8: Writing a Port Scanner

- **Task**: Implement a port scanner for specific IP ranges.

Day 9: ARP Spoofing

- **Task**: Write a script using scapy to perform ARP spoofing.

Day 10: Banner Grabbing

- **Task**: Modify the port scanner to grab service banners.

Day 11: Web Recon and HTTP Requests

- **Task**: Write a script to perform GET/POST requests on web forms.

Day 12: Directory and File Enumeration

- **Task**: Write a directory brute-forcing script like DirBuster.

Day 13: SQL Injection Basics

- **Task**: Create a Python script to test SQL injection vulnerabilities.

Day 14: Week 2 Review & Challenge

- **Task**: Build a full network recon tool combining scanning and exploitation.

Week 3: Exploitation and Automation

Day 15: Brute Forcing Login Pages

- **Task**: Write a Python script to brute force login forms.

Day 16: Automating Nmap Scans

- **Task**: Write a Python script to run and parse Nmap results.

Day 17: Building a Basic Keylogger

- **Task**: Create a Python keylogger that saves keystrokes to a file.

Day 18: Developing Exploit Scripts

- **Task**: Write a Python exploit script for a controlled vulnerability.

Day 19: Writing a Custom Password Cracker

- **Task**: Create a dictionary-based password cracker for hashed passwords.

Day 20: Fuzzing for Input Vulnerabilities

- **Task**: Build a fuzzer to send random inputs to a target service.

Day 21: Week 3 Review & Challenge

- **Task**: Develop a Python tool that performs scanning, brute forcing, and exploitation.

Week 4: Advanced Automation and Final Project

Day 22: Automating Web Scraping with Python

- **Task**: Write a script using BeautifulSoup to scrape data from websites.

Day 23: Custom Vulnerability Scanning Tool

- **Task**: Write a tool that automates vulnerability detection using Python.

Day 24: Reporting and Logging Results

- **Task**: Create a script to generate a report of findings from a test.

Day 25: Final Project Preparation

- **Task**: Begin building a full penetration testing tool.

Day 26: Final Project - Full Recon and Exploit Tool

- **Task**: Complete your custom tool for network and web testing.

Day 27: Testing the Final Project

- **Task**: Test your tool on a vulnerable machine (Metasploitable or DVWA).

Day 28: Final Review and Improvements

- **Task**: Refine your tool and ensure all functions work smoothly.

Key Course Goals

- Automate network and web penetration testing tasks using Python.
- Build a custom toolkit for reconnaissance, scanning, and exploitation.
- Gain hands-on experience with real-world penetration testing challenges.