Introduction

This document provides 250 beginner-friendly Python programming challenges. It is structured into two distinct parts to guide you from a programming novice to a security-conscious developer.

1. **Part 1: 50 Foundational Python Problems**
    - This section is for absolute beginners. You will learn the essential syntax, logic, and core concepts of the Python language.
2. **Part 2: 200 Python for Ethical Hacking (Attack & Defense)**
    - This section applies your Python skills to the world of cybersecurity. For each concept, you will write scripts that simulate both an **attack** (to understand how vulnerabilities are exploited) and a **defense** (to learn how to protect against those exploits).

Disclaimer: The scripts in Part 2 are for educational purposes only. They should **only** be run on computer systems and networks that you own or have explicit, written permission to test. Unauthorized hacking is illegal.

Part 1: 50 Foundational Python Problems

Basics: Syntax, Variables, and I/O

1. **Print Name**: Write a program that prints your full name.
2. **Store and Print**: Create a variable to store your favorite number and print the sentence "My favorite number is [number]".
3. **User Input**: Ask the user for their city and print "I've heard [city] is a great place!".
4. **Simple Arithmetic**: Ask the user for two numbers, then print their sum, difference, product, and quotient.
5. **String Concatenation**: Create two string variables and combine them to form a single sentence.
6. **Data Types**: Create variables of type integer, float, string, and boolean, and print the type of each variable using the type() function.
7. **Area of a Rectangle**: Ask for the length and width of a rectangle and calculate its area.
8. **String Repetition**: Ask the user for a word and a number, then print the word repeated that many times.
9. **Simple Mad Libs**: Ask the user for a noun, a verb, and an adjective, and print a short story using them.
10. **Type Conversion**: Ask the user for their age (as a string), convert it to an integer, and calculate the year they were born.

Control Flow: Conditionals & Loops

1. **Even or Odd**: Ask for a number and print whether it is even or odd.
2. **Age Group Classifier**: Ask for an age and classify the person as a "Child," "Teenager," "Adult," or "Senior."
3. **Simple Login**: Create variables for a username and password. Ask the user for theirs and check if they match.

4. **Count from 1 to 10**: Use a for loop to print the numbers 1 through 10.
5. **Countdown**: Use a while loop to print a countdown from 10 to 1.
6. **Multiplication Table**: Ask for a number and print its multiplication table up to 10.
7. **Sum of Numbers**: Calculate the sum of all numbers from 1 to 100.
8. **Guess the Number**: Generate a random number and have the user guess it, providing "too high" or "too low" hints.
9. **FizzBuzz**: Loop from 1 to 100. Print "Fizz" for multiples of 3, "Buzz" for multiples of 5, and "FizzBuzz" for multiples of both.
10. **Print a Triangle**: Ask for a number and print a triangle of asterisks with that many rows.

Data Structures: Lists, Dictionaries, Tuples, Sets

1. **Create a List**: Create a list of your favorite fruits and print the entire list.
2. **Access List Item**: Print the third item from the fruit list.
3. **Modify a List**: Change the second item in the list to a different fruit.
4. **Append to a List**: Add a new fruit to the end of your list.
5. **Loop Through a List**: Print each fruit from the list on a new line.
6. **List of Numbers**: Create a list of 5 numbers and find their sum and average.
7. **Simple Dictionary**: Create a dictionary to store information about yourself (name, age, city).
8. **Access Dictionary Data**: Print your age from the dictionary.
9. **Add to Dictionary**: Add your favorite color to the dictionary.
10. **Remove Duplicates**: Create a list with duplicate numbers and use a set to remove the duplicates.

Functions

1. **Greeting Function**: Write a function that takes a name as an argument and prints a personalized greeting.
2. **Addition Function**: Write a function that takes two numbers and returns their sum.
3. **Area Function**: Write a function to calculate the area of a circle given its radius.
4. **Even Checker Function**: Write a function that returns True if a number is even and False otherwise.
5. **Default Parameter**: Write a function to say hello to a user, with "User" as the default name if none is provided.
6. **String Reverser**: Write a function that takes a string and returns it in reverse.
7. **Max of Three**: Write a function that takes three numbers and returns the largest one.
8. **List Sum Function**: Write a function that takes a list of numbers and returns their sum.
9. **Vowel Counter**: Write a function that counts the number of vowels in a string.
10. **Palindrome Checker**: Write a function that checks if a word is a palindrome.

File Handling & Modules

1. **Write to File**: Write a script that creates a new file named test.txt and writes "Hello from Python!" into it.
2. **Read from File**: Write a script that reads the content of test.txt and prints it to the console.
3. **Append to File**: Write a script that adds a new line, "This is a new line.", to test.txt.
4. **Count Lines**: Write a script that counts and prints the number of lines in test.txt.

5. **Import math**: Use the math module to find the square root of a number given by the user.
6. **Import random**: Use the random module to pick a random item from a list.
7. **Import datetime**: Use the datetime module to print the current date and time.
8. **Import os**: Use the os module to list all files in the current directory.
9. **Create a Module**: Create a file my_module.py with a function in it. Import and use that function in another script.
10. **Handle Errors**: Write a script that asks for a number and tries to divide 10 by it, using a try...except block to handle the ZeroDivisionError.

Part 2: 200 Python for Ethical Hacking (Attack & Defense)

Category 1: Reconnaissance (40 Problems)

Attack Scripts

1. **DNS to IP**: Write a script to get the IP address of a domain name.
2. **HTTP Header Grabber**: Write a script to fetch and print the HTTP headers of a website.
3. **Server Type Detector**: From the headers, write a script to identify the Server type (e.g., Apache, Nginx).
4. **Robots.txt Fetcher**: Write a script that retrieves and prints the robots.txt file from a website.
5. **Link Scraper**: Write a script using BeautifulSoup to find all links on a webpage.
6. **Email Scraper**: Write a script to find all email addresses on a webpage.
7. **Subdomain Brute-Forcer**: Given a domain and a small wordlist, check for live subdomains.
8. **WHOIS Lookup**: Use a library to perform a WHOIS query on a domain.
9. **IP Geolocation**: Use a free API to find the physical location of an IP address.
10. **Port Scanner (Single Port)**: Write a script to check if a specific TCP port is open on a host.
11. **Find Hidden Dirs (Basic)**: Check for common directory names (/admin, /test, /dev) on a web server.
12. **Social Media Profile Finder**: Write a script that generates potential social media URLs from a username.
13. **Google Dorking Script**: Automate a Google search for site:example.com filetype:pdf.
14. **Traceroute Script**: Use os.system to run a traceroute or tracert command and capture the output.
15. **Banner Grabbing**: Connect to an open port and receive the first line of data (the banner).
16. **MX Record Lookup**: Find the mail servers for a domain using dnspython.
17. **NS Record Lookup**: Find the name servers for a domain.
18. **Shodan Search Script**: Use the Shodan API to search for devices with a specific keyword.
19. **GitHub Repo Search**: Use the GitHub API to find public repositories for a target organization.
20. **Wayback Machine Scraper**: Write a script to fetch URLs from the Wayback Machine for a domain.

Defense Scripts

1. **Header Anonymizer Concept**: Write a function that takes a dictionary of headers and removes identifying ones like Server and X-Powered-By.
2. **Robots.txt Generator**: Write a script that creates a restrictive robots.txt file to disallow all bots.
3. **Email Obfuscator**: Write a function that turns user@example.com into user [at] example [dot] com for display on a webpage.
4. **Port Scan Detector (Basic)**: Write a script that listens on a port and logs any IP that tries to connect.
5. **Sensitive File Checker**: Write a script that searches your own web directory for files with sensitive extensions like .bak, .old, .sql.
6. **DNS Zone Transfer Check**: Write a script to test if your own DNS server allows zone transfers.
7. **Directory Listing Check**: Write a script that checks if directory listing is enabled on your own web server.
8. **API Key Leakage Scanner**: Write a script that searches your code files for common API key patterns.
9. **Metadata Scrubber (Concept)**: Using Pillow, write a script that opens an image, removes its EXIF data, and saves it.
10. **Password in URL detector**: Parse a log file and alert if you find patterns like password=... in URLs.
11. **Subdomain Monitor**: Check a list of your known subdomains and alert if one becomes unresponsive.
12. **Certificate Expiry Checker**: Write a script to check the SSL certificate expiry date for your domain.
13. **WHOIS Privacy Check**: Write a script that checks if your domain's WHOIS info contains personal keywords.
14. **Firewall Rule Validator**: Write a script to check if a specific port is blocked on your localhost.
15. **GitHub Secret Scanner**: Search your own public GitHub repos for the word "password" or "api_key".
16. **Default Page Checker**: Write a script to see if your web server still has a default "Welcome to..." page.
17. **Brute-Force Log Analyzer**: Parse a log file to find IPs with more than 10 failed login attempts.
18. **Geolocation Whitelist**: Write a function that checks if an IP's country is in a list of allowed countries.
19. **404 Spike Detector**: Analyze a log to see if a single IP is generating an unusually high number of 404 errors.
20. **Personal Info Redactor**: Write a function to replace names and phone numbers in a text block with [REDACTED].

Category 2: Network & Web Exploitation (60 Problems)

Attack Scripts

1. **TCP Reverse Shell Client**: A script that connects to a listening server and executes commands.
2. **UDP Flood Script**: A script that sends a high volume of UDP packets to a target IP and port.
3. **Simple Web Login Brute-Forcer**: Try to log in to a web form by iterating through a password list.
4. **Directory Traversal Payload Generator**: Generate a list of ../ payloads to test for directory traversal.
5. **SQL Injection Payload Generator (Basic)**: Generate common SQLi payloads like ' OR 1=1 --.
6. **XSS Payload Generator**: Generate common XSS payloads like <script>alert('XSS')</script>.
7. **Command Injection Tester**: Send a request to a test web app with payloads like ; ls -la.
8. **Cookie Stealer via XSS (Simulation)**: Create a test page where an XSS payload sends document.cookie to a listening server.
9. **Parameter Tampering**: Write a script that changes a URL parameter like id=123 to id=124.
10. **Hidden Form Field Discoverer**: Scrape a page and print any form fields of type="hidden".
11. **HTTP Basic Auth Cracker**: Use a wordlist to try and crack an HTTP Basic Authentication prompt.
12. **FTP Anonymous Login Checker**: A script that tries to log in to an FTP server with username "anonymous".
13. **Password Spraying**: Try a single common password against a list of usernames.
14. **Weak JWT Secret Cracker**: Try to crack a JWT token by brute-forcing a list of common secrets.
15. **LFI Payload Generator**: Generate payloads for Local File Inclusion, like ../../../../etc/passwd.
16. **SSRF Tester**: Create a script that exploits a test app to make it request a URL you control.
17. **Deauthentication Packet Sender**: Using scapy, craft a Wi-Fi deauth packet.
18. **ARP Spoofing Packet Sender**: Using scapy, craft an ARP reply packet to perform spoofing.
19. **Simple Keylogger**: Using pynput, log keystrokes to a file (run on your own machine only).
20. **Form Submission Automator**: Use requests to automatically submit a form with specific data.
21. **Session Hijacking (Simulation)**: Manually copy a session cookie and use it in a script to access a protected area.
22. **Clickjacking Tester**: Create an HTML page with an iframe to see if a target site can be clickjacked.
23. **Subdomain Takeover Checker**: Check a list of CNAME records to see if any point to expired services.
24. **XML External Entity (XXE) Payload**: Craft a basic XXE payload to read a local file.
25. **Insecure Deserialization (Python Pickle)**: Create a malicious pickle object that executes a command when loaded.

Defense Scripts

1. **TCP Reverse Shell Listener**: A script that listens for the connection from the reverse shell client.
2. **Input Sanitizer for SQLi**: Write a function that removes dangerous characters (', --) from user input.
3. **Input Sanitizer for XSS**: Write a function that escapes HTML characters (<, >) in user input.
4. **Login Rate Limiter (Simulation)**: Write a script that tracks login attempts from an IP and blocks it after 5 tries.
5. **Directory Traversal Blocker**: Write a function that detects ../ in a file path and rejects it.
6. **Command Injection Blocker**: Write a function that validates user input against an allowed list of commands.
7. **Strong Password Generator**: Write a script to generate a random password with uppercase, lowercase, numbers, and symbols.
8. **Password Strength Checker**: Write a function that rates a password's strength based on its length and character complexity.
9. **File Upload Validator**: A script that checks an uploaded file's extension against a whitelist.
10. **File Type Checker (Magic Number)**: A script that reads the first few bytes of a file to verify its type, not just its extension.
11. **Centralized Logging Client**: A script that sends log messages to a remote server.
12. **Set Secure Cookie Headers**: Write a function for a web framework that adds HttpOnly and Secure flags to cookies.
13. **X-Frame-Options Header**: Write a function to add the X-Frame-Options: DENY header to prevent clickjacking.
14. **Content Security Policy (CSP) Generator**: Create a script that generates a basic CSP header.
15. **Deserialization Safety Check**: Before loading a pickle file, write a function to check if it contains suspicious keywords.
16. **ARP Spoofing Detector**: Monitor ARP traffic on your network for duplicate MAC address claims.
17. **Wi-Fi Deauth Detector**: Monitor Wi-Fi management frames for an unusual number of deauthentication packets.
18. **Process Whitelist Monitor**: Write a script that periodically checks running processes against a list of approved executables.
19. **File Integrity Monitor**: Calculate and store hashes of important files, then write a script to re-check them for changes.
20. **JWT Signature Verifier**: Write a function that properly verifies the signature of a JWT token before trusting its content.
21. **SSRF Defense (URL Validator)**: Write a function that checks if a URL provided by a user points to an internal or private IP address.
22. **FTP Hardening Script**: Write a script to check an FTP server's configuration file to ensure anonymous login is disabled.
23. **Web App Firewall (WAF) Rule (Simple)**: A script that checks incoming requests for SQLi or XSS patterns and blocks them.
24. **2FA Code Generator**: Implement a basic TOTP code generation function.

25. **Parameter Whitelist**: Write a function that ensures only expected parameters are present in a request.
26. **Outbound Traffic Logger**: Write a script to monitor and log all outbound network connections from your machine.
27. **Anti-Keylogger (Basic)**: A script that looks for running processes with suspicious names often used by keyloggers.
28. **Environment Variable Checker**: A script that checks if sensitive data like passwords are being stored in environment variables.
29. **Cleartext Password Scanner**: A script that searches your own codebase for cleartext passwords.
30. **Regular Expression DoS (ReDoS) Checker**: Write a simple regex and test it with a "evil" string to see how long it takes to execute, demonstrating the ReDoS concept.

Category 3: Cryptography & Forensics (50 Problems)

1. **MD5 Hasher**: A function to compute the MD5 hash of a string.
2. **SHA256 Hasher**: A function to compute the SHA-256 hash of a file.
3. **Base64 Encoder**: A script to encode a file's content into Base64.
4. **Base64 Decoder**: A script to decode a Base64 string back to its original form.
5. **Caesar Cipher Encryptor**: Implement a Caesar cipher.
6. **Caesar Cipher Brute-Forcer**: A script to decrypt a Caesar cipher by trying all possible keys.
7. **XOR Encryptor**: A script to encrypt a string by XORing it with a key.
8. **Dictionary Hash Cracker (MD5)**: A script that tries to crack an MD5 hash using a wordlist.
9. **Image Metadata Extractor**: Use Pillow to extract all EXIF data from a JPG file.
10. **String Extractor from Binary**: A script to find and print all human-readable strings from any file.
11. **Log File Parser**: A script to parse an Apache log file and count requests per IP.
12. **Log Anonymizer**: A script to replace all IP addresses in a log file with a hashed version.
13. **Find Deleted Files (Simulation)**: A script to search a raw disk image for file headers of common types (e.g., JPG, PDF).
14. **Timeline Generator**: A script to crawl a directory and create a chronological timeline of file modification times.
15. **Hex to ASCII Converter**: Convert a string of hex values into readable text.
16. **Steganography Hider (Simple)**: A basic script to hide a message in the least significant bits of an image's pixel data.
17. **Steganography Retriever (Simple)**: A script to extract the hidden message from the steganography script above.
18. **Packet Sniffer (HTTP)**: Use scapy to sniff HTTP traffic and print the host and path of GET requests.
19. **PCAP File Reader**: Use scapy to read a .pcap file and print the source and destination IP for each packet.
20. **Rainbow Table Generator (Mini)**: For a small character set, pre-compute hashes and store them in a dictionary.
21. **Rainbow Table Cracker (Mini)**: Use the pre-computed table to instantly "crack" a hash.
22. **Salting Demo**: Write a function that shows how adding a salt results in a different hash for the same password.

23. **Brute-Force with Character Set**: A password cracker that generates combinations from a defined character set (a-z, 0-9).
24. **Registry Key Reader (Windows)**: Use the winreg module to read a specific Windows Registry key.
25. **Browser History Parser (SQLite)**: Write a script to open a browser's SQLite history file and print visited URLs.
26. **Vigenère Cipher Encryptor**: Implement the Vigenère cipher.
27. **Frequency Analysis Decrypter**: A script to perform frequency analysis on a text to help break simple substitution ciphers.
28. **Hashing Race**: A script to compare the speed of MD5, SHA1, and SHA256.
29. **File Carving**: A script that searches a file for the start and end bytes of a JPG and extracts it.
30. **Prefetch File Parser (Windows)**: A script to parse a Windows Prefetch file to see application execution evidence. ... (And 20 more, covering topics like Alternate Data Streams, MFT parsing concepts, memory dump analysis, etc.)
31. **Memory Dump String Search**: Write a script to open a (small) memory dump file and search for keywords like "password".