

أهلاً بك، صديقي العزيز، في رحلة تعلم مجال الأمن السيبراني (Cybersecurity). هذه التمارين تم إعدادها في مارس 2025، وأعتقد أنها ستظل صالحة حتى يحدث تغيير جذري في المجال بالكامل، وهو أمر unlikely: D، لذا اقرأ وأنت مطمئن.

هذه هي خطوتك الأولى في رحلة طويلة، لكنها ستكون ممتعة بإذن الله.

خطة التعلم

الخطة تعتمد على اكتساب مهارة جديدة كل أسبوع، بإذن الله. هذا هو رابط الفيديو المرتبط بهذه التمارين:

[رابط الفيديو](#)

من المفترض أن تأخذ وقتك وراحتك أثناء حل التمارين، فلا تتسرع 😊.

ما نوعية التمارين؟

التمارين متنوعة وتشمل:

- تمارين مباشرة على ما تم شرحه.
- تحديات مأخوذة من مسابقات CTFs الشهيرة.
- مشكلات في حل التحديات (Problem Solving) مفيدة في المسابقات والحياة العملية.

بعد انتهائك من هذه التمارين، يُفترض أن تكون قادرًا على كتابة معظم السكريبتات (Scripts) للمستويات القادمة، بالإضافة إلى فهم الأساسيات البرمجية لمعظم اللغات التي سنستخدمها.

مستويات التمارين

التمارين مقسمة إلى سبعة مستويات:

✓ يجب عليك إكمال تحديات المستويات 1، 2، و3 دون أي مشاكل، لأن جميع المصطلحات والمفاهيم قد تم شرحها بالفعل في الفيديو. هذا هو الحد الأدنى المطلوب للانتقال إلى المهارة التالية.

! أما المستويات 4، 5، 6، و7 فستتطلب منك البحث والتعلم الذاتي لتقنيات جديدة. هذه التقنيات سيتم شرحها لاحقًا، لكن تم إدراجها هنا لاختبار مدى شغفك في البحث عن المعلومة بنفسك.

🔍 **التعلم الذاتي هو مفتاح النجاح في هذا المجال!** يجب أن تعود نفسك على البحث عن المعلومات بنفسك. وكما يُقال: "Go and get it" – لا تنتظر أن يأتيك كل شيء جاهزًا.

! إذا لم تكن لديك الجرأة الكافية الآن، لا بأس! ستصبح الأمور أسهل مع تعلم المهارات القادمة، ويمكنك العودة إلى هذه التحديات لاحقًا وحلها.

متابعة الحلول

🌸 بعد كل أسبوع، وقبل نزول المهارة التالية، سأقوم بمشاركة حلول التحديات السابقة.

قواعد البحث عن الحلول

💡 استخدم **Google** أولاً للبحث عن المشاكل.
💡 إذا واجهت صعوبة كبيرة ولم تجد الحل، يمكنك الاستفسار من **ChatGPT** لكن فقط للتوضيح وليس لحل السؤال بالكامل.

⚠️ **تنبيه مهم:**

إذا قمت بنسخ السؤال ولصقه في **ChatGPT** مباشرة للحصول على الحل، فرجاء قم
بالغاء اشتراكك والانسحاب مبكراً من الرحلة 😊.
صدقني، أنا أساعدك على توفير وقتك ومجهودك "غير المرني".

🚀 بداية سعيدة، وموفق في رحلتك!

🔥 Level 1: Python Basics & Data Types

Topics: Variables, Data Types, Strings, Numbers, Booleans, Input/Output

♦ Exercises:

1. Print `"Hello, Hacker!"` to the console.
2. Ask the user for their name and print `"Welcome, [name]!"`.
3. Convert `"1337"` (string) to an integer and add 10.
4. Create a program that asks for two numbers and prints their sum, difference, product, and quotient.
5. Reverse the string `"rekcah_repus"` **without** using built-in functions.
6. Write a program that checks if a number is even or odd.
7. Create a variable that stores a boolean value (`True` or `False`) and print it.
8. Convert the string `"100101"` to a decimal number.
9. Print `"H4ck3r"` with alternating uppercase and lowercase letters.
10. Given `password = "P@ssw0rd"`, replace all vowels with `"*"`.

🔥 Level 2: Control Flow & Loops

Topics: If-else, For loops, While loops, Logical Operators

♦ Exercises:

1. Write a script that asks for a password and only allows access if it matches `"s3cr3t"`.
2. Print all numbers from 1 to 100 **except** numbers divisible by 4.

3. Create a script that prints every 4-digit PIN code (`0000-9999`).
4. Check if a given year is a **leap year**.
5. Find prime numbers between `1` and `100`.
6. Simulate a **login attempt system** that locks after 3 failed attempts.
7. Create a **basic number guessing game**.
8. Print numbers `1-100`, replacing multiples of `3` with `"Fizz"`, multiples of `5` with `"Buzz"`, and both with `"FizzBuzz"`.
9. Use a `while` loop to **continuously ask for a password** until the correct one is entered.
10. Check if a string is a **palindrome** (same forward and backward).

Level 3: Lists, Tuples, and Dictionaries

Topics: Indexing, Iteration, Dictionary Lookups

◆ Exercises:

1. Create a list of 10 hacker tools (`["Nmap", "Metasploit", "Wireshark", "Burp Suite", "JohnTheRipper", ...]`).
2. Print the **3rd item** in the list.
3. Create a dictionary storing **common HTTP status codes** and their meanings (e.g., `{200: "OK", 404: "Not Found"}`).
4. Write a program that **counts** how many times each letter appears in a string.
5. Sort a list of random numbers **without** using `.sort()`.
6. Store **ports and their corresponding services** in a dictionary and allow the user to query by port number.
7. Write a function that removes duplicates from a list.
8. Convert a list into a comma-separated string.
9. Write a function that finds the **longest word** in a list.
10. Given a dictionary of usernames and passwords, write a script that asks for a username and prints the stored password.

Level 4: Functions & Modules

Topics: Function Definitions, Arguments, Return Values, Importing Modules

◆ Exercises:

1. Write a function that **reverses a string**.

2. Create a function that **checks if a password is strong** (at least 8 characters, includes a number, and a special character).
3. Write a function that **generates a random password** of 12 characters.
4. Use the `hashlib` module to create an **MD5 hash** of a user-entered string.
5. Write a function that checks if an **IP address** is valid.
6. Create a function that generates **random MAC addresses**.
7. Build a function that performs a **simple XOR encryption** on text.
8. Use Python's `uuid` module to generate a **random device ID**.
9. Write a function that takes a **hostname** and resolves it to an IP address.
10. Write a function that extracts **all vowels "aeiouAEIOU"** from a string.

Level 5: File Handling & OS Module

Topics: Reading/Writing Files, OS Commands

◆ Exercises:

1. Write a script that **reads a file** and prints its contents.
2. Create a script that **logs user input** to a file (`keystroke.log`).
3. Write a program that **lists all files** in the current directory.
4. Read a file and **count how many times** a given word appears.
5. Create a script that **renames all .txt files** in a folder to `.bak`.
6. Write a script that **creates a new text file**, writes a message to it, and then reads it.
7. Make a Python script that **deletes all files in a directory** (be careful!).
8. Write a script that **compresses** a file into a ZIP archive.
9. Find the **largest file** in a directory.
10. Scan a directory for files containing a **specific keyword**.

Level 6: Networking & Sockets

Topics: IP Addresses, Ports, Sockets, Requests Module

◆ Exercises:

1. Fetch and print your **public IP address** using an API.
2. Use `socket` to **scan open ports** on `scanme.nmap.org` for ports `20-100`.
3. Write a script that sends an **HTTP request** to `example.com` and prints the response headers.
4. Create a **simple client-server** program using `socket`.
5. Write a program that **resolves a domain name** to an IP address.
6. Create a program that sends **fake HTTP headers** to mimic different browsers.

7. Write a script that checks if a **website is online or down**.
8. Implement a script that finds **live hosts** on a local network.
9. Write a program that pings a **list of IPs** and prints their status.
10. Use Python to **brute-force a login page** with different passwords.

Level 7: Hacking Simulations & Mini Projects

Topics: Bringing It All Together

♦ Exercises:

1. Generate **10 fake email addresses**.
2. Simulate a **Caesar cipher encryption/decryption** tool.
3. Build a script that **scrapes all links from a webpage** using **BeautifulSoup**.
4. Create a simple **keylogger** that logs keystrokes to a file (ethical use only).
5. Write a script that detects if a **website is up or down**.
6. Make a tool that **extracts metadata from an image file**.
7. Simulate a **basic phishing email generator**.
8. Create a simple **packet sniffer** that captures network traffic.
9. Write a script that generates **fake credit card numbers** for testing.
10. Simulate a **DDoS attack** script.