🔥 هل أنت مستعد لبدء رحلة لا يخوضها إلا القلّة؟ 🔥
مرحبًا بك في مغامرتك الجديدة مع نظام الويندوز — ولكن ليس كما اعتدت أن تراه!
هنا، لن تقتصر رحلتك على تشغيل النظام وفتح المتصفح فحسب... بل ستغوص في أعماق النظام، وتكشف أسراره وخفاياه كما يفعل المحترفون الحقيقيون.
كل تفصيلة، كل أمر، كل حركة — لها أهمية قصوى، وستكون سلاحك الذي تحتاجه في ميادين الـRed Teaming، واختبار الاختراق (Pentesting)، والدفاع السيبراني (Blue Teaming)، وعمل مراكز العمليات الأمنية (SOC)، وحتى في تحقيقات الأدلة الجنائية الرقمية (Forensics)! 🚀

في عالم الأمن السيبراني، من لا يفهم النظام بعمق، يبقى مجرد هاوٍ.
اختر أن تكون من النخبة، وانطلق في رحلتك الحقيقية نحو الاحتراف! 🌟

🎥 يمكنك مشاهدة الحلقة عبر هذا الرابط:
https://www.youtube.com/watch?v=yRzB0chPnmw&t

✊ إذا كنت مستعدًا للانضمام إلى صفوف الصفوة... فابدأ الآن!

Hint: import platform might help ;)

As usual, you have to solve only 3 days out of 7 to get to the next level, but if you want more challenges you can try to solve it all.

🔥 Day 1: Windows Basics & System Information

Topics: OS Detection, Basic System Info, Environment Variables

🔹 Exercises:

1. Write a script that prints the **Windows version** (e.g., Windows 10, 11).
2. Print the **computer name** and the **current logged-in user**.
3. List all **environment variables** and their values.
4. Get and print the **current working directory**.
5. Find and display the **system architecture** (32-bit or 64-bit).
6. Retrieve the **current user's home directory path**.
7. Write a script to detect if the system is **Windows or not**.
8. Print the **system boot time**.
9. Retrieve and print the **CPU name** and **number of cores**.
10. Write a script that checks if a **specific environment variable** (like `PATH`) exists.

🔥 Day 2: Processes, Tasks, and Services

Topics: Process Listing, Management, Services Control

🔹 **Exercises:**

1. List all **running processes** along with their PID (Process ID).
2. Write a script that **kills** a process by its **name**.
3. Start a new process (like opening **Notepad**) from Python.
4. Monitor and print **CPU usage** of the top 5 processes.
5. Detect if a given process (like `explorer.exe`) is **running**.
6. Write a script to **restart** a Windows service (like `Spooler`).
7. List all **services**, their status (`Running`, `Stopped`), and startup type.
8. Write a script to **start**, **stop**, or **restart** a service programmatically.
9. Create a simple **task manager clone** that refreshes every 5 seconds.
10. Find and terminate **all processes** started by a specific user.

🔥 Day 3: File System & Automation

Topics: File Operations, Directories, File Metadata

🔹 **Exercises:**

1. List all **files and folders** in `C:\Users\Public`.
2. Create a script that **copies a file** from one location to another.
3. Write a Python tool that **monitors a directory** for file changes.

4. Find the **size** of every file inside a specific folder.
5. Automatically create **daily backups** of a folder.
6. Write a script that **searches for a file** by name across a directory tree.
7. Monitor and report any **new files** created in `C:\Windows\Temp`.
8. Create a script that **zips** and **unzips** folders.
9. List all files that were **modified in the last 24 hours**.
10. Write a tool that finds all files **bigger than 100MB** in a directory.

🔥 Day 4: Registry Operations

Topics: Windows Registry Read/Write, Registry Paths

🔹 Exercises:

1. Read the **Windows version** directly from the registry.
2. Write a script that **adds a key** to `HKEY_CURRENT_USER\Software\TestKey`.
3. Modify an existing **registry value**.
4. Delete a specific **registry key** and **confirm** its removal.
5. Search the registry for **startup programs**.
6. Write a tool that lists all **autorun entries** in the registry.
7. Create a script to **disable Task Manager** by editing the registry.
8. Backup a registry key to a `.reg` file.
9. Detect if **Remote Desktop** is enabled via the registry.

10. Write a program that **automatically re-enables** Windows Defender if it's disabled in the registry.

🔥 Day 5: WMI & System Control

Topics: WMI Queries, System Information Gathering, Management

🔹 **Exercises:**

1. Query and list all installed **software applications** using WMI.
2. Get a list of all **network adapters** and their IPs.
3. Retrieve and print **disk usage** (total, used, free) for each drive.
4. Query the system for all **user accounts** on the machine.
5. Find **all connected USB devices**.
6. Write a script that detects **battery percentage** and charging status.
7. Get all **running services** that are set to **auto-start**.
8. Retrieve **motherboard manufacturer** and **BIOS version** information.
9. Detect if the computer is a **virtual machine (VMWare, VirtualBox)**.
10. Write a script that lists all **system restore points**.

🔥 Day 6: Windows Security & Event Logs

Topics: Security Settings, Event Logs, Audit

🔹 **Exercises:**

1. List all users who are **members of the Administrators group**.

2. Write a script that monitors **Windows Security Event Logs** for login failures.
3. Detect if **UAC (User Account Control)** is enabled.
4. Find **recent software installations** using Event Logs.
5. Monitor **firewall status** using Python.
6. Get the **status of BitLocker** for each drive.
7. Find all users who **logged in the last 24 hours**.
8. Write a script that detects **suspicious services** (e.g., services running from unusual locations).
9. Monitor and log any changes to the **hosts file** (`C:\Windows\System32\drivers\etc\hosts`).
10. Write a tool that lists all **open network connections** with their PIDs (like `netstat`).

🔥 Day 7: PowerShell Automation & Advanced Control

Topics: Controlling Windows via Python, PowerShell Commands

🔹 Exercises:

1. Launch a **PowerShell script** from Python.
2. Write a script that runs a **PowerShell command** and captures the output (like `Get-Process`).
3. Create a Python wrapper that runs **`Get-EventLog`** and filters by Event ID.
4. Use Python to **enable or disable services** using PowerShell commands.
5. Automate **Windows Updates** with PowerShell triggered from Python.
6. Write a script to **create a scheduled task** using PowerShell.

7. Create a Python program that **adds a user** to the system via PowerShell.

8. List all **available PowerShell modules** installed on the machine.

9. Write a script that **enables RDP (Remote Desktop Protocol)** using PowerShell commands.

10. Create a Python tool that **sets the system timezone** using PowerShell.