CTF: LazyAdmin
Description: Easy Linux machine to practice your skills
Level: Easy
Type: Web / Linux
Objective: user_flag / root_flag

I began the reconnaissance phase using nmap -sV -v , which revealed two primary services running on the target: an Apache HTTP server and an OpenSSH service, both operating on Ubuntu. With the initial surface mapped, I proceeded with directory enumeration using gobuster dir -u -w common.txt, which identified the /content directory. A second enumeration pass against /content using the same wordlist uncovered two key paths: /content/as and /content/inc. These became the main entry vectors for the exploitation process.

Inside /content/as, I discovered an instance of the SweetRice CMS. Meanwhile, /content/inc contained internal CMS structures, including a backup directory with SQL files. These files stored the SweetRice admin credentials in hashed form. After cracking the hash locally and retrieving the plaintext password, I gained administrative access to the SweetRice dashboard. Within the dashboard, I noticed that the advertisement management feature allowed uploading files directly into /content/inc/ads without proper validation, creating an unfiltered Remote Code Execution vector. I prepared a standard PHP reverse shell payload, configured it to connect back to my listener, and uploaded it through the ads section. Once the file was accessible via the web server and my Netcat listener was active, I received an initial remote shell. I upgraded it to a functional interactive terminal using Python's pty module to spawn a /bin/bash session.

With initial access established, I explored the home directory of the user itguy and retrieved the user.txt flag. In the same directory, I identified the script backup.sh, which became central to the privilege escalation stage. A closer inspection revealed that backup.sh executed /etc/copy.sh with sudo privileges and without requiring a password. After checking its permissions, I confirmed that /etc/copy.sh was writable by the current compromised user. This exposed a privilege escalation vector based on a misconfigured sudo call combined with a writable root-owned script. I replaced the content of copy.sh with a reverse shell payload relying on mkfifo and bidirectional sh communication over Netcat, pointing back to my machine. With my listener ready, executing backup.sh triggered the modified copy.sh, resulting in a new shell — this time with full root privileges.

From there, I accessed the /root directory, obtained the root.txt flag, and successfully completed the CTF, covering the full exploitation chain: web enumeration, CMS misconfiguration discovery, RCE through unsafe file upload, initial shell access, script analysis, privilege escalation via insecure sudo usage, and final root compromise.