Penetration Testing Report: Chill Hack

Executive Summary

During a sanctioned security assessment of the target IP 10.10.253.164, a comprehensive analysis revealed multiple security vulnerabilities. An aggressive Nmap scan uncovered open ports for FTP, SSH, and HTTP services. Further enumeration confirmed an anonymous FTP login was allowed, presenting an initial vector for exploitation. This led to the discovery of a file hinting at potential usernames and a server-side command filtering mechanism.

Exploitation of the HTTP service revealed a command execution vulnerability within a /secret directory. This vulnerability was exploited to gain an initial shell on the system as the www-data user, bypassing command filters through syntax manipulation. Subsequent enumeration facilitated the extraction of sensitive data, including user credentials from a MySQL database and files via a covertly established Python HTTP server.

Privilege escalation was achieved by exploiting the user's membership in the docker group, ultimately granting root access to the host machine. The compromised credentials, coupled with the misconfigured services, highlighted systemic security oversights. All vulnerabilities were documented in detail, and appropriate recommendations have been provided to bolster the target's security posture.

All penetration testing activities were conducted with strict adherence to ethical guidelines and within the bounds of the authorized testing scope. Post-assessment, the target system was returned to its pre-test state to ensure continuity of the client's operations.

NMAP Enumeration and Exploitation

An aggressive scan was conducted on the target IP **10.10.253.164** using Nmap with the **-A** flag for comprehensive service detection, version detection, OS detection, and script scanning. The results indicated that several ports were open, including FTP (21), SSH (22), and HTTP (80).

Enumeration Results:

- **FTP Enumeration (Port 21):**
    - The FTP service **vsftpd 3.0.3** was found to be open and allowed anonymous login.
    - A file named **note.txt** was retrieved, hinting at two potential usernames and a command input filter in place.
- **SSH Enumeration (Port 22):**
    - OpenSSH 7.6 was running, indicating the potential for SSH access if credentials could be obtained.
- **HTTP Enumeration (Port 80):**
    - An Apache web server was hosting a soccer-related website with several directories available.

- Notably, a **/secret** directory presented a command input interface to the user as **www-data**.
- Basic command execution was possible, bypassing certain command filters with special characters.
- The user **www-data** was able to view all users on the machine, revealing command filters in place.

Exploitation and Foothold:

- **FTP Server Exploitation:**
  - Leveraged anonymous access to retrieve **note.txt**, which contained potential usernames and hints at input filtering.
- **HTTP Server Exploitation:**
  - A Gobuster directory scan revealed several standard directories and a **/secret** endpoint.
  - The **/secret** endpoint allowed for command execution, which led to discovering filtered commands.
  - A reverse shell was successfully deployed using obfuscated commands to bypass filters.
  - A stable shell was obtained using **python3** for a better interactive shell.
- **File and Directory Discovery:**
  - Sensitive images were discovered in a directory, which were then exfiltrated to the local machine using a Python HTTP server and **wget**.
  - Steghide was used to extract data from the images without a passphrase, revealing a **backup.zip** file.
  - **fcrackzip** was used to crack the zip file, revealing a password **pass1word**.
- **MySQL Database Enumeration:**
  - Found credentials within a PHP file allowed access to the MySQL database.
  - The **webportal** database was selected, and user data including password hashes was extracted.
  - Password hashes were cracked using an online service, yielding two plaintext passwords.

Privilege Escalation:

- **SSH Access as User 'anurodh':**
  - SSH access was established using the credentials obtained from the database.

    - The user was found to be a member of the **docker** group, suggesting a possible privilege escalation vector.
- **Docker Group Exploitation:**
    - GTFOBins was referenced for an exploit granting root shell access through the docker group.
    - Executed a docker container with mounted root directory to gain root access on the host machine.

Conclusion and Recommendations: The target IP exhibited multiple vulnerabilities, from an insecure FTP configuration to inadequate command filtering mechanisms and poor password hygiene. It is recommended to:

- Disable anonymous FTP access and ensure strong, unique passwords for all user accounts.
- Harden command execution environments to prevent reverse shell deployment.
- Implement robust input validation to mitigate command injection vulnerabilities.
- Regularly audit database access and encrypt sensitive information to prevent unauthorized access.
- Review and restrict group memberships and privileges, especially for the docker group.