Started with the usual nmap -A <your-ip> which showed 2 open ports, 22 and 80.

A screenshot of a computer

AI-generated content may be incorrect.

Upon reading the content in the web app, it got me thinking about a potential file upload vulnerability but first, we need to find a way inside.

I used bot dirsearch -u http://nocturnal and gobuster dir -u http://nocturnal.htb -w /usr/share/wordlists/seclists/Discovery/Web-Content/common.txt

A screenshot of a computer

AI-generated content may be incorrect.

Most of the directories had the access restricted or they redirected me to the /login.php page.

The only way inside should be through a vulnerability inside the register/login page.

“Regular Backups: Your files are backed up regularly, preventing loss and ensuring reliability.” This was mentioned in the web app so, assuming it communicates to a database, I thought of sql injecton.

This wasn’t the case, apparently all you needed to do was creating an account and bam, where in.

A screenshot of a computer

AI-generated content may be incorrect.

So yeah, maybe a file upload vulnerability? I started my Burp to capture the request in case we had some allowed/not allowed types of files.

A simple shell from revshells.com should do ( php pentestmonkey 😉 ).

Sudo nano nocturnal.php => paste => working exploit. (hopefully)

A screenshot of a computer

AI-generated content may be incorrect.

As I thought, only certain types of files are allowed.

After creating a basic pdf i uploaded it and captured the request on burp.

A screenshot of a computer

AI-generated content may be incorrect.

After that, I uploaded my exploit and edited the content-type field entering the type for the pdf and bam, nothing.

Either I’m not doing it correctly or, the filter is able to read the content maybe? i dunno.

Maybe I will return to this later and try another method,maybe obfuscating the content.

For now, I return to what I found with dirsearch and again, nothing worked.

I returned to the upload part, renamed the exploit with .pdf and it worked, meaning that there is no filter in place which scans for content.

Apparently all I had to do was just change the extension from .php to something else accepted.

The problem is that the files can be only downloaded, not executed.There are other directories available but I can’t access them because lack of permission

I noticed that when trying to create username “admin” it doesn’t let me so, I used turbo intruder and rockyou.txt for brute force the password for admin but still nothing.

A screenshot of a computer

AI-generated content may be incorrect.

Nothing worked so I looked in page source : “view.php?username=noct&file=noct.pdf”

I’ve already seen view.php in dirsearch’s results and I knew admin is an actual user so maybe there’s another user.

I used turbo intruder for brute-forcing the username field and I found : amanda,admin and tobias but only amanda had files in her directory (privacy.odt).

So it was an LFI not file upload vulnerability after all.

Inside I found a message from the it team with the temporary password but the resolution was so bad I couldn’t read it so I used grep -r “pass” and it gave me the password.

A screenshot of a computer

AI-generated content may be incorrect.

Amanda was also an administrator of the web-app so we should find something interesting,maybe user flag?

I created a backup with amanda’s password and downloaded it to my pc, in those files was nocturnal\_database.db.

A screenshot of a computer

AI-generated content may be incorrect.

We have some passwords stored as hash so we’ll use an online hash cracker, or john.

After cracking the hash, we manage’d so ssh inside the system finding the user flag.

A screenshot of a computer

AI-generated content may be incorrect.