**This is writeup for basic pentesting THM CTF**

- CTF Level: Easy

- Date: 6/29/2025

- Platform: THM CTF

- Category: jeopardy style

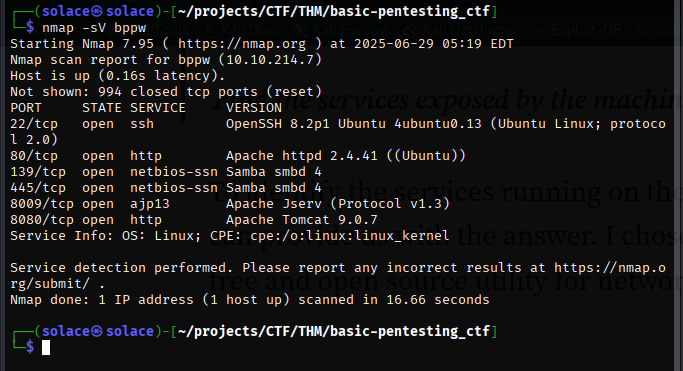
- IP: I have changed in to host(bppw)

- CTF Description: This CTF can makes observant, analytical, investigative, and to understand vulnerabilities.

**First we start with information gathering.**

**NB.** bppw is host of the target ip. Use IP in place of bppw.

To identify the services running on the target machine: nmap –sV bppw



I got some open ports with their version:

22/tcp open **ssh** OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)

80/tcp open **http** Apache httpd 2.4.41 ((Ubuntu))

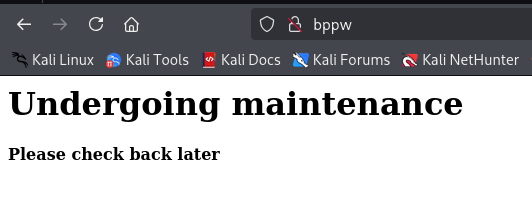
139/tcp open **netbios-ssn** Samba smbd 4

445/tcp open **netbios-ssn** Samba smbd 4

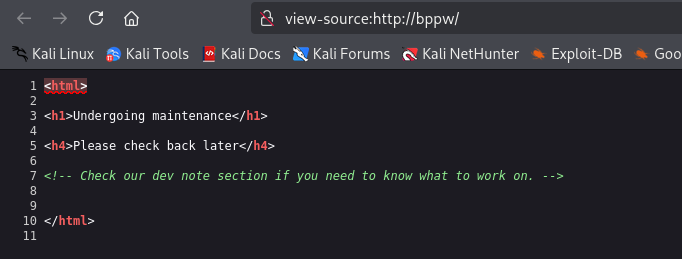
8009/tcp open **ajp13** Apache Jserv (Protocol v1.3)

8080/tcp open **http** Apache Tomcat 9.0.7

**When I visited the website on port 80/http, I came across these messages:**

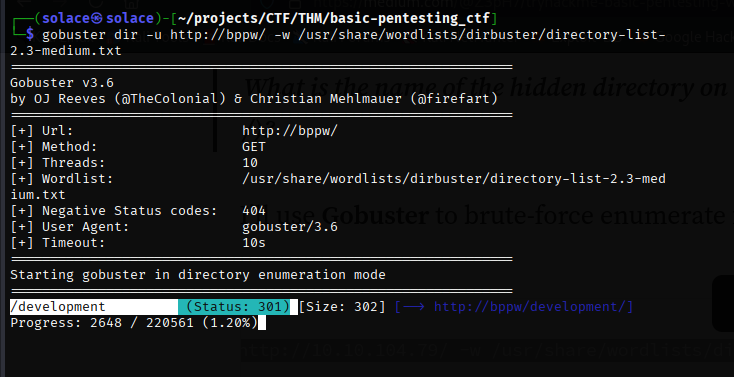


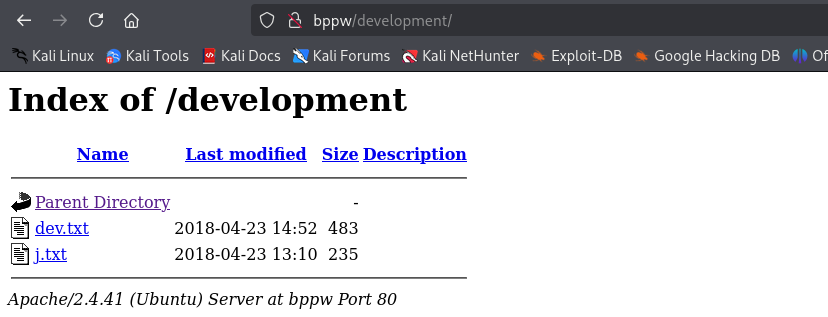
So, I decided to see view page source.



There’s a small hint here that makes us curious. Let’s goto the Gobuster directory scan and check it out.

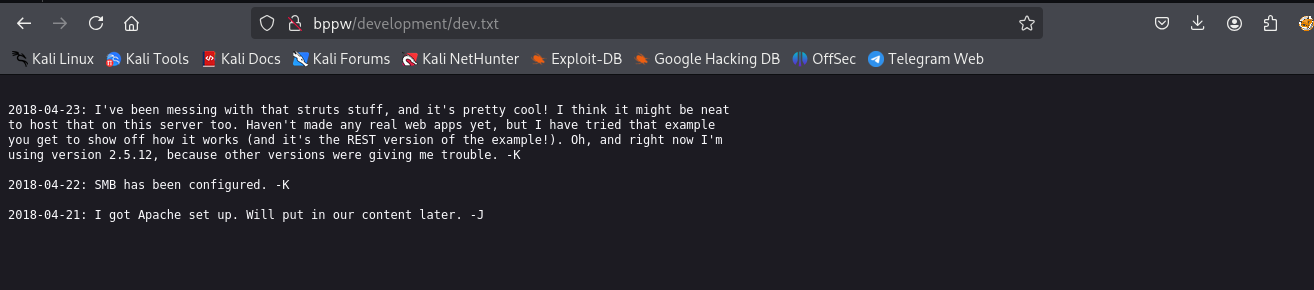
use **Gobuster** to brute-force enumerate files and directories:



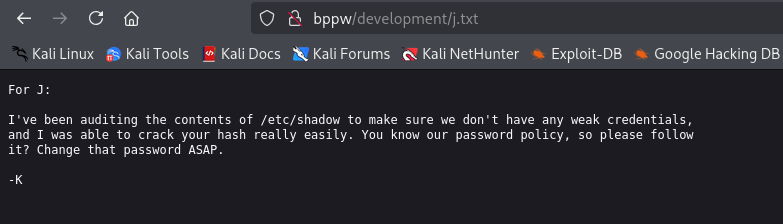


it’s a conversation among the developers, likely a report.

Let’s check dev.txt:

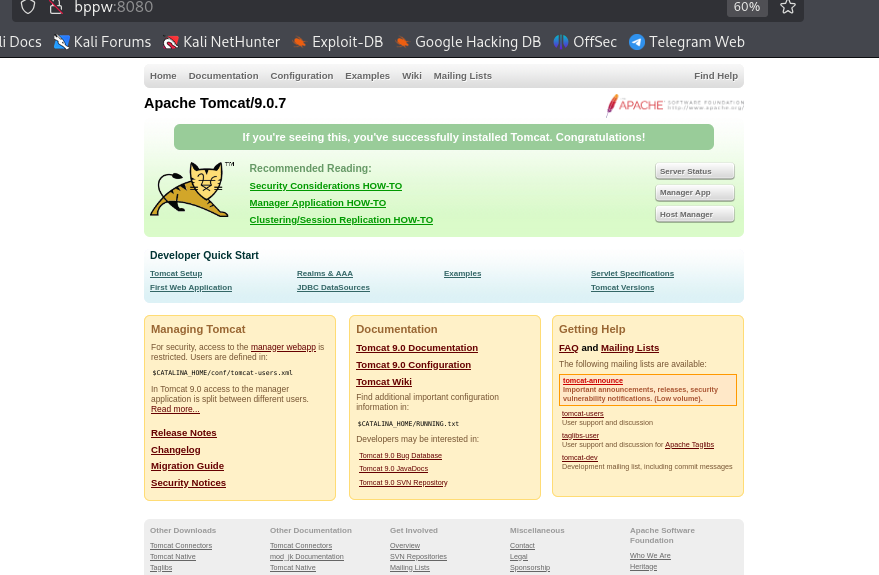


they’re reporting about weak passwords. Let’s check j.txt if they’ve changed it yet. :



Alright, based on the exploration of port 80, we gathered useful informations for exploitation, but don’t be too confident until we’ve explored each corner.

Now, **let’s visit the website on port 8080/http. :**

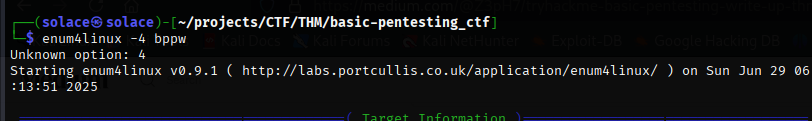


It looks like an Apache Tomcat Version 9.0.7 page.

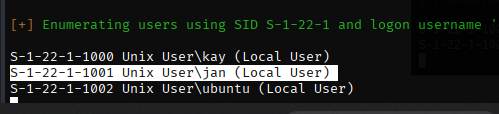
**username & password finding (User brute-forcing)**

At the nmap scan result, the samba service is running So I’ll use **enum4linux** to find users. :

Do all simple enumeration(-a): enum4linux -4 bppw

****

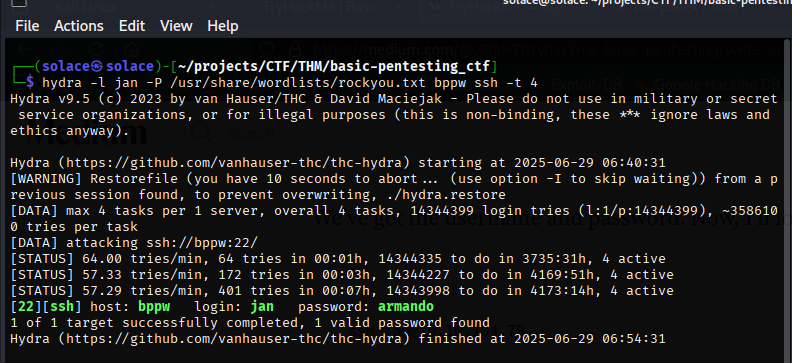
It takes some time ….finally:

****

I got the **Username** = jan

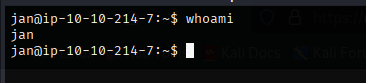
But we need to know the password, so let’s use Hydra for effective password cracking.

hydra -l jan -P /usr/share/wordlists/rockyou.txt bppw ssh -t 4



Now, I will log in via SSH. I have the **username** and **password**.

**ssh jan@10.10.104.79**



Now, I want the user.txt flag.

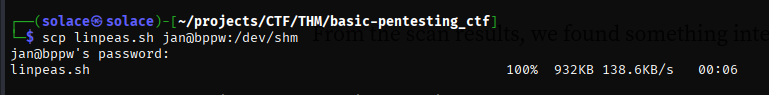
**Enumerate the machine to find any vectors for privilege escalation**

[**LinPeas**](https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS) is a shortcut to identify vulnerabilities or possible ways to escalate privileges to root.

Another method to transfer files would be using **scp**, granted we have obtained ssh user credentials on the remote host.

scp linpeas.sh jan@10.10.104.79:/dev/shm

We can do so as follows:

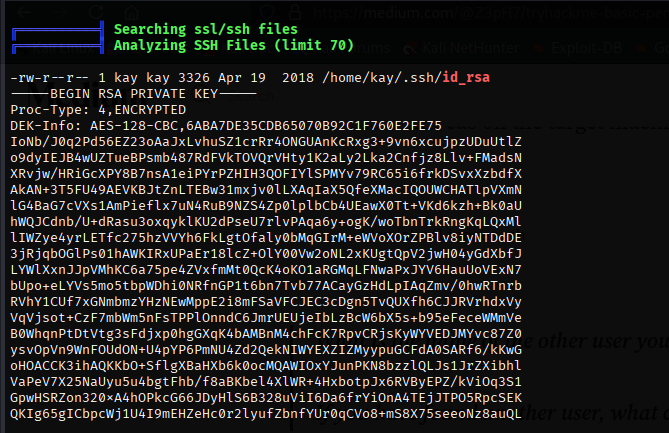
****

And, then run **LinPeas** on the **target** machine.

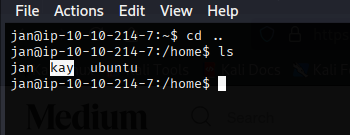
If not works give it +x permission then run again

**./dev/shm/linpeas.sh**

From the scan results, we found something interesting — kay’s id\_rsa key.



**What is the name of the other user you found(all lower case)?**



**If you have found another user, what can you do with this information?**

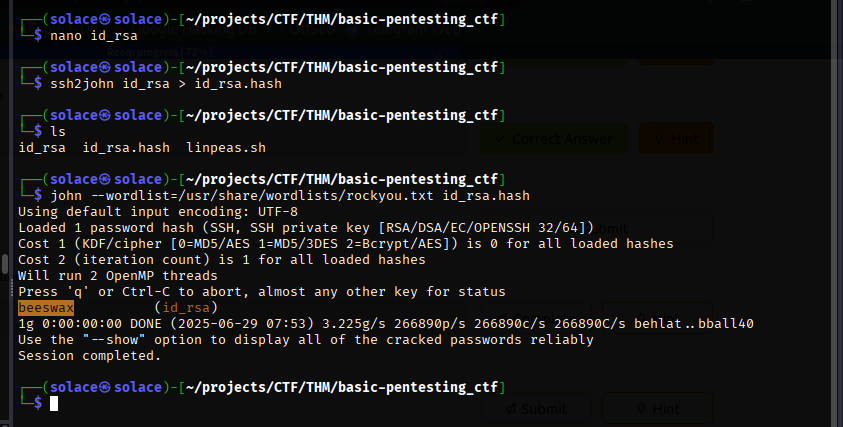
Copy the above key and create an id\_rsa file on our machine. I’ll use John the Ripper to crack this SSH hash.

For SSH hashes, you need to use ssh2john to make it easier to crack with John.

**ssh2john id\_rsa > id\_rsa.hash**

**So let’s crack the hash with John:**

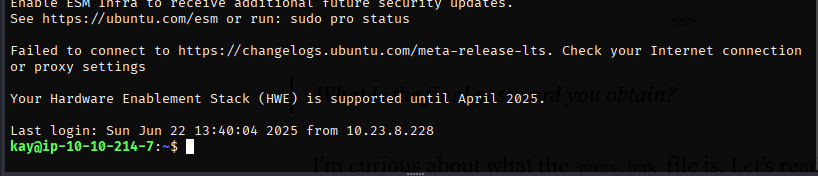
john --wordlist=/usr/share/wordlists/rockyou.txt id\_rsa.hash

****

**I found** kay’s password.

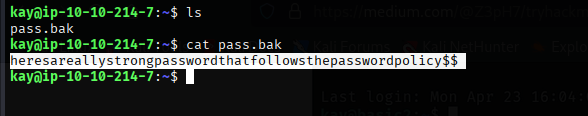
Let’s proceed by logging in to SSH on the jan machine.

**ssh -i /home/kay/.ssh/id\_rsa kay@10.10.214.7**

****

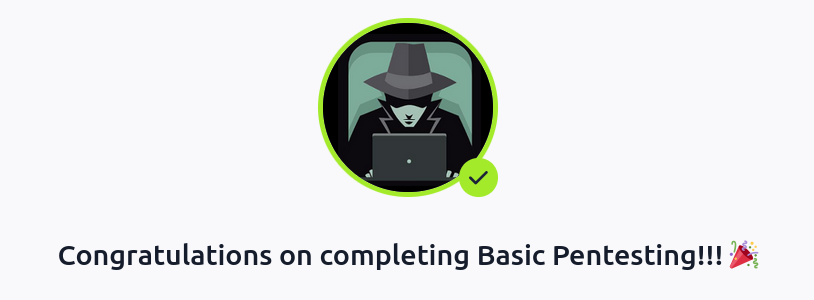
**What is the final password you obtain?**

what the pass.bak file is. Let’s read it.

****

**Finally I DID IT!!**

**CTF Mission accomplished!**

****

**Name**: Solomon Tesfaye

Ethical Hacker

**Types of Attacks:**

Brute-Force Attack, Privilege Escalation AND SSH and Samba Services

## Severity Level: High