🛡️ INET IntSec / Cowrie Honeypot Project — Step-by-Step Guide

🎯 Project Goal

The goal of this project is to simulate and monitor SSH attacks using a **honeypot** (fake SSH server) called **Cowrie**, running inside a **Virtual Machine (VM)**. This helps us:

- Log fake login attempts (both failed and successful)
- Analyze those logs to detect patterns and report them

You can simulate attacks from the **host machine**, and monitor them from the **VM**.

🧩 What It Does

- Only allows one login combo (`s1001` / `123asd`) — anything else is rejected.
- If the credentials are accepted, Cowrie opens a **fake terminal**.
- All login attempts (valid or invalid) are saved in a JSON log file.
- A Python script analyzes those logs and shows useful stats.
- Another script simulates brute-force style SSH attacks.

🔧 How to Set It Up

1. Install Ubuntu Server in a VM

- Use **VirtualBox** to create a VM and install **Ubuntu Server (no GUI needed)**.
- Enable **Port Forwarding**: port 2222

---

2. Install Cowrie inside the VM (check official website for more details)

---

3. Configure userdb.txt (default behaviour in userdb.example)

In `cowrie/etc/userdb.txt`, write:

```
s1001::123asd
*:*
```

This means:

- Only accept username `s1001` with password `123asd`
- Reject all other combinations

---

4. Activate the virtual environment for python and run Cowrie

```
cd ~/cowrie
source cowrie-env/bin/activate
bin/cowrie start
```

Use this anytime you reboot and want to start Cowrie.

---

5. Simulate Attacks from the Host

Run this on your main machine (outside the VM):

```
python3 simulate_attack.py
```

This script sends fake SSH login attempts to the VM via port 2222.

---

6. Analyze the Logs in the VM

In the VM cowrie directory, place and run `analyze_logs.py`:

```
python3 analyze_logs.py
```

This script parses `cowrie.log` and outputs:

- Top usernames attempted
- Top passwords
- IP addresses
- Timestamps of login attempts

---

7. View Screenshots and Outputs

You can check the folder:

- `fake_terminal.png` – shows the fake terminal after correct login (specified in userdb.txt)
- `login_rejected.png` – failed login message
- `results.png` – output from log analysis (analyze_logs.py)
