

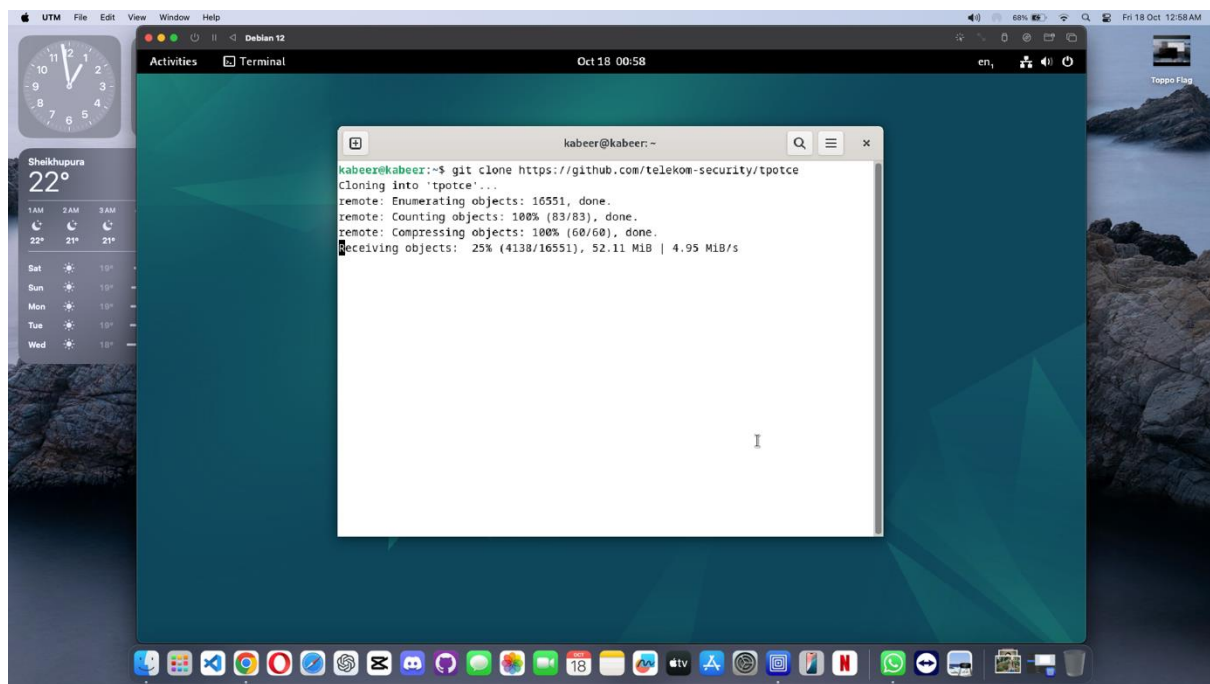
Information Security
Class Activity
HoneyPot

Name: Kabeer Ahmad

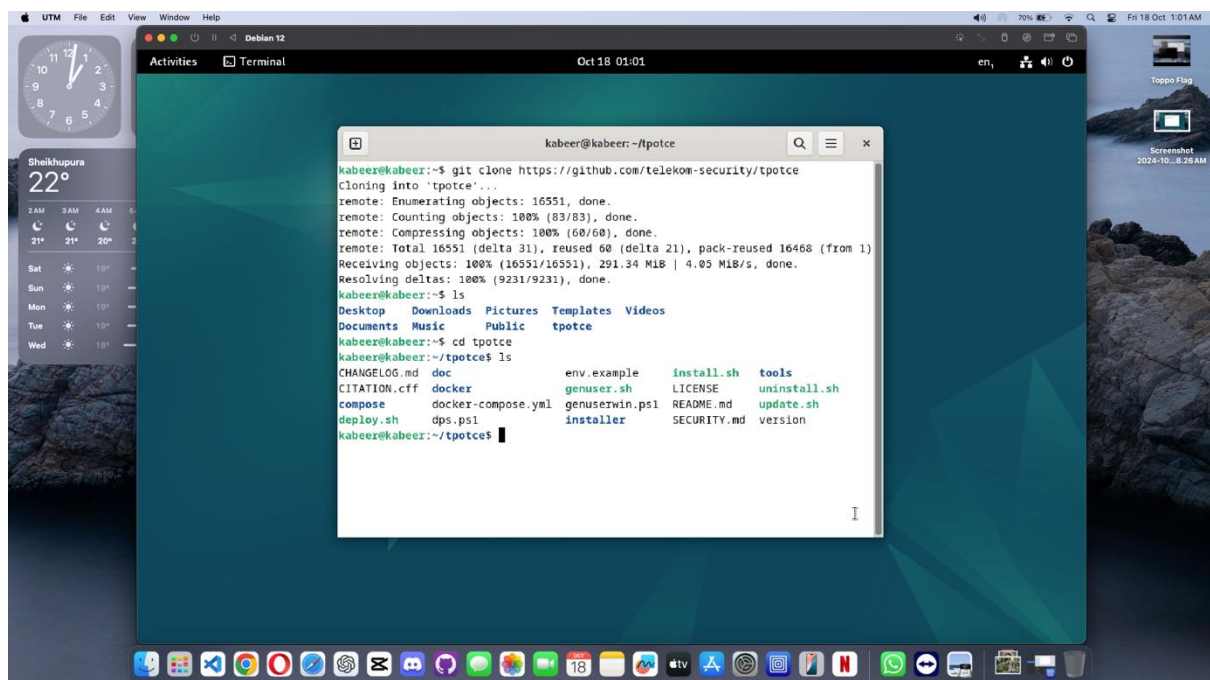
Roll No: 21L-6184

1. Clone The Repo:

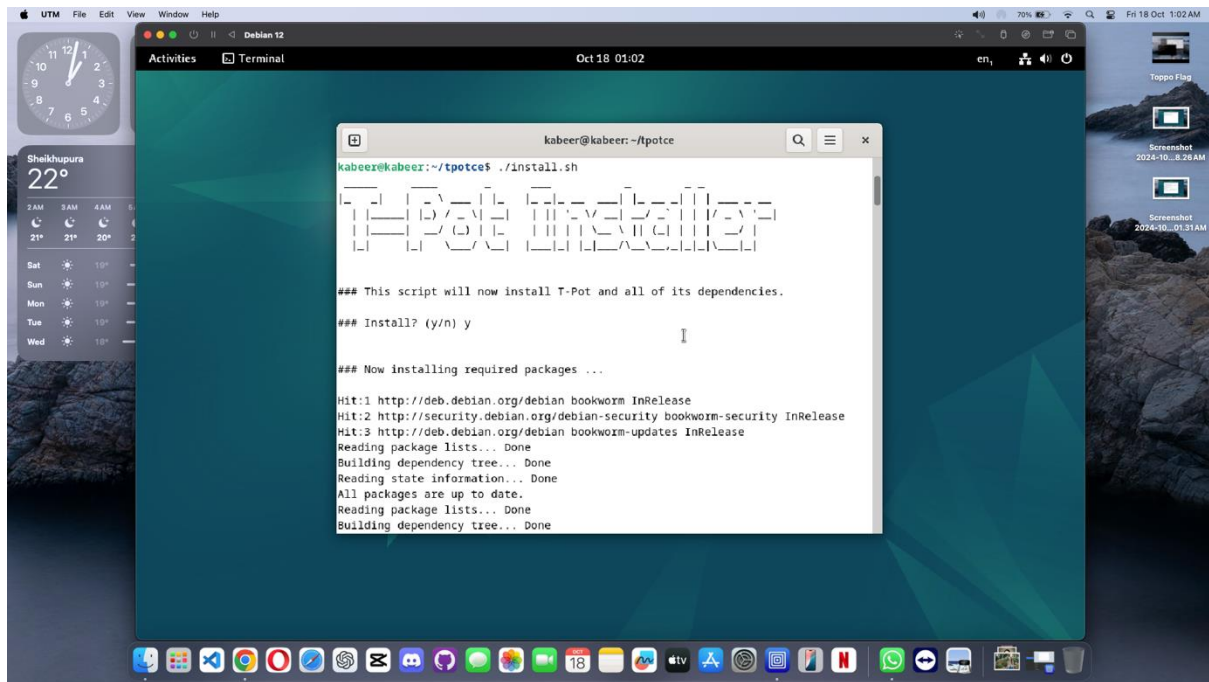
Sudo git clone <https://github.com/telekom-security/tpotce>



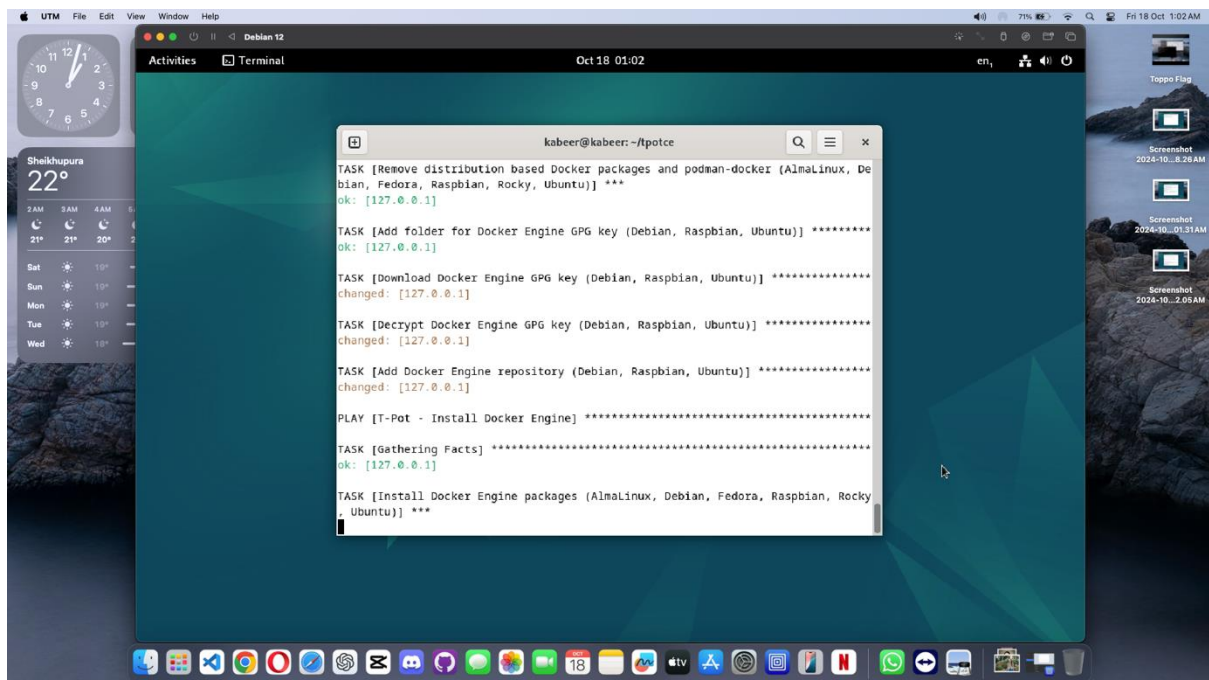
2. List Files using “ls” Command:



3. Run Installer by “./install.sh”

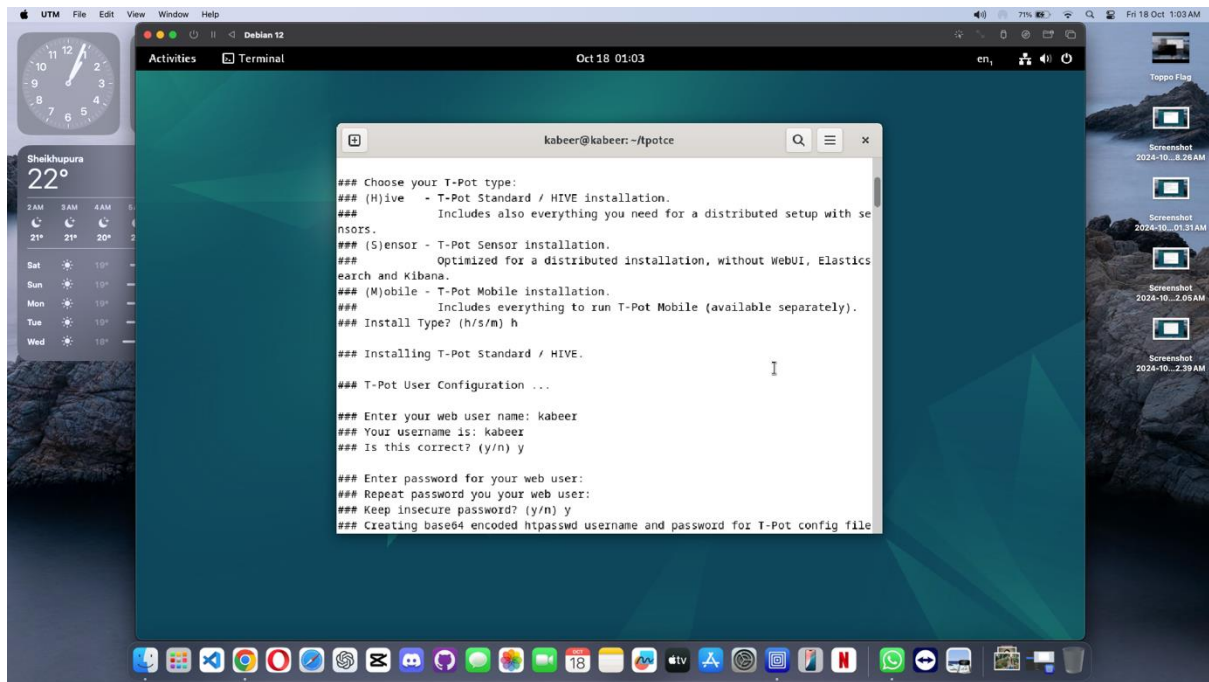


Let the installer do its work.....

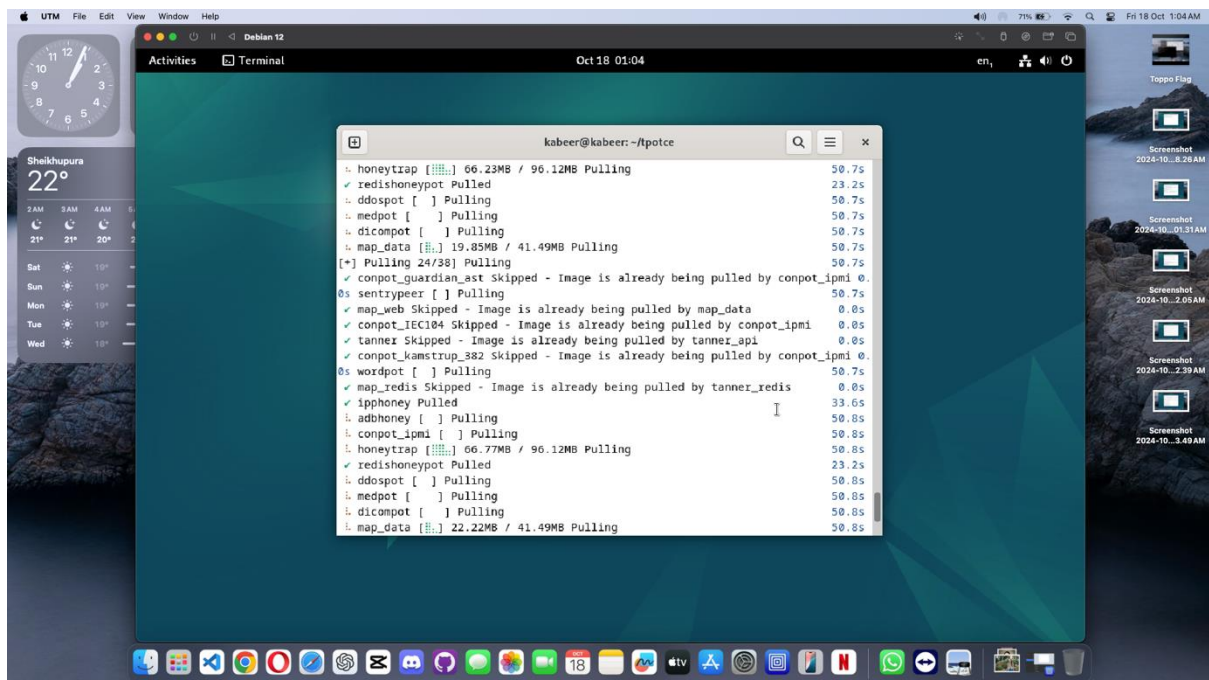


4. Use Hive “h” to install full version.

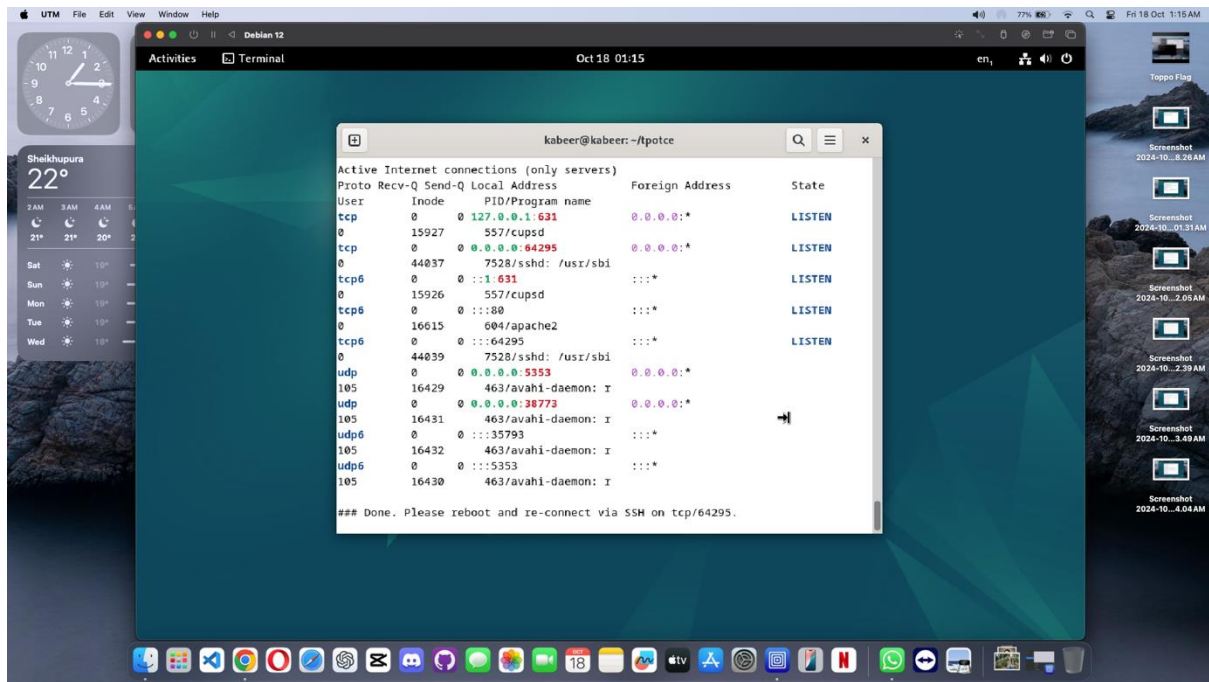
5. Enter Username and Password to Create A Account.



Now, again it will install all requirements.....

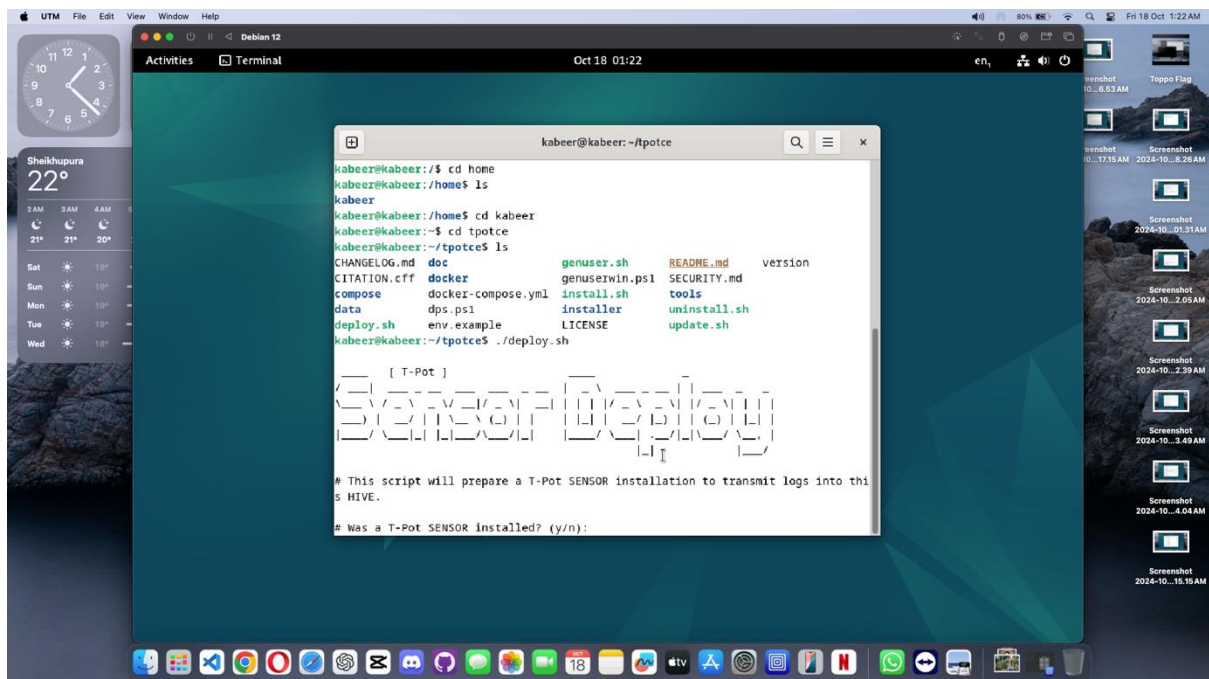


Now, it has completed the installation process, and we can see all the ports details where honeypot and other features will work.

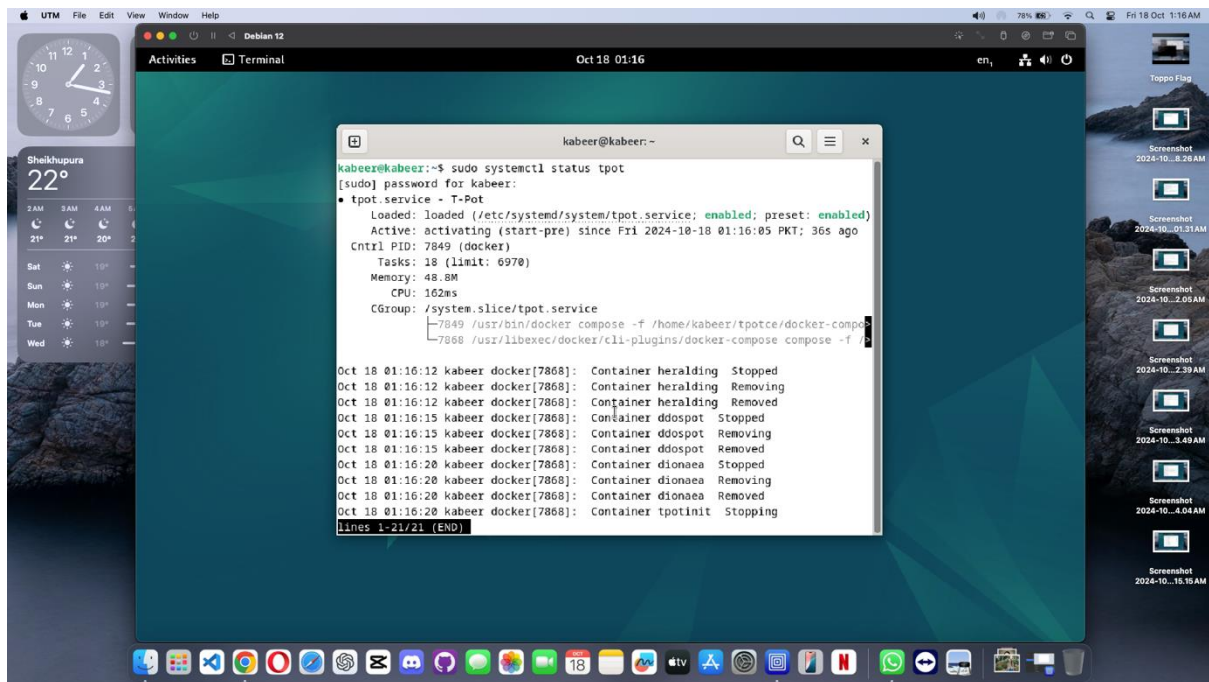


6. The Basic step is done now, for further access via other system, we can Deploy the sensor too.

Using : `./deploy.sh`



7. Check Status using “sudo systemctl status tpot”



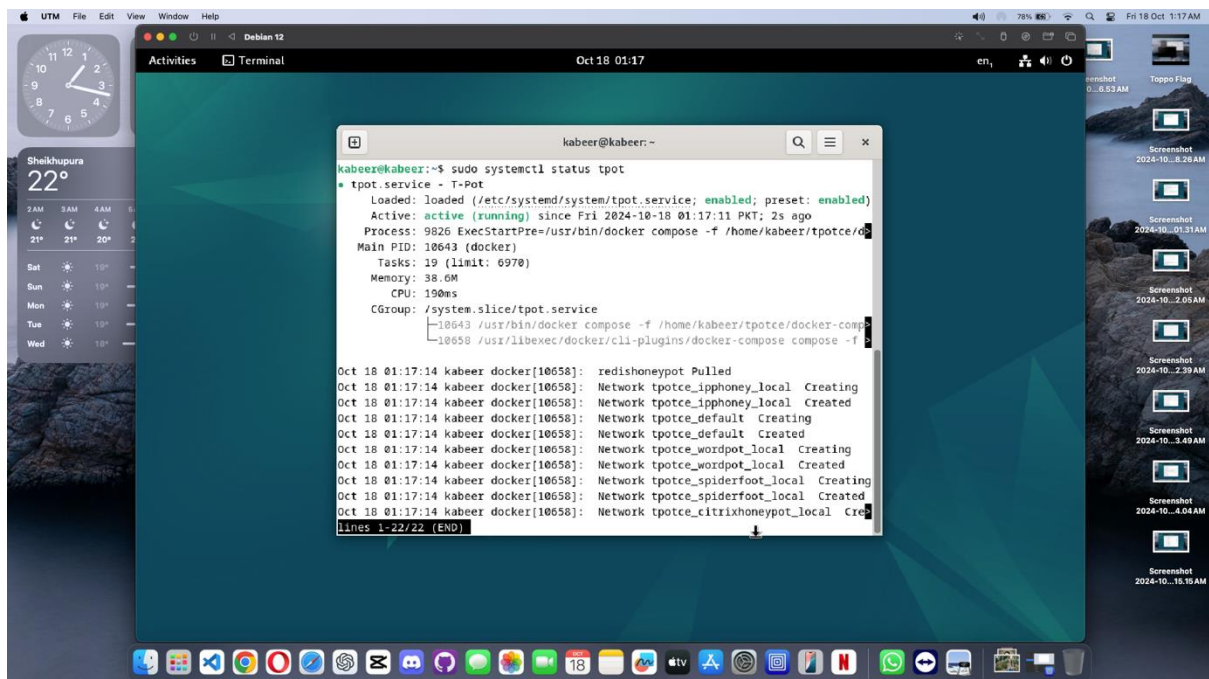
```
kabeer@kabeer:~$ sudo systemctl status tpot
[sudo] password for kabeer:
• tpot.service - T-Pot
   Loaded: loaded (/etc/systemd/system/tpot.service; enabled; preset: enabled)
   Active: activating (start-pre) since Fri 2024-10-18 01:16:05 PKT; 36s ago
     Cntrl PID: 7849 (docker)
     Tasks: 18 (limit: 6970)
    Memory: 48.8M
       CPU: 162ms
    CGroup: /system.slice/tpot.service
           └─7849 /usr/bin/docker compose -f /home/kabeer/tpotce/docker-compo
              7868 /usr/libexec/docker/cli-plugins/docker-compose compose -f /

Oct 18 01:16:12 kabeer docker[7868]: Container heralding Stopped
Oct 18 01:16:12 kabeer docker[7868]: Container heralding Removing
Oct 18 01:16:12 kabeer docker[7868]: Container heralding Removed
Oct 18 01:16:15 kabeer docker[7868]: Container ddospot Stopped
Oct 18 01:16:15 kabeer docker[7868]: Container ddospot Removing
Oct 18 01:16:15 kabeer docker[7868]: Container ddospot Removed
Oct 18 01:16:20 kabeer docker[7868]: Container dionaea Stopped
Oct 18 01:16:20 kabeer docker[7868]: Container dionaea Removing
Oct 18 01:16:20 kabeer docker[7868]: Container dionaea Removed
Oct 18 01:16:20 kabeer docker[7868]: Container tpotinit Stopping
lines 1-21/21 (END)
```

We can see, yet it is in activation state, so we can restart it.

Using: “sudo systemctl restart tpot”

Again then checking status:



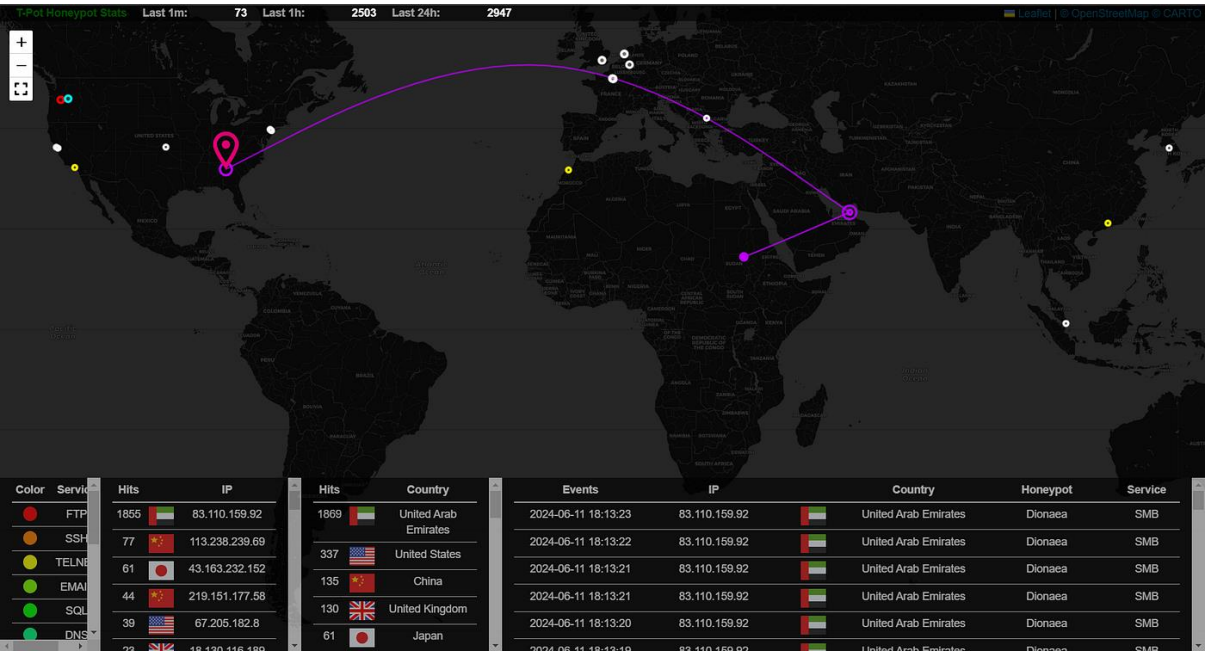
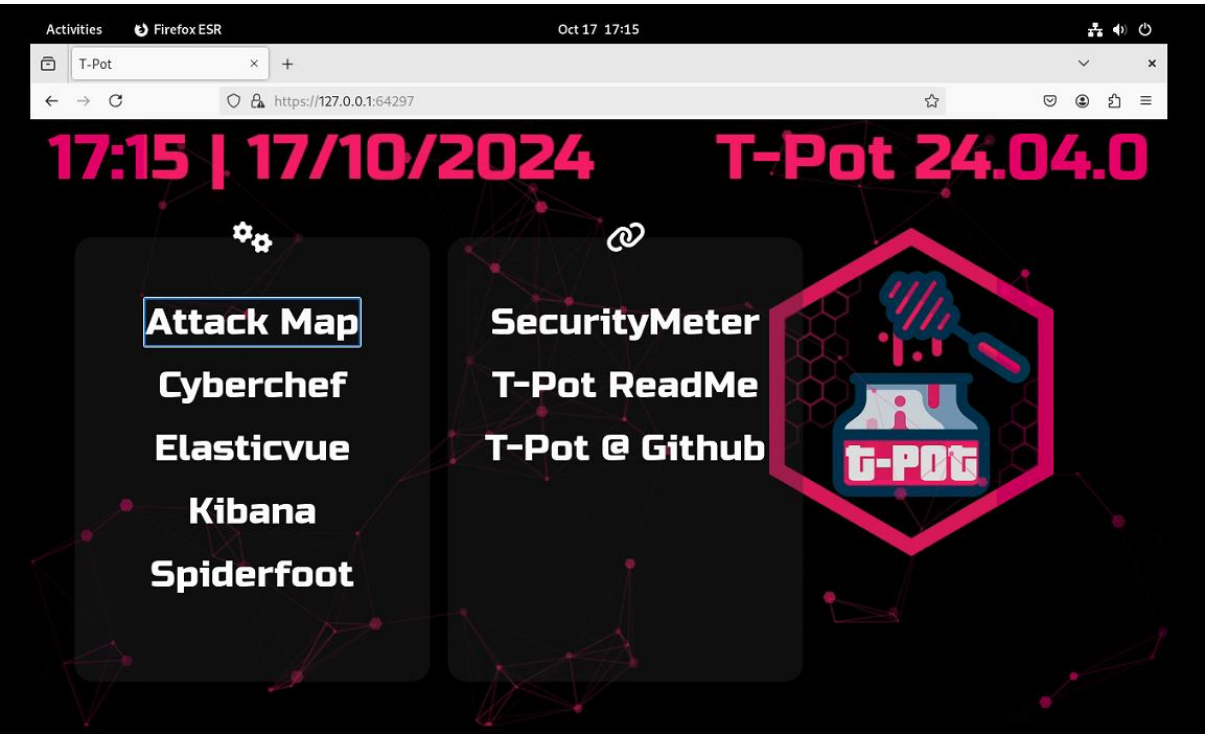
```
kabeer@kabeer:~$ sudo systemctl status tpot
• tpot.service - T-Pot
   Loaded: loaded (/etc/systemd/system/tpot.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-10-18 01:17:11 PKT; 2s ago
     Process: 9826 ExecStartPre=/usr/bin/docker compose -f /home/kabeer/tpotce/d
    Main PID: 10643 (docker)
     Tasks: 19 (limit: 6970)
    Memory: 38.6M
       CPU: 190ms
    CGroup: /system.slice/tpot.service
           └─10643 /usr/bin/docker compose -f /home/kabeer/tpotce/docker-compo
              10658 /usr/libexec/docker/cli-plugins/docker-compose compose -f /

Oct 18 01:17:14 kabeer docker[10658]: redishoney_pot Pulled
Oct 18 01:17:14 kabeer docker[10658]: Network tpotce_iphoney_local Creating
Oct 18 01:17:14 kabeer docker[10658]: Network tpotce_iphoney_local Created
Oct 18 01:17:14 kabeer docker[10658]: Network tpotce_default Creating
Oct 18 01:17:14 kabeer docker[10658]: Network tpotce_default Created
Oct 18 01:17:14 kabeer docker[10658]: Network tpotce_wordpot_local Creating
Oct 18 01:17:14 kabeer docker[10658]: Network tpotce_wordpot_local Created
Oct 18 01:17:14 kabeer docker[10658]: Network tpotce_spiderfoot_local Creating
Oct 18 01:17:14 kabeer docker[10658]: Network tpotce_spiderfoot_local Created
Oct 18 01:17:14 kabeer docker[10658]: Network tpotce_citrixhoney_pot_local Cre
lines 1-22/22 (END)
```

Now, we can see it is **ACTIVE** And **Running**.

We Can now Access It by our browser.

Using: <https://127.0.0.1:64297>



Below are some general services that can be used by TPOT, these can be used easily using few more steps, they are already installed, just have to access them.

1. Dionaea

- **Objective:** A low-interaction honeypot designed for malware capture.
- **Capabilities:** Simulates multiple protocols such as SMB, HTTP, and FTP to attract and collect malware. It is frequently used to identify worms and network-propagating malware. Dionaea focuses on obtaining samples for analysis.

2. Conpot

- **Objective:** Honeypot for Industrial Control Systems (ICS).
- **Capabilities:** Emulates elements typically found in critical infrastructure like Programmable Logic Controllers (PLCs). Conpot assists in identifying attacks aimed at industrial environments and operational technology.

3. Cowrie

- **Objective:** Honeypot for SSH and Telnet services.
- **Capabilities:** Simulates SSH and Telnet servers, capturing login attempts and recording commands entered by attackers. It provides key insights into brute force attempts and post-compromise activities such as malware execution. Sessions can be logged for further analysis.

4. Heralding

- **Objective:** Multi-protocol authentication honeypot.
- **Capabilities:** Emulates multiple services including FTP, SSH, VNC, and RDP, capturing login attempts across these protocols. This service is particularly useful for detecting brute-force attacks across various protocols.

5. Honeytrap

- **Objective:** Event-driven, flexible honeypot.
- **Capabilities:** Supports various protocols through plugins, capturing connection attempts. Honeytrap is versatile, allowing for identification of attack vectors over a broad range of protocols.

6. Glastopf

- **Objective:** Web application honeypot.
- **Capabilities:** Simulates vulnerable web applications to attract attacks targeting web exploits like SQL injection and cross-site scripting (XSS). It provides an overview of web-based threats aimed at specific application vulnerabilities.

7. Mailoney

- **Objective:** Open mail relay honeypot.
- **Capabilities:** Simulates an open SMTP server to attract spammers. Captures and analyzes spam emails, aiding in understanding spam and phishing tactics.

8. Suricata

- **Objective:** Network intrusion detection and prevention system (IDS/IPS).

- **Capabilities:** Monitors network traffic to identify and report known threats in real-time, enabling early detection of ongoing attacks on the honeypot environment.

9. **Tanner**

- **Objective:** Botnet command and control (C2) analysis.
- **Capabilities:** Monitors botnet C2 traffic, offering insights into communication methods and behavior of botnets. This helps in studying and mitigating botnet activities.

10. **Elastic Stack (Elasticsearch, Logstash, Kibana)**

- **Objective:** Data collection and visualization platform.
- **Capabilities:** Collects logs from all honeypot services, organizes and stores them in Elasticsearch, and visualizes them in Kibana. This provides real-time monitoring and analysis of honeypot data.

11. **EWS (Elasticpot, Wordpot, Sitepot)**

- **Objective:** A collection of web server honeypots.
- **Capabilities:** Emulates specific web platforms like WordPress and ElasticSearch, attracting attacks that exploit known vulnerabilities in popular content management systems.

12. **Kippo**

- **Objective:** An older SSH honeypot.
- **Capabilities:** Similar to Cowrie, Kippo captures SSH login attempts and records interactions, helping to analyze attacker behavior and techniques.

13. **Vulnwhisperer**

- **Objective:** Vulnerability management tool.
- **Capabilities:** Gathers and processes vulnerability data from multiple sources, integrating it into Elasticsearch for further analysis and visualization.

14. **Spiderfoot**

- **Objective:** Automated threat intelligence gathering.
- **Capabilities:** Collects intelligence on IPs, domains, and other entities using over 100 data sources to provide comprehensive threat intelligence and reconnaissance data.

15. **Cyтомic Falcon Sensor**

- **Objective:** Endpoint detection and response (EDR) tool.
- **Capabilities:** Monitors system behavior and detects suspicious activity, providing real-time detection and response to security incidents on endpoints.

16. **IPTables and Fail2Ban**

- **Objective:** Basic security and firewalling.
- **Capabilities:** Protects the honeypot system by blocking malicious IPs and limiting unauthorized access attempts. Fail2Ban helps in dynamically banning IPs after repeated failed login attempts.

17. **Kibana Dashboards**

- **Objective:** Data visualization and monitoring.

- **Capabilities:** Customizable dashboards that allow security analysts to monitor attacks, trends, and collected data from various honeypots in an easy-to-understand graphical format.