

SOC Analyst  
**PROJECT: Shadow Sentry**

Student: David Lim

Student code: s7

SOC Analyst trainer: James

Class code: CFC020823

## SOC Project

This project outlines the setup and deployment of a Security Operations Center (SOC) using Elastic Cloud on DigitalOcean. The goal is to detect and analyze network threats through the installation of a honeypot and the development of penetration testing scripts. Key steps include:

- Project Objectives:**
1. **Elastic Cloud Deployment:** Installation of Elastic Cloud on DigitalOcean, configuring Elasticsearch, Logstash, Kibana for effective communication.
  2. **Honeypot Installation:** Selection and deployment of a honeypot solution (e.g., Cowrie, Honeyd) on DigitalOcean, including network and logging configurations to mimic real services.
  3. **Honeypot Security:** Harden the honeypot server by disabling unused services, updating security measures, and configuring firewalls.
  4. **Attack Scripts:** Creation of scripts to simulate different types of cyber attacks, with a focus on automation and user interaction for target selection.
  5. **Integration and Alerting:** Setup of Logstash for log ingestion and Elasticsearch for log storage. Creation of Kibana dashboards for real-time monitoring and alert configurations based on known threats.
  6. **Testing and Validation:** Conduct penetration tests using the created scripts, monitor for alerts within the Elastic Stack, and validate the system's effectiveness.
  7. **Documentation:** Document the setup process and findings, providing insights and recommendations for security improvements.

This concise approach focuses on establishing a robust SOC framework to enhance network security through detection, analysis, and proactive threat management.

- 1) Setting up digital ocean and install honeypot
- 2) Creating an attack script containing 3 types of attack
- 3) Documentation

Name of honeypot chosen: T-Pot

Attack chosen:

1. Bruteforce
2. DDOS
3. Sql Injection

Honeypot chosen: T-pot

T-Pot CE: A Comprehensive Cybersecurity Tool

Introduction to T-Pot CE

T-Pot CE, or T-Pot Community Edition, is a comprehensive honeypot platform developed and maintained by Deutsche Telekom's Security Team. It's designed as an all-in-one multi-honeypot platform, enabling the deployment and management of multiple honeypots to simulate various systems and services that could be targeted by attackers. It's crafted to act as a digital decoy, attracting hackers and analyzing their tactics. This approach allows organizations to detect, analyze, and understand attack methods by observing how attackers interact with these decoy systems. This platform is essentially a virtual battleground, designed to understand and mitigate cyber threats by simulating vulnerable systems.

The platform integrates more than 20 different honeypots, including well-known ones like Dionaea, Cowrie, and Honeytrap, alongside various security and visualization tools such as the Elastic Stack (Elasticsearch, Logstash, and Kibana) for data analysis and visualization, CyberChef for data encoding and encryption, and Suricata for network security monitoring. T-Pot is designed to provide a seamless experience with a pre-configured environment where all these components work together out of the box. Making it a versatile tool for simulating a range of systems and services that hackers might target, this diversity helps in capturing a wide array of attack methods and understanding the hackers' approach.

T-Pot can be installed on physical hardware, in virtual environments, or cloud platforms using a prebuilt ISO image provided by the developers or by creating your own installation media. The system is optimized for easy deployment, requiring minimal user interaction during setup, and is designed to run without much maintenance.

T-Pot is a valuable tool for cybersecurity professionals and researchers, offering a sophisticated platform for threat detection and analysis. Its community-driven nature ensures that it stays updated with the latest honeypot technologies and attack detection methods. The community-driven nature of T-Pot ensures that it remains updated with cutting-edge honeypot technologies and defensive tactics.

Conclusion

T-Pot CE is more than just a cybersecurity tool; it's a proactive approach to understanding and combating cyber threats. Its comprehensive design, combined with ease of deployment and a wide range of integrated tools, makes it an essential platform for cybersecurity defense and research. By employing T-Pot, organizations can significantly enhance their ability to detect, analyze, and protect against cyber attacks, ensuring a more secure digital environment.

## Choosing DigitalOcean to host the honeypot.

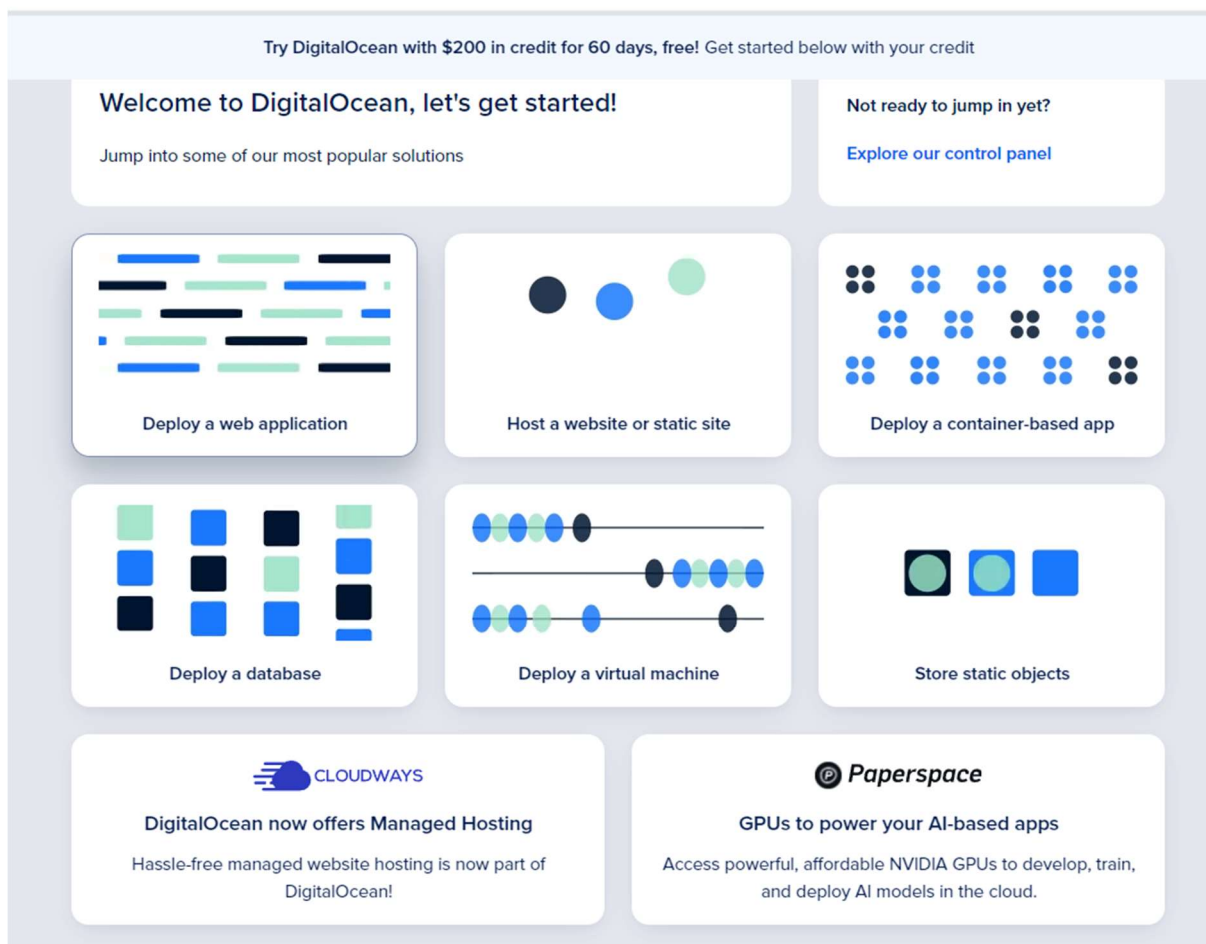


Figure 1.1.1 Screenshot of Creating droplets with the following setting.

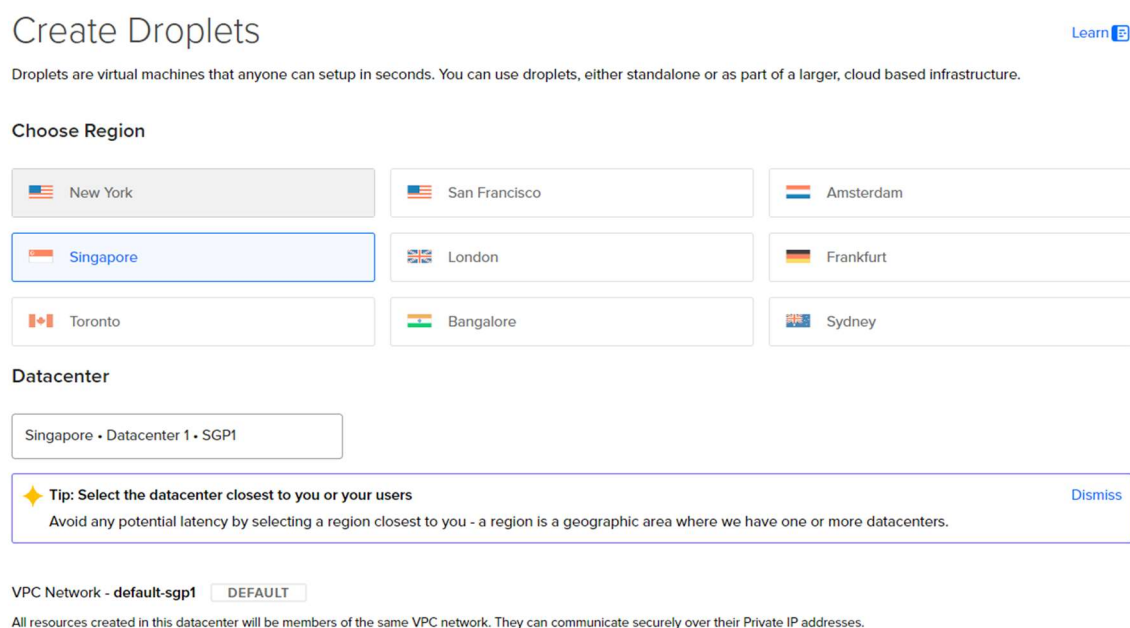





Figure 1.1.2 Screenshot of Choosing the region (Singapore)


### Choose an image


OS Marketplace Custom images


Ubuntu

Fedora

Debian

CentOS

AlmaLinux

Rocky Linux

Version  
11 x64

**Figure 1.1.3** Screenshot of Choosing Debian 11 as the OS image.

### Choose Size

Need help picking a plan? [Help me choose](#)

#### Droplet Type

SHARED CPU	DEDICATED CPU			
<b>Basic</b> (Plan selected)	General Purpose	CPU-Optimized	Memory-Optimized	Storage-Optimized

Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

#### CPU options

☒ **Regular**  
Disk type: SSD

☐ **Premium Intel**  
Disk: NVMe SSD

☐ **Premium AMD**  
Disk: NVMe SSD

\$6/mo \$0.009/hour	\$12/mo \$0.018/hour	\$18/mo \$0.027/hour	\$24/mo \$0.036/hour	<b>\$48/mo \$0.071/hour</b>	\$96/mo \$0.143/hour
1 GB / 1 CPU 25 GB SSD Disk 1000 GB transfer	2 GB / 1 CPU 50 GB SSD Disk 2 TB transfer	2 GB / 2 CPUs 60 GB SSD Disk 3 TB transfer	4 GB / 2 CPUs 80 GB SSD Disk 4 TB transfer	<b>8 GB / 4 CPUs 160 GB SSD Disk 5 TB transfer</b>	16 GB / 8 CPUs 320 GB SSD Disk 6 TB transfer

**Figure 1.1.4** Screenshot of Choosing the number of CPU and ram (4cpu/8gb)

Access

Power

Volumes

Resize

Networking

Backups

## Droplet Console

Use the Droplet Console for native-like terminal access to yo the new console.

Log in as...  
root

**Launch Droplet Console**

**Figure 1.1.5** Screenshot of Choosing Click onto Access, then Launch Droplet Console to start the cloud machine.

## Installing T-Pot CE

First, “apt-get update && apt-get install git -y” to install git

Follow by “git clone https://github.com/telekom-security/tpotce”

Cd into the dir and start the process by “cd tpotce/iso/installer/ ; ./install.sh --type=user”

```
root@debian-s-4vcpu-8gb-sgp1-01:~# git clone https://github.com/telekom-security/tpotce
-bash: git: command not found
root@debian-s-4vcpu-8gb-sgp1-01:~# apt-get update
Hit:1 http://security.debian.org/debian-security bullseye-security InRelease
Hit:2 http://deb.debian.org/debian bullseye InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Hit:4 http://deb.debian.org/debian bullseye-backports InRelease
Hit:5 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Reading package lists... Done
root@debian-s-4vcpu-8gb-sgp1-01:~# apt-get install git
```

Figure 1.2.1 Screenshot of Updating Package Lists and Installing Git

```
root@debian-s-4vcpu-8gb-sgp1-01:~# git clone https://github.com/telekom-security/tpotce
Cloning into 'tpotce'...
remote: Enumerating objects: 15218, done.
remote: Counting objects: 100% (352/352), done.
remote: Compressing objects: 100% (216/216), done.
remote: Total 15218 (delta 157), reused 314 (delta 130), pack-reused 14866
Receiving objects: 100% (15218/15218), 241.57 MiB | 14.78 MiB/s, done.
Resolving deltas: 100% (8423/8423), done.
root@debian-s-4vcpu-8gb-sgp1-01:~# ls
tpotce
root@debian-s-4vcpu-8gb-sgp1-01:~# cd tpotce/iso/installer/
root@debian-s-4vcpu-8gb-sgp1-01:~/tpotce/iso/installer# ./install.sh --type=user
```

Figure 1.2.2 Screenshot of Navigating to the Installer Directory and Initiating the Installation Process

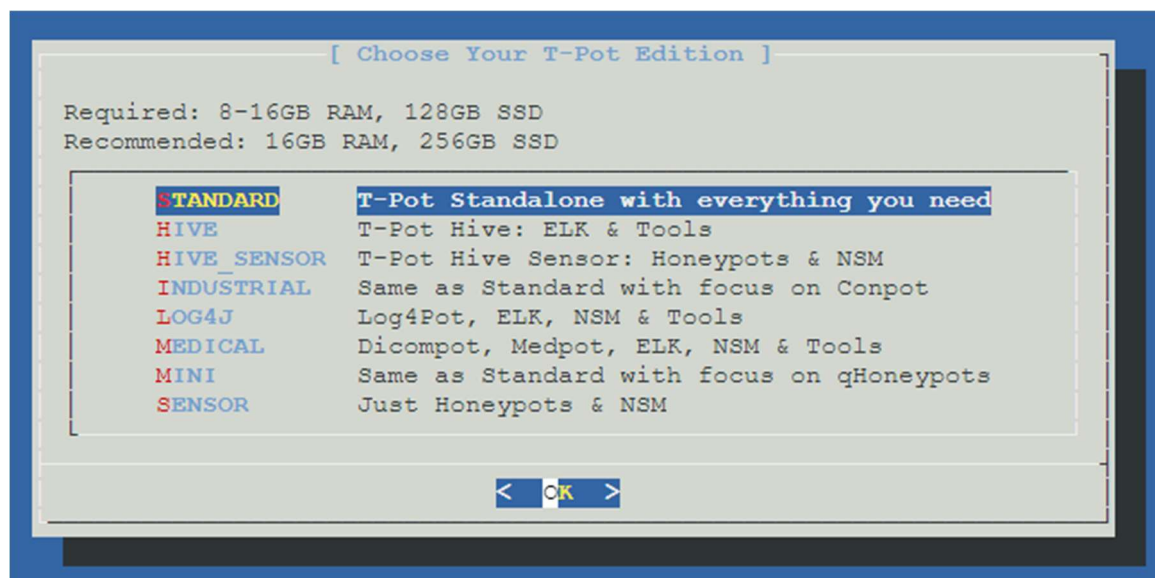
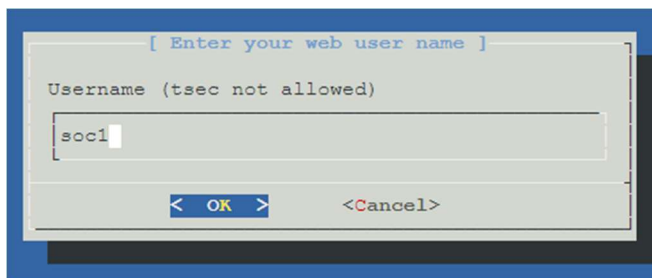
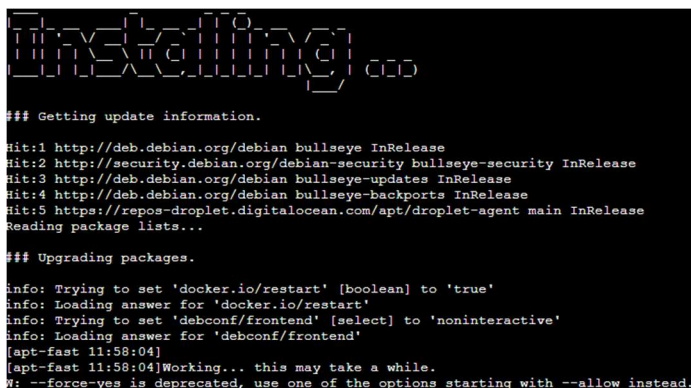


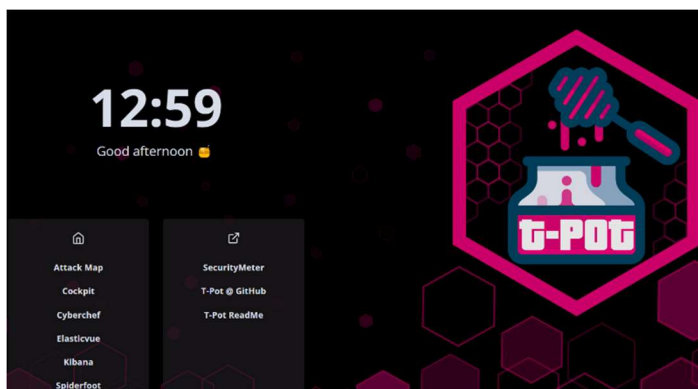
Figure 1.2.3 Screenshot of Selecting the 'STANDARD' Option from the T-Pot Installation Menu



**Figure 1.2.4** Screenshot of Inputting 'soc1' as the Web User Name in the Setup Configuration

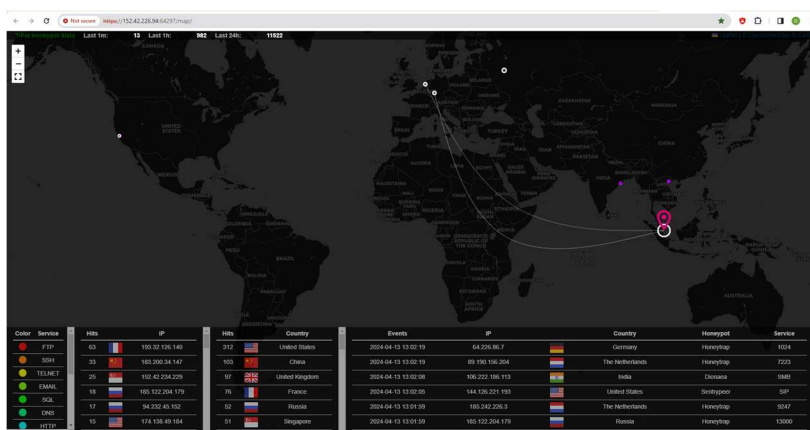


**Figure 1.2.5** Screenshot of Updating System Packages and Installing T-Pot Dependencies.

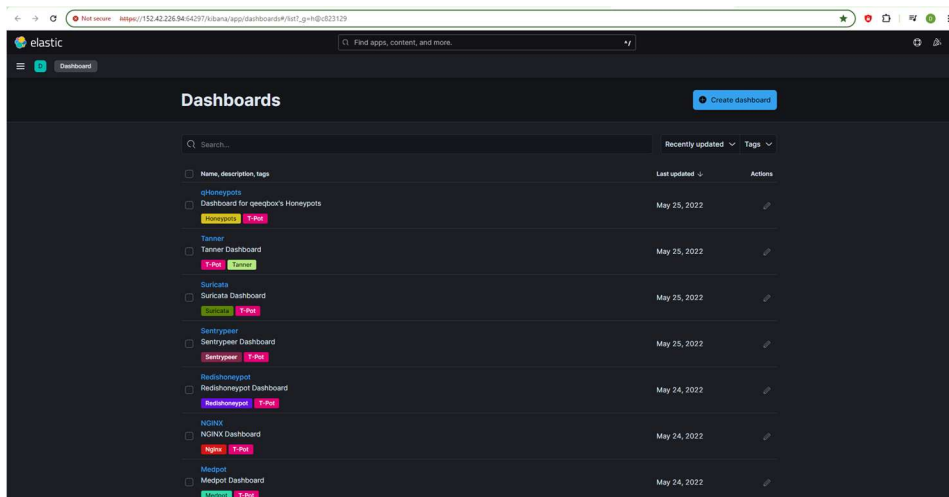


- 1) Navigate to Attack Map
- 2) Navigate to Cockpit
- 3) Navigate to Cyberchief
- 4) Navigate to Elasticvue
- 5) Navigate to Kibana
- 6) Navigate to Spiderfoot

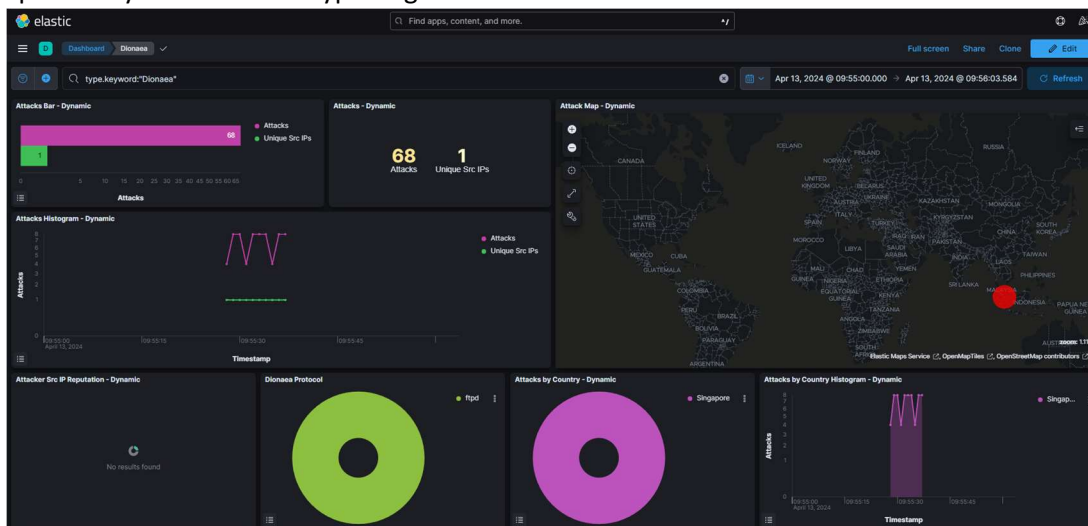
**Figure 1.2.6** Screenshot of T-Pot Dashboard Interface After Successful Installation (access the page via `<ip>:64297`)



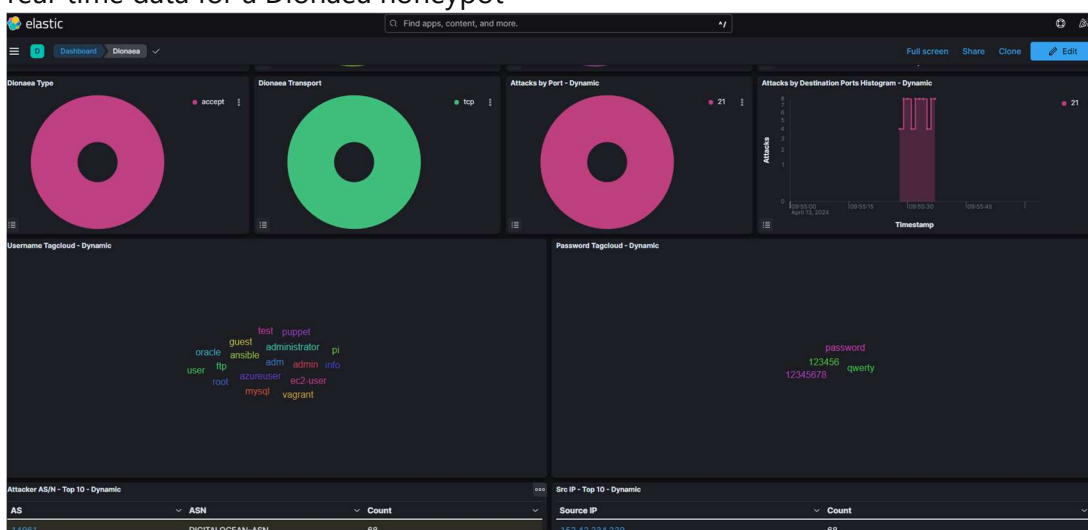
**Figure 1.2.7** Screenshot of Displaying the Real-time Attack Map on T-Pot Dashboard (access the attack map via `https://<ip>:64297/map/`)



**Figure 1.2.8** Screenshot of Kibana dashboard displaying a list of various dashboards, with one specifically noted for honeypot logs.



**Figure 1.2.9** Screenshot of a comprehensive Kibana security analytics dashboard displaying real-time data for a Dionaeea honeypot



**Figure 1.2.10:** Kibana dashboard for a Dionaeea honeypot, illustrating attack transport protocols, targeted ports, and common usernames and passwords. Displays top attacker IPs and ASNs.



Creating a new droplet (to run the attack script on the honey pot)

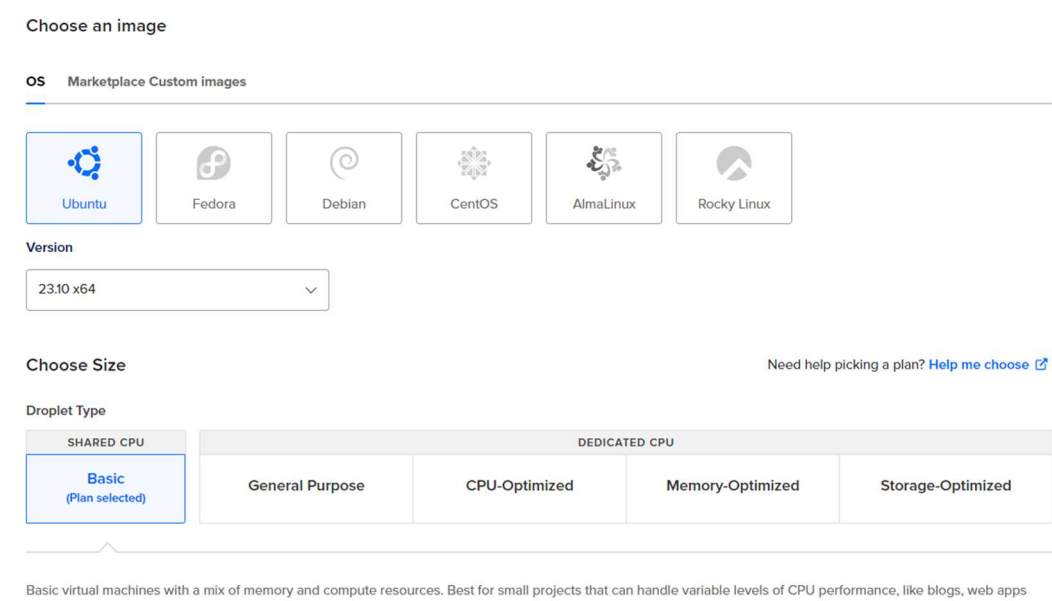


Figure 1.3.1 Screenshot of Choosing ubuntu as the OS image.

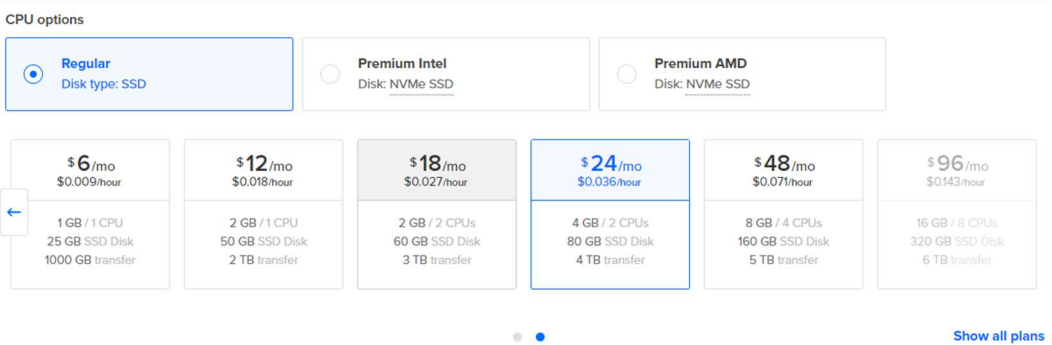


Figure 1.3.2 Screenshot of Choosing the number of CPU and ram (2cpu/4gb)

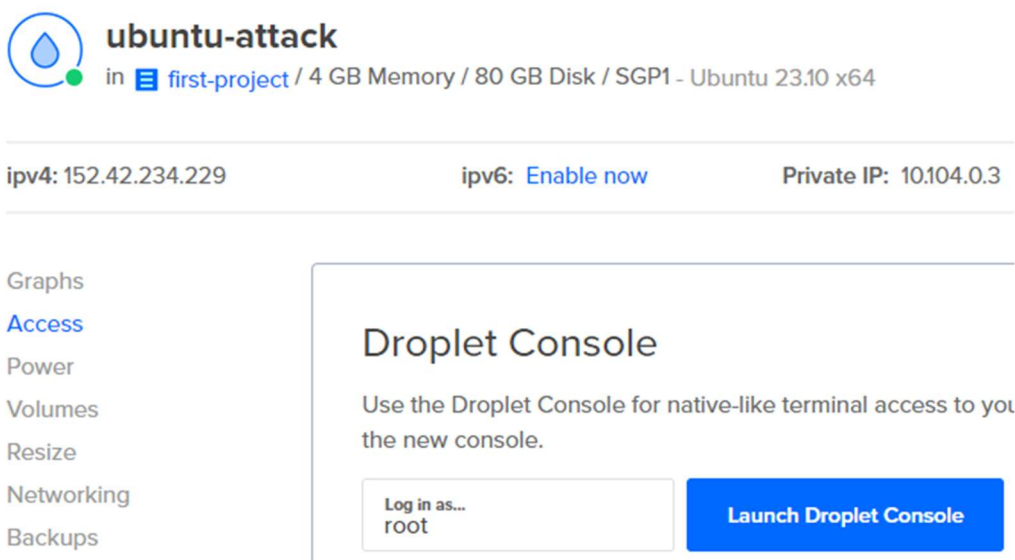


Figure 1.3.3 Screenshot of Droplet created

## Cyber Attack Simulations and Analysis

This section of the report documents the various cyber attack simulations performed using the T-Pot honeypot platform. Each simulation aims to test and validate the detection capabilities of our SOC setup. Detailed logs and analyses of each attack type are presented to demonstrate the SOC's responsiveness and to refine our defensive strategies.

### Brute Force Attack Simulation

```
Main Menu - Choose an option:
1) Run Brute Force
2) Denial of Service Attack
3) SQL Injection
4) Exit

Your choice: 1

Enter the IP Address to target (press Enter for random, or type 'exit' to quit): 152.42.226.94

Targeting user-specified IP: 152.42.226.94
Common services and their default ports for reference:
Service      | Port
-----+-----
ftp           | 21
ftps          | 21
ssh           | 22
telnet        | 23
smtp          | 25
pop3          | 110
imap          | 143
smb           | 445
smtps         | 465
mssql         | 1433
oracle        | 1521
mysql         | 3306
rdp           | 3389
postgres     | 5432
mongodb       | 27017
-----+-----

Enter the details for your custom service and port.
Enter service name (or press Enter for random, or type 'custom' to enter a custom service): ftp
Service ftp on port 21 selected.
Service ftp selected on Port 21

Choose the wordlist source:
1) Searching for wordlist from seclist
2) Please specify the custom wordlist paths.
Choice: 1
Do you wish to continue with the attack? (y/n): [ ]
```

Figure 2.1.1 Screenshot of running the brute force menu.

```
2024-04-13 01:55:38 - Target IP: 152.42.226.94:21//ftp Password found: azureuser/password
[21][ftp] host: 152.42.226.94 login: azureuser password: 12345678
2024-04-13 01:55:38 - Target IP: 152.42.226.94:21//ftp Password found: azureuser/12345678
[21][ftp] host: 152.42.226.94 login: azureuser password: qwerty
2024-04-13 01:55:38 - Target IP: 152.42.226.94:21//ftp Password found: azureuser/qwerty
1 of 1 target successfully completed, 68 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-13 01:55:38
Time taken: 10 seconds.
Hydra attack completed. Exiting script.
```

Figure 2.1.2 Screenshot of Output of a Successful Hydra Brute-Force Attack Indicating Compromised Credentials

Events	IP	Country	Honeypot	Service
2024-04-13 09:55:37	152.42.234.229	 Singapore	Dionaea	FTP

Figure 2.1.3 Screenshot of the attack map of T-pot showing that Dionaea honeypot received and logged the attack

```
[root@slightmice:/data/dionaea/log]# grep -iRn "152.42.234.229" | tail -n 1
grep: dionaea.sqlite: binary file matches
dionaea.json:72:{"connection": {"protocol": "ftpd", "transport": "tcp", "type": "accept"}, "dst_ip": "172.20.0.2", "dst_port": 21, "src_hostname": "", "src_ip": "152.42.234.229", "src_port": 38340, "timestamp": "2024-04-13T01:55:37.804196", "ftp": {"commands": {"arguments": ["azureuser", "qwerty"], "command": ["USER", "PASS"]}, "credentials": {"username": ["azureuser"], "password": ["qwerty"]}}
```

Figure 2.1.4 Screenshot of the Log of Dionaea saved at /data/dionaea/log/dionaea.sqlite on the honeypot machine.

```
root@ubuntu-attack:/var/log# cat security_assessment.log | tail -n 1
2024-04-13 01:55:38 - Attack executed: Bruteforce, Target IP: 152.42.226.94, Port: 21, Services: ftp, Details: 68 password(s) found for ftp
```

Figure 2.1.5 Screenshot of attack is logged at “/var/log/security\_assessment.log” of the attacker machine.

```

root@ubuntu-attack:/var/log# cat security_assessment_hydra.log | tail -n 5
2024-04-13 01:55:37 - Target IP: 152.42.226.94:21//ftp Password found: vagrant/qwerty
2024-04-13 01:55:38 - Target IP: 152.42.226.94:21//ftp Password found: azureuser/123456
2024-04-13 01:55:38 - Target IP: 152.42.226.94:21//ftp Password found: azureuser/password
2024-04-13 01:55:38 - Target IP: 152.42.226.94:21//ftp Password found: azureuser/12345678
2024-04-13 01:55:38 - Target IP: 152.42.226.94:21//ftp Password found: azureuser/qwerty

```

Figure 2.1.6 Screenshot of User/password found stored at '/var/log/security\_assessment\_hydra.log'

## Denial of Service (DoS) Attack Simulation

```

Main Menu - Choose an option:
1) Run Brute Force
2) Denial of Service Attack
3) SQL Injection
4) Exit

Your choice: 2

Enter the IP Address to target (press Enter for random, or type 'exit' to quit): 152.42.226.94

Targeting user-specified IP: 152.42.226.94
Select the attack simulation to run:
1. DNS Amplification
2. NTP Reconnaissance/Reflection
3. SSDP Amplification
Choice: 3

Running SSDP Amplification attack simulation...
HPING 152.42.226.94 (eth0 152.42.226.94): udp mode set, 28 headers + 87 data bytes
[main] memlockall(): No such file or directory
Warning: can't disable memory paging!
len=246 ip=152.42.226.94 ttl=61 DF id=7164 seq=0 rtt=7.4 ms

--- 152.42.226.94 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 7.4/7.4/7.4 ms

SSDP Amplification attack completed

```

Figure 2.2.1 Screenshot of DoS attack menu

Events	IP	Country	Honeypot	Service
2024-04-13 10:56:31	152.42.234.229	Singapore	Ddospot	1900

Figure 2.2.2 Screenshot of the attack map of T-pot showing that Ddospot honeypot received and logged the attack.

```

[root@slightmice:/data/ddospot/log]# grep -iRn "152.42.234.229" | tail -n 1
ssdpot.log:6:{"time": "2024-04-13 02:56:31.548752", "src_ip": "152.42.234.229", "src_port": 1527, "st": "ssdp:a
11", "mx": "1", "req_size": 87, "resp_size": 712, "req_pkt": "TS1TRUFSQ0ggKiBIVFRQLzEuMQpIT1NUOiAyMzkuMjU1LjI1N
S4yNTA6MTkwMApNQ46ICJzc2RwOmRpc2NvdmVyIgpNWDogMQpTVDogc3NkcDphbGwK"}

```

Figure 2.2.3 Screenshot of the Log of Ddospot saved at /data/ddospot/log/ssdpot.log on the honeypot machine.

```

root@ubuntu-attack:/var/log# cat security_assessment.log | tail -n 1
2024-04-13 02:56:31 - Attack executed: SSDP Amplification attack simulation completed on 152.42.226.94

```

Figure 2.2.4 Screenshot of attack is logged at "/var/log/security\_assessment.log" of the attacker machine.

## SQL Injection Attack Simulation

```
seclists directory /opt/SecLists already exists. Skipping installation.
ntpd command is available. Skipping installation.
Main Menu - Choose an option:
1) Run Brute Force
2) Denial of Service Attack
3) SQL Injection
4) Exit

Your choice: 3

Enter the IP Address to target (press Enter for random, or type 'exit' to quit): 152.42.226.94
```

Figure 2.3.1 Screenshot of SQL Injection attack menu

```
2024-04-13 03:56:58 - Resource Not Found: The path does not exist.
2024-04-13 03:56:58 - Performing Standard SQL Injection on http://152.42.226.94:/login.php with payload: '', Res
response: 404
2024-04-13 03:56:58 - Resource Not Found: The path does not exist.
2024-04-13 03:56:58 - Performing Data Extraction SQL Injection on http://152.42.226.94:/login.php with payload: '
'', Response: 404
2024-04-13 03:56:58 - Resource Not Found: The path does not exist.
2024-04-13 03:56:58 - Performing Command Execution SQL Injection on http://152.42.226.94:/login.php with payload:
'', Response: 404
2024-04-13 03:56:58 - Resource Not Found: The path does not exist.
2024-04-13 03:56:58 - Attack completed. Check the /var/log for details.
```

Figure 2.3.2 Screenshot of Output of the SQL Injection attack

Events	IP	Country	Honeypot	Service
2024-04-13 11:56:58	152.42.234.229	 Singapore	Tanner	HTTP

Figure 2.3.3 Screenshot of the attack map of the T-pot showing that Tanner honeypot received and logged the attack.

```
[root@slightmice:/data/tanner/log]# grep -iRn "152.42.234.229" | tail -n 1
tanner_report.json:24:{"method": "POST", "path": "/login.php", "headers": {"host": "152.42.226.94", "user-agent":
"curl/8.2.1", "accept": "*/*", "content-length": "9", "content-type": "application/x-www-form-urlencoded"}, "ui
d": "d1d0c071-4351-45f2-bef3-1243631b7aea", "peer": {"ip": "152.42.234.229", "port": 46106}, "status": 200, "post
_data": {"param": "", "cookies": {"sess_uid": null}, "response_msg": {"version": "0.6.0", "response": {"messag
e": {"detection": {"name": "unknown", "order": 0, "type": 1, "version": "0.6.0"}, "sess_uid": "f156205c-64e7-422
b-9cef-3a9103cdf8f6"}}, "timestamp": "2024-04-13T03:56:58.206026"}}
```

Figure 2.3.4 Screenshot of the Log of Tanner saved at

‘/data/tanner/log/ tanner\_report.json’ on the honeypot machine

```
root@ubuntu-attack:/var/log# cat security_assessment.log | tail -n 1
2024-04-13 03:56:58 - Attack executed: Sql Injection attack simulation completed on 152.42.226.94
```

Figure 2.3.5 Screenshot of the attack is logged at “/var/log/security\_assessment.log” of the attacker machine.

```
root@ubuntu-attack:/var/log# cat security_assessment_sql_injection.log | tail -n 5
2024-04-13 03:56:58 - Resource Not Found: The path does not exist.
2024-04-13 03:56:58 - Performing Data Extraction SQL Injection on http://152.42.226.94:/login.php with payload: '
'', Response: 404
2024-04-13 03:56:58 - Resource Not Found: The path does not exist.
2024-04-13 03:56:58 - Performing Command Execution SQL Injection on http://152.42.226.94:/login.php with payload:
'', Response: 404
2024-04-13 03:56:58 - Resource Not Found: The path does not exist.
```

Figure 2.3.6 Screenshot of SQL attack has been logged at

‘/var/log/security\_assessment\_sql\_injection.log’

## Credits and Acknowledgments

Project Title: SOC Project: Shadow Sentry

Student: David Lim

Student Code: s7

SOC Analyst Trainer: James

Class Code: CFC020823

### Project Acknowledgments:

**Self-Driven Initiative:** This project was independently researched, designed, and executed by me, David Lim. I appreciate the opportunity to apply theoretical knowledge in a practical, real-world setting and extend my gratitude to my educational institution for fostering an environment that encourages self-directed learning and innovation.

**DigitalOcean:** My sincere thanks to DigitalOcean for their reliable cloud hosting solutions. Their platform was essential for deploying and running the various technologies used in this project, such as the Elastic Stack and T-Pot honeypots.

**Deutsche Telekom's Security Team:** A special acknowledgment to Deutsche Telekom's Security Team for developing the T-Pot honeypot platform, a key component in our cybersecurity defense strategy. Their comprehensive solution allowed for effective simulation and analysis of potential security threats.

**Google:** I extend my thanks to Google for their powerful search engine and cloud services, which were instrumental in the research phase and implementation of this project. Google's resources provided valuable information and technical support, enabling efficient problem-solving and access to a wide array of data that supported my decisions and designs.

**Dedicated to:** This project is dedicated to all aspiring cybersecurity professionals. It serves as a testament to the fact that with determination, access to cutting-edge tools, and a supportive learning environment, significant educational and professional advancements are achievable.

### Conclusion:

The "Shadow Sentry" project not only enhanced my understanding of cybersecurity mechanisms but also highlighted the importance of continuous learning and application of new technologies in the field. This project has prepared me to further contribute to the field of cybersecurity, aiming to make digital environments safer for all users.