

# Alexander Chait



I'll start with DigitalOcean, I'll create a cloud based Ubuntu machine



24.04 (LTS) x64

I'll create firewall Rules that will only allow my own public IP address access to the machine using PuTTY SSH

## Inbound Rules

Set the Firewall rules for incoming traffic. Only the specified ports will accept inbound connections. All other traffic will be blocked.

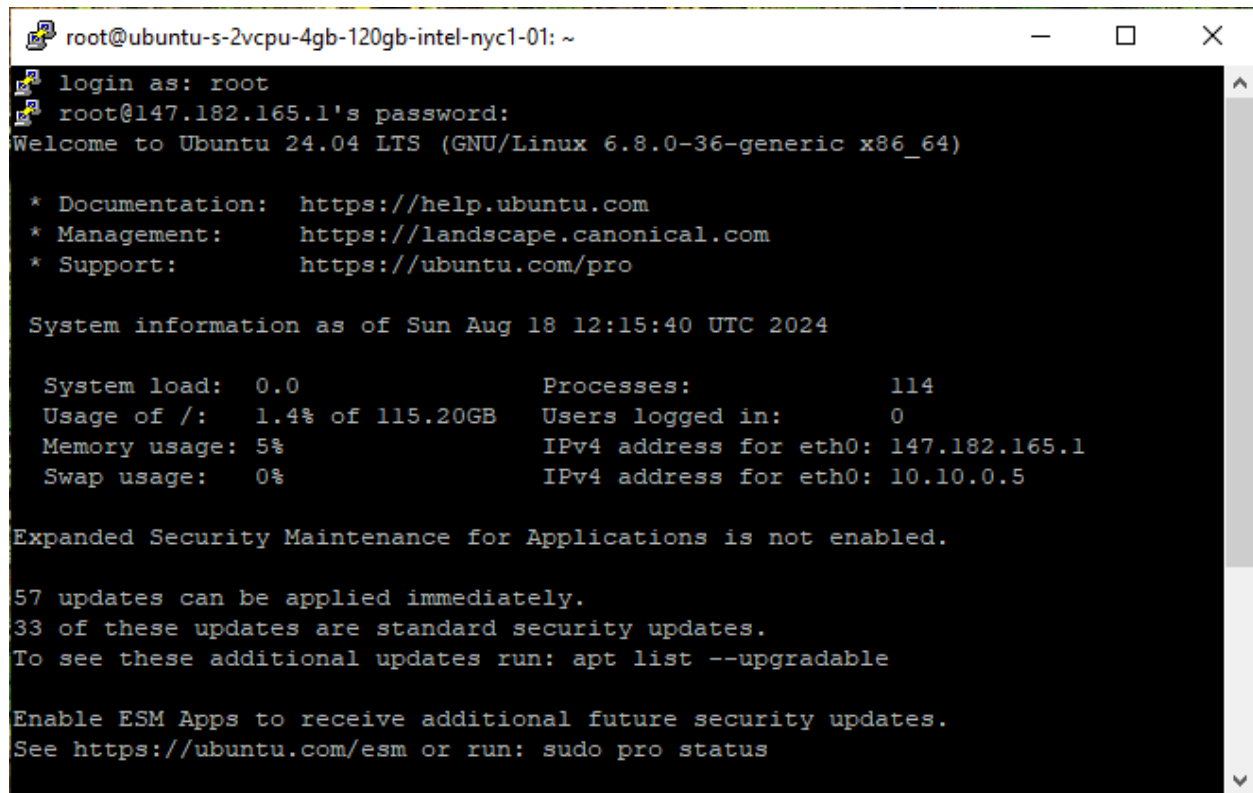
Type	Protocol	Port Range	Sources	
All TCP	TCP	All ports	<div></div>	<a href="#">More</a> <span>▼</span>
All UDP	UDP	All ports	<div></div>	<a href="#">More</a> <span>▼</span>
<div>New rule <span>▼</span></div>				

## Outbound Rules

Set the Firewall rules for outbound traffic. Outbound traffic will only be allowed to the specified ports. All other traffic will be blocked.

Type	Protocol	Port Range	Destinations	
ICMP	ICMP		<div>All IPv4 All IPv6</div>	<a href="#">More</a> <span>▼</span>
All TCP	TCP	All ports	<div>All IPv4 All IPv6</div>	<a href="#">More</a> <span>▼</span>
All UDP	UDP	All ports	<div>All IPv4 All IPv6</div>	<a href="#">More</a> <span>▼</span>

Now I'll Log in using PuTTY & I'll start installing T-Pot which is an all-in-one multiplatform honeypot solution

A terminal window titled 'root@ubuntu-s-2vcpu-4gb-120gb-intel-nyc1-01: ~' with standard window controls. The terminal shows a login sequence for 'root' at IP '147.182.165.1'. After the password is entered, it displays the Ubuntu 24.04 LTS welcome message and system information. The system info includes load, processes, memory usage, and network addresses. It also lists available updates and provides links for enabling ESM (Expanded Security Maintenance) for applications.

```
root@ubuntu-s-2vcpu-4gb-120gb-intel-nyc1-01: ~
login as: root
root@147.182.165.1's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Aug 18 12:15:40 UTC 2024

System load:  0.0               Processes:           114
Usage of /:   1.4% of 115.20GB   Users logged in:    0
Memory usage: 5%               IPv4 address for eth0: 147.182.165.1
Swap usage:   0%               IPv4 address for eth0: 10.10.0.5

Expanded Security Maintenance for Applications is not enabled.

57 updates can be applied immediately.
33 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

Installing T-Pot on Ubuntu 24.04 via SSH involves several steps. T-Pot is a multi-honeypot platform from Deutsche Telekom AG that includes various honeypots, IDS/IPS, and other security tools.

I'll be following these steps to install the T-Pot dependencies and prerequisites:

### **Step 1: Update and Upgrade the System**

```
sudo apt-get update && sudo apt-get upgrade -y
```

### **Step 2: Install Dependencies**

```
sudo apt-get install -y curl wget git
```

### **Step 3: Install Docker**

**Add Docker's official GPG key:**

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo  
gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

**Set up the stable Docker repository:**

```
echo "deb [arch=$(dpkg --print-architecture)  
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs)  
stable" | sudo tee /etc/apt/sources.list.d/docker.list >  
/dev/null
```

**Install Docker Engine:**

```
sudo apt-get update  
sudo apt-get install -y docker-ce docker-ce-cli containerd.io
```

**Start and enable Docker:**

```
bash
```

```
sudo systemctl start docker  
sudo systemctl enable docker
```

**Add your user to the Docker group:**

```
sudo usermod -aG docker $USER
```

**Log out and back in to apply the group change.**

**Download the Docker Compose binary:**

```
sudo curl -L  
"https://github.com/docker/compose/releases/download/v2.20.0/doc  
ker-compose-$(uname -s)-$(uname -m)" -o  
/usr/local/bin/docker-compose
```

**Apply executable permissions:**

```
sudo chmod +x /usr/local/bin/docker-compose
```

**Verify the installation:**

```
docker-compose --version
```

**Step 5: Download and Install T-Pot**

**Clone the T-Pot repository:**

```
git clone https://github.com/telekom-security/tpotce  
cd tpotce
```

**Run the installation script:**

```
sudo ./install.sh --type=user
```

^ That didn't work so I had to delete a machine and start a new one, the TL:DR way is easier 😊

## 🔗 TL;DR

1. Meet the [system requirements](#). The T-Pot installation needs at least 8-16 GB RAM, 128 GB free disk space as well as a working (outgoing non-filtered) internet connection.
2. [Download](#) or use a running, supported distribution.
3. Install the ISO with as minimal packages / services as possible ( `ssh` required)
4. Install `curl`: `$ sudo [apt, dnf, zypper] install curl` if not installed already
5. Run installer as non-root from `$HOME` :

```
env bash -c "$(curl -sL https://github.com/telekom-security/tpotce/raw/master/install.sh)"
```



Nice, I'll begin installing T-Pot

```
john@ubuntu-s-2vcpu-4gb-120gb-intel-nyc1-01: ~/tpotce
remote: Enumerating objects: 16162, done.
remote: Counting objects: 100% (174/174), done.
remote: Compressing objects: 100% (124/124), done.
remote: Total 16162 (delta 66), reused 139 (delta 49), pack-reused 15988 (from 1
)
Receiving objects: 100% (16162/16162), 278.91 MiB | 25.10 MiB/s, done.
Resolving deltas: 100% (8991/8991), done.
john@ubuntu-s-2vcpu-4gb-120gb-intel-nyc1-01:~/tpotce$
john@ubuntu-s-2vcpu-4gb-120gb-intel-nyc1-01:~/tpotce$
john@ubuntu-s-2vcpu-4gb-120gb-intel-nyc1-01:~/tpotce$ sudo ./install.sh --type=u
ser
This script should not be run as root. Please run it as a regular user.
john@ubuntu-s-2vcpu-4gb-120gb-intel-nyc1-01:~/tpotce$ ./install.sh --type=user
T-Pot Installation
### This script will now install T-Pot and all of its dependencies.
### Install? (y/n) y
```

After the installation, we need to re-connect with port 64295 using SSH

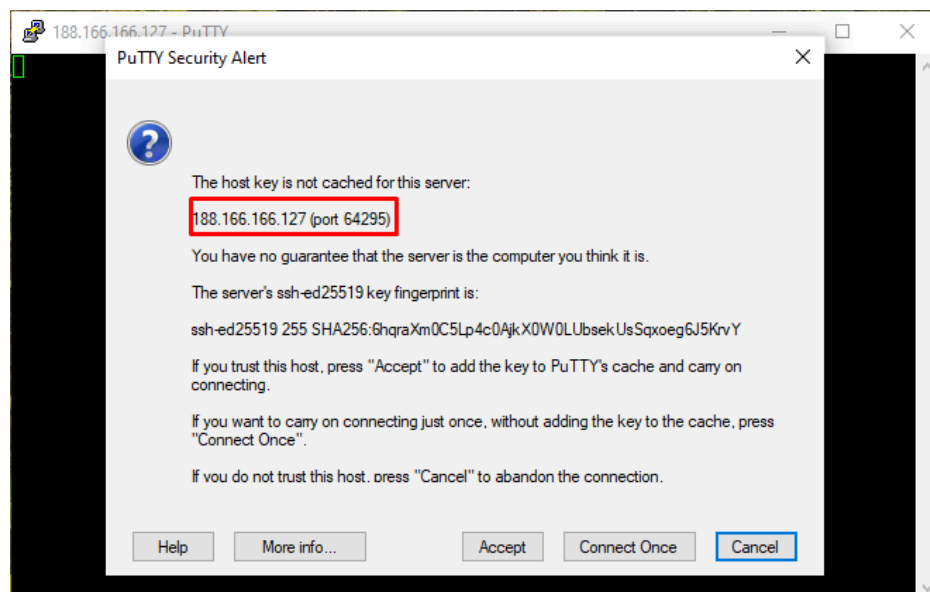
```
### Please review for possible honeypot port conflicts.
### While SSH is taken care of, other services such as
### SMTP, HTTP, etc. might prevent T-Pot from starting.

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
User      Inode      PID/Program name
tcp        0          0 0.0.0.0:64295           0.0.0.0:*               LISTEN
0          50773      18196/sshd: /usr/sb
tcp6       0          0 :::64295                :::*                     LISTEN
0          50775      18196/sshd: /usr/sb

### Done. Please reboot and re-connect via SSH on tcp/64295.

newusername@TopYenoh:~$
```

Connecting using PuTTY & Port 64295

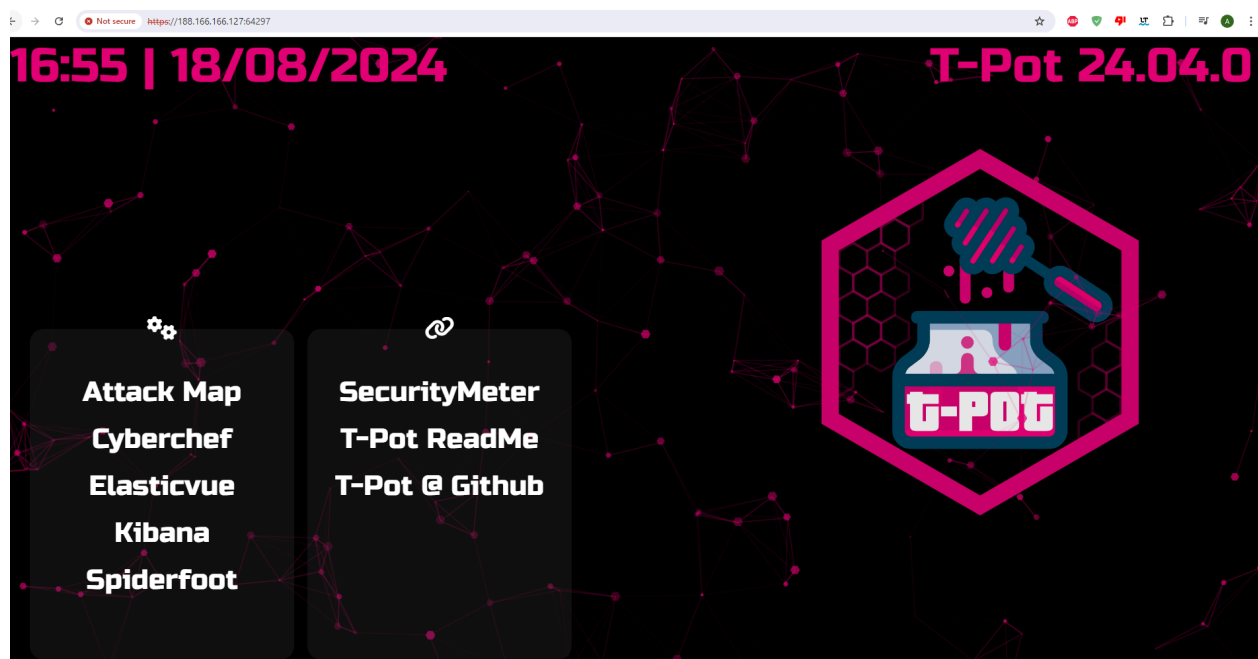


## Starting the T-Pot service after the restart

```
newusername@TopYenoh:~$ sudo systemctl status tpot.service
[sudo] password for newusername:
Sorry, try again.
[sudo] password for newusername:
○ tpot.service - T-Pot
   Loaded: loaded (/etc/systemd/system/tpot.service; enabled; preset: enabled)
   Active: inactive (dead)
newusername@TopYenoh:~$
newusername@TopYenoh:~$
newusername@TopYenoh:~$ sudo systemctl start tpot.service
newusername@TopYenoh:~$ sudo systemctl status tpot.service
● tpot.service - T-Pot
   Loaded: loaded (/etc/systemd/system/tpot.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-08-18 13:49:38 UTC; 9s ago
   Process: 19526 ExecStartPre=/usr/bin/docker compose -f /home/newusername/tp>
   Main PID: 19553 (docker)
   Tasks: 16 (limit: 9489)
   Memory: 27.7M (peak: 28.1M)
   CPU: 651ms
   CGroup: /system.slice/tpot.service
           └─19553 /usr/bin/docker compose -f /home/newusername/tpotce/docker>
             └─19566 /usr/libexec/docker/cli-plugins/docker-compose compose -f >

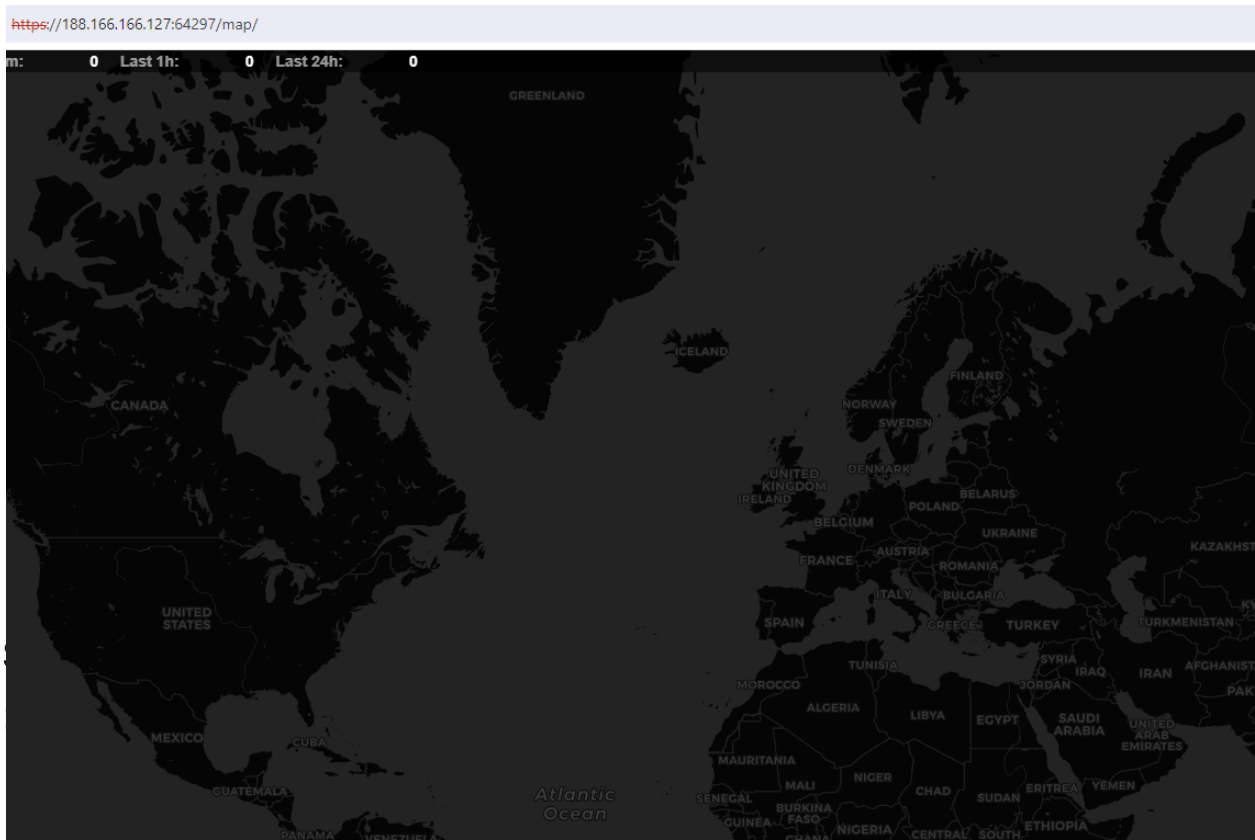
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
Aug 18 13:49:47 TopYenoh docker[19566]: tpotinit
```

Great! now I'm able to use the Web interface GUI





Right now the Attack map isn't showing any information, because the Firewall rules are blocking all incoming traffic

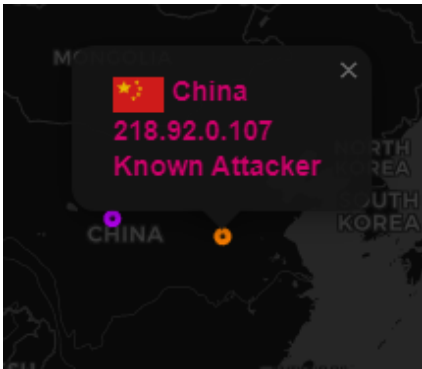
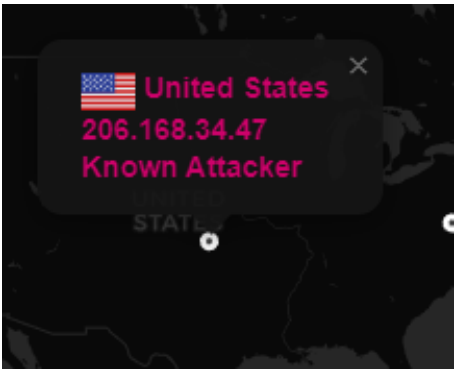
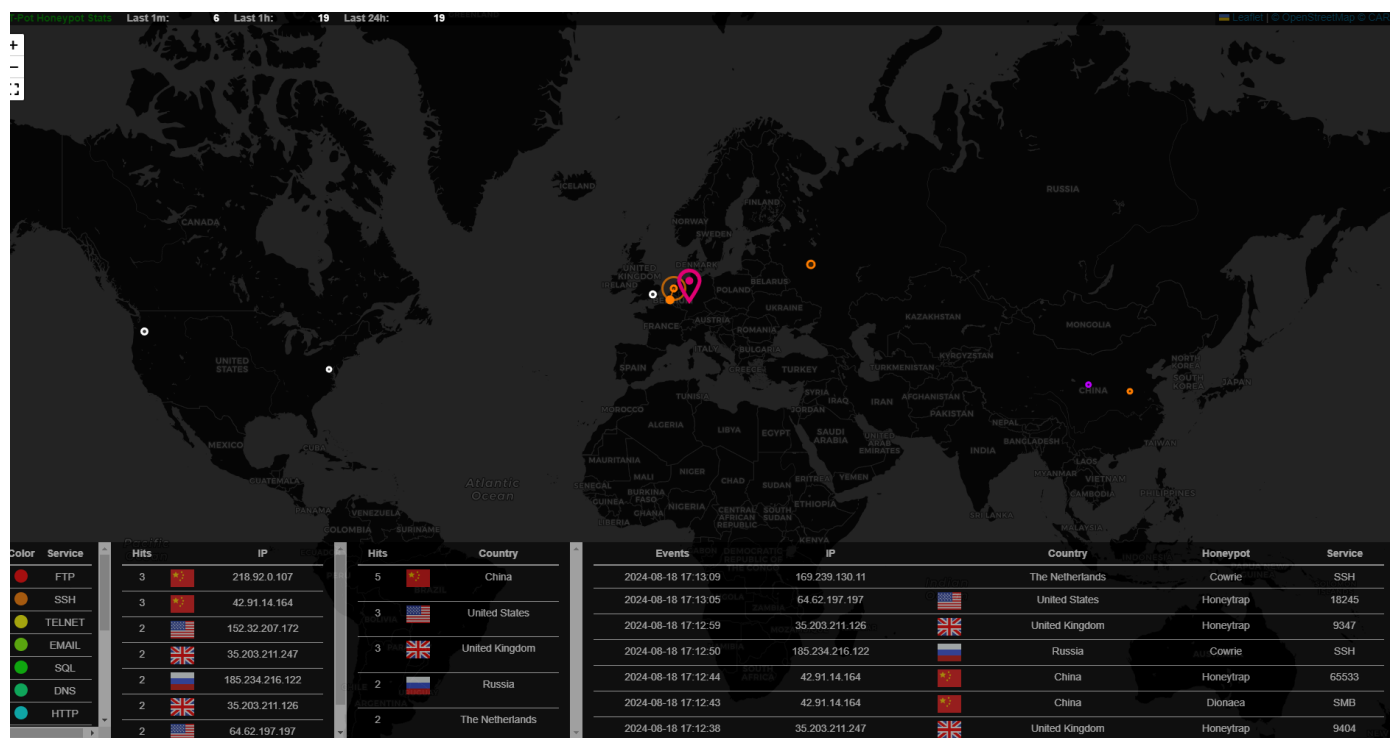


I'll create two new rules to make sure only my public IP address can access these specific ports, while removing the other rules to allow incoming traffic to the Honeypot

Type	Protocol	Port Range	Sources	
All TCP	TCP	All ports	All IPv4 All IPv6	More ▾
SSH	TCP	22		More ▾
Custom	TCP	64297		More ▾
All UDP	UDP	All ports	All IPv4 All IPv6	More ▾

After changing the firewall Rules to allow incoming traffic in I should be seeing attacks coming in any second now

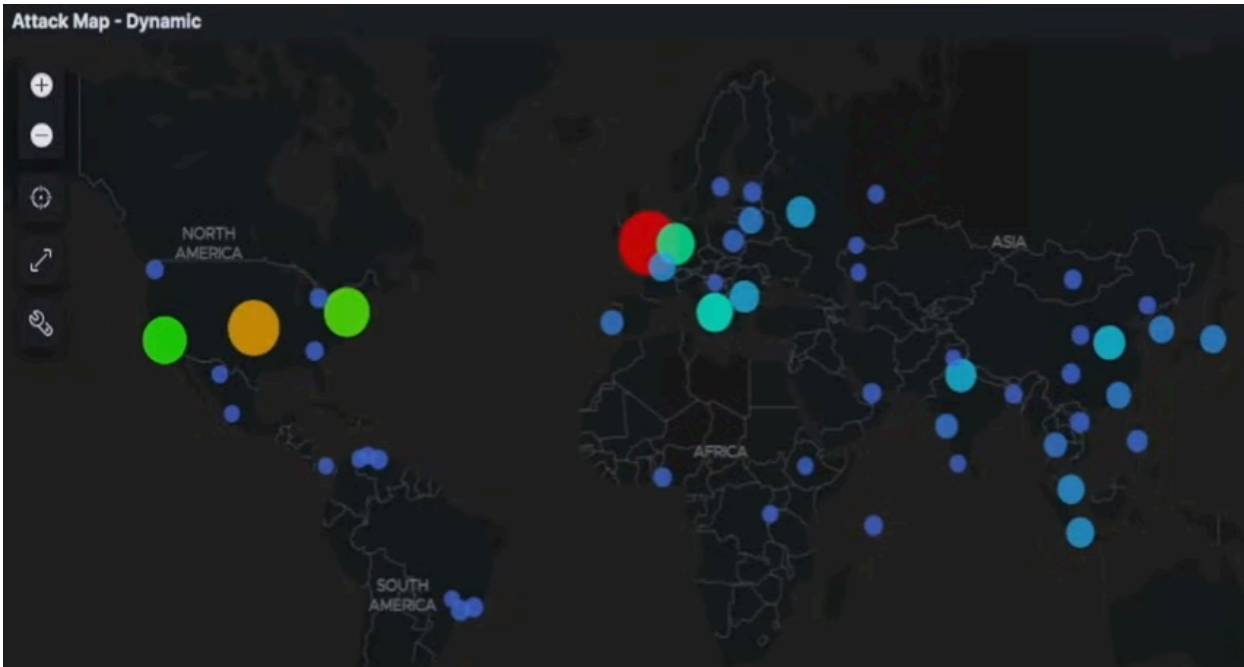
Yes, we can see attacks flowing for all over into our HoneyPot



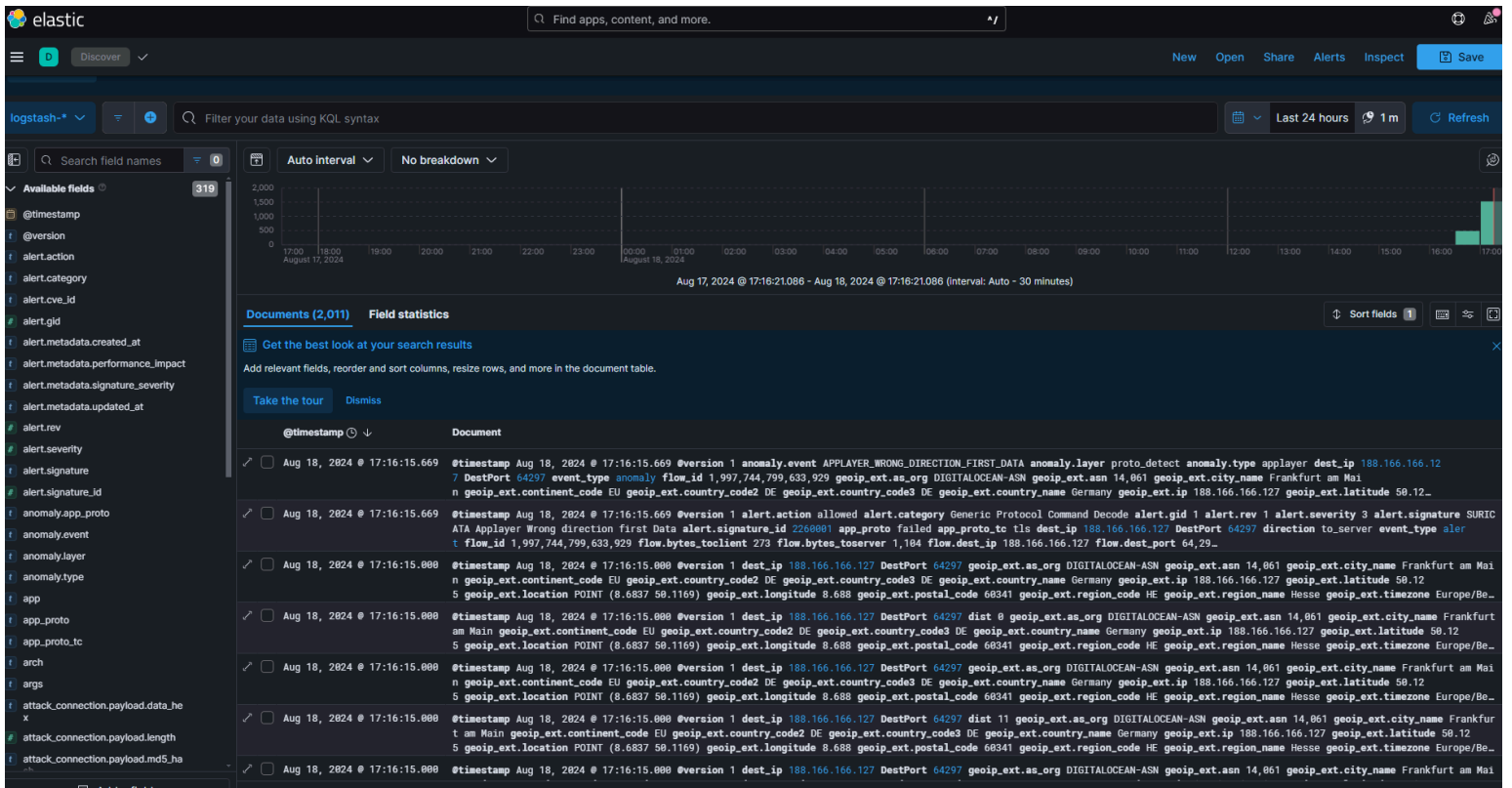
There have been 6 attacks in the last minute alone.



Here we can see a Dynamic map of the locations



I can also see Alert Queries starting to stack up using the elastic feature, which also allows for deep investigations of TTP's



## HoneyPot Project Summary: T-Pot Implementation

The goal of this project was to create an effective honeypot environment using T-Pot to attract and log attacker activities. A honeypot is a security mechanism that creates a decoy system to lure cybercriminals and analyze their behavior, helping organizations to understand and mitigate threats.

T-Pot is an advanced open-source platform that combines multiple honeypots with a comprehensive set of tools for analysis and reporting.

### Objectives

1. Set up a T-Pot environment to simulate multiple vulnerabilities and attract various types of cyber attacks.
2. Log and Analyze Attacker Actions: Capture and document attacker activities to study attack methods and patterns.
3. Improve Security Posture: Use insights gained from the honeypot to strengthen real systems and develop better defensive strategies.

This was a fun one!