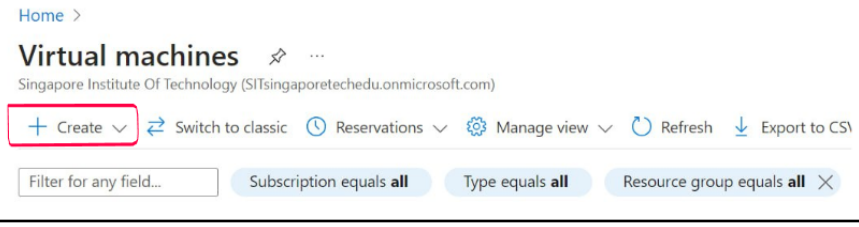
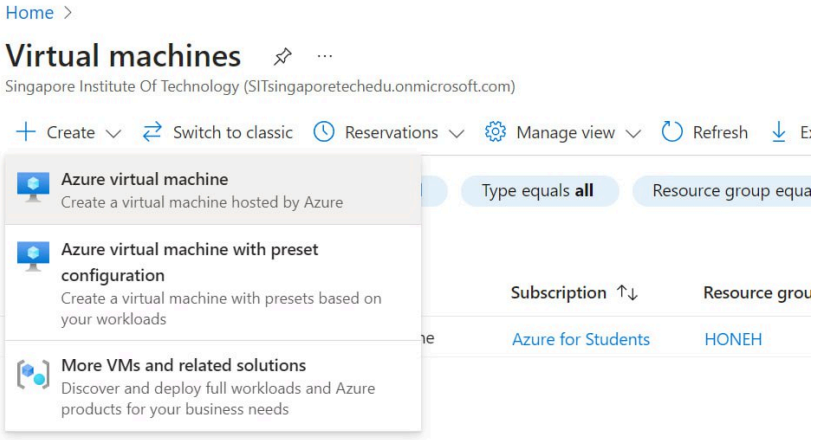
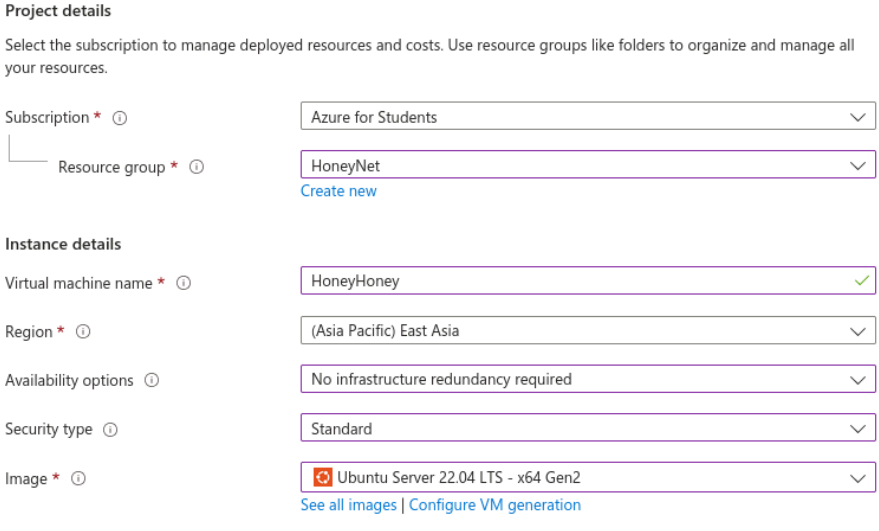
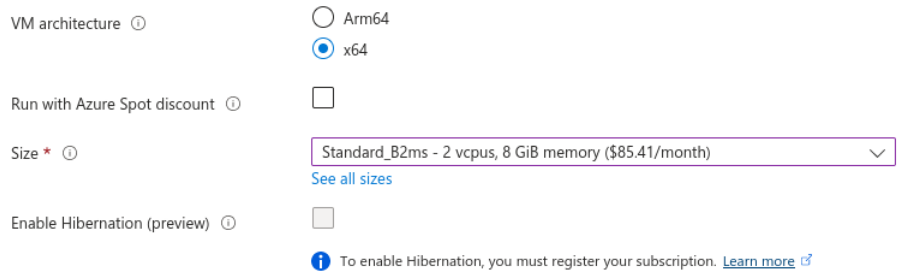
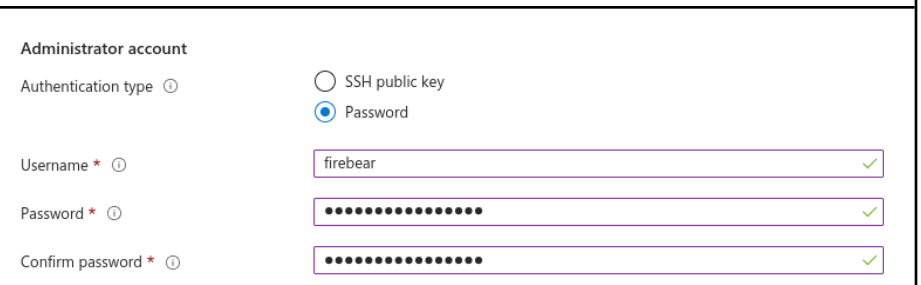
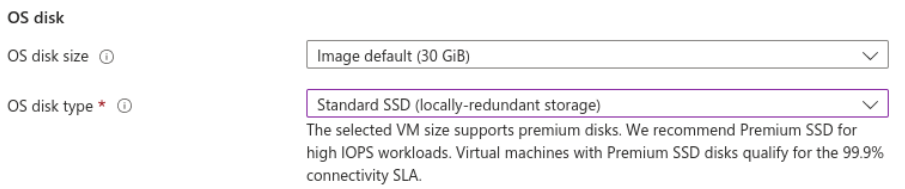
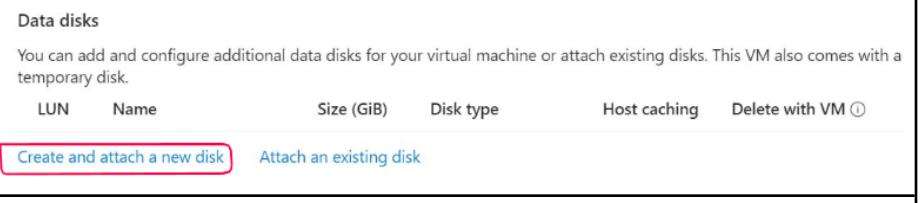
Microsoft Azure HoneyPot (T-CUP)

Creating a Virtual Machine

Head to the microsoft azure webpage to create a Virtual Machine
Link: https://azure.microsoft.com/en-us/

Follow the steps to create the virtual machines:

<table>
<tr><td rowspan="2">Click on create → Azure Virtual Machine</td><td> 
</td></tr>
<tr><td> 
</td></tr>
<tr><td colspan="2">On [Basic] tab</td></tr>
<tr><td>Select image as Ubuntu Server</td><td> 
</td></tr>
</table>

For the Size, base on what you need	
Select Password and create a username and password	
On [**Disk**] tab	
Select Standard SDD (Locally-Redundant Storage)	
Under DataDisk Click on “Create and attach a new disk” Select the size as “**128 GiB**”	 
Leave Everything as default and click on “Review + Create”	

Importing HoneyPots

These are the few HoneyPots that are Interesting

[Installing Dockers]
Link: https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04

[Main List of HoneyPots]
Link: https://github.com/paralax/awesome-honeypots

[Secondary List of HoneyPots]
Link: https://github.com/Correia-jpv/fucking-awesome-honeypots

[SSH HoneyPots]
Link: https://github.com/cowrie/cowrie

[ICS Honeypot to collect intelligence about motives]
Link: https://github.com/mushorg/conpot

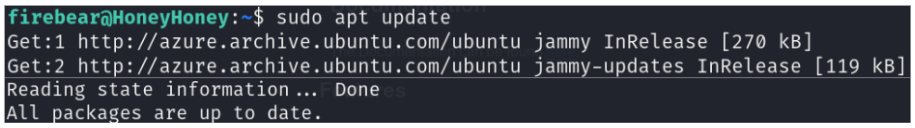
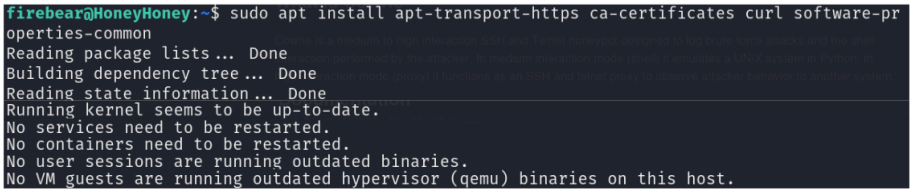
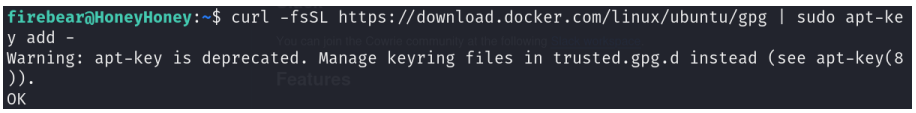
[HTTP Basic Authentication HoneyPot]
Link: https://github.com/bjeborn/basic-auth-pot

[Install VSFTPD]
Link: https://www.youtube.com/watch?v=XNjOSY-wcb0&t=316s

[FTP-HoneyPot]
Link: https://github.com/alexbredo/honeypot-ftp

Installing Dockers

Link: https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04

Update the list of Existing Packages **sudo apt update**	
Install a few pre-requisites packages **sudo apt install apt-transport-https ca-certificates curl software-properties-common**	
Adding GPG key for official Docker Repo **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -**	
Adding Docker Repo to APT sources **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"**	
Make sure you are about the install from docker repo instead of the default Ubuntu Repo **apt-cache policy docker-ce**	

Notice that the "**docker-ce**" is not installed, Install docker-ce

sudo apt install docker-ce

```
firebear@HoneyHoney:~$ sudo apt install docker-ce
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
```

→ Check the Status

sudo systemctl status docker

```
firebear@HoneyHoney:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-03-09 03:22:24 UTC; 51s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 4447 (dockerd)
```

Using Docker commands:

docker [option] [command] [arguments]

There are more output, not just few of these.

```
Output

  attach      Attach local standard input, output, and error streams to a
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesys
  create      Create a new container
  diff        Inspect changes to files or directories on a container's fil
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
```

To view Options available to specific command

docker docker-subcommand --help

```
firebear@HoneyHoney:~$ docker attach --help

Usage:  docker attach [OPTIONS] CONTAINER

Attach local standard input, output, and error streams to a running container

Aliases:
  docker container attach, docker attach

Options:
      --detach-keys string   Override the key sequence for detaching a container
      --no-stdin             Do not attach STDIN
      --sig-proxy            Proxy all received signals to the process (default true)
```

→ To view system-wide information about Docker

docker info

```
firebear@HoneyHoney:~$ docker info
Client: Docker Engine - Community
 Version:    25.0.4
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.13.0
    Path:     /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version:  v2.24.7
    Path:     /usr/libexec/docker/cli-plugins/docker-compose
```

To check if you are able to access and download images from docker hub

sudo docker run hello-world

```
firebear@HoneyHoney:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:d000bc569937abbe195e20322a0bde6b2922d805332fd6d8a68b19f524b7d21d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

The docker was initially unable to find the **hello-world** image locally therefore it downloaded the image from the Docker Hub

→ To search for images on DockerHub

sudo docker search ubuntu

```
firebear@HoneyHoney:~$ sudo docker search ubuntu
NAME                      DESCRIPTION                                     STARS     OFFICIAL
ubuntu                    Ubuntu is a Debian-based Linux operating sys…   16928     [OK]
ubuntu-debootstrap        DEPRECATED; use "ubuntu" instead                52        [OK]
open-liberty              Open Liberty multi-architecture images based…   64        [OK]
neurodebian               NeuroDebian provides neuroscience research s…   106       [OK]
websphere-liberty         WebSphere Liberty multi-architecture images …   298       [OK]
ubuntu-upstart            DEPRECATED, as is Upstart (find other proces…   115       [OK]
ubuntu/nginx              Nginx, a high-performance reverse proxy & we…   112
ubuntu/squid              Squid is a caching proxy for the Web. Long-t…   84
ubuntu/cortex             Cortex provides storage for Prometheus. Long…   4
ubuntu/prometheus         Prometheus is a systems and service monitori…   58
```

Execute the command to download images to your computer

sudo docker pull ubuntu

```
firebear@HoneyHoney:~$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

→ To see images that you have downloaded

sudo docker images

```
firebear@HoneyHoney:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
ubuntu        latest    ca2b0f26964c   10 days ago     77.9MB
hello-world   latest    d2c94e258dcb   10 months ago   13.3kB
```

→ Running Docker Container

-i -t switches gives you interactive shell access into the container

-d to keep running on background

sudo docker run -it ubuntu
sudo docker run -dit ubuntu

```
firebear@HoneyHoney:~$ sudo docker run -it ubuntu
root@a2fa55caa3ea:/#
```

The container ID in the command prompt "**a2fa55caa3ea**", You will need this container ID to identify the container if you want to remove it. Simply type "**Exit**" if you want to get out from the container

→ Managing Docker Containers

sudo docker ps

```
firebear@HoneyHoney:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

To view all containers

sudo docker ps -a

```
firebear@HoneyHoney:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND       CREATED          STATUS
a2fa55caa3ea   ubuntu        "/bin/bash"   5 minutes ago    Exited (127) About a minute ago
52fbfc6e2940   hello-world   "/hello"      16 minutes ago   Exited (0) 16 minutes ago
```

To START STOP a container

sudo docker start <container ID>
sudo docker stop <container ID>

```
firebear@HoneyHoney:~$ sudo docker start a2fa55caa3ea
a2fa55caa3ea
firebear@HoneyHoney:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS     NAMES
a2fa55caa3ea   ubuntu    "/bin/bash"   8 minutes ago   Up 3 seconds             loving_leakey
```

```
firebear@HoneyHoney:~$ sudo docker stop a2fa55caa3ea
a2fa55caa3ea
firebear@HoneyHoney:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

To REMOVE a container

sudo docker rm <container ID>

```
firebear@HoneyHoney:~$ sudo docker rm 52fbfc6e2940
52fbfc6e2940
firebear@HoneyHoney:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS         PORTS     NAMES
a2fa55caa3ea   ubuntu    "/bin/bash"   12 minutes ago   Up 4 minutes             loving_leakey
```

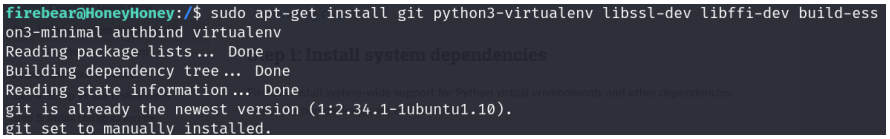
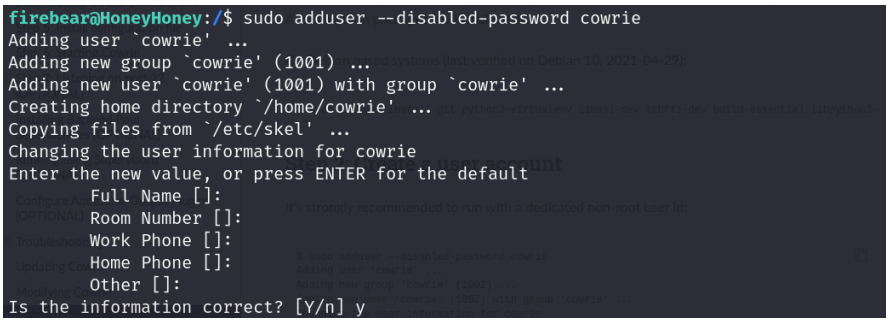
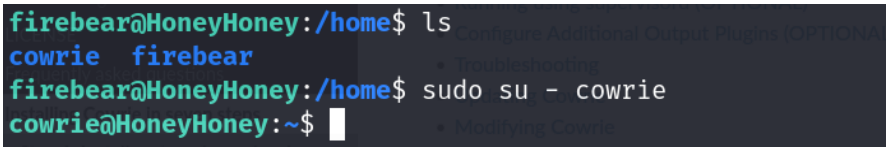
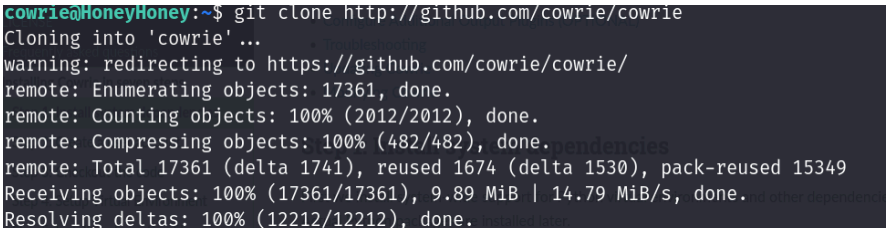
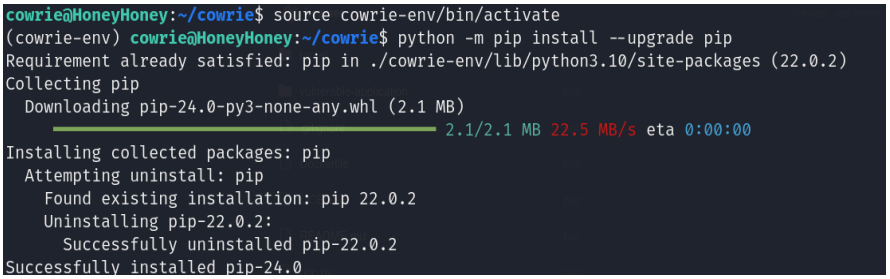
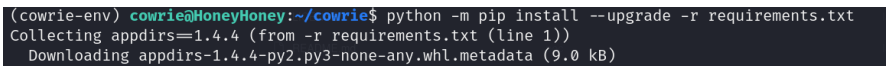
Installing Cowrie

[SSH HoneyPots]

Link: https://github.com/cowrie/cowrie

[Documentation]

Link: https://cowrie.readthedocs.io/en/latest/index.html

Install Dependencies **sudo apt-get install git python3-virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv**	
Create User Accounts **sudo adduser --disabled-password cowrie**	
Using superuser - cowrie **sudo su - cowrie**	
Pull the cowrie from github **git clone http://github.com/cowrie/cowrie** **cd cowrie**	
Set up the environment pwd python -m venv cowrie-env	
Activate virtual environment **source cowrie-env/bin/activate** **python -m pip install --upgrade pip** **python -m pip install --upgrade -r requirements.txt**	  

Install configuration file, change the directory to **"/etc"** and create a **cowrie.cfg** file

[telnet]
enable = true

```
(cowrie-env) cowrie@HoneyHoney:~/cowrie$ cd etc
(cowrie-env) cowrie@HoneyHoney:~/cowrie/etc$ nano cowrie.cfg
```

```
  GNU nano 6.2                                    cowrie.cfg *
[telnet]
enable = true
```

→ Start cowrie

bin/cowrie start

```
(cowrie-env) cowrie@HoneyHoney:~/cowrie$ bin/cowrie start

Join the Cowrie community at: https://www.cowrie.org/slack/

Using activated Python virtual environment "/home/cowrie/cowrie/cowrie-env"
Starting cowrie: [twistd  --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile
.logger cowrie ]...
/home/cowrie/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:106: Cr
yptographyDeprecationWarning: Blowfish has been deprecated and will be removed in a future release
  b"blowfish-cbc": (algorithms.Blowfish, 16, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:110: Cr
yptographyDeprecationWarning: CAST5 has been deprecated and will be removed in a future release
  b"cast128-cbc": (algorithms.CAST5, 16, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:115: Cr
yptographyDeprecationWarning: Blowfish has been deprecated and will be removed in a future release
  b"blowfish-ctr": (algorithms.Blowfish, 16, modes.CTR),
```

[Attackers POV]

Once started, on the other VM normally attackers use Nmap to scan for open ports, but let's not waste time and straight away scan port 2222

sudo nmap -p 2222 -sV 23.99.107.240

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p 2222 -sV 23.99.107.240
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-09 00:42 E
ST
Nmap scan report for 23.99.107.240
Host is up (0.0062s latency).

PORT     STATE SERVICE VERSION
2222/tcp open  ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2
.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Using SSH on port 2222, any users will do, doesn't matter

ssh -p 2222 root@23.99.107.240

```
┌──(kali㉿kali)-[~]
└─$ ssh -p 2222 root@23.99.107.240
The authenticity of host '[23.99.107.240]:2222 ([23.99.107.240]:
2222)' can't be established.
ED25519 key fingerprint is SHA256:Z7rPXD2h4V9E68Bu3fleJLEMPXHIij
Se27GD/f3BrxU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprin
t])? yes
```

Issue any commands on the **CMD**

```
root@svr04:~# ls
root@svr04:~#
root@svr04:~# cd ..
root@svr04:/# ls
bin         boot        dev         etc         home
initrd.img  lib         lost+found  media       mnt
opt         proc        root        run         sbin
selinux     srv         sys         test2       tmp
usr         var         vmlinuz
root@svr04:/# cd opt
```

[Cowries POV]

To view the **cowrie.log file** change the directory to

cd var/log/cowrie

```
(cowrie-env) cowrie@HoneyHoney:~/cowrie$ cd var/log/cowrie/
(cowrie-env) cowrie@HoneyHoney:~/cowrie/var/log/cowrie$
```

View the log file

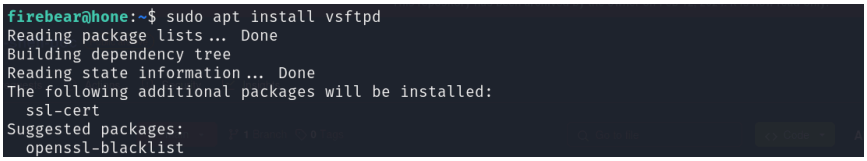
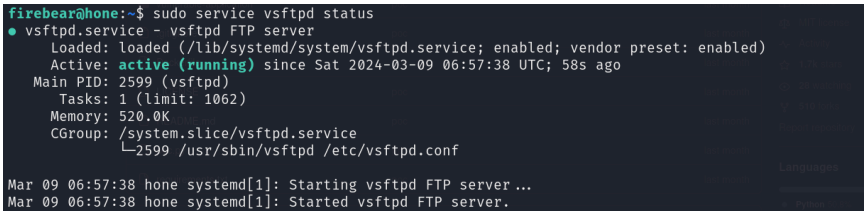
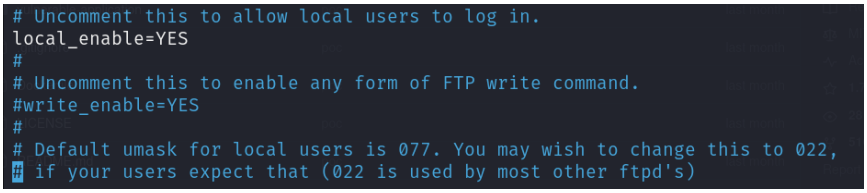
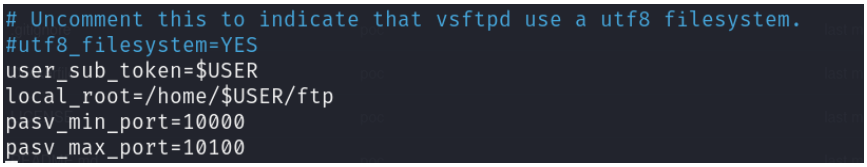
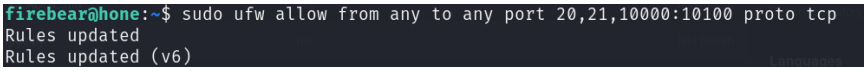
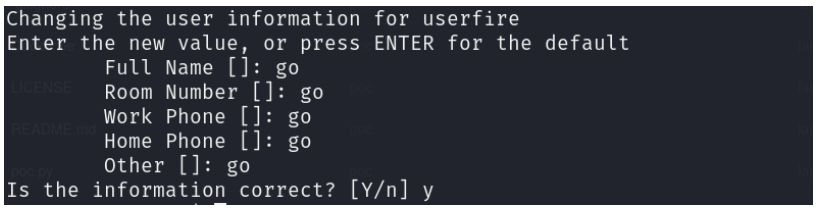
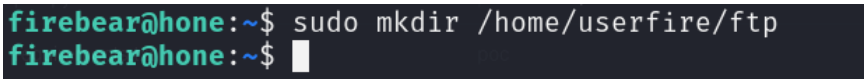
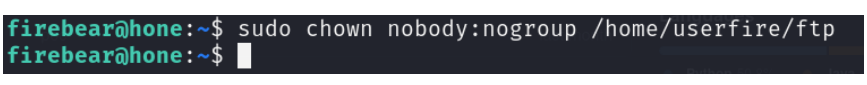
cat cowrie.log

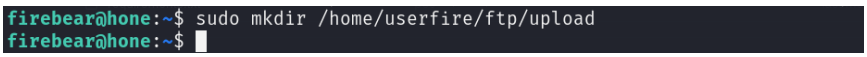
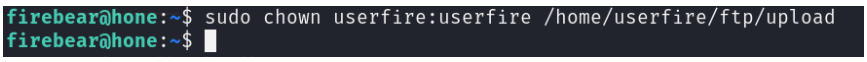
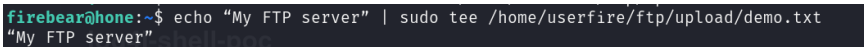
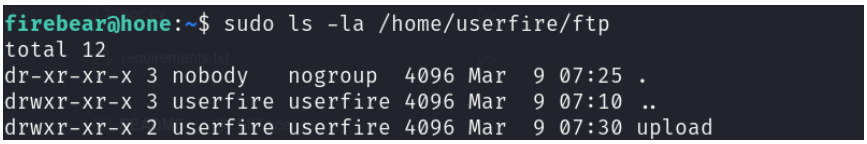
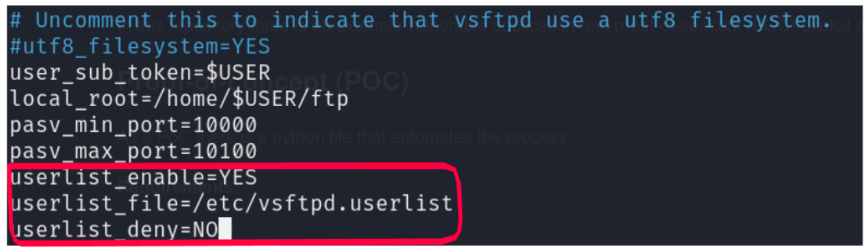
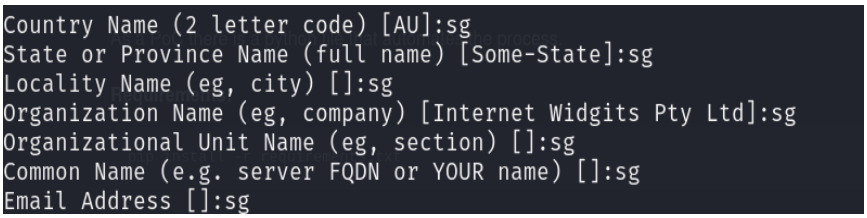
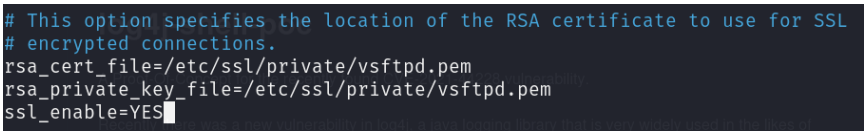
```
2024-03-09T06:24:36.832056Z [HoneyPotSSHTransport,3,116.14.113.125] Command found: cd ..
2024-03-09T06:24:37.651107Z [HoneyPotSSHTransport,3,116.14.113.125] CMD: ls
2024-03-09T06:24:37.651901Z [HoneyPotSSHTransport,3,116.14.113.125] Command found: ls
2024-03-09T06:24:40.044623Z [HoneyPotSSHTransport,3,116.14.113.125] CMD: cd opt
2024-03-09T06:24:40.045337Z [HoneyPotSSHTransport,3,116.14.113.125] Command found: cd opt
2024-03-09T06:24:40.760844Z [HoneyPotSSHTransport,3,116.14.113.125] CMD: ls
```

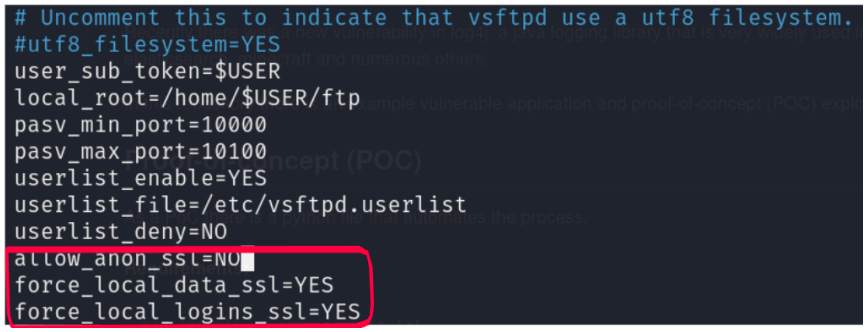
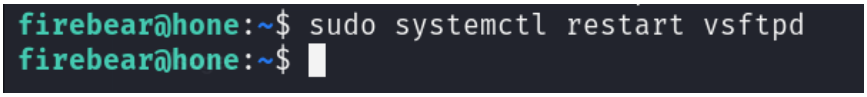
Installing VSFTPD

[Install VSFTPD]

Link: https://www.youtube.com/watch?v=XNjOSY-wcb0&t=316s

Install VSFTPD **sudo apt install vsftpd**	
Check VSFTPD STATUS **sudo service vsftpd status**	
Change the settings of the vsftpd.conf file **sudo nano /etc/vsftpd.conf** **local_enable=YES** **write_enable=YES** **chroot_local_user=YES**	
Add the command at the bottom, and save the file **user_sub_token=$USER** **local_root=/home/$USER/ftp** **pasv_min_port=10000** **pasv_max_port=10100**	
Allow TCP traffic on ports 20,21 and range 10000 to 10100 from any source IP address to any destination IP address. **sudo ufw allow from any to any port 20,21,10000:10100 proto tcp**	
Add new username and password **sudo adduser <username>**	 
Create a new directory named "**FTP**" **sudo mkdir /home/userfire/ftp**	
Change the ownership to nobody user, nogroup group **sudo chown nobody:nogroup /home/userfire/ftp**	

Removes the write permission for all user **sudo chmod a-w /home/userfire/ftp**	
Creates directory named “upload” directory **sudo mkdir /home/userfire/ftp/upload**	
Change the ownership of the directory to userfire **sudo chown userfire:userfire /home/userfire/ftp/upload**	
Write the text to a file name **echo “My FTP server” | sudo tee /home/userfire/ftp/upload/demo.txt**	
List all files and directories in the /home/userfire/ftp **sudo ls -la /home/userfire/ftp**	
Appends the string “userfire” to the file **echo “userfire” | sudo tee -a /etc/vsftpd.userlist**	
→ Restart VSFTPD **sudo systemctl restart vsftpd**	
Edit the **vsftpd.conf** **sudo nano /etc/vsftpd.conf** Add in these codes, and save the configuration file **userlist_enable=YES** **userlist_file=/etc/vsftpd.userlist** **userlist_deny=NO**	
Generate self signed certificates **sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem**	 
Edit the **vsftpd.conf** **sudo nano /etc/vsftpd.conf** Add these lines **rsa_cert_file=/etc/ssl/private/vsftpd.pem** **rsa_private_key_file=/etc/ssl/private/vsftpd.pem** **ssl_enable=YES**	

Add in couple more lines, and save the **config file** **allow_anon_ssl=NO** **force_local_data_ssl=YES** **force_local_logins_ssl=YES**	
Restart the VSFTPD **sudo systemctl restart vsftpd**	

Installing ConPot

[ICS Honeypot to collect intelligence about motives]

Link: https://github.com/mushorg/conpot

Link: https://conpot.readthedocs.io/en/latest/installation/quick_install.html

Pull the Clone ConPot from github, and change directory to conpot

git clone https://github.com/mushorg/conpot.git

cd conpot

```
firebear@HoneyHoney:~$ git clone https://github.com/mushorg/conpot.git
Cloning into 'conpot' ...
remote: Enumerating objects: 9004, done.
remote: Counting objects: 100% (552/552), done.
remote: Compressing objects: 100% (253/253), done.
remote: Total 9004 (delta 307), reused 491 (delta 297), pack-reused 8452
Receiving objects: 100% (9004/9004), 3.03 MiB | 11.88 MiB/s, done.
Resolving deltas: 100% (6076/6076), done.
```

Run docker and build conpot

sudo docker build -t conpot .

```
firebear@HoneyHoney:~/conpot$ sudo docker build -t conpot .
[+] Building 2.4s (1/3)                                          docker:default
 ⇒ [internal] load build definition from Dockerfile                        0.1s
 ⇒ ⇒ transferring dockerfile: 1.02kB                                       0.0s
 ⇒ [internal] load metadata for docker.io/library/python:3.8-slim          2.3s
 ⇒ [internal] load metadata for docker.io/library/python:3.8               2.3s
```

→ Start conpot

sudo docker run -dit -p 8443:8800 -p 102:10201 -p 502:5020 -p 161:16100/udp -p 47808:47808/udp -p 623:6230/udp -p 2121:2121 -p 69:6969/udp -p 44818:44818 --network=bridge conpot > docker_run_log.txt 2>&1

sudo docker logs 032d0c9eb48c06861012e1ca7f5508c1380d8cfbc96c6993d5891a1c26f490ea

```
firebear@HoneyHoney:~/conpot$ sudo docker run -it -p 8443:8800 -p 102:10201 -p 502:5020 -p
161:16100/udp -p 47808:47808/udp -p 623:6230/udp -p 2121:2121 -p 69:6969/udp -p 44818:4481
 --network=bridge conpot
                    _
 ___ ___ ___ ___ ___|_|_
|  _| . |   | . | . | | |
|___|___|_|_|  _|___|_|
            |_|

Version 0.6.0
MushMush Foundation
```

[Attackers POV]

Run the command nmap

sudo nmap -sS 23.99.107.240

OR

sudo nmap -Pn 23.99.107.240

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sS 23.99.107.240
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-09 06
:57 EST
Nmap scan report for 23.99.107.240
Host is up (0.038s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
2121/tcp open  ccproxy-ftp
2222/tcp open  EtherNetIP-1
8443/tcp open  https-alt
```

[ConPot POV]

When attacker doing a nmap scan, if you are not able to see, check the **ufw** settings

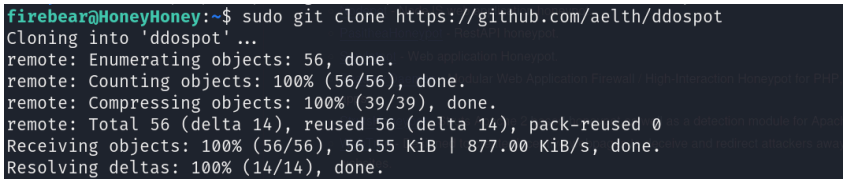
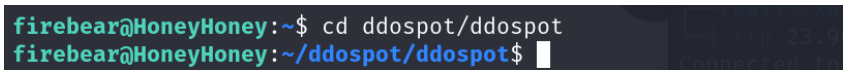
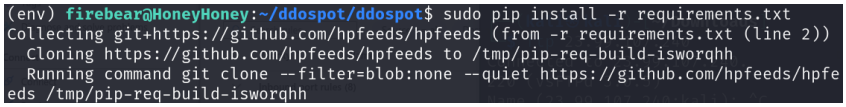
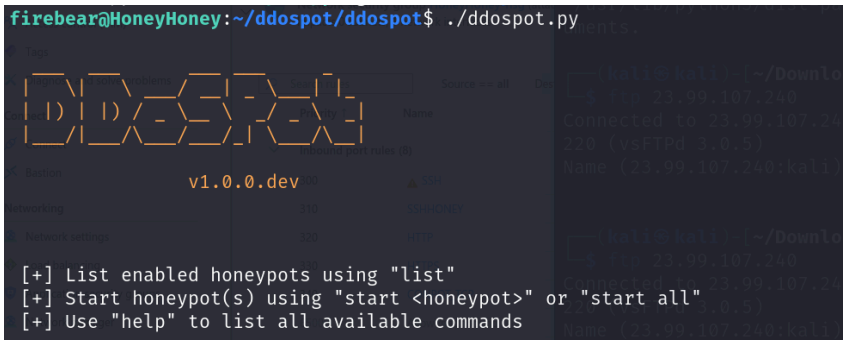
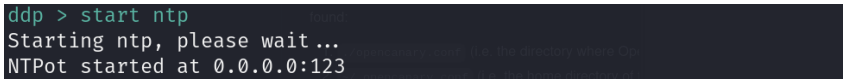
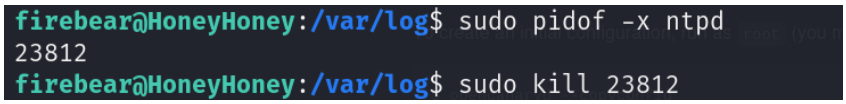
sudo ufw allow <port>/tcp
sudo ufw allow <port>/udp

```
2024-03-09 11:57:46,398 FTP server started on: ('0.0.0.0', 2121)
2024-03-09 11:57:46,398 HTTP server started on: ('0.0.0.0', 8800)
2024-03-09 11:57:46,399 IPMI server started on: ('0.0.0.0', 6230)
2024-03-09 11:57:46,399 Modbus server started on: ('0.0.0.0', 5020)
2024-03-09 11:57:46,399 S7Comm server started on: ('0.0.0.0', 10201)
2024-03-09 11:57:46,738 SNMP server started on: ('0.0.0.0', 16100)
2024-03-09 11:57:46,739 Starting TFTP server at ('0.0.0.0', 6969)
b174f0)
2024-03-09 11:57:57,490 New FTP connection from 116.14.113.125:25286. (0c6f26e2-d4f5-483f-a
ecb-f291ddb174f0)
2024-03-09 11:57:57,490 FTP traffic to ('116.14.113.125', 25286): {'response': b'200 FTP se
rver ready.\r\n'} (0c6f26e2-d4f5-483f-aecb-f291ddb174f0)
```

Installing DDoSPot

[Tracking and Monitoring UDP-Based DDoS]

Link: https://github.com/aelth/ddospot

Installation using Docker image, clone the repo from github, and change the directory to "**ddospot/ddospot**" **sudo git clone https://github.com/aelth/ddospot** **cd ddospot/**	 
Change to the python3 environment **python3 -m venv /home/firebear/ddospot/venv** **source /home/firebear/ddospot/venv/bin/activate** Install the requirement text **sudo pip install -r requirements.txt**	 
Allow permission to overwrite the folder **sudo chmod -R 777 /home/firebear/ddospot/ddospot/logs**	
Configuration file is well commented and has several sections. - General section that specifies the listening interface and port of the plugins	
→ Start DDoSPot **sudo ./ddospot.py**	
Type in the command **start ntp**	
If can't start, identify the ntpd pid and kill the PID **sudo pidof -x ntpd** **sudo kill <PID>**	

→ To see the status of the NTP

status ntp

```
ddp > status ntp

ntp status:

    ____________________________  -
    Number of IPs                 0
    Number of attacks             0
    Total num. of packets recv.   0
    First attack                  -
    Latest attack                 -
    Average attack duration       -
    Longest continuous attack     -
    Largest continuous attack     -
    Top target (by pkt. count)    -
    ____________________________  -
```

To solve for DNS

sudo systemctl stop systemd-resolved
sudo systemctl disable systemd-resolved

```
firebear@HoneyHoney:~/ddospot/ddospot$ sudo systemctl stop systemd-resolved
sudo systemctl disable systemd-resolved
sudo: unable to resolve host HoneyHoney: Temporary failure in name resolution
Removed /etc/systemd/system/dbus-org.freedesktop.resolve1.service.
Removed /etc/systemd/system/multi-user.target.wants/systemd-resolved.service.
```

→ Start DDoSPot DNS

start dns

```
ddp > start dns
Starting dns, please wait...
DNSPot started at 0.0.0.0:53
```

[Attackers POV]

Flood the DNS of the webpage

sudo hping3 -S --flood -V -p 53 29.99.107.240

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo hping3 -S --flood -V -p 53 29.99.107.240
using eth0, addr: 192.168.1.1, MTU: 1500
HPING 29.99.107.240 (eth0 29.99.107.240): S set, 40 headers
```

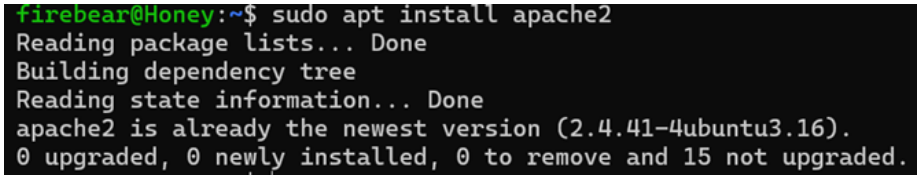
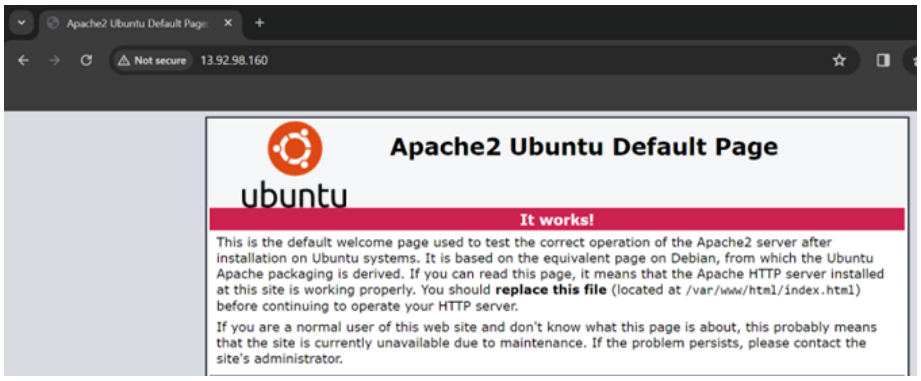
[DDoSPot POV]

To check the status of the DNS

status dns

```
ddp > status dns

dns status:

    ____________________________  ____________________________________________
    Number of IPs                 1
    Number of attacks             36
    Total num. of packets recv.   191
    First attack                  127.0.0.1 @ 2024-03-10 16:32:10.718762
    Latest attack                 127.0.0.1 @ 2024-03-10 16:59:53.587139
    Average attack duration       1 minute, 7 seconds
    Longest continuous attack     127.0.0.1, 3 minutes, 35 seconds (2024-
16 (0.07 pps)
    Largest continuous attack     127.0.0.1, 3 minutes, 35 seconds (2024-
16 (0.07 pps)
    Top target (by pkt. count)    127.0.0.1 (2024-03-10 16:32:10.718762 -
    Average DNS amplification     1.4
    Top domains by amp
```

Installing Apache2

Install dependencies **sudo apt install apache2**	
Verify that apache2 is working by visiting the server URL	

Navigate to the /var/www/html

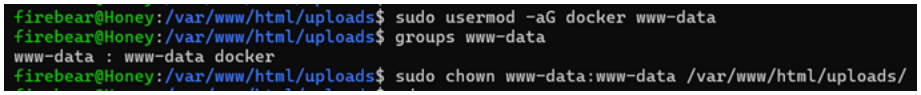
Create the php files needed for file upload

sudo nano fileupload.php

```
 GNU nano 4.8                                                fileupload.php
<!DOCTYPE html>
<html>
<body>

<form action="upload.php" method="post" enctype="multipart/form-data">
    Select File to Upload:
    <input type="file" name="file">
    <input type="submit" name="submit" value="Upload">
</form>

</body>
</html>
```

sudo nano upload.php

```
 GNU nano 4.8                                                upload.php
<?php

$statusMsg = '';

// Temporary upload path on the host
$targetDir = "/var/www/html/uploads/";
$fileName = basename($_FILES["file"]["name"]);
$targetFilePath = $targetDir . $fileName;
$fileType = pathinfo($targetFilePath, PATHINFO_EXTENSION);

if (isset($_POST["submit"]) && !empty($_FILES["file"]["name"])) {
    // Check if there are any errors with the file.
    if ($_FILES["file"]["error"] != 0) {
        $statusMsg = "Error: " . $_FILES["file"]["error"];
    } else {
        // Upload file to temporary location on server
        if (move_uploaded_file($_FILES["file"]["tmp_name"], $targetFilePath)) {
            // Specify the Docker container ID
            $dockerID = 'honey_vsftpd';
            $dockerTargetPath = '/var/ftp/anon_upload/' . $fileName;

            // Construct the Docker cp command
            $copyCmd = "docker cp \"{$targetFilePath}\" \"{$dockerID}:{$dockerTargetPath}\"";
            exec($copyCmd, $output, $return_var);

            // Check if the copy was successful
            if ($return_var == 0) {
                // Delete the temporary file from the host
                unlink($targetFilePath);
                $statusMsg = "The file " . $fileName . " has been uploaded successfully.";
            } else {
                $statusMsg = "Sorry, there was an error uploading your file. Please try again.";
            }
        } else {
            $statusMsg = "Sorry, there was an error uploading your file. Please try again.";
        }
    }
} else {
    $statusMsg = 'Please select a file to upload.';
}

// Display status message
echo $statusMsg;

echo '<br><button onclick="window.location.href=\'fileupload.php\'">Back</button>';
echo '<br><button onclick="window.location.href=\'index.html\'">Home</button>';

?>
```

Change necessary permissions for ‘www-data’ user **sudo usermod -aG docker www-data** **sudo chown www-data:www-data /var/www/html/uploads/**	

Installing Docker and vsftpd

Create new directory called vsftpd

Create Dockerfile in vsftpd directory

```
  GNU nano 4.8                  Dockerfile
FROM ubuntu:latest

# Update the package list and install vsftpd
RUN apt-get update && \
    apt-get install -y vsftpd

# Create the secure_chroot_dir directory
RUN mkdir -p /var/run/vsftpd/empty
RUN mkdir -p /var/ftp
RUN usermod -d /var/ftp ftp
RUN mkdir /var/ftp/anon_upload

# Copy the vsftpd configuration files
COPY vsftpd.conf /etc/vsftpd.conf

# Create a directory for FTP users
RUN chown root:root /var/ftp
RUN chmod og-w /var/ftp
RUN chown ftp:ftp /var/ftp/anon_upload
RUN chmod 0777 /var/ftp/anon_upload

# Expose FTP ports 20 and 21
EXPOSE 20 21

# Start vsftpd in the foreground
CMD ["vsftpd", "/etc/vsftpd.conf"]
```

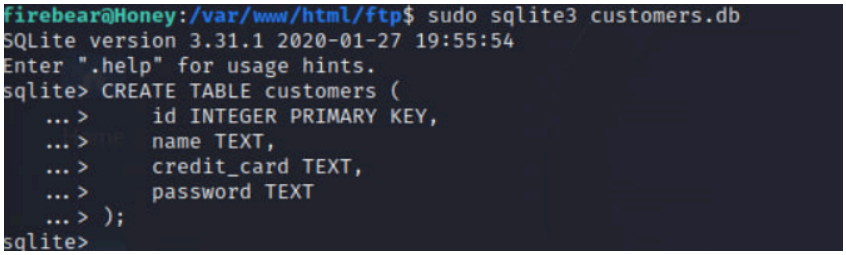
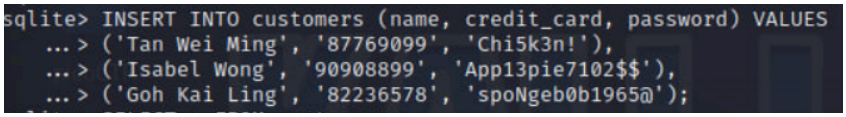
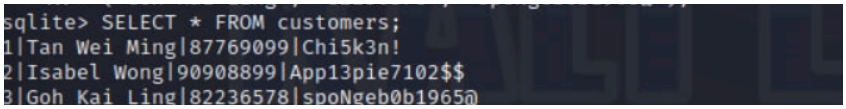
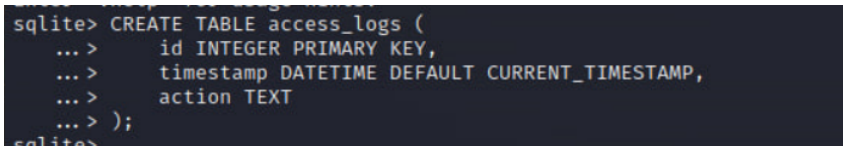
Create vsftpd.conf in vsftpd directory	GNU nano 4.8 vsftpd.conf listen=NO listen_ipv6=YES anonymous_enable=YES local_enable=NO write_enable=YES anon_upload_enable=YES anon_mkdir_write_enable=YES dirmessage_enable=YES use_localtime=YES xferlog_enable=YES connect_from_port_20=YES chown_uploads=YES chown_username=ftp secure_chroot_dir=/var/run/vsftpd/empty pam_service_name=vsftpd ssl_enable=NO rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.k
Build the docker image **sudo docker build -t vsftpd .**	
Build the docker container **sudo docker run -d --name honey_vsftpd vsftpd**	firebear@Honey:~/vsftpd$ sudo docker run -d --name honey_vsftpd vsftpd c5e1ee502e384a30b4c9ebb123bdba11145d2920ce4ad130b9a4ceb54cea8436 **Obtain the docker container id:** c5e1ee502e384a30b4c9ebb123bdba11145d2920ce4ad130b9a4ceb54cea8436

Check if docker container is running **sudo docker ps**	

Creating a .TXT with fake information

Creating a text file to store VIP information **sudo touch vip_into.txt**	
Adding VIP information into the text file **nano vip_info.txt** Show contents of text file **Cat vip_info.txt**	

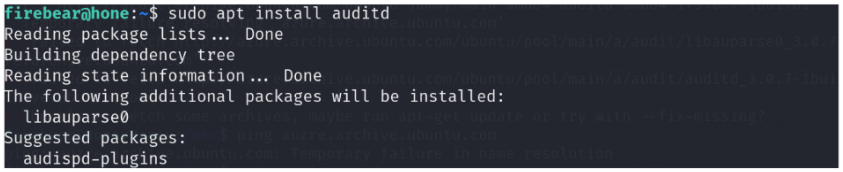
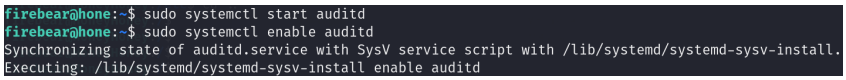
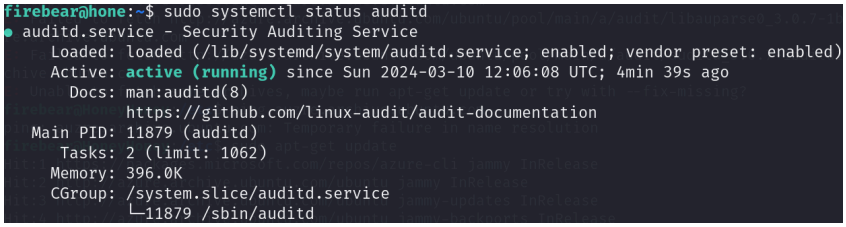
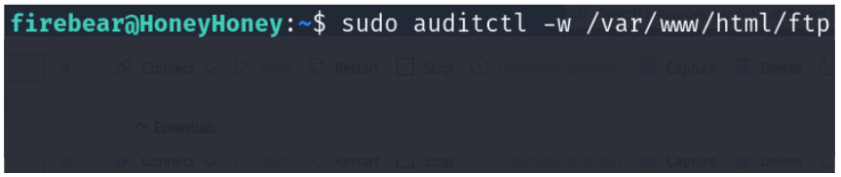
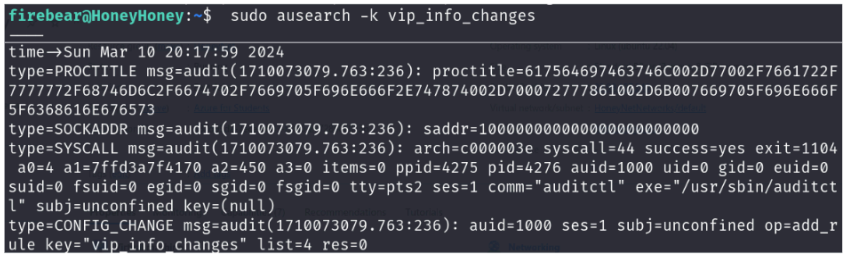
Creating a DB with fake information

Change the directory where the FTP is **cd var/www/html/ftp**	
Creating a database using sqlite3, give a name for the data base **sudo sqlite3 customer.db** Create a table for customers	
Inserting data into the customer.db	
To view the data **SELECT * FROM customers;**	
Create access_logs for customers.db	
View access logs **SELECT * FROM access_logs;**	
To Exit from the SQLITE **.exit**	

Installation of Auditd

[Collecting and Writing audit log file records]

Link: https://sematext.com/glossary/auditd/

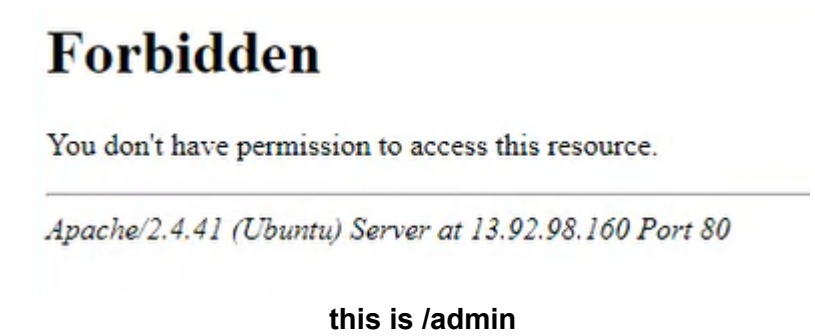
Install Auditd **sudo apt install auditd**	
→ Start & Enable Auditd **sudo systemctl start auditd** **sudo systemctl enable auditd**	
→ Verifying the status **sudo systemctl status auditd**	
To monitor changes to **vip_info.txt** file **sudo auditctl -w /var/www/html/ftp/vip_info.txt -p rwa -k vip_info_changes** **-w** - specifies the files or directory **-p rwa** - sepcifies the permission to monitor for the file, read,write,attribute **-k vip_info_changes** - assigns a unique key to the rule	
To check if the rule is being applied correctly **sudo auditctl -l**	
→ To view the audit logs for events **sudo ausearch -k vip_info_changes** **ausearch** - command line tool for searching audit logs **-k vip_info_changes** - specifies the keys to search	

Creating Honeytoken: Fake Admin Page (Webhook, Tracking, Logging)

Pre-requisites

- PHP v 8.1 (minimum)
- libapache2-mod-php : Install php on the server to ensure php pages show up properly, and not as text.
- Server time updated to SG time UTC +08 for easier logging.
- Install php-sqlite3

→ **.htaccess**



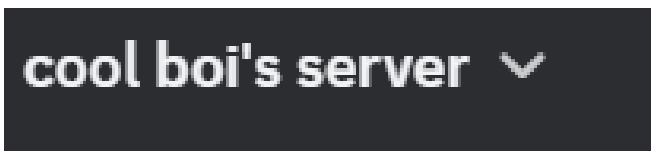
this is /admin

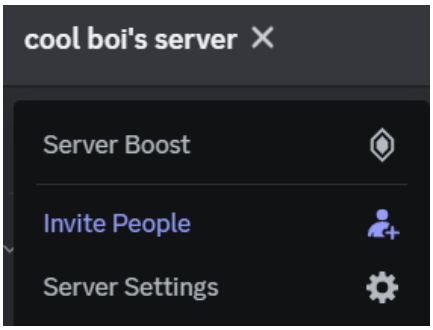
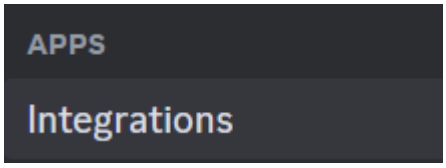
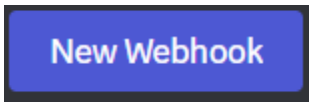
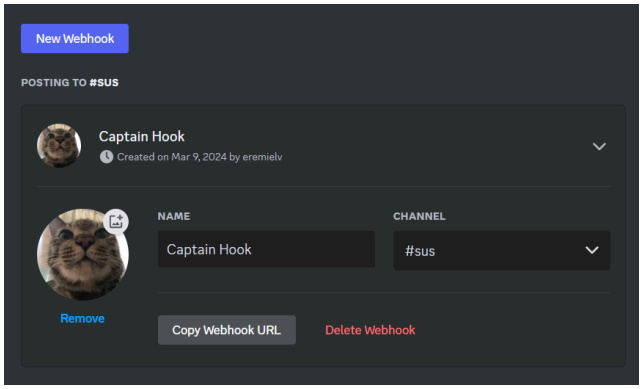
```
Require all denied

<Files "admin.php">
   Require all granted
</Files>

<Files "login_success.php">
   Require all granted
</Files>

<Files "attack_map.html">
   Require all granted
</Files>

<Files "beacon.php">
   Require all granted
</Files>

<Files "fetch_attacks.php">
   Require all granted
</Files>

<Files "attack_map.js">
   Require all granted
</Files>

<Files "beacon.js">
   Require all granted
</Files>
```

Setting up WebHook

Setting up WebHook in discord.

Other alternatives:
1. Slack



2. Microsoft Teams 3. Telegram Bots 4. Email 5. SMS Open Discord and either create a new server or select an existing server where you have permissions to add webhooks. Choose or create a channel where the notifications will be posted.	
Click on Server settings > Integrations > View Webhooks > New Webhook	   
Set a picture and specify a channel from the server. Copy Webhook URL!	

Creating a fake admin page (admin.php) which serves as the honeytoken.

this is /admin/admin.php

→ admin.php

- Admin login page
- Credentials: (admin:iloveyou)
  - Can use hydra and rockyou.txt to brute force as shown below, but DO NOT ATTEMPT (Azure will flag the server and take it down)
  - Password is not too far up the list, thus won't overload the webhook.
- Code will send login/ brute force attempts to discord webhook.

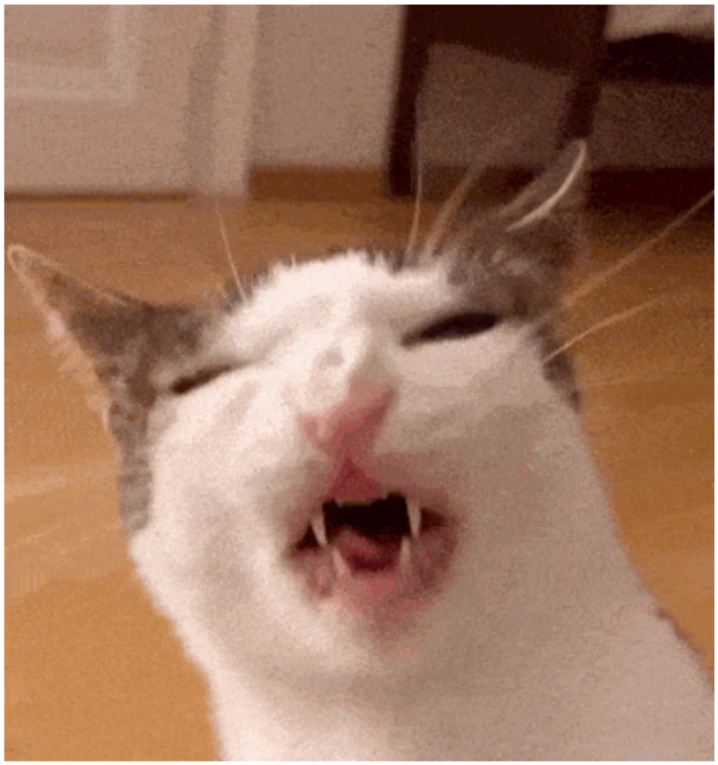
```
<?php
require_once __DIR__ . '/vendor/autoload.php';

// Connect to the SQLite database
$db = new SQLite3('/var/www/html/admin/adminDB.db');

$loginSuccessful = false; // Flag to check login status

// Check for a POST request
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Assuming attackers try to post credentials
    $user = $_POST['username'] ?? 'unknown';
    $pass = $_POST['password'] ?? 'unknown';
    $ip = $_SERVER['REMOTE_ADDR'];

    // Check if the credentials match any fake credentials
    $stmt = $db->prepare('SELECT * FROM admin_users WHERE username = :username AND password =
:password');
    $stmt->bindValue(':username', $user);
    $stmt->bindValue(':password', $pass);
    $result = $stmt->execute();

    // Check if any row matches the provided credentials
    if ($result->fetchArray()) {
        // Credentials are correct
        $loginSuccessful = true;
        $message = "Successful login attempt to admin panel from $ip using username: $user";
    } else {
        // Credentials are incorrect
        $message = "Failed login attempt to admin panel from $ip using username: $user and
password: $pass";
    }

    // Discord webhook URL
    $webhookUrl =
"https://discord.com/api/webhooks/1215952182546010132/5vjCoIUFZ-4wLJU2l6ROrxc2Yuc5pfAaWeKJmTf8ntA
osFpGxY0Vxs7Jlva9R-ef1orV";
```

```
    $postData = json_encode(['content' => $message]);
    $ch = curl_init($webhookUrl);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-type: application/json'));
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $postData);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    curl_exec($ch); // Send the webhook
    curl_close($ch);

    // Redirect based on login success
    if ($loginSuccessful) {
        header('Location: /admin/login_success.php'); // Redirect to a fake dashboard for
successful login
    } else {
        // Optionally, redirect to a "failed login" page or show a message (staying on the current
page for this example)
        $failedLogin = true;
    }

    // Close the database connection
    $db->close();
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Admin Panel</title>
</head>
<body>
    <h1>Admin Login</h1>
    <?php if (!empty($failedLogin)): ?>
        <p style="color: red;">Login failed. Please try again.</p>
    <?php endif; ?>
    <img src="/admin/beacon.php" alt="" style="border: none;">
    <script src="/admin/beacon.js"></script>
    <form action="admin.php" method="post">
        Username: <input type="text" name="username"><br>
        Password: <input type="password" name="password"><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

Creating a fake login success page (login_success.php). Page expires and redirects to home.

Login Successful!

Welcome to the admin panel. Enjoy this cat while we redirect you.



this is /admin/login_success.php

→ login_success.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login Successful</title>
</head>
<body>
    <h2>Login Successful!</h2>
    <p>Welcome to the admin panel. Enjoy this cat while we redirect you.</p>
    <img src="https://media1.tenor.com/m/8QsOaiMKvWQAAAAd/cat-crunch.gif" alt="Crunching Cat">
    <script>
        setTimeout(function() {
            window.location.href = '/'; // Redirect to the home page after displaying the meme
        }, 5000); // Adjust the time as needed (5000 milliseconds = 5 seconds)
    </script>
</body>
</html>
```

[Attacker's POV]

*This is an earlier version of the project. DO NOT ATTEMPT TO BRUTE FORCE. This is just for demonstration.

Running a hydra command to brute force:

hydra -l lobby -P /usr/share/wordlists/rockyou.txt melonet.ddns.net http-post-form "/admin/admin.php:username=^USER^&password=^PASS^:F=error" -s 443 -S

```
┌──(kali㉿kali)-[~]
└─$ hydra -l lobby -P /usr/share/wordlists/rockyou.txt melonet.ddns.net http-post-form "/admin/admin.php:username=^USER^&password=^PASS^:F=error" -s 443 -S

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-10 15:04:26
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-forms://melonet.ddns.net:443/admin/admin.php:username=^USER^&password=^PASS^:F=error
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: 12345
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: rockyou
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: monkey
```

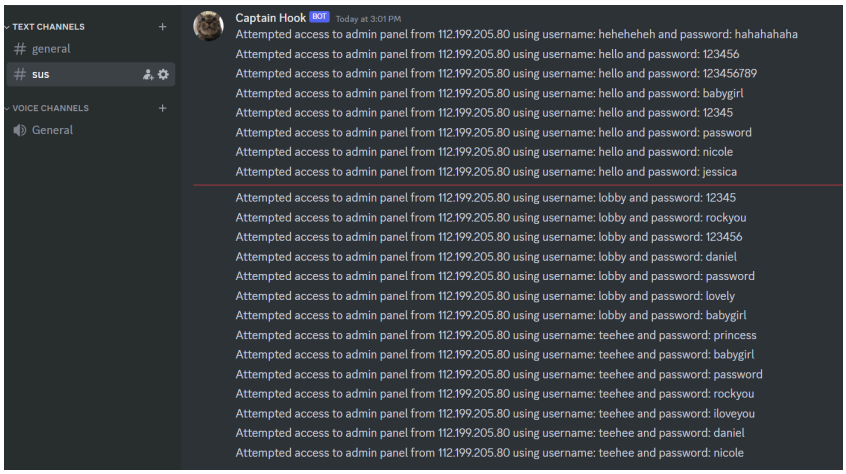
Fake login success page is shown. Page expires and redirects to home.

Login Successful!

Welcome to the admin panel. Enjoy this cat while we redirect you.

[Discord WebHook's POV]

We receive each attempt to login/ brute force in the Discord channel.



Setting up Beacon

Beacon used: 1x1 transparent pixel (https://en.wikipedia.org/wiki/File:1x1.png#/media/File:1x1.png)

→ beacon.js

- Implementing client-side tracking in beacon.js.

```
document.addEventListener("DOMContentLoaded", function() {
    // Generate or retrieve a unique session ID
    let sessionId = sessionStorage.getItem('sessionId');
    if (!sessionId) {
        sessionId = generateUID(); // Generate a new UID
        sessionStorage.setItem('sessionId', sessionId);
    }

    let clickCount = parseInt(sessionStorage.getItem('clickCount') || "0");
    let fakeEventTriggered = sessionStorage.getItem('fakeEventTriggered') === 'true';
```

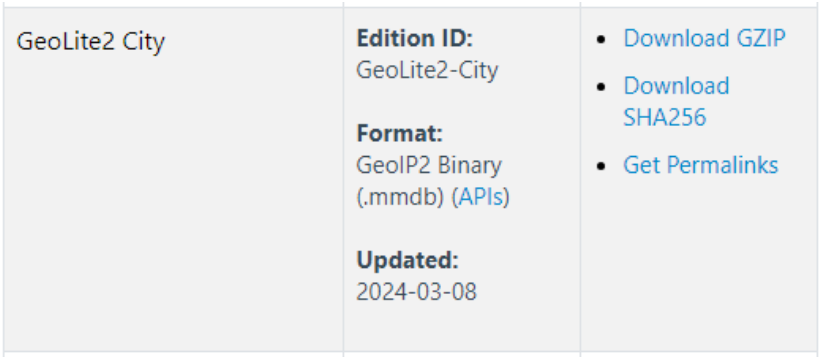
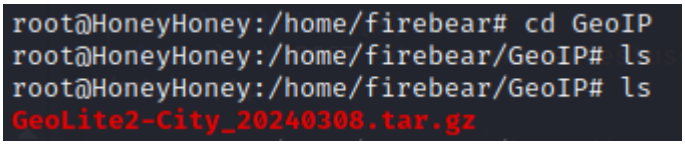
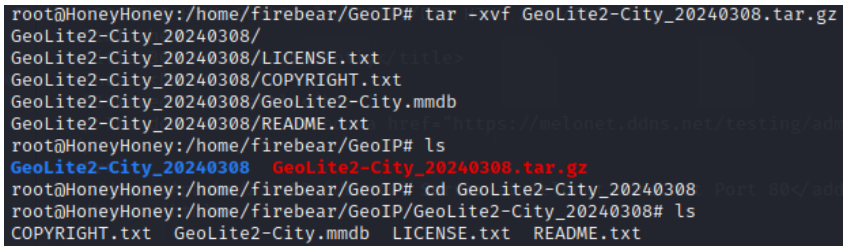
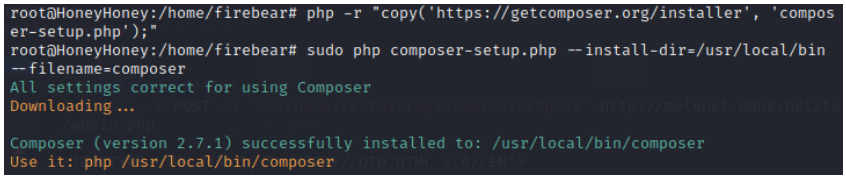
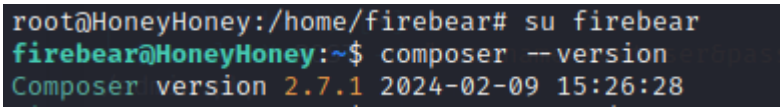
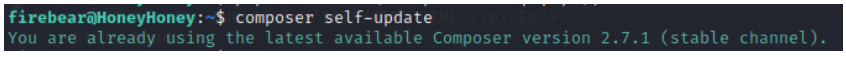
```
    document.addEventListener('click', function() {
        clickCount++;
        sessionStorage.setItem('clickCount', clickCount.toString());
        sendData();
    });

    var fakeButton = document.createElement('button');
    fakeButton.style.position = 'absolute';
    fakeButton.style.left = '-9999px';
    fakeButton.id = 'fakeButton';
    document.body.appendChild(fakeButton);

    fakeButton.addEventListener('click', function() {
        fakeEventTriggered = true;
        sessionStorage.setItem('fakeEventTriggered', 'true');
        sendData();
    });

    // Function to generate a unique ID
    function generateUID() {
        return 'id-' + Math.random().toString(36).substr(2, 16);
    }

    sendData();

    function sendData() {
        var data = {
            sessionId: sessionId,
            screenWidth: window.screen.width,
            screenHeight: window.screen.height,
            referrer: document.referrer,
            clickCount: clickCount,
            fakeEventTriggered: fakeEventTriggered,
        };

        fetch("/admin/beacon.php", {
            method: "POST",
            headers: {"Content-Type": "application/json"},
            body: JSON.stringify(data)
        });
    }


});
```

→ beacon.php

- Setting up server-side tracking and beacon functions handling(logging).

Download GeoIP (MaxMind) for GeoIP information.

- Free
- Downloadable database, allowing for offline IP address to location mappings. This can be faster and more privacy-friendly since it doesn't require sending IP addresses to a third-party service.
- Reasonably accurate for country-level geolocation, and generally acceptable for city-level.
- Requires manual updates to the database to stay current, although automated update options are available with additional setup.

Sign up for an account. Then head to this page: https://www.maxmind.com/en/accounts/984758/geoip/downloads . Download GeoLite2 City (GZIP) into Kali Linux.	
Create folder under /home/firebear called GeoIP	
Upload file from Kali Linux to server. **scp /home/kali/Downloads/GeoLite2-City_20240308.tar.gz firebear@23.99.107.240:/home/firebear/GeoIP**	
Unzip file. Make sure to check if GeoLite2-City.mmdb is inside. Check using command. **ls /home/firebear/GeoIP/GeoLite2-City_20240308/GeoLite2-City.mmdb**	 
Enter the /home/firebear directory. We need Composer to proceed with using GeoLite2. Set up Composer. **php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"** **sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer**	
Check if Composer installed. (Check as non-root user) **composer --version**	
Check for the latest version. **composer self-update**	

Remove Composer installer script.

php -r "unlink('composer-setup.php');"

```
firebear@HoneyHoney:~$ php -r "unlink('composer-setup.php');"
```

Now we can proceed with the GeoLite2 setup. Set up a composer.json file.

composer init

```
firebear@HoneyHoney:/var/www/html/admin$ composer init

  Welcome to the Composer config generator

This command will guide you through creating your composer.json config.
```

Set up with necessary details. I only specified geoip2 as my dependency. I left the rest blank.

```
Package name (<vendor>/<name>) [firebear/admin]:
Description []:
Author [n to skip]: n
Minimum Stability []:
Package Type (e.g. library, project, metapackage, composer-plugin) []:
License []:
```

```
Define your dependencies.

Would you like to define your dependencies (require) interactively [yes]? yes
Search for a package: geoip2

Found 15 packages matching geoip2

  [0] geoip2/geoip2
  [1] maxmind-db/reader
  [2] tronovav/geoip2-update
  [3] gpslab/geoip2
  [4] geocoder-php/geoip2-provider
  [5] bobey/geoip2-geolite2-composer
  [6] cravler/maxmind-geoip-bundle
  [7] danielme85/laravel-geoip2
  [8] tobai/magento2-geo-ip2 Abandoned. No replacement was suggested.
  [9] atchondjo/geoip2country
 [10] leo108/geolite2-db
 [11] dotkernel/dot-geoip
 [12] tuandm/geoip2
 [13] overals/yii2-geoip2
 [14] bonvga/geoip2 Abandoned. No replacement was suggested.

Enter package # to add, or the complete package name if it is not listed: 0
Enter the version constraint to require (or leave blank to use the latest version):
```

```
Using version ^3.0 for geoip2/geoip2
Search for a package:
Would you like to define your dev dependencies (require-dev) interactively [yes]? no
Add PSR-4 autoload mapping? Maps namespace "Firebear\Admin" to the entered relative path.
 [src/, n to skip]:

{
    "name": "firebear/admin",
    "require": {
        "geoip2/geoip2": "^3.0"
    },
    "autoload": {
        "psr-4": {
            "Firebear\\Admin\\": "src/"
        }
    }
}

Do you confirm generation [yes]? yes
Would you like to install dependencies now [yes]? yes
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 0 updates, 0 removals
  - Locking composer/ca-bundle (1.4.1)
  - Locking geoip2/geoip2 (v3.0.0)
  - Locking maxmind-db/reader (v1.11.1)
  - Locking maxmind/web-service-common (v0.9.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 4 installs, 0 updates, 0 removals
  - Downloading composer/ca-bundle (1.4.1)
  - Downloading maxmind/web-service-common (v0.9.0)
  - Downloading maxmind-db/reader (v1.11.1)
  - Downloading geoip2/geoip2 (v3.0.0)
  - Installing composer/ca-bundle (1.4.1): Extracting archive
  - Installing maxmind/web-service-common (v0.9.0): Extracting archive
  - Installing maxmind-db/reader (v1.11.1): Extracting archive
  - Installing geoip2/geoip2 (v3.0.0): Extracting archive
3 package suggestions were added by new dependencies, use `composer suggest` to see detai
ls.
Generating autoload files
1 package you are using is looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found.
PSR-4 autoloading configured. Use "namespace Firebear\Admin;" in src/
Include the Composer autoloader with: require 'vendor/autoload.php';
```

Check to see if Composer has successfully included geoip2.

composer show

```
firebear@HoneyHoney:/var/www/html/admin$ composer show
composer/ca-bundle          1.4.1   Lets you find a path to the system CA bundle, and ...
geoip2/geoip2               v2.13.0 MaxMind GeoIP2 PHP API
maxmind-db/reader           v1.11.1 MaxMind DB Reader API
maxmind/web-service-common v0.9.0  Internal MaxMind Web Service API
```

```
< ?php
ini_set('display_errors', 1);
error_reporting(E_ALL);

require_once
__DIR__.
'/vendor/autoload.php';
use
GeoIp2\Database\Reader;

// Function
to
analyze
the
user
agent
function
isLikelyBot($userAgent) {
$botPatterns = [
   '/googlebot/i', '/bingbot/i', '/slurp/i', '/yahoo/i', '/yandex/i', '/baiduspider/i',
   '/facebookexternalhit/i', '/twitterbot/i', '/linkedinbot/i', '/curl/i', '/python/i',
   '/php/i', '/ruby/i', '/java/i', '/perl/i', '/wget/i', '/headlesschrome/i'
];

foreach($botPatterns as $pattern) {
if (preg_match($pattern, $userAgent)) {
return true;
}
}

// Additional
specific
checks
for headless browsers
if (strpos($userAgent, 'HeadlessChrome') !== false) {
return true;
}

return false;
}

function
writeToDatabase($data) {
$dbPath = '/home/firebear/GeoIP/GeoLite2-City_20240308/GeoLite2-City.mmdb'; // Update
to
your
path
to
the
GeoIP2
database
$reader = new
Reader($dbPath);
$retryCount = 0;
$maxRetries = 5;
$done = false;

while (!$done & & $retryCount < $maxRetries) {
```

```
try {
$db = new SQLite3('/var/www/html/admin/attack_map.db', SQLITE3_OPEN_READWRITE |
SQLITE3_OPEN_CREATE);

$ip = $_SERVER['REMOTE_ADDR'];

// Add GeoIP2 code here
$reader = new Reader($dbPath);
$geoInfo = $reader->city($ip);
$country = $geoInfo->country->name ?? 'Unknown';
$city = $geoInfo->city->name ?? 'Unknown';
$latitude = $geoInfo->location->latitude ?? 'Unknown';
$longitude = $geoInfo->location->longitude ?? 'Unknown';

$db->exec("CREATE TABLE IF NOT EXISTS logs (
id INTEGER PRIMARY KEY AUTOINCREMENT,
session_id TEXT UNIQUE,
datetime TEXT,
ip TEXT,
user_agent TEXT,
referer TEXT,
country TEXT,
city TEXT,
latitude TEXT,
longitude TEXT,
click_count INTEGER,
fake_event_triggered BOOLEAN,
is_likely_bot BOOLEAN
)");

$sessionId = $data['sessionId'] ?? exit('Session ID is required.');
$dateTime = date('Y-m-d H:i:s');
$ip = $_SERVER['REMOTE_ADDR'];
$userAgent = $_SERVER['HTTP_USER_AGENT'];
$referrer = $_SERVER['HTTP_REFERER'] ?? 'Direct or no referrer';
$clickCount = $data['clickCount'] ?? 0;
$fakeEventTriggered = $data['fakeEventTriggered'] ? 1: 0;
$
   isLikelyBot = isLikelyBot($userAgent) ? 1: 0;

$stmt = $db->prepare(
   "INSERT OR REPLACE INTO logs (session_id, datetime, ip, user_agent, referer, country, city,
latitude, longitude, click_count, fake_event_triggered, is_likely_bot) VALUES (?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?)");

$stmt->bindValue(1, $sessionId);
$stmt->bindValue(2, $dateTime);
$stmt->bindValue(3, $ip);
$stmt->bindValue(4, $userAgent);
$stmt->bindValue(5, $referrer);
$stmt->bindValue(6, $country);
$stmt->bindValue(7, $city);
$stmt->bindValue(8, $latitude);
$stmt->bindValue(9, $longitude);
$stmt->bindValue(10, $clickCount, SQLITE3_INTEGER);
$stmt->bindValue(11, $fakeEventTriggered, SQLITE3_INTEGER);
$stmt->bindValue(12, $isLikelyBot, SQLITE3_INTEGER);

$stmt->execute();
$done = true;
} catch(Exception $e) {
                    // Wait
for a moment before retrying
```

```
usleep(500000); // 500ms
$retryCount++;
} finally {
if ($db ?? false) {
$db->close();
}
}
}
}

$data = json_decode(file_get_contents('php://input'), true);
writeToDatabase($data);

header('Content-Type: image/gif');
echo
base64_decode('R0lGODlhAQABAIAAAP///wAAACH5BAEAAAAALAAAAAABAAEAAAICRAEAOw==');
exit;
? >
```

***Make sure both beacon.php and beacon.js are embedded in the html portion of admin.php.**

```
<!DOCTYPE html>
<html>
<head>
   <title>Admin Panel</title>
</head>
<body>
   <h1>Admin Login</h1>
   <?php if (!empty($failedLogin)): ?>
       <p style="color: red;">Login failed. Please try again.</p>
   <?php endif; ?>
   <img src="/admin/beacon.php" alt="" style="border: none;">
   <script src="/admin/beacon.js"></script>
   <form action="admin.php" method="post">
       Username: <input type="text" name="username"><br>
       Password: <input type="password" name="password"><br>
       <input type="submit" value="Login">
   </form>
</body>
</html>
```

***Make sure the file directory looks like this, including composer folders. Set necessary file permissions. This part is crucial for everything to work.**

```
root@Honey:/var/www/html/admin# ls -la
total 108
drwxr-xr-x 3 www-data www-data  4096 Apr  4 03:42 .
drwxr-xr-x 5 root     root      4096 Apr  4 00:49 ..
-rw-r--r-- 1 www-data www-data   424 Apr  4 03:12 .htaccess
-rw-r--r-- 1 www-data www-data    95 Mar 10 00:31 1x1.png
-rw-r--r-- 1 www-data www-data  2775 Mar 31 21:40 admin.php
-rw-rw-rw- 1 www-data www-data 12288 Mar 31 21:44 adminDB.db
-rw-r--r-- 1 www-data www-data     0 Mar 31 21:30 adminDB.sqlite
-rw-rw-r-- 1 www-data www-data 32768 Apr  4 03:18 attack_map.db
-rw-r--r-- 1 www-data www-data  1131 Mar 23 00:40 attack_map.html
-rw-r--r-- 1 www-data www-data  3018 Mar 31 16:50 attack_map.js
-rw-r--r-- 1 www-data www-data  1961 Apr  1 05:29 beacon.js
-rw-r--r-- 1 www-data www-data  4073 Apr  4 02:49 beacon.php
-rw-r--r-- 1 www-data www-data    88 Mar 22 17:20 composer.json
-rw-r--r-- 1 www-data www-data  9773 Mar 22 17:20 composer.lock
-rw-r--r-- 1 www-data www-data   702 Mar 31 18:53 fetch_attacks.php
-rw-r--r-- 1 www-data www-data   570 Mar 22 16:28 login_success.php
drwxrwxr-x 6 www-data www-data  4096 Mar 22 17:20 vendor
```

[Attackers POV]

Beacon is unseen.
User is tracked the moment the page is *accessed*, as this page is under a forbidden directory (/admin).

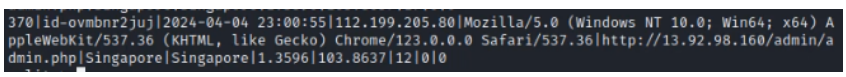
Properties tracked across beacon.js and beacon.php:

- Date and Time
- IP
- User-Agent
- Referer
- GeoIP (Country, City, Latitude, Longitude)
- Click Count
- Hidden JS event
- Simple bot detection

this is /admin/admin.php

[Logs in database collected from beacon POV]

Information from both beacon.js and beacon.php are successfully logged on page access and updated according to session-id.



Attack Map

Create a database to store the logged GeoIP information from the beacons.

Create a table in the SQLite prompt.
Check if database is populated (after triggering /admin/admin.php)

sqlite3 attack_map.db

Display all entries in the SQLite prompt.

SELECT * FROM logs;

```
290|session-1712241106000|2024-04-04 22:31:47|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64
; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160
/admin/admin.php|Singapore|Singapore|1.3596|103.8637|3|1|0
291|id-os7vp4vuqt|2024-04-04 22:33:00|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64; x64) A
ppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160/admin/a
dmin.php|Singapore|Singapore|1.3596|103.8637|0|0|0
296|id-h8gxgmjhm1m|2024-04-04 22:33:37|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160/admin/
admin.php|Singapore|Singapore|1.3596|103.8637|4|0|0
303|id-k3xox331lll|2024-04-04 22:34:43|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160/admin/
admin.php|Singapore|Singapore|1.3596|103.8637|9|0|0
310|id-z5p0ow73zoa|2024-04-04 22:37:54|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160/admin/
admin.php|Singapore|Singapore|1.3596|103.8637|6|0|0
317|id-ulnp0ctr5xh|2024-04-04 22:43:15|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160/admin/
admin.php|Singapore|Singapore|1.3596|103.8637|9|0|0
327|id-ai12wxh84wq|2024-04-04 22:44:14|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160/admin/
admin.php|Singapore|Singapore|1.3596|103.8637|7|0|0
328|id-4u5k0acrjkj|2024-04-04 22:44:35|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160/admin/
admin.php|Singapore|Singapore|1.3596|103.8637|0|0|0
346|id-rlloeta524p|2024-04-04 22:51:20|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160/admin/
admin.php|Singapore|Singapore|1.3596|103.8637|2|0|0
347|session-1712242283959|2024-04-04 22:51:24|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64
; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160
/admin/admin.php|Singapore|Singapore|1.3596|103.8637|3|1|0
355|id-7gbfmgcgw6a|2024-04-04 22:53:02|112.199.205.80|Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36|http://13.92.98.160/admin/
admin.php|Singapore|Singapore|1.3596|103.8637|17|0|0
```

→ attack_map.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Live Attack Map</title>
   <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"/>
   <link href="https://unpkg.com/leaflet-gesture-handling/dist/leaflet-gesture-handling.min.css"
type="text/css" rel="stylesheet">
   <link href="https://cdn.jsdelivr.net/npm/leaflet-pulse-icon@0.1.0/dist/L.Icon.Pulse.css"
rel="stylesheet">
   <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
   <script src="https://d3js.org/d3.v6.min.js"></script>
   <script src="https://unpkg.com/leaflet-gesture-handling"></script>
   <script
src="https://cdn.jsdelivr.net/npm/leaflet-pulse-icon@0.1.0/dist/L.Icon.Pulse.js"></script>
   <style>
       #map {
           height: 100vh;
       }
       .leaflet-container {
           /* This ensures the map is interactive (zoom and pan) only when using ctrl+scroll or
two fingers on touch devices. */
           touch-action: none;
       }
   </style>
</head>
<body>
   <div id="map"></div>
   <script src="attack_map.js"></script> <!-- Your custom JS file for the attack map -->
</body>
</html>
```

→ attack_map.js

```
document.addEventListener('DOMContentLoaded', function() {
   var map = L.map('map').setView([1.3521, 103.8198], 12); // Center the map around Singapore
   L.tileLayer('https://{s}.basemaps.cartocdn.com/dark_all/{z}/{x}/{y}{r}.png', {
```

```
        maxZoom: 19,
        attribution: '© OpenStreetMap contributors'
    }).addTo(map);

    var svgLayer = L.svg({clickable: true}).addTo(map);
    var svg = d3.select("#map").select("svg");
    var g = svg.append("g");

    var targetLocation = [1.3521, 103.8198]; // Server location: Singapore

    function drawAttack(attacks) {
        g.selectAll('*').remove(); // Clear existing paths and pulses

        attacks.forEach(function(attack) {
            var from = map.latLngToLayerPoint(new L.LatLng(attack.latitude, attack.longitude));
            var to = map.latLngToLayerPoint(targetLocation);

            // Draw the curved attack line
            var path = g.append("path")
                .attr("fill", "none")
                .attr("stroke", "red")
                .attr("stroke-width", 2);

            // Define the curve
            var curve = d3.line().curve(d3.curveNatural)([
                [from.x, from.y],
                [(from.x + to.x) / 2, from.y - 100], // This creates the arch effect
                [to.x, to.y]
            ]);

            path.attr("d", curve);

            // Animate the attack line
            var totalLength = path.node().getTotalLength();
            path.attr("stroke-dasharray", totalLength + " " + totalLength)
                .attr("stroke-dashoffset", totalLength)
                .transition()
                    .duration(3000)
                    .attr("stroke-dashoffset", 0)
                    .on("end", function() { pulseTarget(to.x, to.y); });
        });
    }

    function pulseTarget(x, y) {
        var pulse = g.append("circle")
            .attr("cx", x)
            .attr("cy", y)
            .attr("r", 1)
            .style("fill", "red");

        animatePulse(pulse);
    }

    function animatePulse(circle) {
        // Animate the pulse
        circle.transition()
            .duration(1000)
            .attr("r", 10)
            .style("opacity", 0)
            .transition()
                .duration(1000)
                .attr("r", 1)
                .style("opacity", 1)
```

```
            .on("end", function() { animatePulse(circle); }); // Repeat the animation
    }

    function fetchAttacks() {
        fetch('fetch_attacks.php')
            .then(response => response.json())
            .then(data => {
                // Assuming 'data' is an array of attack objects with 'latitude' and 'longitude'
properties
                drawAttack(data);
            })
            .catch(error => console.error('Error fetching attack data:', error));
    }

    // Fetch attacks data every 30 seconds to update the map
    fetchAttacks(); // Initial fetch
    setInterval(fetchAttacks, 30000);

    // Redraw attacks whenever the map view changes
    map.on('moveend', fetchAttacks);
});
```

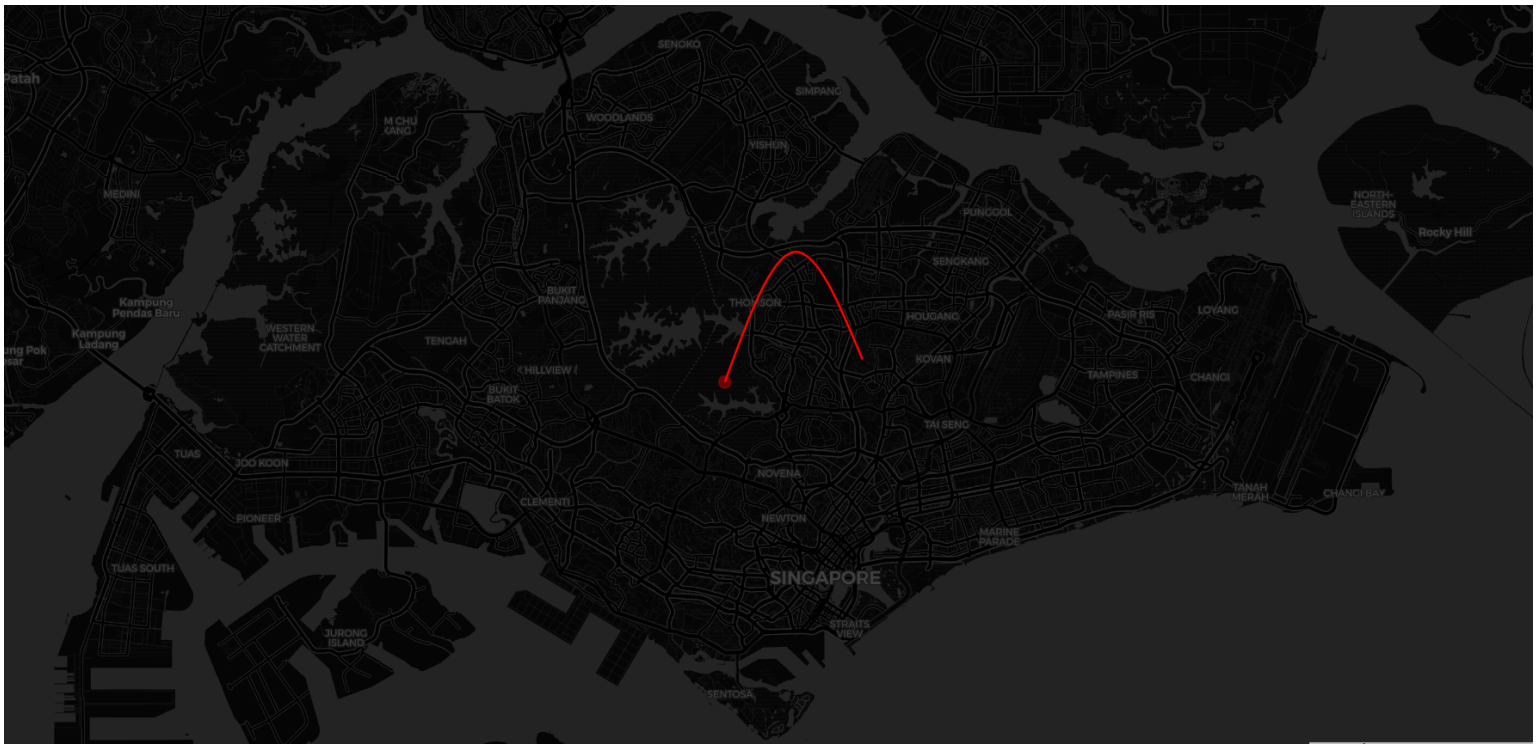
→ fetch_attacks.php

```
 <?php
header('Content-Type: application/json');

$db = new SQLite3('/var/www/html/admin/attack_map.db');

// Get the current time minus 5 minutes
$fiveMinutesAgo = new DateTime('-5 minutes');

// Prepare SQL statement with a parameter for time comparison
$stmt = $db->prepare('SELECT datetime, latitude, longitude, click_count, fake_event_triggered
FROM logs WHERE datetime >= :fiveMinutesAgo AND latitude != "Unknown" AND longitude != "Unknown"
ORDER BY datetime DESC');
$stmt->bindValue(':fiveMinutesAgo', $fiveMinutesAgo->format('Y-m-d H:i:s'));

$results = $stmt->execute();

$attacks = [];
while ($row = $results->fetchArray(SQLITE3_ASSOC)) {
    $attacks[] = $row;
}

echo json_encode($attacks);
```

[Attack Map POV]

- Powered by logs which track access to /admin/admin.php
    - Can be expanded to include other databases logs
    - However, preferable if data is powered from centralized logging server
- Centered around Singapore, but whole map is viewable
- Only attacks from last 5 minutes appear
    - Global attacks available, but CAUTION USING VPN. (Azure may flag the server and take it down)
- Source of attack is geo-located and an arching line simulates the attack detection.
- Server location which is the target, pulses after an attack is detected.


NORTH AMERICA
ASIA
AFRICA
SOUTH AMERICA
Indian Ocean


SENOKO
SIMPANG
WOODLANDS
YISHUN
PUNGGOL
SENGKANG
NORTH-EASTERN ISLANDS
Rocky Hill
MEDINI
Kampung Pendas Baru
Kampung Ladang
WESTERN WATER CATCHMENT
TENGAH
BUKIT PANJANG
THOMSON
HOUGANG
PASIR RIS
LOYANG
KOVAN
TAMPINES
CHANGI
HILLVIEW
BUKIT BATOK
TAI SENG
TUAS
JOO KOON
NOVENA
TANAH MERAH
CHANGI BAY
CLEMENTI
PIONEER
NEWTON
MARINE PARADE
TUAS SOUTH
SINGAPORE
JURONG ISLAND
STRAITS VIEW
SENTOSA