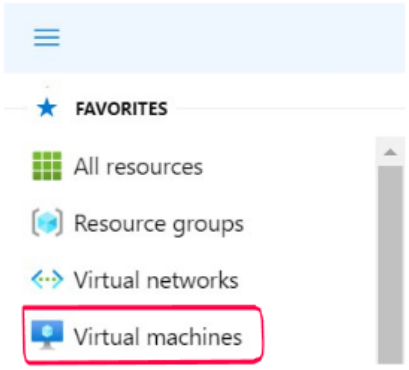
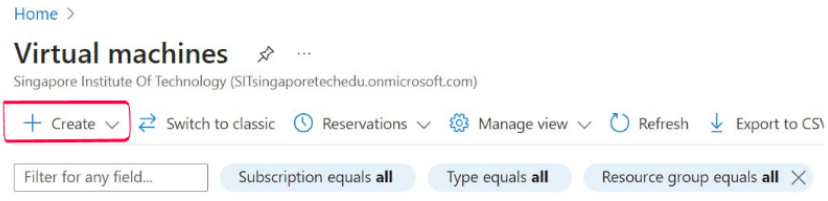
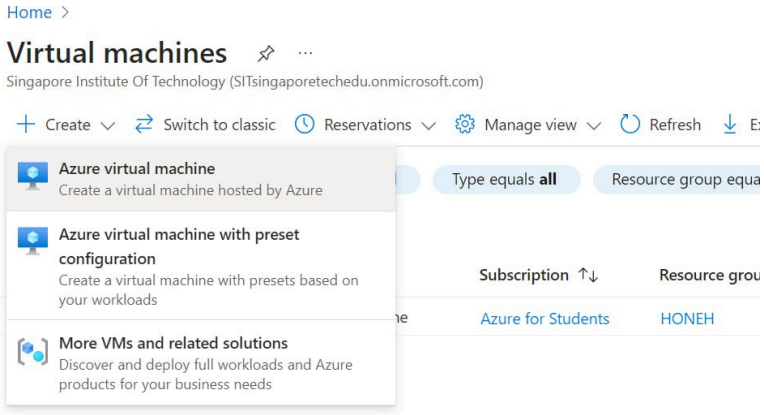
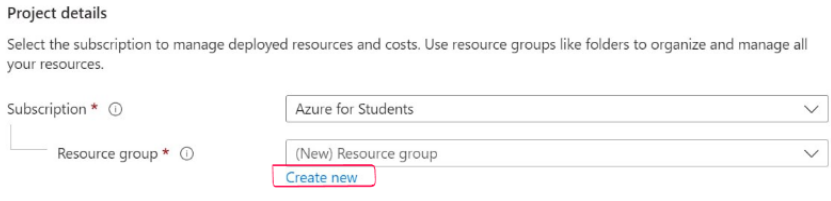
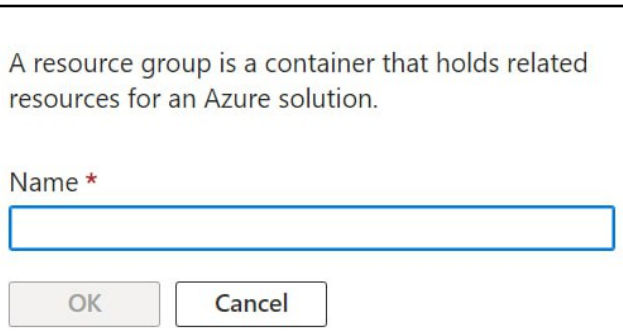
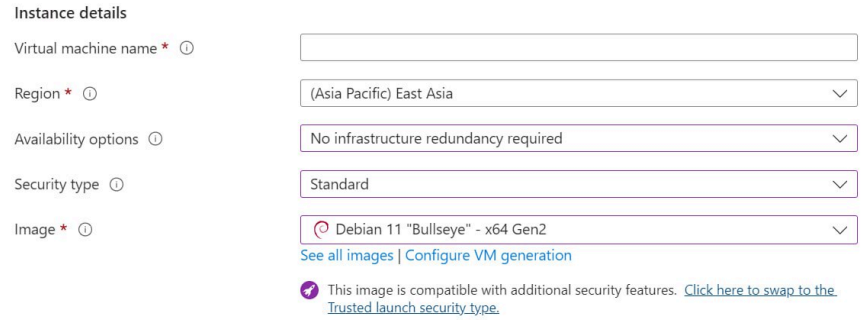
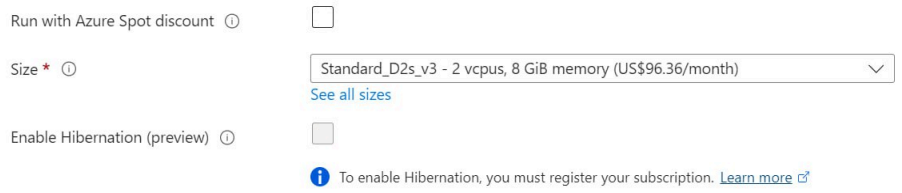
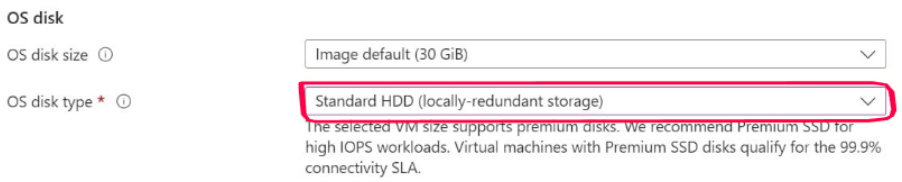
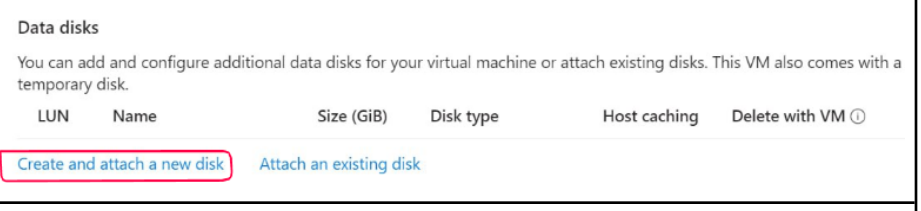
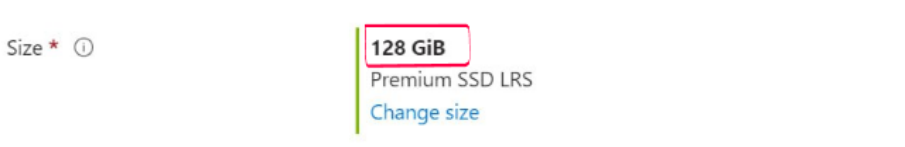
Microsoft Azure HoneyPot (T-Pot)

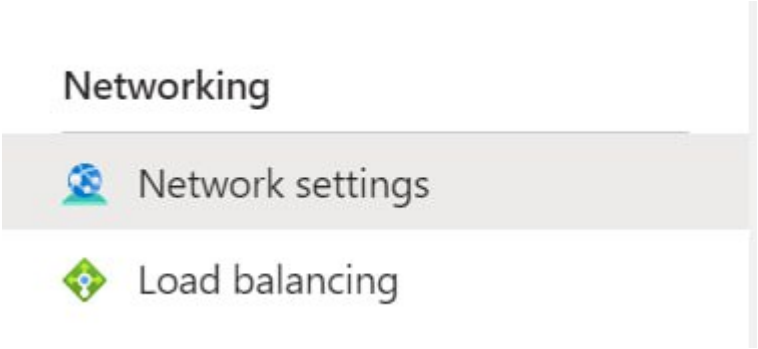
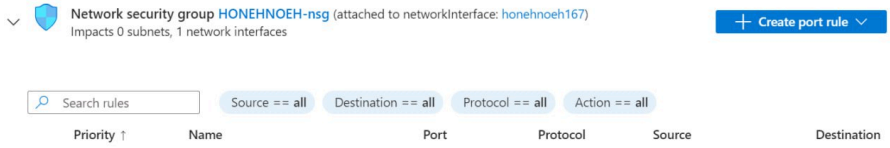
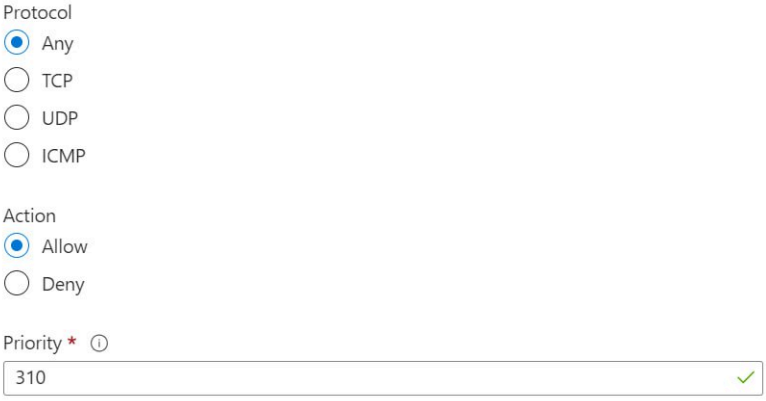
Creating a VM

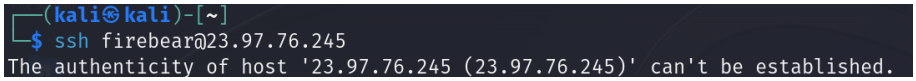
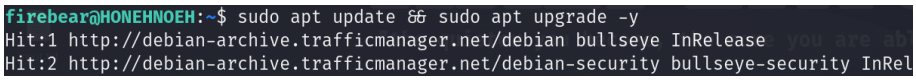
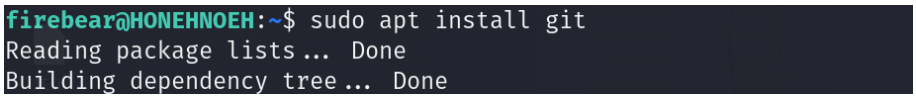
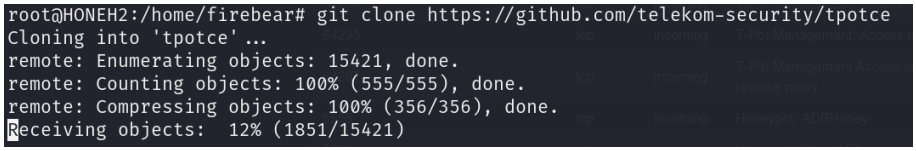
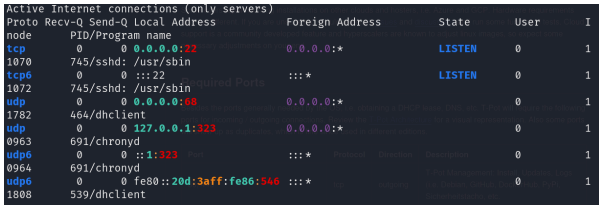
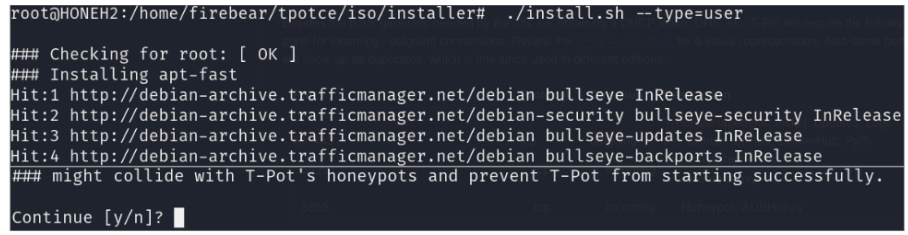
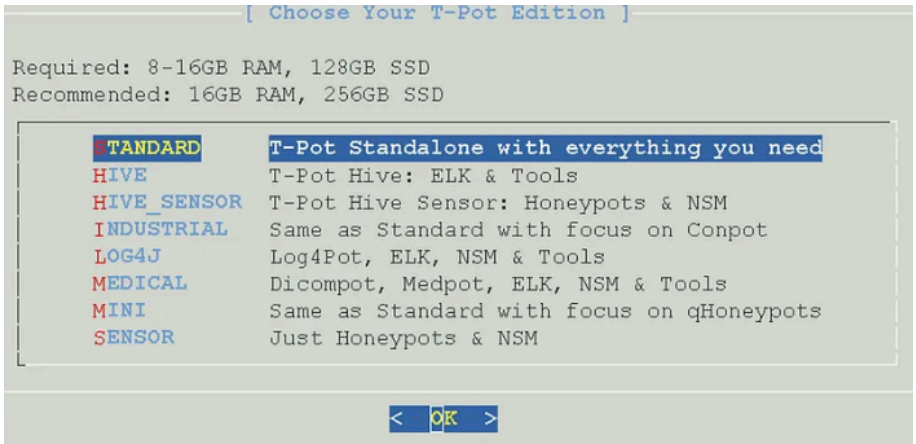
Head to the microsoft azure webpage to create a Virtual Machine

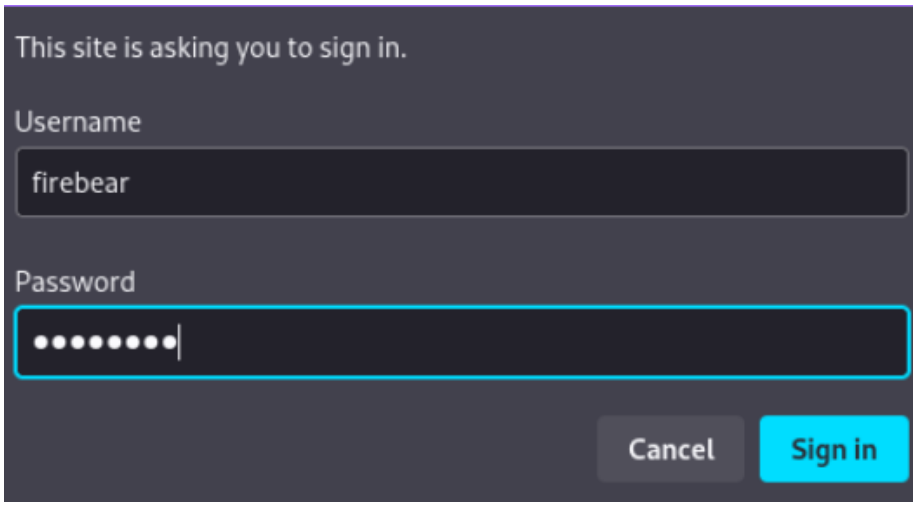
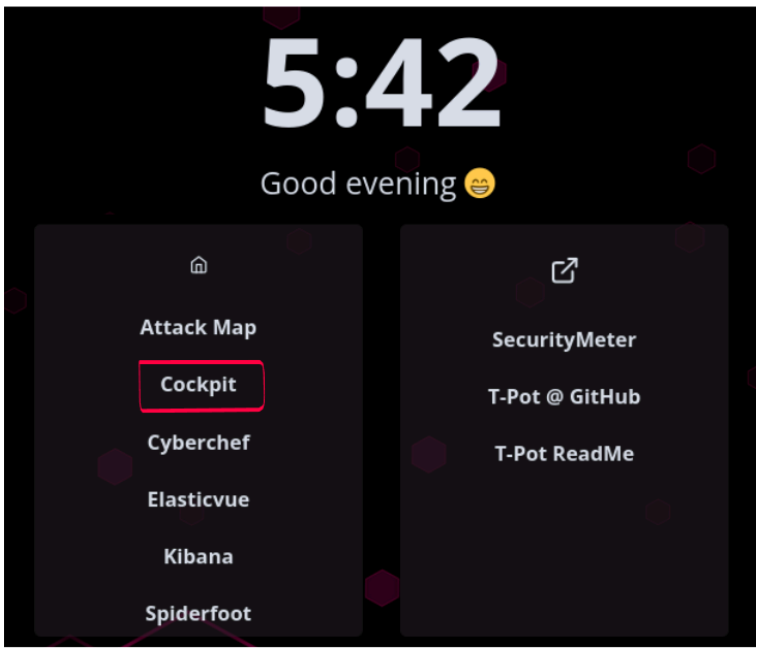
- https://azure.microsoft.com/en-us/

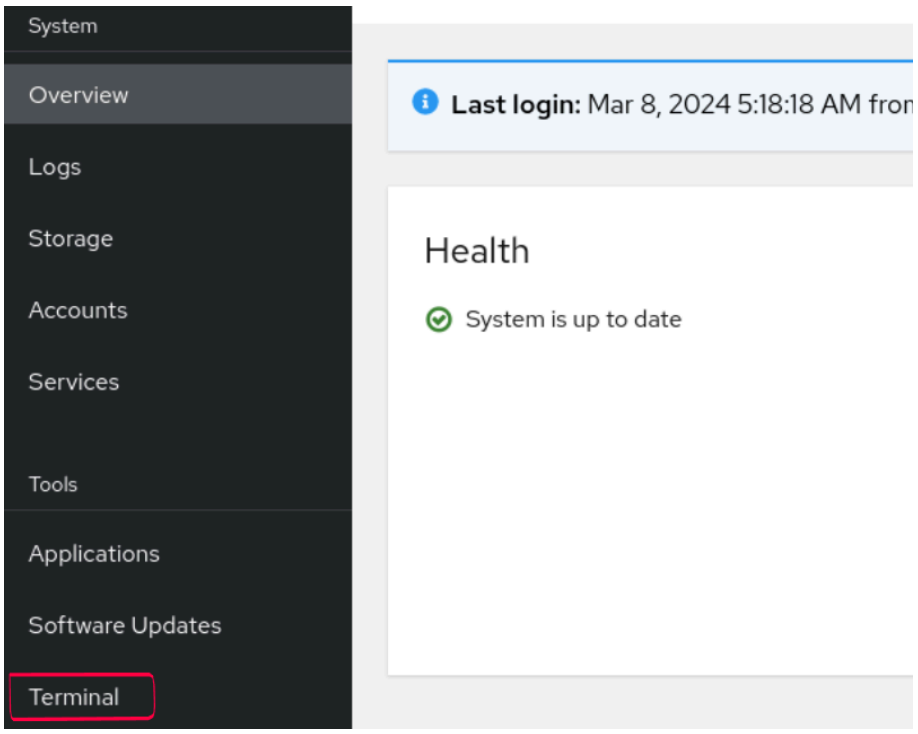
<table>
<tr><td>Once you've logged in, navigate to the sidebar and locate "Virtual Machine"</td><td> 
</td></tr>
<tr><td rowspan="2">Click on create → Azure Virtual Machine</td><td> 
</td></tr>
<tr><td> 
</td></tr>
<tr><td colspan="2">[BASICS] Tabs</td></tr>
<tr><td>Click on Create new</td><td> 
</td></tr>
</table>

Give any name you want	
Select Name: <Name your virtual machine> Region: (Asia Pacific) East Asia Avail Option: No infrastructure Redundancy Required Security Type: Standard Image: Debian 11 'Bullseye' x64 Gen2	
Select the size Standard_D2s_V3 - 2vcpus, 8 GiB memory	
Select Authentication type: Password (For SSH & Cockpit) Create a username & password	
[DISK] Tabs	
Change the OS Disk Type to **Standard HDD (Locally-Redundant Storage)**	
Under DataDisk Click on "Create and attach a new disk" Select the size as "**128 GiB**"	 

Click next and leave everything as default, and click on “**Review + Create**”	
Once your VM is created, navigate to the VM under networking → Network settings	
Click on **Create Port Rule**	
Follow these settings	  

On Kali-Linux	
SSH to your public ip **ssh <usr>@<public-ip>**	
Update and Upgrade the Debian **sudo apt update && sudo apt upgrade -y**	
Install Git **sudo apt install git**	
Install TPOT **git clone https://github.com/telekom-security/tpotce**	
Change to the directory cd tpotce/iso/installer	
It will show you the active Internet Connections	
Execute the install.sh sudo ./install.sh --type=user After the installation, type in the “**y**” during the installation	
Select “Standard”	

Create Username and Password	
Once done installing, you will be kicked out from port 22, its normal	
Open web browser head to your <public ip> https://<public-ip>:64297	
Select Cockpit	

Head to the terminal	

https://medium.com/@alisefer/t-pot-installation-and-use-f359b9f39a93

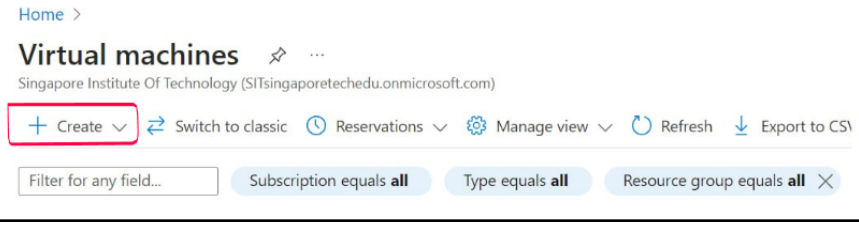
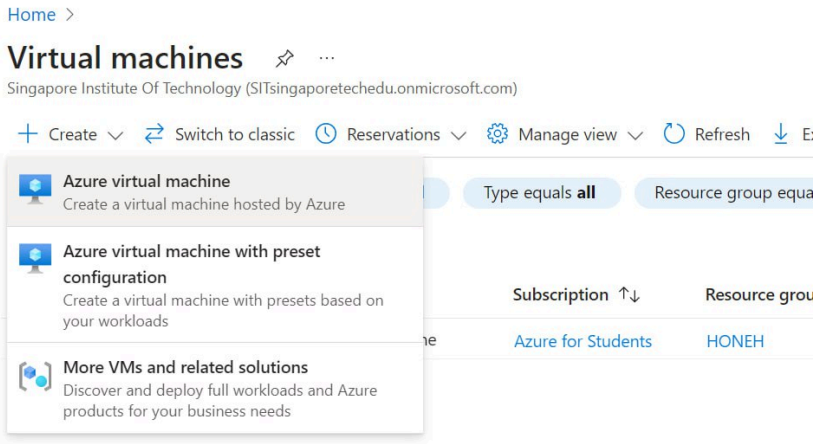
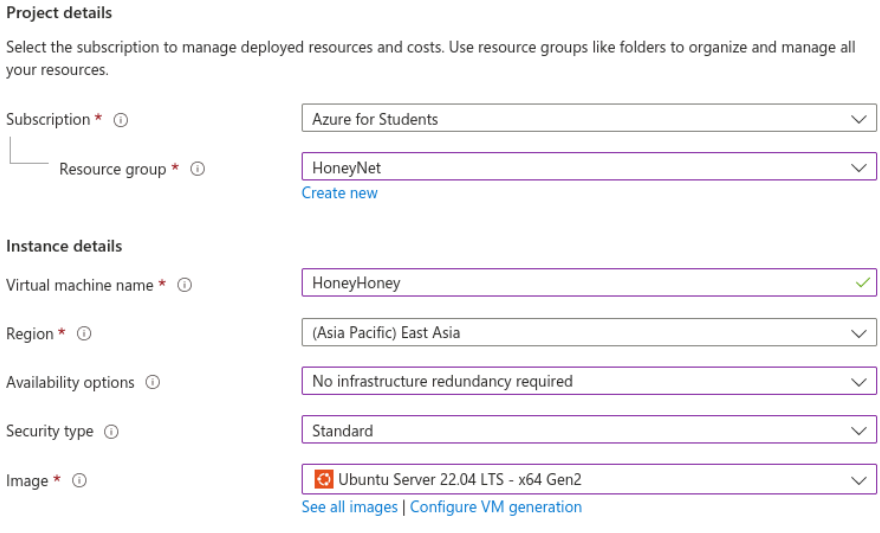
https://www.linkedin.com/pulse/setup-t-pot-honeypot-azure-less-than-30-minutes-sigmund

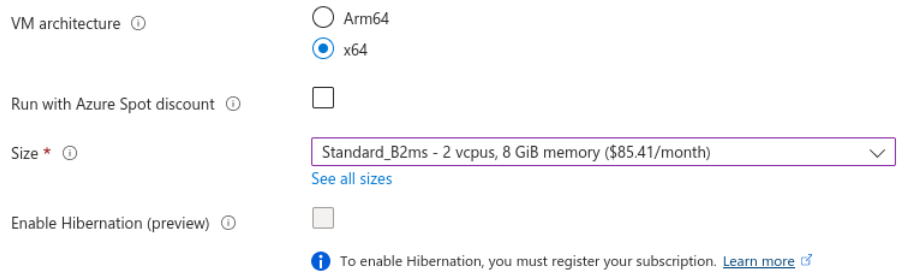
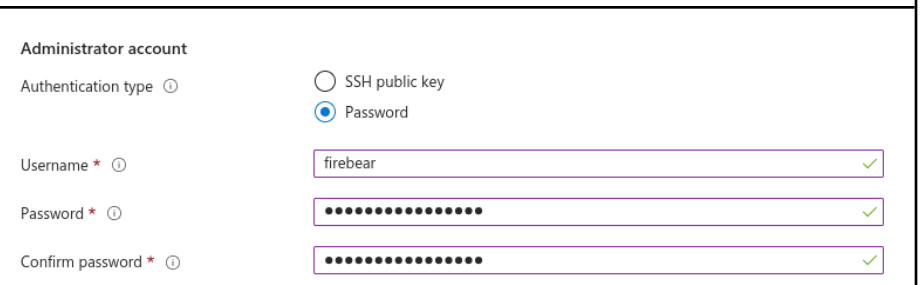
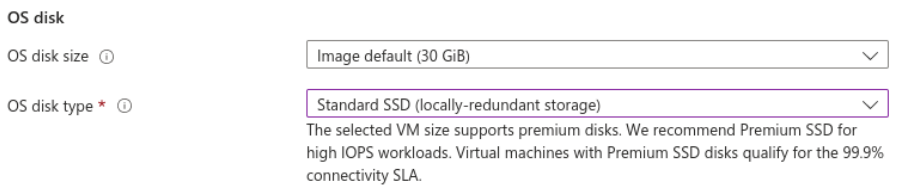
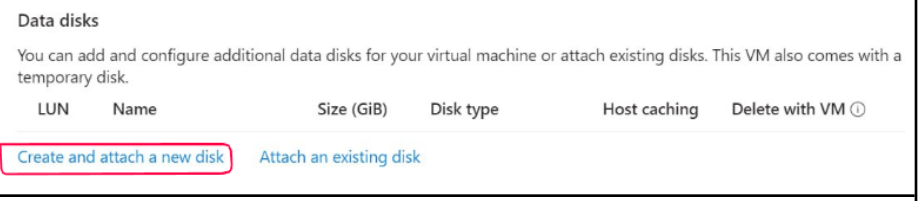
Microsoft Azure HoneyPot (T-CUP)

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

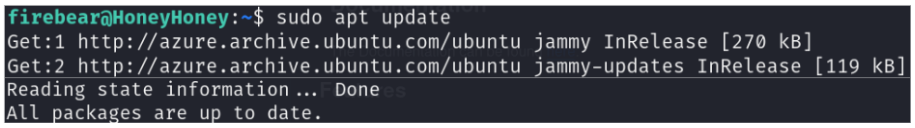
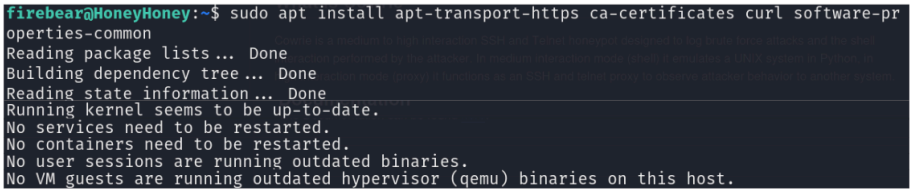
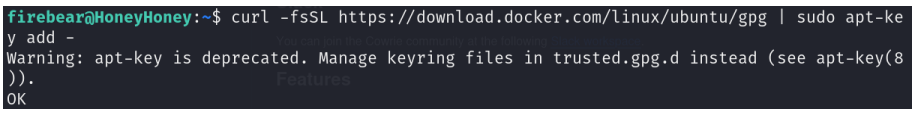
[HTTP Basic Authentication HoneyPot]
Link: https://github.com/bjeborn/basic-auth-pot

[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
```

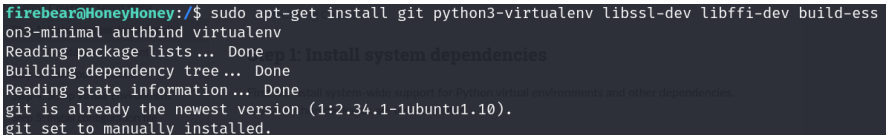
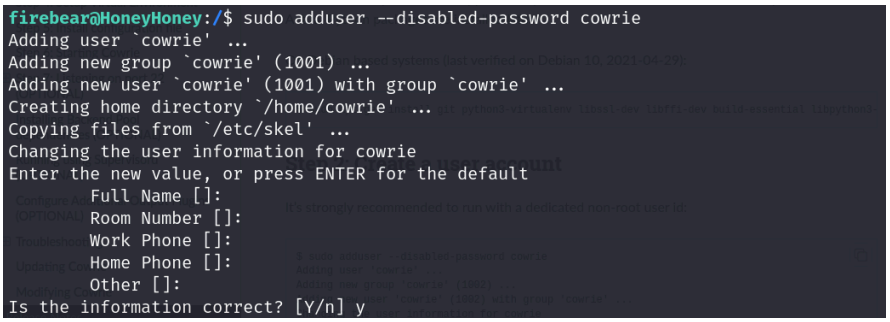
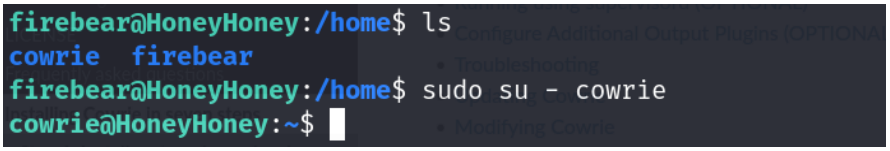
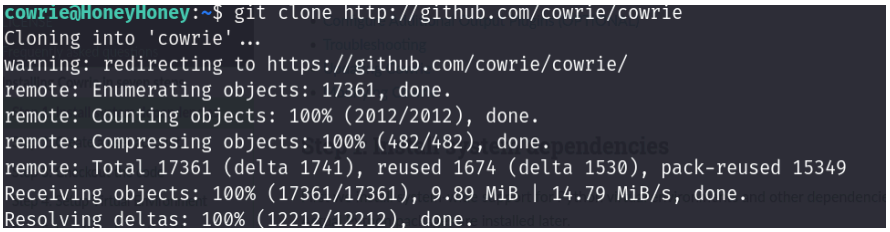
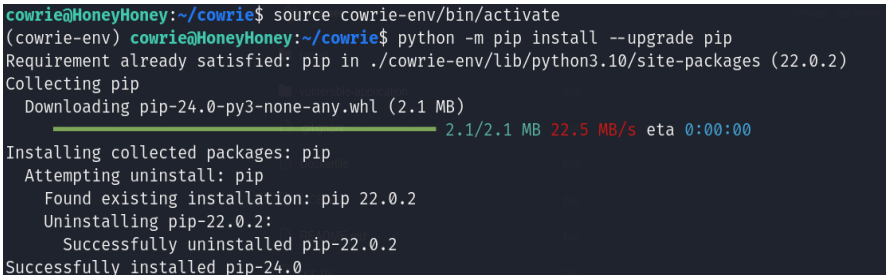
Installing Cowrie

[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

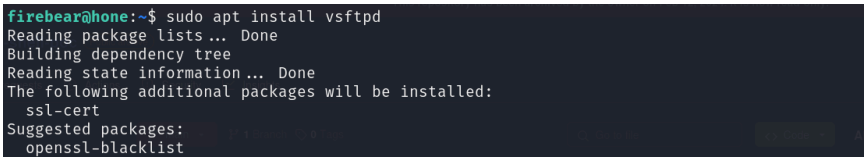
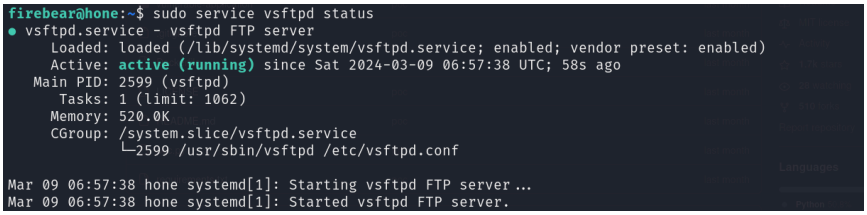
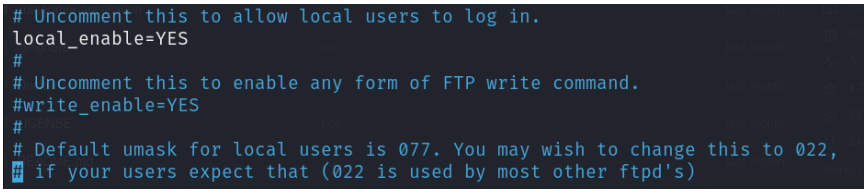
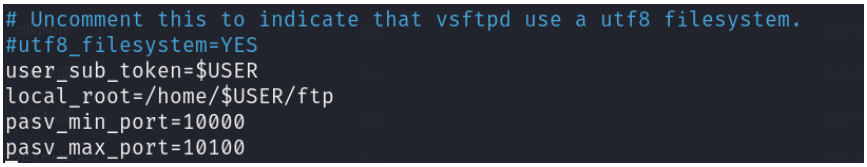
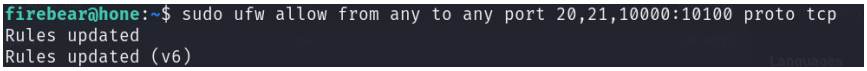
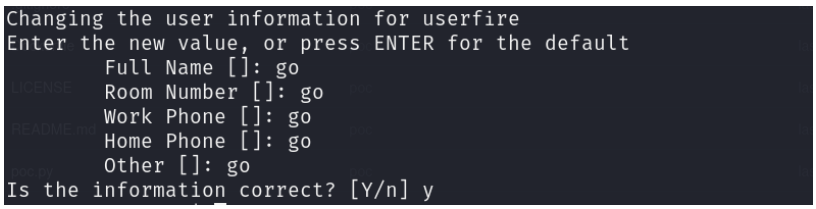
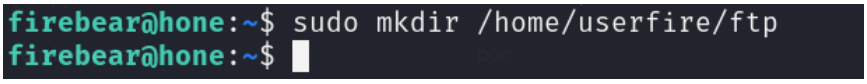
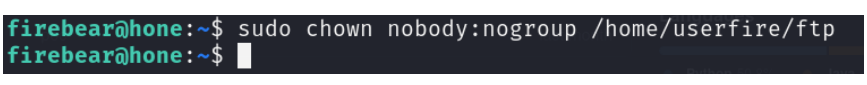
cat cowrie.log

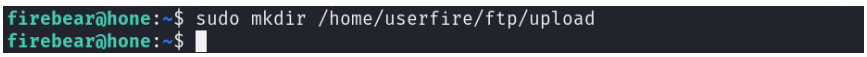
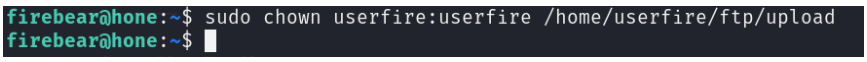
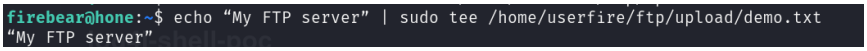
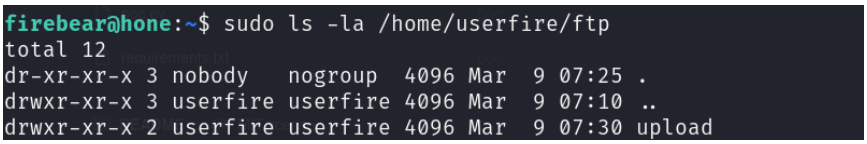
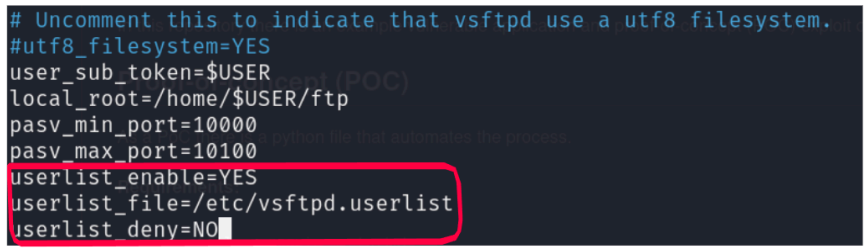
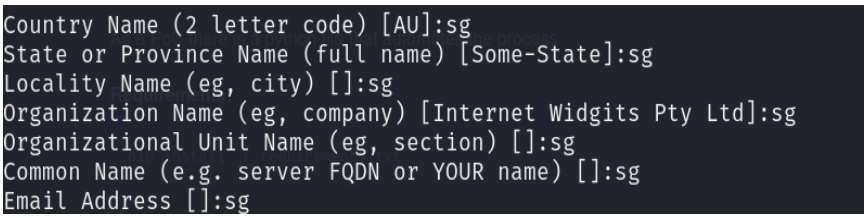
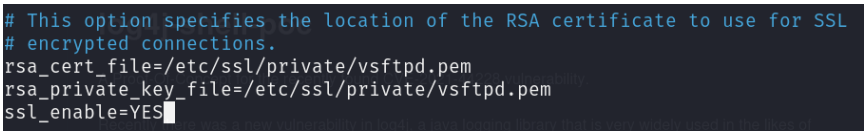
```
2024-03-09T06:24:36.832056Z [HoneyPotSSHTransport,3,116.14.113.125] Command found: cd ..
2024-03-09T06:24:37.651107Z [HoneyPotSSHTransport,3,116.14.113.125] CMD: ls
2024-03-09T06:24:37.651901Z [HoneyPotSSHTransport,3,116.14.113.125] Command found: ls
2024-03-09T06:24:40.044623Z [HoneyPotSSHTransport,3,116.14.113.125]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

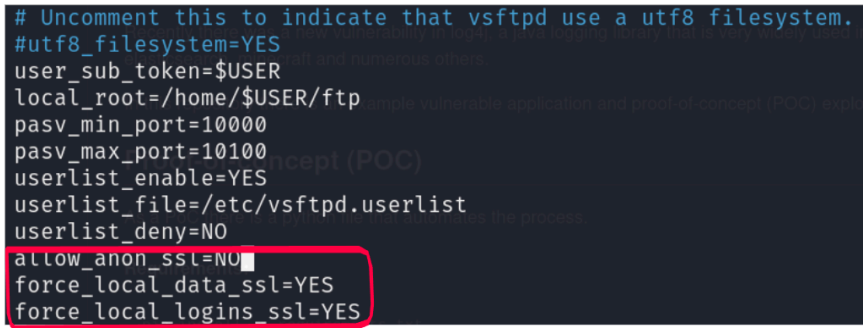
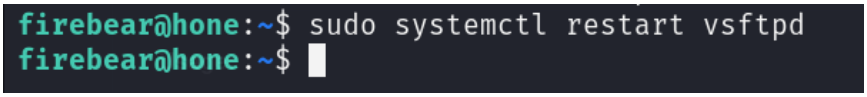
Installing VSFTPD

[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

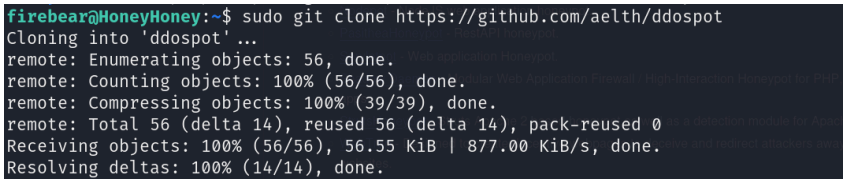
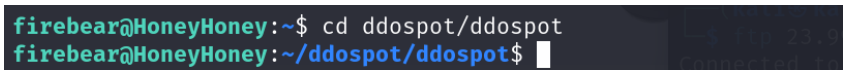
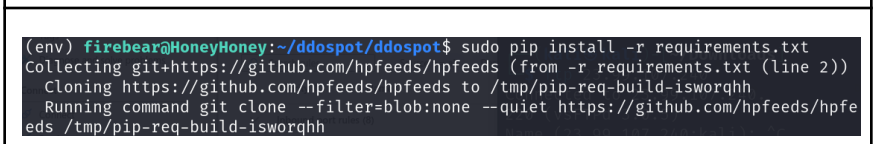
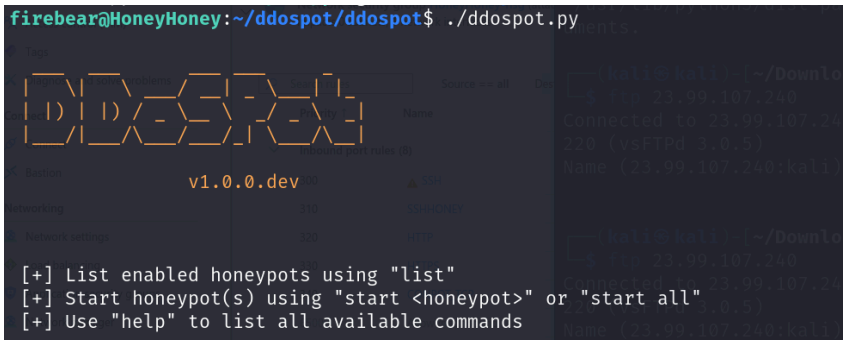
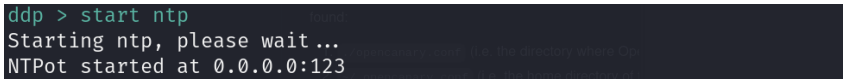
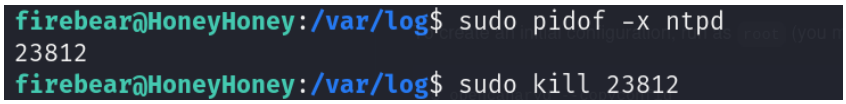
sudo ufw allow <port>/tcp
sudo ufw allow <port>/udp

```
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
```

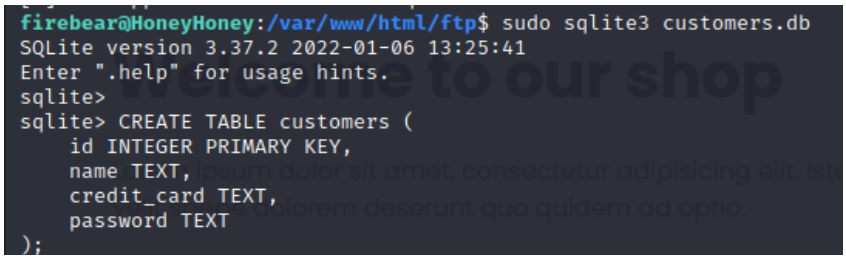
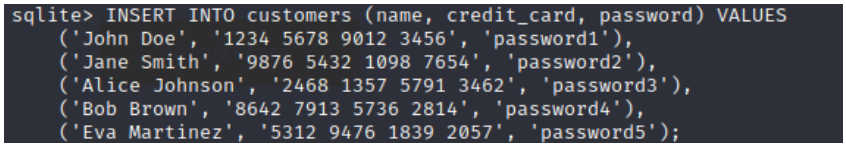
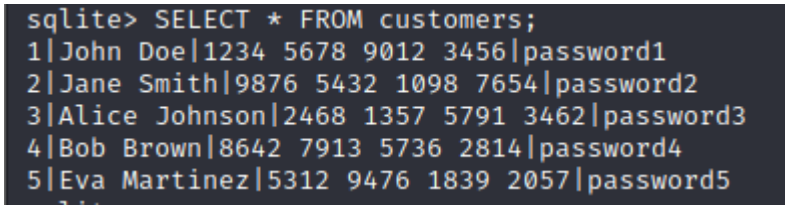
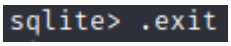
[DDoSPot POV]

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
```

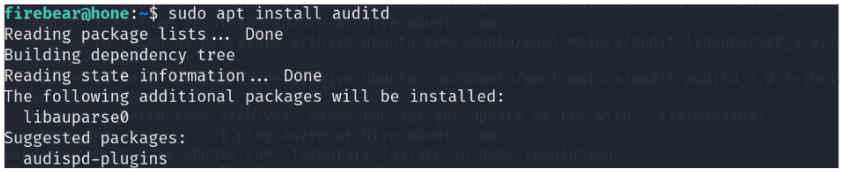
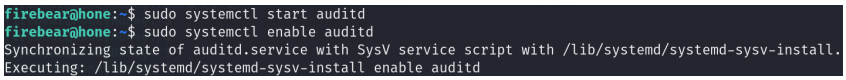
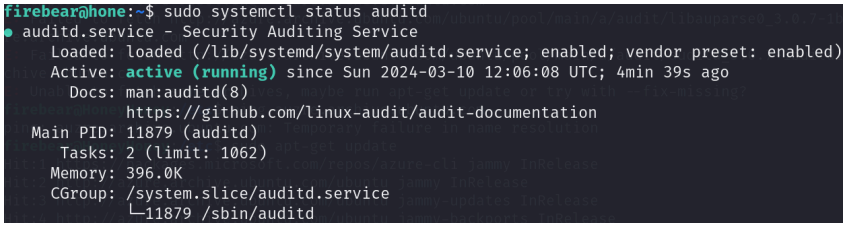
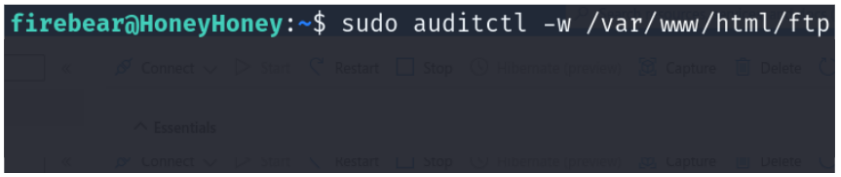
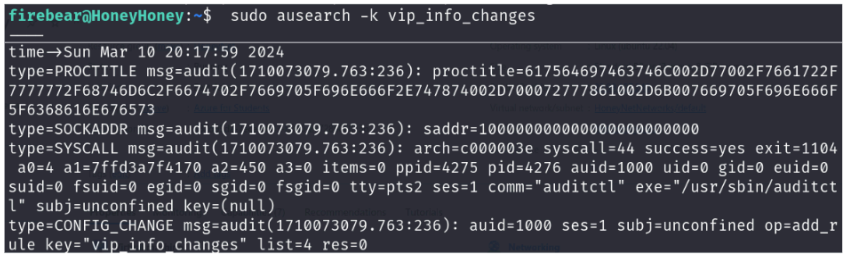
Creating a DB with fake information

Change the directory where the FTP is **cd var/www/html/ftp**	
Creating a database using sqlite3, give a random name for the data base **sudo sqlite3 customer.db**	
Inserting datas into the customer.db	
To view the data **SELECT * FROM customers;**	
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 rwxa** - 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)

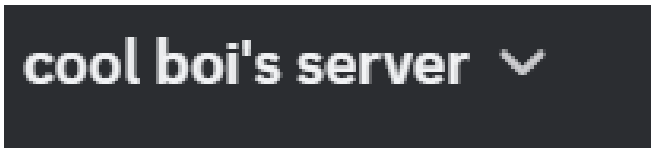
[Inspired by https://github.com/0x4D31/honeylambda]

Setting up WebHook

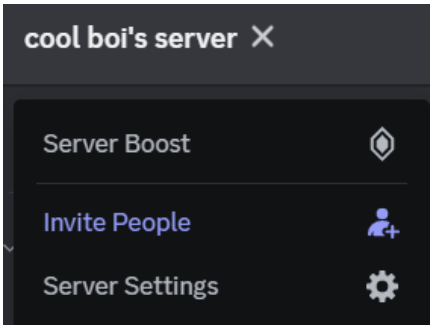
Setting up WebHook in discord.

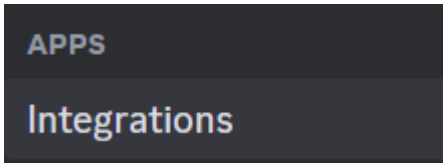
Other alternatives:
1. Slack
2. Microsoft Teams
3. Telegram Bots
4. Email
5. SMS

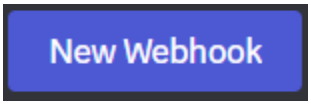
Open Discord and either create a new server or select an existing server where you have permissions to add webhooks. Choose or create a channel where the notifications will be posted.



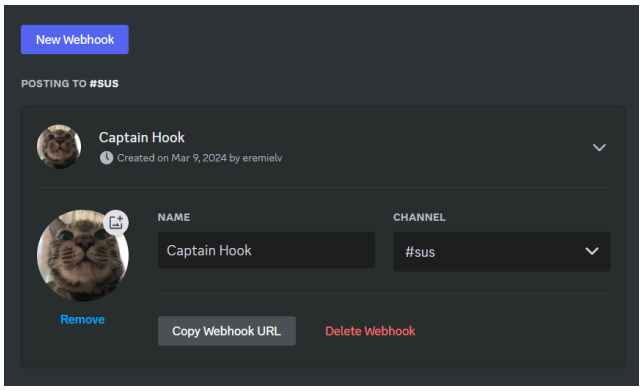
Click on Server settings > Integrations > View Webhooks > New Webhook







Set a picture and specify a channel from the server. Copy Webhook URL!



Creating a fake admin page which serves as the honeytoken.

→ Code will send brute force attempts to discord webhook.

Install php on the server to ensure php pages show up properly, and not as text.

sudo apt update

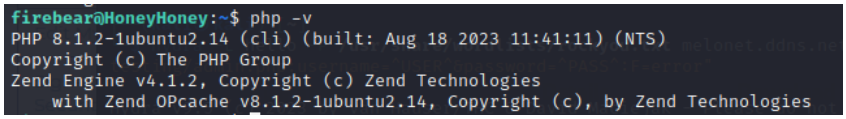
sudo apt install php libapache2-mod-php

Restart the server. [BE CAREFUL WHEN RUNNING THIS COMMAND]

sudo systemctl restart apache2

Check PHP version on server.

php -v




```php
<?php
// This is a fake admin panel page
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Assuming attackers try to post credentials
    $user = $_POST['username'] ?? 'unknown';
    $pass = $_POST['password'] ?? 'unknown';
    $ip = $_SERVER['REMOTE_ADDR'];

    $message = "Attempted access to admin panel from $ip using username: $user and password: $pass";

    // Discord webhook URL
    $webhookUrl = "https://discord.com/api/webhooks/1215952182546010132/5vjCoIUFZ-4wLJU2l6ROrxc2Yuc5pfAaWeKJmTf8ntAosFpGxY0Vxs7Jlva9R-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);

    $response = curl_exec($ch);
    // Log the response from Discord if needed
    // error_log($response);
```

```
    curl_close($ch);

    // Optionally, redirect the user to a "real" page to avoid suspicion
    header('Location: /admin/login_success.php');
    exit;
}

// Basic fake login form
?>
<!DOCTYPE html>
<html>
<head>
    <title>Admin Panel</title>
</head>
<body>
    <h1>Admin Login</h1>
//add beacons
    <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>
```

[Attackers POV]

*There is no authentication for now. This is just for demonstration.

Running a hydra command to brute force:

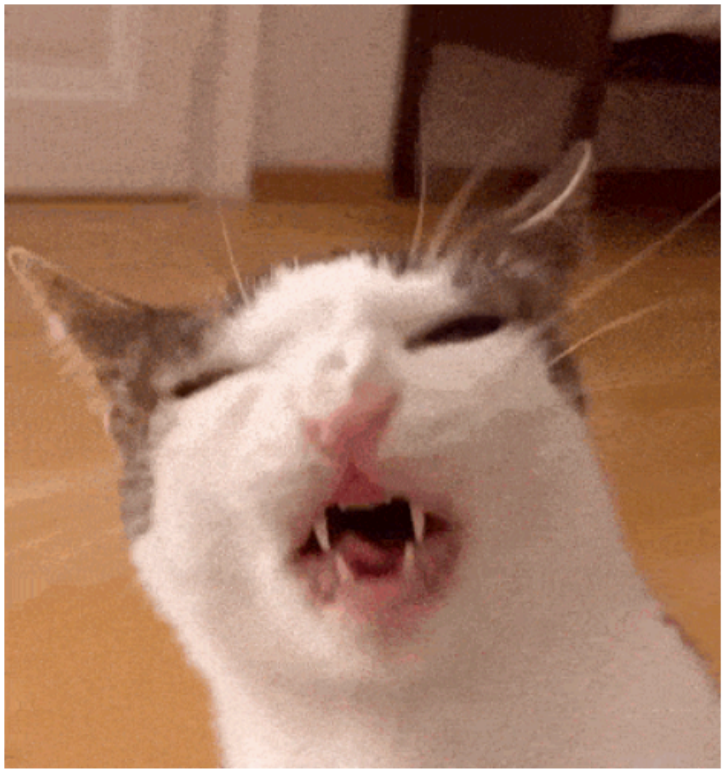
hydra -l hello -P /usr/share/wordlists/rockyou.txt melonet.ddns.net http-post-form "/admin/admin.php:username=^USER^&password=^PASS^:F=error" -s 443 -S

```
┌──(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
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: 123456
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: abc123
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: daniel
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: iloveyou
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: jessica
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: 1234567
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: 123456789
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: password
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: lovely
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: nicole
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: princess
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: 12345678
[443][http-post-form] host: melonet.ddns.net   login: lobby   password: babygirl
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-10 15:04:33
```

Fake login success page is shown. [Code not included] Page expires after 10 seconds.

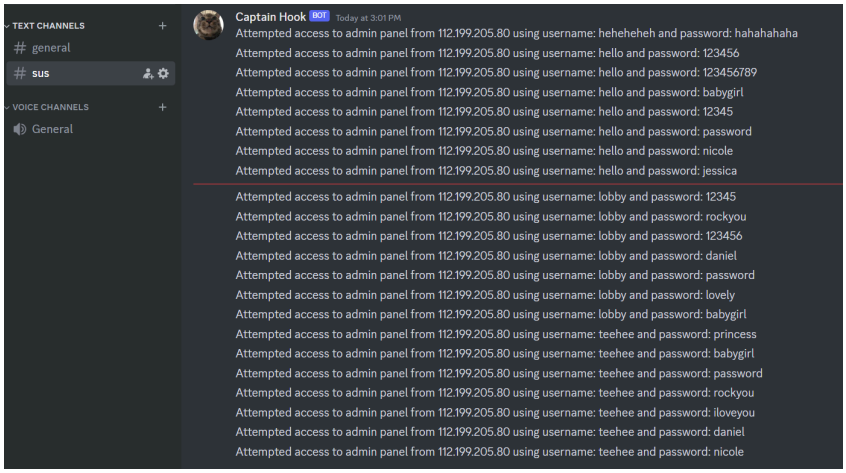
Login Successful!

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



[Discord POV]

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



Discord notifications into Logs FILE??? haven't tried

Setting up Beacon

Beacon used: 1x1 transparent pixel (Link to download)

Server time updated to SG time UTC +08 for easier logging

Upload beacon to server from terminal:

sudo scp /home/kali/Pictures/1x1.png firebear@23.99.107.240:/var/www/html/admin

```
root@HoneyHoney:/var/www/html/admin# ls -la
total 32
drwxrwxrwx 2 root     root     4096 Mar 10 14:45 .
drwxrwxrwx 7 root     root     4096 Mar 10 14:29 ..
-rwxrwxrwx 1 firebear firebear   95 Mar 10 00:40 1×1.png
```

→ Implementing client-side tracking in beacon.js.

```
// Example beacon.js content
document.addEventListener("DOMContentLoaded", function() {
   let clickCount = 0;

   // Increment click count on each click event on the page
   document.addEventListener('click', function() {
       clickCount++;
   });

   // Collect and send data when the user is about to leave the page or after a set time
   window.addEventListener('beforeunload', sendData);
   setTimeout(sendData, 5000); // Example: send data after 5 seconds

   function sendData() {
       var data = {
           screenWidth: window.screen.width,
           screenHeight: window.screen.height,
           referrer: document.referrer,
           clickCount: clickCount, // Include the click count
           // You can add more properties here as needed
       };

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

       // Prevent multiple sends
       window.removeEventListener('beforeunload', sendData);
   }
});
```

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

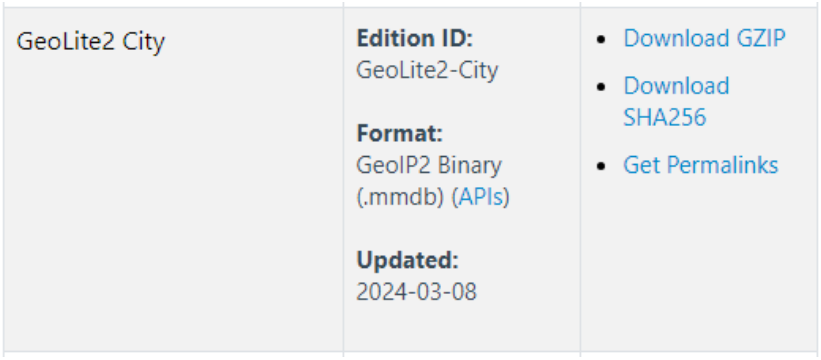
Download GeoIP (MaxMind) for GeoIP information.

- 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

```
root@HoneyHoney:/# cd /home/firebear/
root@HoneyHoney:/home/firebear# mkdir GeoIP
root@HoneyHoney:/home/firebear# ls
GeoIP    go            https          package-lock.json  server.csr  ssheametest
conpot   honeyLambda   https.pub      package.json       server.key
ddospot  honeypot-ftp  node_modules   server.crt         snap
```

Upload file from Kali Linux to server.

scp /home/kali/Downloads/GeoLite2-City_20240308.tar.gz firebear@23.99.107.240:/home/firebear/GeoIP

```
root@HoneyHoney:/home/firebear# cd GeoIP
root@HoneyHoney:/home/firebear/GeoIP# ls
root@HoneyHoney:/home/firebear/GeoIP# ls
GeoLite2-City_20240308.tar.gz
```

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

```
root@HoneyHoney:/home/firebear/GeoIP# tar -xvf GeoLite2-City_20240308.tar.gz
GeoLite2-City_20240308/
GeoLite2-City_20240308/LICENSE.txt
GeoLite2-City_20240308/COPYRIGHT.txt
GeoLite2-City_20240308/GeoLite2-City.mmdb
GeoLite2-City_20240308/README.txt
root@HoneyHoney:/home/firebear/GeoIP# ls
GeoLite2-City_20240308  GeoLite2-City_20240308.tar.gz
root@HoneyHoney:/home/firebear/GeoIP# cd GeoLite2-City_20240308
root@HoneyHoney:/home/firebear/GeoIP/GeoLite2-City_20240308# ls
COPYRIGHT.txt  GeoLite2-City.mmdb  LICENSE.txt  README.txt
```

```
firebear@HoneyHoney:/var/www/html/admin$ ls /home/firebear/GeoIP/GeoLite2-City_20240308/GeoLite2-City.mmdb
/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');"

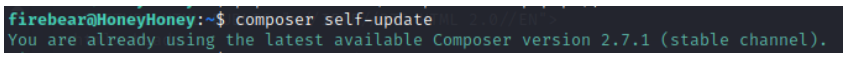
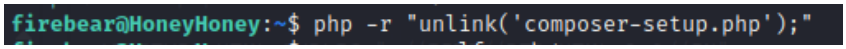
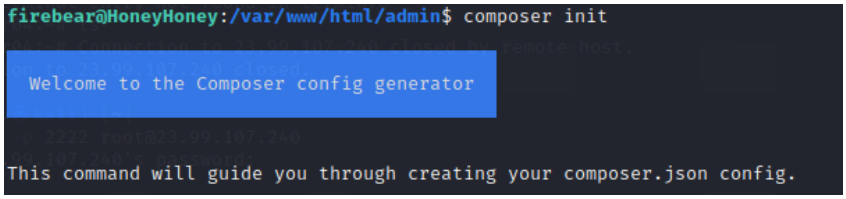
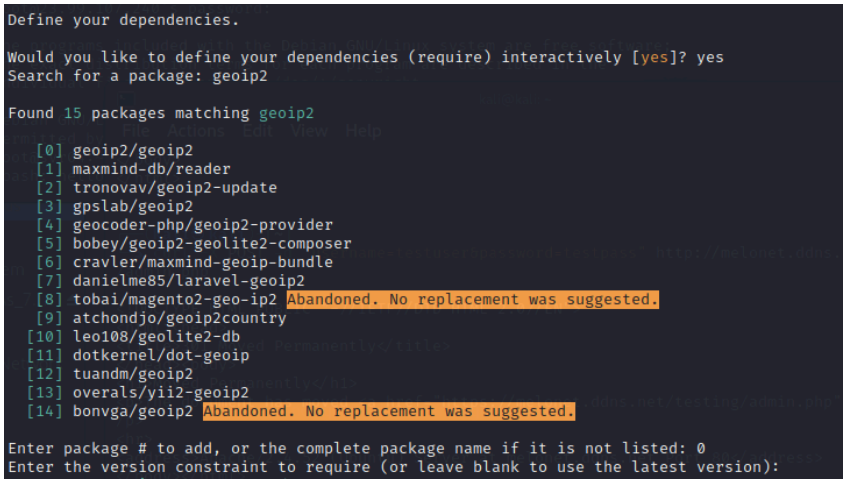
sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer

```
root@HoneyHoney:/home/firebear# php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
root@HoneyHoney:/home/firebear# sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
All settings correct for using Composer
Downloading ...

Composer (version 2.7.1) successfully installed to: /usr/local/bin/composer
Use it: php /usr/local/bin/composer
```

Check if Composer installed. (Check as non-root user)

composer --version

```
root@HoneyHoney:/home/firebear# su firebear
firebear@HoneyHoney:~$ composer --version
Composer version 2.7.1 2024-02-09 15:26:28
```

Check for the latest version. **composer self-update**	
Remove Composer installer script. **php -r "unlink('composer-setup.php');"**	
Now we can proceed with GeoLite2 setup. Set up a composer.json file. **composer init**	
Set up with necessary details. For me, I only specified geoip2 as my dependency. I left the rest blank.	 

```
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
```

Set necessary file permissions. This part is crucial.

sudo chown www-data:www-data /home/firebear/GeoIP/GeoLite2-City_20240308/GeoLite2-City.mmdb

sudo chmod 644 /home/firebear/GeoIP/GeoLite2-City_20240308/GeoLite2-City.mmdb

sudo chmod 755 /home/firebear/

sudo chmod 755 /home/firebear/GeoIP/

sudo chmod 755 /home/firebear/GeoIP/GeoLite2-City_20240308/

```
firebear@HoneyHoney:/var/www/html/admin$ sudo chown www-data:www-data /home/firebear/GeoI
P/GeoLite2-City_20240308/GeoLite2-City.mmdb
firebear@HoneyHoney:/var/www/html/admin$ sudo chmod 644 /home/firebear/GeoIP/GeoLite2-Cit
y_20240308/GeoLite2-City.mmdb
firebear@HoneyHoney:/var/www/html/admin$ sudo chmod 755 /home/firebear/
firebear@HoneyHoney:/var/www/html/admin$ sudo chmod 755 /home/firebear/GeoIP/
firebear@HoneyHoney:/var/www/html/admin$ sudo chmod 755 /home/firebear/GeoIP/GeoLite2-Cit
y_20240308/
```

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

// Path to your GeoLite2 database
$dbPath = '/home/firebear/GeoIP/GeoLite2-City_20240308/GeoLite2-City.mmdb';

// Initialize variables to default values
$ip = $_SERVER['REMOTE_ADDR'];
$userAgent = $_SERVER['HTTP_USER_AGENT'];
$referrer = $_SERVER['HTTP_REFERER'] ?? 'Direct or no referrer';
```

```
$dateTime = date('Y-m-d H:i:s');
$additionalInfo = '';

// Perform GeoIP lookup
try {
    $reader = new Reader($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';

    // Append GeoIP info to the additional info string
    $additionalInfo .= ", Country: $country, City: $city, Latitude: $latitude, Longitude:
$longitude";
} catch (\Exception $e) {
    // Handle exceptions, e.g., database file not found or IP address not in database
    $additionalInfo .= ", GeoIP Lookup Failed: " . $e->getMessage();
}

// Existing logic for handling JSON payload from beacon.js
if (isset($_SERVER['CONTENT_TYPE']) && $_SERVER['CONTENT_TYPE'] === 'application/json') {
    $jsonPayload = file_get_contents('php://input');
    $data = json_decode($jsonPayload, true);
    $clickCount = $data['clickCount'] ?? 'unknown';
    $additionalInfo .= ", Clicks: $clickCount";
}

// Log file
$logFile = 'access_log.txt';
$logMessage = "$dateTime, $ip, $referrer, $userAgent$additionalInfo\n";
file_put_contents($logFile, $logMessage, FILE_APPEND);

// Serve the 1x1 pixel image for both direct and JS beacon requests
header('Content-Type: image/png');
$pixel = '1x1.png';
echo file_get_contents($pixel);
exit;
```

Embedding beacon in fake admin page: admin.php.

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

```
<?php
// This is a fake admin panel page
// [COPY FROM ABOVE: admin.php]

// Basic fake login form
?>
<!DOCTYPE html>
<html>
<head>
    <title>Admin Panel</title>
</head>
<body>
    <h1>Admin Login</h1>
<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 the right permissions.**

```
firebear@HoneyHoney:/var/www/html/admin$ ls -la
total 68
drwxrwxrwx 4 root     root      4096 Mar 11 02:30 .
drwxrwxrwx 9 root     root      4096 Mar 11 13:24 ..
-rwxrwxrwx 1 firebear firebear    95 Mar 10 00:40 1×1.png
-rwxrwxrwx 1 www-data www-data 13151 Mar 11 17:03 access_log.txt
-rwxrwxrwx 1 root     root      1696 Mar 11 02:05 admin.php
-rwxrwxrwx 1 firebear firebear  1101 Mar 10 14:35 beacon.js
-rw-rw-r-- 1 firebear firebear  1772 Mar 11 02:30 beacon.php
-rwxrwxrwx 1 firebear firebear   183 Mar 11 01:54 composer.json
-rwxrwxrwx 1 firebear firebear  9808 Mar 11 01:23 composer.lock
-rwxrwxrwx 1 root     root       571 Mar 10 01:13 login_success.php
drwxrwxrwx 2 firebear firebear  4096 Mar 11 00:21 src
-rw-rw-r-- 1 firebear firebear     0 Mar 11 01:58 test.php
drwxrwxrwx 7 firebear firebear  4096 Mar 11 01:23 vendor
```

[Attackers POV]

Beacon is unseen.
User is tracked the moment the page is accessed, updated every 5 seconds.

Actions/ Items tracked:

- Date and Time
- IP
- User-Agent
- Referer
- GeoIP (Country, City, Latitude, Longitude)
- Click Count
- ISP

[Logs in access_log.txt collected from Beacons POV]

Information from both beacon.js and beacon.php are logged.

```
2024-03-11 17:03:29, 112.199.205.80, https://melonet.ddns.net/admin/admin.php, Mozilla/5.
0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 S
afari/537.36, Country: Singapore, City: Singapore, Latitude: 1.3596, Longitude: 103.8637,
 Clicks: 17
```

Attack Map

Link:https://github.com/cecirio/azure-sentinel-attack-map [Not using]

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

sudo sqlite3 attack_map.db

Create a table in the SQLite prompt.

CREATE TABLE attack_map (
id INTEGER PRIMARY KEY,
ip TEXT,
country TEXT,
city TEXT,
latitude REAL,
longitude REAL,
dateTime TEXT,
userAgent TEXT,
referrer TEXT,
clicks INTEGER
);

```
firebear@HoneyHoney:/var/www/html/admin$ sqlite3 attack_map.db
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
sqlite> CREATE TABLE access_log (
    id INTEGER PRIMARY KEY,
    ip TEXT,
    country TEXT,
    city TEXT,
    latitude REAL,
    longitude REAL,
    dateTime TEXT,
    userAgent TEXT,
    referrer TEXT,
    clicks INTEGER
);
sqlite> .quit
```

Install php-sqlite3

sudo apt-get install php-sqlite3

```
firebear@HoneyHoney:/var/www/html/admin$ sudo apt-get install php-sqlite3
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  php8.1-sqlite3
The following NEW packages will be installed:
  php-sqlite3 php8.1-sqlite3
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 34.0 kB of archives.
After this operation, 154 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-sqlite3 amd64
 8.1.2-1ubuntu2.14 [32.2 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu jammy/main amd64 php-sqlite3 all 2:8.1+92ubun
tu1 [1840 B]
Fetched 34.0 kB in 1s (37.1 kB/s)
Selecting previously unselected package php8.1-sqlite3.
(Reading database ... 149160 files and directories currently installed.)
Preparing to unpack .../php8.1-sqlite3_8.1.2-1ubuntu2.14_amd64.deb ...
Unpacking php8.1-sqlite3 (8.1.2-1ubuntu2.14) ...
Selecting previously unselected package php-sqlite3.
Preparing to unpack .../php-sqlite3_2%3a8.1+92ubuntu1_all.deb ...
Unpacking php-sqlite3 (2:8.1+92ubuntu1) ...
Setting up php8.1-sqlite3 (8.1.2-1ubuntu2.14) ...

Creating config file /etc/php/8.1/mods-available/sqlite3.ini with new version

Creating config file /etc/php/8.1/mods-available/pdo_sqlite.ini with new version
Setting up php-sqlite3 (2:8.1+92ubuntu1) ...
Processing triggers for libapache2-mod-php8.1 (8.1.2-1ubuntu2.14) ...
Processing triggers for php8.1-cli (8.1.2-1ubuntu2.14) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Check if database is populated (after triggering /admin/admin.php)

sqlite3 attack_map.db

Display all entries in the SQLite prompt.

SELECT * FROM attack_map;

```
firebear@HoneyHoney:/var/www/html/admin$ sqlite3 attack_map.db
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
sqlite> SELECT * FROM attack_map;
1|112.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:13|Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/53
7.36|https://melonet.ddns.net/admin/admin.php|
2|112.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:13|Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/53
7.36|https://melonet.ddns.net/admin/admin.php|
3|112.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:14|Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/53
7.36|https://melonet.ddns.net/admin/admin.php|
4|112.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:14|Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/53
7.36|https://melonet.ddns.net/admin/admin.php|
5|112.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:14|Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/53
7.36|https://melonet.ddns.net/admin/admin.php|
6|112.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:14|Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/53
7.36|https://melonet.ddns.net/admin/admin.php|
7|112.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:19|Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/53
7.36|https://melonet.ddns.net/admin/admin.php|
8|112.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:19|Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/53
7.36|https://melonet.ddns.net/admin/admin.php|
sqlite> .quit
```

Create an attack_map.html. this displays the map.

```
<!DOCTYPE html>
<html>
<head>
   <title>Live Attack Map</title>
   <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"/>
   <style>
       #map {
           height: 100vh;
       }
   </style>
</head>
<body>
   <div id="map"></div>
   <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
   <script src="attack_map.js"></script> <!-- Your custom JS file -->
</body>
</html>
```

Create attack_map_sse.php. This sends updates using the SSE protocol. SSE allows the server to push updates to the client whenever new data is available.

```
<?php
header('Content-Type: text/event-stream');
header('Cache-Control: no-cache');

$dbFile = __DIR__ . '/attack_map.db';
$db = new PDO("sqlite:$dbFile");

// Fetch the latest entry or entries from the database
$query = "SELECT * FROM attack_map ORDER BY dateTime DESC LIMIT 5"; // Adjust LIMIT based on your
needs
$result = $db->query($query);
$data = $result->fetchAll(PDO::FETCH_ASSOC);

// Send data as a formatted event
echo "data: " . json_encode($data) . "\n\n";
flush();
```

Create attack_map.js. This script will listen for data from your SSE PHP script and update the map accordingly.

```
//var map = L.map('map').setView([1.3521, 103.8198], 11); // Singapore POV
var map = L.map('map').setView([0, 0], 2); // Regular map
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', { maxZoom: 19, }).addTo(map);

// Initialize an EventSource
var source = new EventSource('attack_map_sse.php');

source.onmessage = function(event) {
   // Parse the incoming data
   var data = JSON.parse(event.data);

   data.forEach(item => {
       L.marker([item.latitude, item.longitude]).addTo(map)
           .bindPopup(`IP: ${item.ip}<br>Country: ${item.country}<br>City: ${item.city}<br>Date
and Time: ${item.dateTime}`);
   });
};
```

Ensure the file directory looks like this.

```
firebear@HoneyHoney:/var/www/html/admin$ ls -la
total 100
drwxrwxrwx 4 root     root      4096 Mar 12 03:24 .
drwxrwxrwx 9 root     root      4096 Mar 11 13:24 ..
-rwxrwxrwx 1 firebear firebear    95 Mar 10 00:40 1×1.png
-rw-r--r-- 1 www-data www-data 11171 Mar 12 03:24 access_log.txt
-rwxrwxrwx 1 root     root      1696 Mar 11 02:05 admin.php
-rw-rw-r-- 1 www-data www-data 24576 Mar 12 03:24 attack_map.db
-rw-rw-r-- 1 firebear firebear   431 Mar 12 02:05 attack_map.html
-rw-rw-r-- 1 firebear firebear   647 Mar 12 02:48 attack_map.js
-rw-rw-r-- 1 firebear firebear   480 Mar 12 02:41 attack_map_sse.php
-rwxrwxrwx 1 firebear firebear  1101 Mar 10 14:35 beacon.js
-rw-rw-r-- 1 firebear firebear  2429 Mar 12 01:15 beacon.php
-rwxrwxrwx 1 firebear firebear   183 Mar 11 01:54 composer.json
-rwxrwxrwx 1 firebear firebear  9808 Mar 11 01:23 composer.lock
-rwxrwxrwx 1 root     root       571 Mar 10 01:13 login_success.php
drwxrwxrwx 2 firebear firebear  4096 Mar 11 00:21 src
drwxrwxrwx 7 firebear firebear  4096 Mar 11 01:23 vendor
```

[Attack Map POV]

Map is live! :-)) https://melonet.ddns.net/admin/attack_map.html