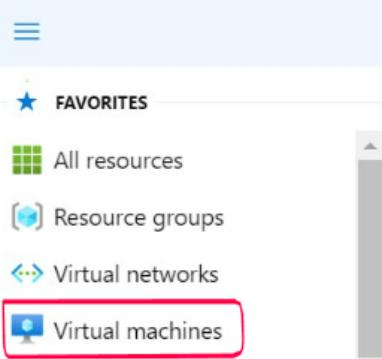
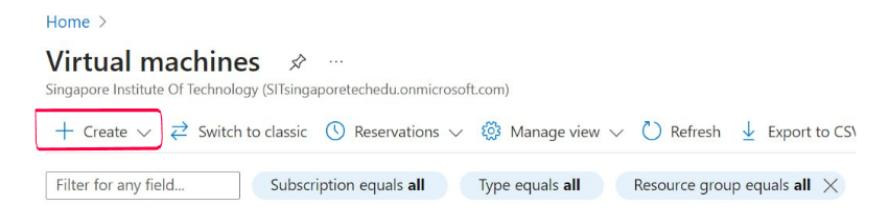
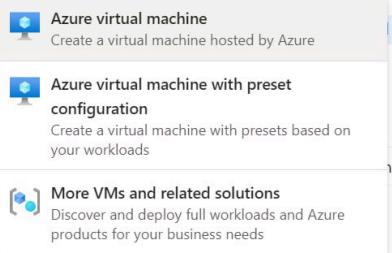
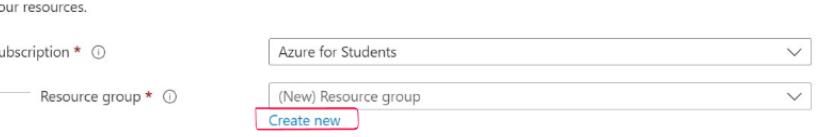


# Microsoft Azure HoneyPot (T-Pot)

## Creating a VM

Head to the microsoft azure webpage to create a Virtual Machine

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

Once you've logged in, navigate to the sidebar and locate “Virtual Machine”	
	
Click on create → Azure Virtual Machine	
<b>[BASICS] Tabs</b>	
Click on Create new	

Give any name you want	<p>A resource group is a container that holds related resources for an Azure solution.</p> <p>Name *</p> <input type="text"/> <p><b>OK</b> <b>Cancel</b></p>												
<p>Select</p> <p>Name: &lt;Name your virtual machine&gt;</p> <p>Region: (Asia Pacific) East Asia</p> <p>Avail Option: No infrastructure Redundancy Required</p> <p>Security Type: Standard</p> <p>Image: Debian 11 ‘Bullseye’ x64 Gen2</p>	<p><b>Instance details</b></p> <p>Virtual machine name * ⓘ <input type="text"/></p> <p>Region * ⓘ (Asia Pacific) East Asia <input type="text"/></p> <p>Availability options ⓘ No infrastructure redundancy required <input type="text"/></p> <p>Security type ⓘ Standard <input type="text"/></p> <p>Image * ⓘ Debian 11 “Bullseye” - x64 Gen2 <input type="text"/></p> <p><a href="#">See all images</a>   <a href="#">Configure VM generation</a></p> <p><input checked="" type="checkbox"/> This image is compatible with additional security features. <a href="#">Click here to swap to the Trusted launch security type</a>.</p>												
<p>Select the size</p> <p>Standard_D2s_V3 - 2vcpus, 8 GiB memory</p>	<p>Run with Azure Spot discount ⓘ <input type="checkbox"/></p> <p>Size * ⓘ Standard_D2s_v3 - 2 vcpus, 8 GiB memory (US\$96.36/month) <input type="text"/></p> <p><a href="#">See all sizes</a></p> <p>Enable Hibernation (preview) ⓘ <input type="checkbox"/></p> <p><small>To enable Hibernation, you must register your subscription. <a href="#">Learn more</a></small></p>												
<p>Select Authentication type: Password (For SSH &amp; Cockpit)</p> <p>Create a username &amp; password</p>	<p>Authentication type ⓘ <input checked="" type="radio"/> Password <input type="radio"/> SSH public key</p> <p>Username * ⓘ hello <input type="text"/></p> <p>Password * ⓘ ..... <input type="text"/></p> <p>Confirm password * ⓘ ..... <input type="text"/></p>												
<b>[DISK] Tabs</b>	<p><b>OS disk</b></p> <p>OS disk size ⓘ Image default (30 GiB) <input type="text"/></p> <p>OS disk type * ⓘ Standard HDD (locally-redundant storage) <input type="text"/></p> <p>The selected VM size supports premium disks. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.</p> <p><b>Data disks</b></p> <p>You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.</p> <table border="1"> <thead> <tr> <th>LUN</th> <th>Name</th> <th>Size (GiB)</th> <th>Disk type</th> <th>Host caching</th> <th>Delete with VM ⓘ</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p><a href="#">Create and attach a new disk</a> <a href="#">Attach an existing disk</a></p> <p>Size * ⓘ <b>128 GiB</b> <small>Premium SSD LRS</small> <a href="#">Change size</a></p>	LUN	Name	Size (GiB)	Disk type	Host caching	Delete with VM ⓘ						
LUN	Name	Size (GiB)	Disk type	Host caching	Delete with VM ⓘ								

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

## Networking

 Network settings

 Load balancing

Click on **Create Port Rule**

Network security group **HONEHNOEH-nsg** (attached to networkinterface: honehnoeh167)  
Impacts 0 subnets, 1 network interfaces

+ Create port rule

Priority ↑	Name	Port	Protocol	Source	Destination
------------	------	------	----------	--------	-------------

### Source ⓘ

Any

### Source port ranges \* ⓘ

\*

### Destination ⓘ

Any

### Service ⓘ

Custom

### Destination port ranges \* ⓘ

\*

### Protocol

- Any
- TCP
- UDP
- ICMP

### Action

- Allow
- Deny

### Priority \* ⓘ

310



### Name

AllowAnyCustomAnyInbound

### Description

Save

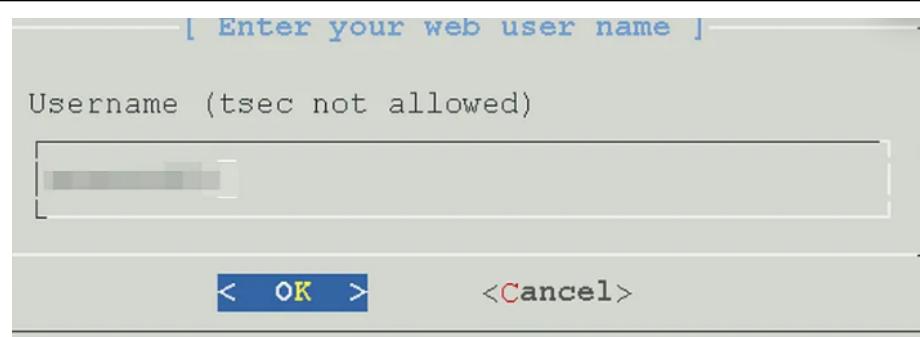
Cancel

 Give feedback

## On Kali-Linux

SSH to your public ip  <b>ssh &lt;usr&gt;@&lt;public-ip&gt;</b>	<pre>(kali㉿kali)-[~] \$ ssh firebear@23.97.76.245 The authenticity of host '23.97.76.245 (23.97.76.245)' can't be established.</pre>																
Update and Upgrade the Debian  <b>sudo apt update &amp;&amp; sudo apt upgrade -y</b>	<pre>firebear@HONEHNOE:~\$ sudo apt update &amp;&amp; sudo apt upgrade -y Hit:1 http://debian-archive.trafficmanager.net/debian bullseye InRelease      you are ab Hit:2 http://debian-archive.trafficmanager.net/debian-security bullseye-security InRel</pre>																
Install Git  <b>sudo apt install git</b>	<pre>firebear@HONEHNOE:~\$ sudo apt install git Reading package lists... Done Building dependency tree... Done</pre>																
Install TPOT  <b>git clone https://github.com/telekom-security/tpotce</b>	<pre>root@HONEH2:/home/firebear# git clone https://github.com/telekom-security/tpotce Cloning into 'tpotce'... remote: Enumerating objects: 15421, done. remote: Counting objects: 100% (555/555), done. remote: Compressing objects: 100% (356/356), done. Receiving objects: 12% (1851/15421)</pre>																
Change to the directory  <b>cd tpotce/iso/installer</b>	<pre>root@HONEH2:/home/firebear# cd tpotce/iso/installer root@HONEH2:/home/firebear/tpotce/iso/installer#</pre>																
It will show you the active Internet Connections	<pre>Active Internet connections (only servers) Proto Recv-Q Send-Q Local Address           Foreign Address         State      User   NetDev node  PID/Program name tcp        0      0 0.0.0.0:22              0.0.0.0:*          LISTEN    0      1 1070    745/sshd: /usr/sbin tcp6       0      0 ::1:22                  ::*:*            LISTEN    0      1 1072    745/sshd: /usr/sbin tcp6       0      0 0.0.0.0:68              0.0.0.0:*          LISTEN    0      1 1782    459/dhclient: dhclient udp        0      0 127.0.0.1:323             0.0.0.0:*          LISTEN    0      1 0963    691/chronyd udp6       0      0 ::1:323                ::*:*            LISTEN    0      1 0964    691/chronyd udp6       0      0 fe80::20d:3aff:fe86:546 ::*:*          LISTEN    0      1 1808    539/dhclient</pre>																
Execute the install.sh  <b>sudo ./install.sh --type=user</b>	<pre>root@HONEH2:/home/firebear/tpotce/iso/installer# ./install.sh --type=user T-Pot will require the following ports for incoming + outgoing connections. Review the T-Pot Editions for a visual representation. Also some ports might collide with T-Pot's honeypots and prevent T-Pot from starting successfully.  ### Checking for root: [ OK ] ### Installing apt-fast Hit:1 http://debian-archive.trafficmanager.net/debian bullseye InRelease Hit:2 http://debian-archive.trafficmanager.net/debian-security bullseye-security InRelease Hit:3 http://debian-archive.trafficmanager.net/debian bullseye-updates InRelease Hit:4 http://debian-archive.trafficmanager.net/debian bullseye-backports InRelease ### might collide with T-Pot's honeypots and prevent T-Pot from starting successfully.  Continue [y/n]? </pre>																
Select “Standard”	<p>[ Choose Your T-Pot Edition ]</p> <p>Required: 8-16GB RAM, 128GB SSD Recommended: 16GB RAM, 256GB SSD</p> <table border="1"> <tbody> <tr> <td><b>STANDARD</b></td> <td><b>T-Pot Standalone with everything you need</b></td> </tr> <tr> <td><b>HIVE</b></td> <td>T-Pot Hive: ELK &amp; Tools</td> </tr> <tr> <td><b>HIVE SENSOR</b></td> <td>T-Pot Hive Sensor: Honeypots &amp; NSM</td> </tr> <tr> <td><b>INDUSTRIAL</b></td> <td>Same as Standard with focus on Compot</td> </tr> <tr> <td><b>LOG4J</b></td> <td>Log4Pot, ELK, NSM &amp; Tools</td> </tr> <tr> <td><b>MEDICAL</b></td> <td>Dicompot, Medpot, ELK, NSM &amp; Tools</td> </tr> <tr> <td><b>MINI</b></td> <td>Same as Standard with focus on qHoneypots</td> </tr> <tr> <td><b>SENSOR</b></td> <td>Just Honeypots &amp; NSM</td> </tr> </tbody> </table> <p>&lt; <b>OK</b> &gt;</p>	<b>STANDARD</b>	<b>T-Pot Standalone with everything you need</b>	<b>HIVE</b>	T-Pot Hive: ELK & Tools	<b>HIVE SENSOR</b>	T-Pot Hive Sensor: Honeypots & NSM	<b>INDUSTRIAL</b>	Same as Standard with focus on Compot	<b>LOG4J</b>	Log4Pot, ELK, NSM & Tools	<b>MEDICAL</b>	Dicompot, Medpot, ELK, NSM & Tools	<b>MINI</b>	Same as Standard with focus on qHoneypots	<b>SENSOR</b>	Just Honeypots & NSM
<b>STANDARD</b>	<b>T-Pot Standalone with everything you need</b>																
<b>HIVE</b>	T-Pot Hive: ELK & Tools																
<b>HIVE SENSOR</b>	T-Pot Hive Sensor: Honeypots & NSM																
<b>INDUSTRIAL</b>	Same as Standard with focus on Compot																
<b>LOG4J</b>	Log4Pot, ELK, NSM & Tools																
<b>MEDICAL</b>	Dicompot, Medpot, ELK, NSM & Tools																
<b>MINI</b>	Same as Standard with focus on qHoneypots																
<b>SENSOR</b>	Just Honeypots & NSM																

## Create Username and Password

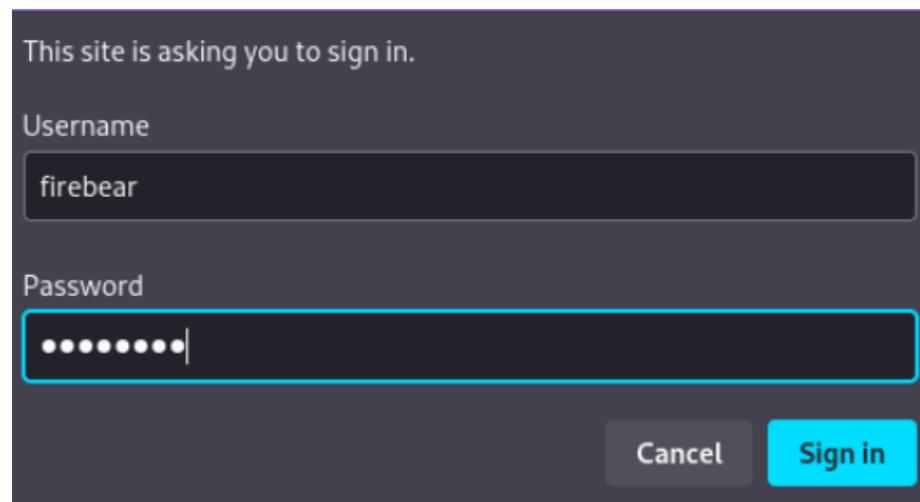


Once done installing, you will be kicked out from port 22, its normal

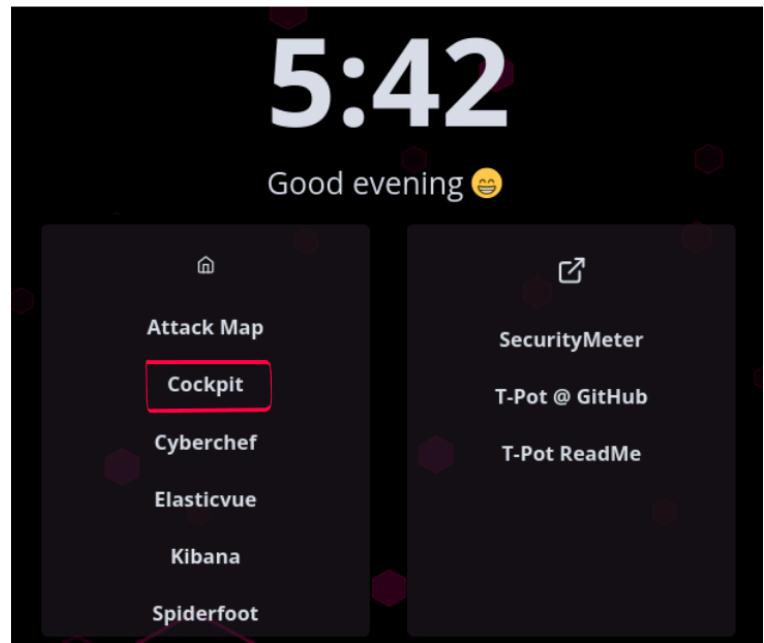


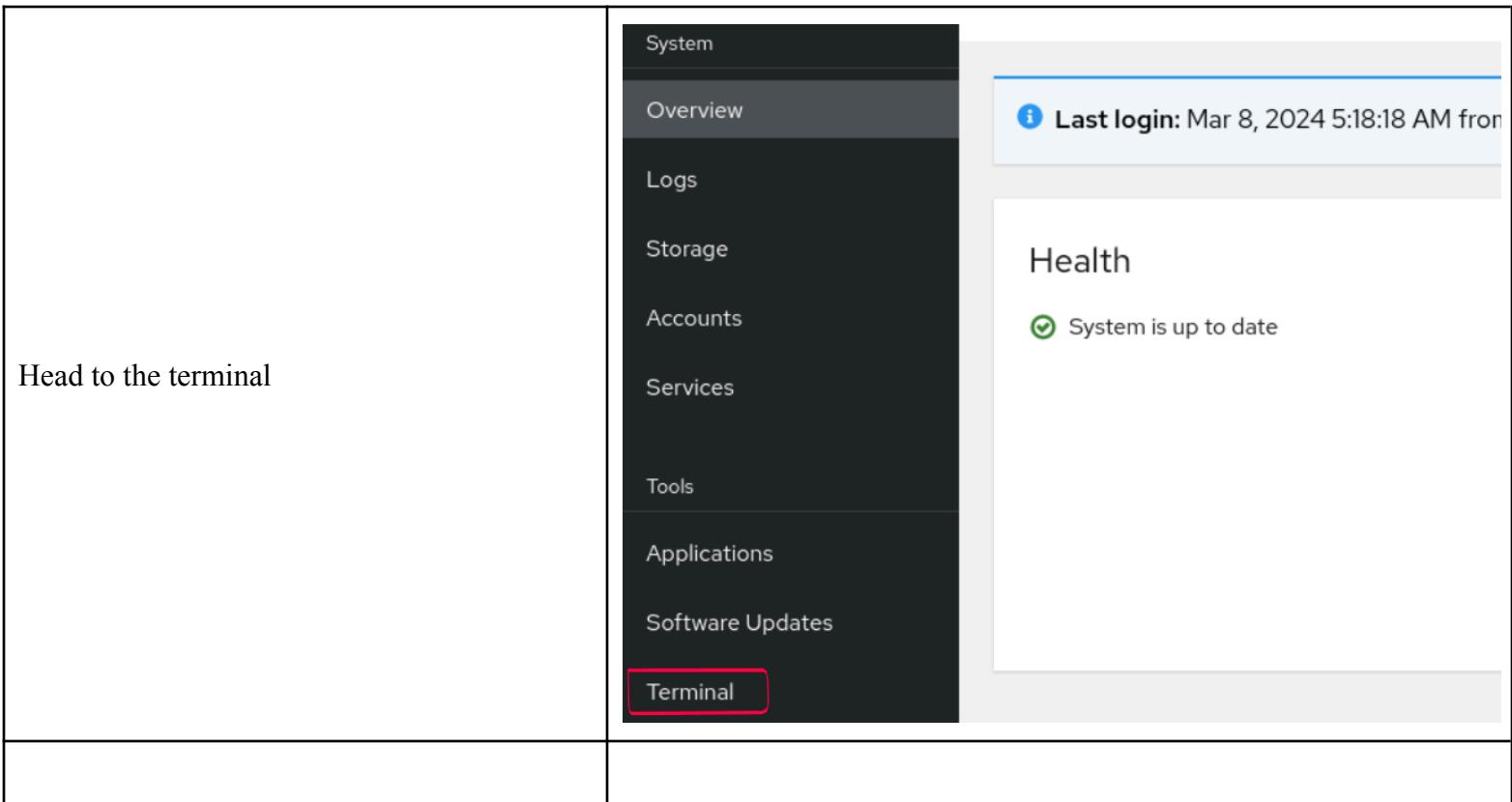
Open web browser head to your <public ip>

<https://<public-ip>:64297>



## Select Cockpit





<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>

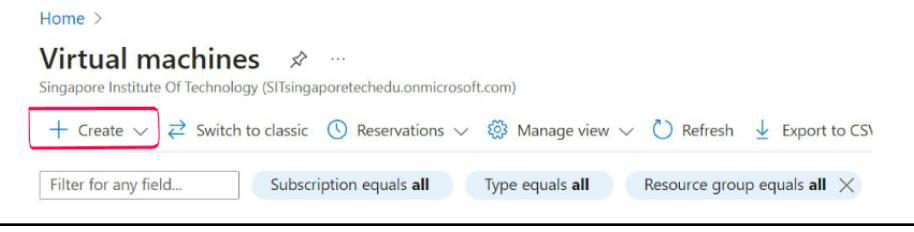
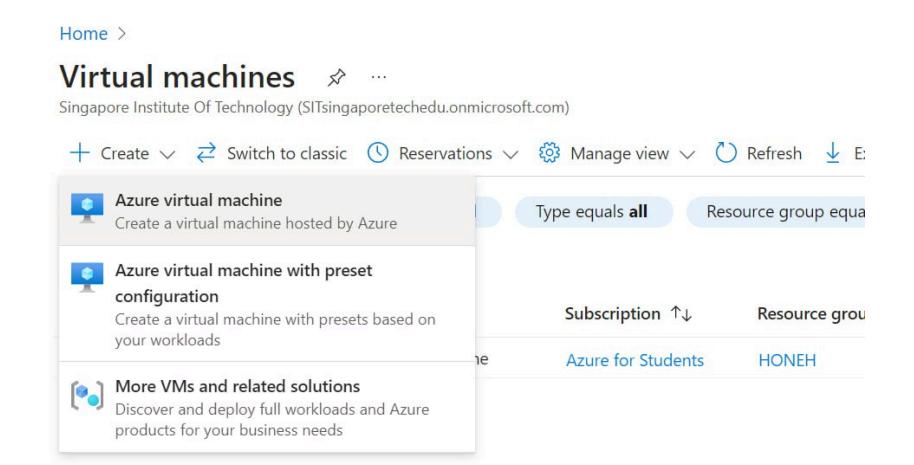
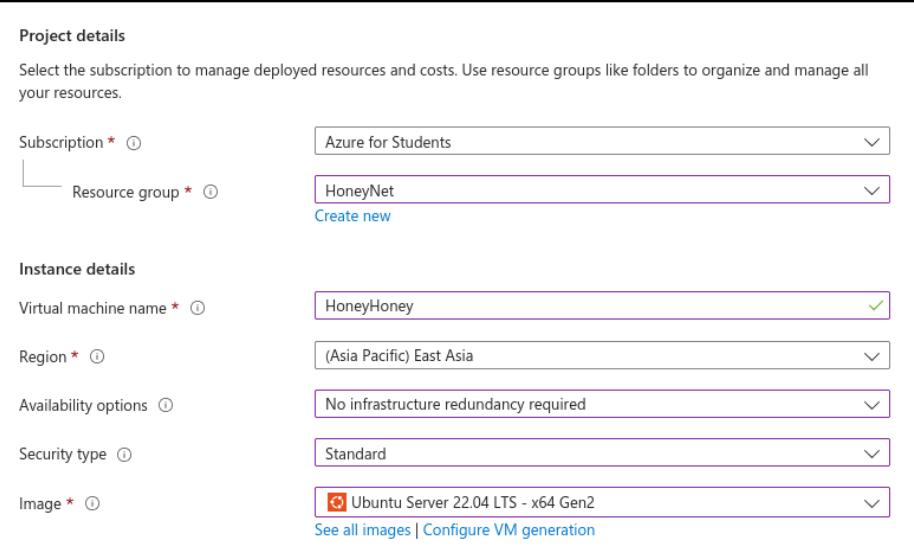
# Microsoft Azure HoneyPot (T-CUP)

## 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:

Click on create → Azure Virtual Machine	 
Select image as Ubuntu Server	<p>On [Basic] tab</p> 

For the Size, base on what you need	<p>VM architecture <input type="radio"/> Arm64 <input checked="" type="radio"/> x64</p> <p>Run with Azure Spot discount <input type="checkbox"/></p> <p>Size * <input type="radio"/> Standard_B2ms - 2 vcpus, 8 GiB memory (\$85.41/month) <input type="button" value="See all sizes"/></p> <p>Enable Hibernation (preview) <input type="checkbox"/> <small>To enable Hibernation, you must register your subscription. <a href="#">Learn more</a></small></p>												
Select Password and create a username and password	<p>Administrator account</p> <p>Authentication type <input type="radio"/> SSH public key <input checked="" type="radio"/> Password</p> <p>Username * <input type="text" value="firebear"/> <input type="button" value="✓"/></p> <p>Password * <input type="password"/> <input type="button" value="✓"/></p> <p>Confirm password * <input type="password"/> <input type="button" value="✓"/></p>												
<b>On [Disk] tab</b>													
Select Standard SSD (Locally-Redundant Storage)	<p>OS disk</p> <p>OS disk size <input type="radio"/> Image default (30 GiB)</p> <p>OS disk type * <input type="radio"/> Standard SSD (locally-redundant storage) <small>The selected VM size supports premium disks. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.</small></p>												
<p>Under DataDisk</p> <p>Click on “Create and attach a new disk”</p> <p>Select the size as “<b>128 GiB</b>”</p>	<p>Data disks</p> <p>You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.</p> <table border="1" data-bbox="670 1115 1584 1353"> <thead> <tr> <th>LUN</th><th>Name</th><th>Size (GiB)</th><th>Disk type</th><th>Host caching</th><th>Delete with VM</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table> <p><a href="#">Create and attach a new disk</a> <a href="#">Attach an existing disk</a></p> <p>Size * <input type="radio"/> <b>128 GiB</b> <small>Premium SSD LRS</small> <a href="#">Change size</a></p>	LUN	Name	Size (GiB)	Disk type	Host caching	Delete with VM						
LUN	Name	Size (GiB)	Disk type	Host caching	Delete with VM								

**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  <b>sudo apt update</b>	<pre>firebear@HoneyHoney:~\$ sudo apt update Get:1 http://azure.archive.ubuntu.com/ubuntu jammy InRelease [270 kB] Get:2 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB] Reading state information... Done All packages are up to date.</pre>
Install a few pre-requisites packages  <b>sudo apt install apt-transport-https ca-certificates curl software-properties-common</b>	<pre>firebear@HoneyHoney:~\$ sudo apt install apt-transport-https ca-certificates curl software-properties-common Reading package lists... Done Building dependency tree... Done Reading state information... Done 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.</pre>
Adding GPG key for official Docker Repo  <b>curl -fsSL https://download.docker.com/linux/ubuntu/gpg   sudo apt-key add -</b>	<pre>firebear@HoneyHoney:~\$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg   sudo apt-key add - Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)). OK</pre>
Adding Docker Repo to APT sources  <b>sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"</b>	<pre>firebear@HoneyHoney:~\$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable" Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable' Description: Docker repository for Ubuntu focal Archive for codename: focal components: stable More info: https://download.docker.com/linux/ubuntu</pre>
Make sure you are about the install from docker repo instead of the default Ubuntu Repo  <b>apt-cache policy docker-ce</b>	<pre>firebear@HoneyHoney:~\$ apt-cache policy docker-ce docker-ce:   Installed: (none)   Candidate: 5:25.0.4-1~ubuntu.20.04~focal   Version table:     5:25.0.4-1~ubuntu.20.04~focal 500       500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages</pre>

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

```
sudo apt install docker-ce
```

→ Check the Status

```
sudo systemctl status docker
```

## Using Docker commands:

**docker [option] [command] [arguments]**

There are more output, not just few of these.

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

```
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
      Docs: https://docs.docker.com
 Main PID: 4447 (dockerd)
TriggeredBy: ● docker.socket
```

## 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 filesystem
create      Create a new container
diff        Inspect changes to files or directories on a container's file
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
       □ virtualbox-application
Attach local standard input, output, and error streams to a running container
       □ (optional)

Aliases:
       □ Dockerfile
         docker container attach, docker attach

Options:
       □ README.md
--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		This repository has been archived by the owner. See README for details.	
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	

<p>Execute the command to download images to your computer</p> <p><b>sudo docker pull ubuntu</b></p>	<pre>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</pre>
<p>→ To see images that you have downloaded</p> <p><b>sudo docker images</b></p>	<pre>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</pre>
<p>→ Running Docker Container</p> <p><b>-i -t</b> switches gives you interactive shell access into the container</p> <p><b>-d</b> to keep running on background</p> <p><b>sudo docker run -it ubuntu</b> <b>sudo docker run -dit ubuntu</b></p>	<pre>firebear@HoneyHoney:~\$ sudo docker run -it ubuntu root@a2fa55caa3ea:/#</pre> <p>The container ID in the command prompt “<b>a2fa55caa3ea</b>”, You will need this container ID to identify the container if you want to remove it. Simply type “<b>Exit</b>” if you want to get out from the container</p>
<p>→ Managing Docker Containers</p> <p><b>sudo docker ps</b></p>	<pre>firebear@HoneyHoney:~\$ sudo docker ps CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES</pre>
<p>To view all containers</p> <p><b>sudo docker ps -a</b></p>	<pre>firebear@HoneyHoney:~\$ sudo docker ps -a CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES 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</pre>
<p>To START STOP a container</p> <p><b>sudo docker start &lt;container ID&gt;</b> <b>sudo docker stop &lt;container ID&gt;</b></p>	<pre>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</pre> <pre>firebear@HoneyHoney:~\$ sudo docker stop a2fa55caa3ea a2fa55caa3ea firebear@HoneyHoney:~\$ sudo docker ps CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES</pre>
<p>To REMOVE a container</p> <p><b>sudo docker rm &lt;container ID&gt;</b></p>	<pre>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</pre>

# Installing Cowrie

[SSH HoneyPots]

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

[Documentation]

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

Install Dependencies	<pre>firebear@HoneyHoney:/\$ sudo apt-get install git python3-virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv</pre>
Create User Accounts	<pre>firebear@HoneyHoney:/\$ sudo adduser --disabled-password cowrie Adding user `cowrie' ... Adding new group `cowrie' (1001) ... Adding new user `cowrie' (1001) with group `cowrie' ... Creating home directory `/home/cowrie' ... Copying files from `/etc/skel' ... Changing the user information for cowrie Enter the new value, or press ENTER for the default   Full Name []: It's strongly recommended to run with a dedicated non-root user id:   (OPTIONAL)   Room Number []:    Troubleshooting:   Work Phone []:   Home Phone []:   Updating Contact Information:   Other []: Is the information correct? [Y/n] y</pre>
Using superuser - cowrie	<pre>firebear@HoneyHoney:/home\$ ls cowrie firebear firebear@HoneyHoney:/home\$ sudo su - cowrie cowrie@HoneyHoney:~\$</pre>
Pull the cowrie from github	<pre>cowrie@HoneyHoney:~\$ git clone http://github.com/cowrie/cowrie Cloning into 'cowrie'... warning: redirecting to https://github.com/cowrie/cowrie/ remote: Enumerating objects: 17361, done. remote: Counting objects: 100% (2012/2012), done. remote: Compressing objects: 100% (482/482), done. remote: Total 17361 (delta 1741), reused 1674 (delta 1530), pack-reused 15349 Receiving objects: 100% (17361/17361), 9.89 MiB   14.79 MiB/s, done. Resolving deltas: 100% (12212/12212), done.</pre>
Set up the environment	<pre>cowrie@HoneyHoney:~/cowrie\$ pwd /home/cowrie/cowrie cowrie@HoneyHoney:~/cowrie\$ python3 -m venv cowrie-env/</pre>
Activate virtual environment	<pre>cowrie@HoneyHoney:~/cowrie\$ source cowrie-env/bin/activate</pre> <pre>cowrie@HoneyHoney:~/cowrie\$ source cowrie-env/bin/activate (cowrie-env) cowrie@HoneyHoney:~/cowrie\$ python -m pip install --upgrade pip Requirement already satisfied: pip in ./cowrie-env/lib/python3.10/site-packages (22.0.2) Collecting pip   Downloading pip-24.0-py3-none-any.whl (2.1 MB)     2.1/2.1 MB 22.5 MB/s eta 0:00:00 Installing collected packages: pip   Attempting uninstall: pip     Found existing installation: pip 22.0.2     Uninstalling pip-22.0.2:       Successfully uninstalled pip-22.0.2 Successfully installed pip-24.0</pre> <pre>(cowrie-env) cowrie@HoneyHoney:~/cowrie\$ python -m pip install --upgrade -r requirements.txt Collecting appdirs==1.4.4 (from -r requirements.txt (line 1))   Downloading appdirs-1.4.4-py2.py3-none-any.whl.metadata (9.0 kB)</pre>

<p>Install configuration file, change the directory to “/etc” and create a <b>cowrie.cfg</b> file</p> <p>[telnet] enable = true</p>	<pre>(cowrie-env) cowrie@HoneyHoney:~/cowrie\$ cd etc (cowrie-env) cowrie@HoneyHoney:~/cowrie/etc\$ nano cowrie.cfg</pre> <p style="text-align: center;">GNU nano 6.2 [telnet] http://shellpoc.com enable = true</p>
---	--

	<pre>(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),</pre>
--	--

## [Attackers POV]

<p>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</p> <p><b>sudo nmap -p 2222 -sV 23.99.107.240</b></p>	<pre>(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 .) Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel</pre>
--	---

<p>Using SSH on port 2222, any users will do, doesn't matter</p> <p><b>ssh -p 2222 root@23.99.107.240</b></p>	<pre>(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:Z7rPXD2h4V9E68Bu3fleJLEMPXHijSe27GD/f3BrxU. This key is not known by any other names. Are you sure you want to continue connecting (yes/no/[fingerprin t])? yes</pre>
---	--

<p>Issue any commands on the <b>CMD</b></p>	<pre>root@svr04:~# ls 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</pre>
---	---

## [Cowries POV]

<p>To view the <b>cowrie.log</b> file change the directory to <b>cd var/log/cowrie</b></p>	<pre>(cowrie-env) cowrie@HoneyHoney:~/cowrie\$ cd var/log/cowrie/ (cowrie-env) cowrie@HoneyHoney:~/cowrie/var/log/cowrie\$</pre>
<p>View the log file</p> <p><b>cat cowrie.log</b></p>	<pre>2024-03-09T06:24:36.832056Z [HoneyPotSSHTTransport,3,116.14.113.125] Command found: cd .. 2024-03-09T06:24:37.651107Z [HoneyPotSSHTTransport,3,116.14.113.125] CMD: ls 2024-03-09T06:24:37.651901Z [HoneyPotSSHTTransport,3,116.14.113.125] Command found: ls 2024-03-09T06:24:40.044623Z [HoneyPotSSHTTransport,3,116.14.113.125] CMD: cd opt 2024-03-09T06:24:40.045337Z [HoneyPotSSHTTransport,3,116.14.113.125] Command found: cd opt 2024-03-09T06:24:40.760844Z [HoneyPotSSHTTransport,3,116.14.113.125] CMD: ls</pre>

# Installing VSFTPD

## [Install VSFTPD]

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

Install VSFTPD  <b>sudo apt install vsftpd</b>	<pre>firebear@hone:~\$ sudo apt install vsftpd Reading package lists... Done Building dependency tree... Reading state information... Done The following additional packages will be installed:   ssl-cert Suggested packages:   openssl-blacklist</pre>
Check VSFTPD STATUS  <b>sudo service vsftpd status</b>	<pre>firebear@hone:~\$ sudo service vsftpd status ● vsftpd.service - vsftpd FTP server    Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)    Active: active (running) since Sat 2024-03-09 06:57:38 UTC; 58s ago      Main PID: 2599 (vsftpd)         Tasks: 1 (limit: 1062)        Memory: 520.0K       CGroup: /system.slice/vsftpd.service               └─2599 /usr/sbin/vsftpd /etc/vsftpd.conf  Mar 09 06:57:38 hone systemd[1]: Starting vsftpd FTP server... Mar 09 06:57:38 hone systemd[1]: Started vsftpd FTP server.</pre>
Change the settings of the vsftpd.conf file  <b>sudo nano /etc/vsftpd.conf</b>  <b>local_enable=YES</b> <b>write_enable=YES</b> <b>chroot_local_user=YES</b>	<pre># Uncomment this to allow local users to log in. local_enable=YES # # Uncomment this to enable any form of FTP write command. #write_enable=YES # # Default umask for local users is 077. You may wish to change this to 022, # if your users expect that (022 is used by most other ftpd's)</pre>
Add the command at the bottom, and save the file  <b>user_sub_token=\$USER</b> <b>local_root=/home/\$USER/ftp</b> <b>pasv_min_port=10000</b> <b>pasv_max_port=10100</b>	<pre># Uncomment this to indicate that vsftpd use a utf8 filesystem. #utf8_filesystem=YES user_sub_token=\$USER local_root=/home/\$USER/ftp pasv_min_port=10000 pasv_max_port=10100</pre>
Allow TCP traffic on ports 20,21 and range 10000 to 10100 from any source IP address to any destination IP address.  <b>sudo ufw allow from any to any port 20,21,10000:10100 proto tcp</b>	<pre>firebear@hone:~\$ sudo ufw allow from any to any port 20,21,10000:10100 proto tcp Rules updated Rules updated (v6)</pre>
Add new username and password  <b>sudo adduser &lt;username&gt;</b>	<pre>firebear@hone:~\$ sudo adduser userfire Adding user `userfire' ... Adding new group `userfire' (1001) ... Adding new user `userfire' (1001) with group `userfire' ... Creating home directory `/home/userfire' ... Copying files from `/etc/skel' ... New password: Retype new password:</pre>
	<pre>Changing the user information for userfire Enter the new value, or press ENTER for the default   Full Name []: go   Room Number []: go   Work Phone []: go   Home Phone []: go   Other []: go Is the information correct? [Y/n] y</pre>
Create a new directory named “FTP”  <b>sudo mkdir /home/userfire/ftp</b>	<pre>firebear@hone:~\$ sudo mkdir /home/userfire/ftp firebear@hone:~\$</pre>
Change the ownership to nobody user, nogroup group  <b>sudo chown nobody:nogroup /home/userfire/ftp</b>	<pre>firebear@hone:~\$ sudo chown nobody:nogroup /home/userfire/ftp firebear@hone:~\$</pre>

Removes the write permission for all user  <b>sudo chmod a-w /home/userfire/ftp</b>	<b>firebear@hone:~\$ sudo chmod a-w /home/userfire/ftp</b> <b>firebear@hone:~\$</b>
Creates directory named “upload” directory  <b>sudo mkdir /home/userfire/ftp/upload</b>	<b>firebear@hone:~\$ sudo mkdir /home/userfire/ftp/upload</b> <b>firebear@hone:~\$</b>
Change the ownership of the directory to userfire  <b>sudo chown userfire:userfire /home/userfire/ftp/upload</b>	<b>firebear@hone:~\$ sudo chown userfire:userfire /home/userfire/ftp/upload</b> <b>firebear@hone:~\$</b>
Write the text to a file name  <b>echo “My FTP server”   sudo tee /home/userfire/ftp/upload/demo.txt</b>	<b>firebear@hone:~\$ echo "My FTP server"   sudo tee /home/userfire/ftp/upload/demo.txt</b> "My FTP server"
List all files and directories in the /home/userfire/ftp  <b>sudo ls -la /home/userfire/ftp</b>	<b>firebear@hone:~\$ sudo ls -la /home/userfire/ftp</b> total 12 dr-xr-xr-x 3 nobody nogroup 4096 Mar 9 07:25 . drwxr-xr-x 3 userfire userfire 4096 Mar 9 07:10 .. drwxr-xr-x 2 userfire userfire 4096 Mar 9 07:30 upload
Appends the string “userfire” to the file  <b>echo “userfire”   sudo tee -a /etc/vsftpd.userlist</b>	<b>firebear@hone:~\$ echo "userfire"   sudo tee -a /etc/vsftpd.userlist</b> "userfire"
→ Restart VSFTPD  <b>sudo systemctl restart vsftpd</b>	<b>firebear@hone:~\$ sudo systemctl restart vsftpd</b> <b>firebear@hone:~\$</b>
Edit the <b>vsftpd.conf</b>  <b>sudo nano /etc/vsftpd.conf</b>  Add in these codes, and save the configuration file  <b>userlist_enable=YES</b> <b>userlist_file=/etc/vsftpd.userlist</b> <b>userlist_deny=NO</b>	# Uncomment this to indicate that vsftpd use a utf8 filesystem. #utf8_filesystem=YES user_sub_token=\$USER local_root=/home/\$USER/ftp pasv_min_port=10000 pasv_max_port=10100 <b>userlist_enable=YES</b> <b>userlist_file=/etc/vsftpd.userlist</b> <b>userlist_deny=NO</b>
Generate self signed certificates  <b>sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem</b>	<b>firebear@hone:~\$ sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem</b> Generating an RSA private key .....+++++ writing new private key to '/etc/ssl/private/vsftpd.pem'
Edit the <b>vsftpd.conf</b>  <b>sudo nano /etc/vsftpd.conf</b>  Add these lines  <b>rsa_cert_file=/etc/ssl/private/vsftpd.pem</b> <b>rsa_private_key_file=/etc/ssl/private/vsftpd.pem</b> <b>ssl_enable=YES</b>	# This option specifies the location of the RSA certificate to use for SSL # encrypted connections. <b>rsa_cert_file=/etc/ssl/private/vsftpd.pem</b> <b>rsa_private_key_file=/etc/ssl/private/vsftpd.pem</b> <b>ssl_enable=YES</b>

Add in couple more lines, and save the **config file**

```
allow_anon_ssl=NO  
force_local_data_ssl=YES  
force_local_logins_ssl=YES
```

```
# Uncomment this to indicate that vsftpd use a utf8 filesystem.  
#utf8_filesystem=YES  
user_sub_token=$USER  
local_root=/home/$USER/ftp  
pasv_min_port=10000  
pasv_max_port=10100  
userlist_enable=YES  
userlist_file=/etc/vsftpd.userlist  
userlist_deny=NO  
allow_anon_ssl=NO  
force_local_data_ssl=YES  
force_local_logins_ssl=YES
```

Restart the VSFTPD

```
sudo systemctl restart vsftpd
```

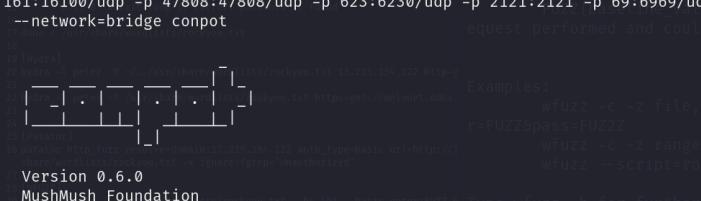
```
firebear@hone:~$ sudo systemctl restart vsftpd  
firebear@hone:~$
```

## Installing ConPot

## **[ICS Honeypot to collect intelligence about motives]**

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

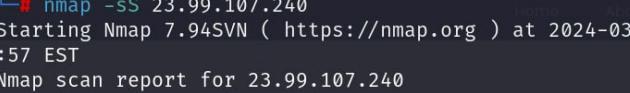
Link: [https://conpot.readthedocs.io/en/latest/installation/quick\\_install.html](https://conpot.readthedocs.io/en/latest/installation/quick_install.html)

<p>Pull the Clone ConPot from github, and change directory to conpot</p> <pre><b>git clone https://github.com/mushorg/conpot.git</b></pre> <pre><b>cd conpot</b></pre>	<pre><b>firebear@HoneyHoney:~\$ git clone https://github.com/mushorg/conpot.git</b> 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.</pre>
<p>Run docker and build conpot</p> <pre><b>sudo docker build -t conpot .</b></pre>	<pre><b>firebear@HoneyHoney:~/conpot\$ sudo docker build -t conpot .</b> [+] Building 2.4s (1/3) docker:default ⇒ [internal] load build definition from Dockerfile ⇒ ⇒ transferring dockerfile: 1.02kB ⇒ [internal] load metadata for docker.io/library/python:3.8-slim ⇒ [internal] load metadata for docker.io/library/python:3.8 0.1s 0.0s 2.3s 2.3s</pre>
<p>→ Start conpot</p> <pre><b>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 &gt; docker_run_log.txt 2&gt;&amp;1</b></pre>	<pre><b>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 44818:44818 --network=bridge conpot</b></pre>
<pre><b>sudo docker logs</b> 032d0c9eb48c06861012e1ca7f5508c1380d8cfbc96c6993d5 891a1c26f490ea</pre>	 <p>The terminal window displays a network diagram consisting of four nodes arranged in a square. Each node is represented by a box with three outgoing lines. Below the diagram, there is a list of URLs and their corresponding responses:</p> <ul style="list-style-type: none"> <li>http://10.219.133.122:35432/basic/peter:rockyou.txt -&gt; 200 OK</li> </ul> <p>At the bottom of the terminal, there is a message: "Type wfuzz -h for further information on how to use this tool."</p>

## [Attackers POV]

Run the command nmap

```
sudo nmap -sS 23.99.107.240
OR
sudo nmap -Pn 23.99.107.240
```

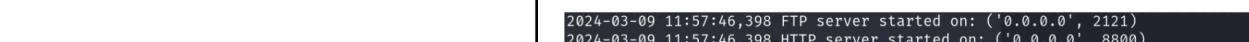


```
(root㉿kali)-[~/home/kali]
# nmap -sS 23.99.107.240
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-09 06:57 EST
Nmap scan report for 23.99.107.240
Host is up (0.038s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   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)  
  
2024-03-09 11:57:47,409 NEW TCP SESSION from 116.14.113.125 (0c0f26e2-d4f5-483f-aecb-f291ddb174f0)  
2024-03-09 11:57:57,490 New FTP connection from 116.14.113.125:25286. (0c0f26e2-d4f5-483f-aecb-f291ddb174f0)  
2024-03-09 11:57:57,490 FTP traffic to ('116.14.113.125', 25286): {'response': b'200 FTP server ready.\r\n'} (0c0f26e2-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/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>
```

```
firebear@HoneyHoney:~$ sudo git clone https://github.com/aelth/ddospot  
Cloning into 'ddospot' ...  
remote: Enumerating objects: 56, done.  
remote: Counting objects: 100% (56/56), done.  
remote: Compressing objects: 100% (39/39), done.  
remote: Total 56 (delta 14), reused 56 (delta 14), pack-reused 0  
Receiving objects: 100% (56/56), 56.55 KiB | 877.00 KiB/s, done.  
Resolving deltas: 100% (14/14), done.
```

```
firebear@HoneyHoney:~$ cd ddospot/ddospot  
firebear@HoneyHoney:~/ddospot/ddospot$
```

```
firebear@HoneyHoney:~/ddospot/ddospot$ sudo python3 -m venv env  
firebear@HoneyHoney:~/ddospot/ddospot$ source env/bin/activate  
(env) firebear@HoneyHoney:~/ddospot/ddospot$
```

```
(env) firebear@HoneyHoney:~/ddospot/ddospot$ sudo pip install -r requirements.txt  
Collecting git+https://github.com/hpfeeds/hpfeeds (from -r requirements.txt (line 2))  
  Cloning https://github.com/hpfeeds/hpfeeds to /tmp/pip-req-build-isworghh  
    Running command git clone --filter=blob:none --quiet https://github.com/hpfeeds/hpfeeds /tmp/pip-req-build-isworghh
```

```
firebear@HoneyHoney:~$ sudo chmod -R 777 /home/firebear/ddospot/logs  
firebear@HoneyHoney:~$
```

```
[general]  
listen_ip = 0.0.0.0  
listen_port = 19
```

```
firebear@HoneyHoney:~/ddospot/ddospot$ ./ddospot.py  
[+] Starting honeypot(s) using "start ntp" found  
Starting ntp, please wait ...  
NTPot started at 0.0.0.0:123  
[+] List enabled honeypots using "list"  
[+] Start honeypot(s) using "start <honeypot>" or "start all"  
[+] Use "help" to list all available commands
```

```
ddp > start ntp  
Starting ntp, please wait ...  
NTPot started at 0.0.0.0:123
```

```
firebear@HoneyHoney:/var/log$ sudo pidof -x ntpd  
23812  
firebear@HoneyHoney:/var/log$ sudo kill 23812
```

<p>→ To see the status of the NTP</p> <p><b>status ntp</b></p>	<pre>ddp &gt; status ntp ntp status: Configuring OpenCanary Number of IPs          Creating the initial configuration 0 Number of attacks       When OpenCanary starts it looks for config files in the 0 Total num. of packets recv. 0 First attack           - 1. open /etc/opencanary.conf (i.e. the directory where Open Latest attack           - Canary starts) Average attack duration - 1 minute, 7 seconds Longest continuous attack - 127.0.0.1, 3 minutes, 35 seconds (2024-03-16 (0.07 pps)) Largest continuous attack - 127.0.0.1, 3 minutes, 35 seconds (2024-03-16 (0.07 pps)) Top target (by pkt. count) - 127.0.0.1 (2024-03-10 16:32:10.718762) </pre>
<p>To solve for DNS</p> <p><b>sudo systemctl stop systemd-resolved</b>  <b>sudo systemctl disable systemd-resolved</b></p>	<pre>firebear@HoneyHoney:~/ddospot/ddospot\$ sudo systemctl stop systemd-resolved sudo systemctl disable systemd-resolved [sudo] password for firebear:  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'.</pre>
<p>→ Start DDoSPot DNS</p> <p><b>start dns</b></p>	<pre>ddp &gt; start dns Starting dns, please wait... DNSPot started at 0.0.0.0:53</pre>
<b>[Attackers POV]</b>	
<p>Flood the DNS of the webpage</p> <p><b>sudo hping3 -S --flood -V -p 53 29.99.107.240</b></p>	<pre>(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</pre>
<b>[DDoSPot POV]</b>	
<p>To check the status of the DNS</p> <p><b>status dns</b></p>	<pre>ddp &gt; status dns dns status: Number of IPs          1 Number of attacks       (main, No. 360 2023, 15214105) [GCC 11.4.0] Total num. of packets recv. 191 03:56.7828672 [-] Python Version 3.10.12 First attack            2024-03-10 16:32:10.718762 Latest attack           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 (2024-03-16 (0.07 pps)) Largest continuous attack 127.0.0.1, 3 minutes, 35 seconds (2024-03-16 (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</pre>

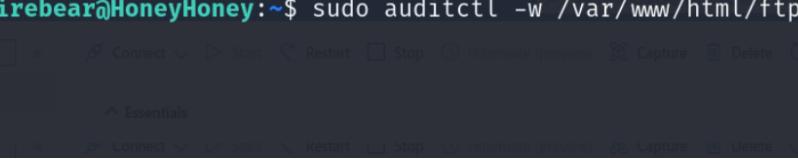
## Creating a DB with fake information

Change the directory where the FTP is <code>cd var/www/html/ftp</code>	<pre>firebear@HoneyHoney:/var/www/html/ftp\$ cd var/www/html/ftp firebear@HoneyHoney:/var/www/html/ftp\$</pre>
Creating a database using sqlite3, give a random name for the data base  <b>sudo sqlite3 customer.db</b>	<pre>firebear@HoneyHoney:/var/www/html/ftp\$ sudo sqlite3 customers.db SQLite version 3.37.2 2022-01-06 13:25:41 Enter ".help" for usage hints. sqlite&gt; CREATE TABLE customers (     id INTEGER PRIMARY KEY,     name TEXT, ipsum dolor sit amet, consectetur adipisicing elit. ist     credit_card TEXT,     password TEXT ); </pre>
Inserting datas into the customer.db	<pre>sqlite&gt; INSERT INTO customers (name, credit_card, password) VALUES ('John Doe', '1234 5678 9012 3456', 'password1'), ('Jane Smith', '9876 5432 1098 7654', 'password2'), ('Alice Johnson', '2468 1357 5791 3462', 'password3'), ('Bob Brown', '8642 7913 5736 2814', 'password4'), ('Eva Martinez', '5312 9476 1839 2057', 'password5');</pre>
To view the data  <b>SELECT * FROM customers;</b>	<pre>sqlite&gt; SELECT * FROM customers; 1 John Doe 1234 5678 9012 3456 password1 2 Jane Smith 9876 5432 1098 7654 password2 3 Alice Johnson 2468 1357 5791 3462 password3 4 Bob Brown 8642 7913 5736 2814 password4 5 Eva Martinez 5312 9476 1839 2057 password5</pre>
To Exit from the SQLITE  <b>.exit</b>	<pre>sqlite&gt; .exit</pre>

## Installation of Auditd

## [Collecting and Writing audit log file records]

Link: <https://sematext.com/glossary/audit/>

Install Auditd	<pre>firebear@hone:~\$ sudo apt install auditd Reading package lists... Done Building dependency tree Reading state information... Done The following additional packages will be installed:   libaudit0 Suggested packages:   audispd-plugins </pre>
→ Start & Enable Auditd	<pre>firebear@hone:~\$ sudo systemctl start auditd firebear@hone:~\$ sudo systemctl enable auditd Synchronizing state of auditd.service with SysV service script with /lib/systemd/systemd-sysv-install. Executing: /lib/systemd/systemd-sysv-install enable auditd</pre>
→ Verifying the status	<pre>firebear@hone:~\$ sudo systemctl status auditd ● auditd.service - Security Auditing Service   Loaded: loaded (/lib/systemd/system/auditd.service; enabled; vendor preset: enabled)     Active: active (running) since Sun 2024-03-10 12:06:08 UTC; 4min 39s ago       Docs: man:auditd(8)              https://github.com/linux-audit/audit-documentation         Main PID: 11879 (auditd)           Tasks: 2 (limit: 1062)          Memory: 396.0K             CGroup: /system.slice/auditd.service                      └─11879 /sbin/auditd</pre>
To monitor changes to <b>vip_info.txt</b> file	<pre>firebear@HoneyHoney:~\$ sudo auditctl -w /var/www/html/ftp/vip_info.txt -p rwx -k vip_info_changes</pre>
-w - specifies the files or directory -p rwx - specifies the permission to monitor for the file, read,write,attribute -k <b>vip_info_changes</b> - assigns a unique key to the rule	
To check if the rule is being applied correctly	<pre>firebear@HoneyHoney:~\$ sudo auditctl -l -w /var/www/html/ftp/vip_info.txt -p rwx -k vip_info_changes firebear@HoneyHoney:~\$ </pre>
→ To view the audit logs for events	<pre>firebear@HoneyHoney:~\$ sudo ausearch -k vip_info_changes time=Sun Mar 10 20:17:59 2024 type=PROCTITLE msg=audit(1710073079.763:236): proctitle=617564697463746C002D77002F7661722F777772F68746D6C2F6674702F7669705F696E666F2E747874002D700072777861002D6B007669705F696E666F5F6368616E676573 type=SOCKADDR msg=audit(1710073079.763:236): saddr=10000000000000000000000000000000 type=SYSCALL msg=audit(1710073079.763:236): arch=c000003e syscall=44 success=yes exit=1104 a0=4 a1=7ffd3a7f4170 a2=450 a3=0 items=0 ppid=4276 pid=4276 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty pts2 ses=1 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null) type=CONFIG_CHANGE msg=audit(1710073079.763:236): auid=1000 ses=1 subj=unconfined op=add_rule key="vip_info_changes" list=4 res=0</pre>
<b>sudo ausearch -k vip_info_changes</b>	
<b>ausearch</b> - command line tool for searching audit logs <b>-k vip_info_changes</b> - 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.

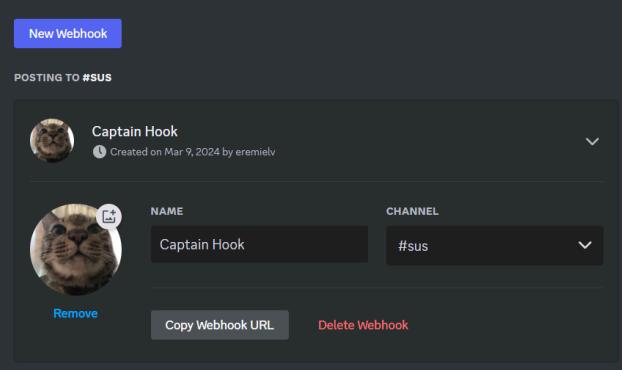
Other alternatives:

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 honeypot.

# Admin Login

Username:

Password:

→ 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**

```
firebear@HoneyHoney:~$ php -v
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies ..
```

```
<?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-4wLJU216R0rxc2Yuc5pfAaWeKJmTf8ntAosFpGxYOVxs7Jlva9R-eflorV";

    $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=^PA
SS^:F:error" -s 443 -S

```

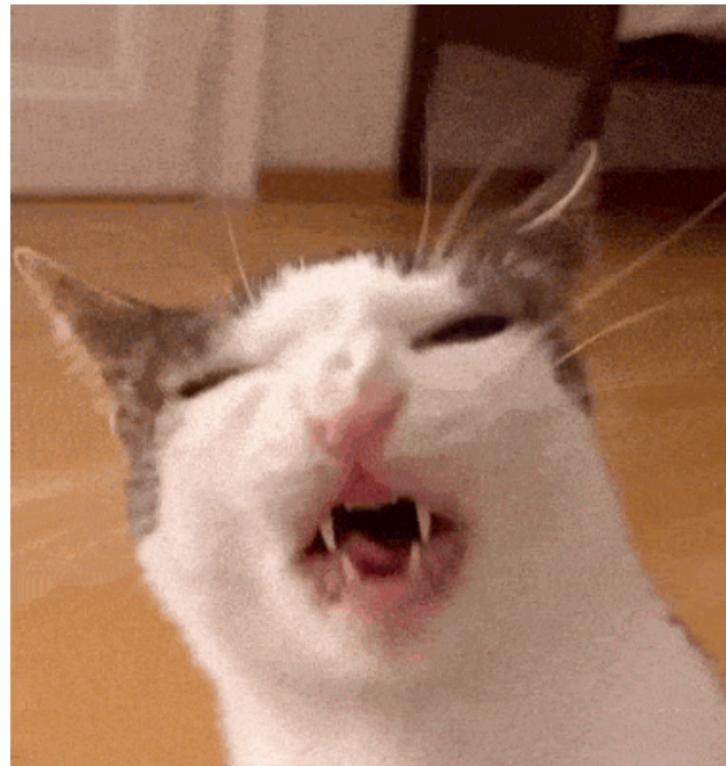
```

[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

```

# Login Successful!

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



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

## [Discord POV]

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

A screenshot of a Discord interface. On the left, there is a sidebar with 'TEXT CHANNELS' and 'VOICE CHANNELS'. Under 'TEXT CHANNELS', there are two channels: '# general' and '# SUS'. The '# SUS' channel has a message from a user named 'Captain Hook' (represented by a skull icon) at 'Today at 3:07 PM'. The message content is a log of failed admin panel login attempts from the IP address 112.199.205.80. The log includes multiple entries for various usernames and passwords, such as 'heheheheh', 'hello', 'babigirl', 'jessica', 'lobby', 'rockyou', 'daniel', 'password', 'lovely', 'princess', 'rockyou', 'iloveyou', 'teehee', and 'daniel'. The log ends with 'Attempted access to admin panel from 112.199.205.80 using username: teehee and password: nicole'.

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 1x1.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 Geoloc (MaxMind) for Geoloc 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.

<p>Sign up for an account. Then head to this page:  <a href="https://www.maxmind.com/en/accounts/984758/geolp/downloads">https://www.maxmind.com/en/accounts/984758/geolp/downloads</a> .</p> <p>Download GeoLite2 City (GZIP) into Kali Linux.</p>	<p><b>GeoLite2 City</b></p> <p><b>Edition ID:</b> GeoLite2-City</p> <p><b>Format:</b> GeolP2 Binary (.mmdb) (APIs)</p> <p><b>Updated:</b> 2024-03-08</p> <ul style="list-style-type: none"> <li>• <a href="#">Download GZIP</a></li> <li>• <a href="#">Download SHA256</a></li> <li>• <a href="#">Get Permalinks</a></li> </ul>
<p>Create folder under /home/firebear called GeoIP</p>	<pre>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</pre>
<p>Upload file from Kali Linux to server.</p> <pre>scp /home/kali/Downloads/GeoLite2-City_20240308.tar.gz firebear@23.99.107.240:/home/firebear/GeoIP</pre>	<pre>root@HoneyHoney:/home/firebear# cd GeoIP root@HoneyHoney:/home/firebear/GeoIP# ls root@HoneyHoney:/home/firebear/GeoIP# ls GeoLite2-City_20240308.tar.gz</pre>
<p>Unzip file. Make sure to check if GeoLite2-City.mmdb is inside.</p> <p>Check using command.</p> <pre>ls /home/firebear/GeoIP/GeoLite2-City_20240308/ GeoLite2-City.mmdb</pre>	<pre>root@HoneyHoney:/home/firebear/GeoIP# tar -xvf GeoLite2-City_20240308.tar.gz GeoLite2-City_20240308/ GeoLite2-City_20240308/LICENSE.txt  title&gt; GeoLite2-City_20240308/COPYRIGHT.txt GeoLite2-City_20240308/GeoLite2-City.mmdb GeoLite2-City_20240308/README.txt &lt;a href="https://melonet.ddns.net/testing/admin.php"&gt;Port 80&lt;/a&gt; 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</pre>
<p>Enter the /home/firebear directory.  We need Composer to proceed with using GeoLite2.  Set up Composer.</p> <pre>php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');" sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer</pre>	<pre>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 //DID HTML 2.0/EN&gt;</pre>
<p>Check if Composer installed. (Check as non-root user)</p> <pre>composer --version</pre>	<pre>root@HoneyHoney:/home/firebear# su firebear firebear@HoneyHoney:~\$ composer --version Composer version 2.7.1 2024-02-09 15:26:28</pre>

Check for the latest version.  <b>composer self-update</b>	<pre>firebear@HoneyHoney:~\$ composer self-update You are already using the latest available Composer version 2.7.1 (stable channel).</pre>
Remove Composer installer script.  <b>php -r "unlink('composer-setup.php');"</b>	<pre>firebear@HoneyHoney:~\$ php -r "unlink('composer-setup.php');" file:///var/www/html/admin/composer-setup.php</pre>
Now we can proceed with GeoLite2 setup. Set up a composer.json file.  <b>composer init</b>	<pre>firebear@HoneyHoney:/var/www/html/admin\$ composer init Connection to 23.99.107.240 closed by remote host.  Welcome to the Composer config generator  This command will guide you through creating your composer.json config.</pre>
Set up with necessary details. For me, I only specified geoip2 as my dependency. I left the rest blank.	<pre>Package name (&lt;vendor&gt;/&lt;name&gt;) [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 File Actions Edit View Help [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 Permanently&lt;/title&gt; [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):</pre>

```

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]: password:

{
    "name": "firebear/admin",
    "require": {
        "geoip2/geoip2": "^3.0"
    },
    "autoload": {"Actions Edit View Help", "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 details.
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**

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$ 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

```

```

firebear@HoneyHoney:/var/www/html/admin$ sudo chown www-data:www-data /home/firebear/GeoIP/GeoLite2-City_20240308/GeoLite2-City.mmdb
firebear@HoneyHoney:/var/www/html/admin$ sudo chmod 644 /home/firebear/GeoIP/GeoLite2-City_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-City_20240308/

```

```

<?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>

<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 1x1.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**

## Admin Login

Username:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Login"/>	

[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 Safari/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
```

[Download Databases](#)

Show archived database files

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+92ubuntu1 [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.
```

[Download Databases](#)

Show archived database files

No

containers need to be restarted.

No user sessions are running outdated binaries.

[Details](#)

[Download Links](#)

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

[Download GZIP](#)

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;**

```
1|12.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:13|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36|https://melonet.ddns.net/admin/admin.php
2|12.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:13|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36|https://melonet.ddns.net/admin/admin.php
3|12.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:14|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36|https://melonet.ddns.net/admin/admin.php
4|12.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:14|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36|https://melonet.ddns.net/admin/admin.php howto look up an IP visit the GeoIP database
5|12.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:14|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36|https://melonet.ddns.net/admin/admin.php
6|12.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:14|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36|https://melonet.ddns.net/admin/admin.php
7|12.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:19|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36|https://melonet.ddns.net/admin/admin.php
8|12.199.205.80|Singapore|Singapore|1.3596|103.8637|2024-03-12 01:23:19|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36|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://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 1x1.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](https://melonet.ddns.net/admin/attack_map.html)

