COWRIE HONEYPOT SETUP

Setup VMS and import into VMWare Workstation

- Make two copies of our Ubuntu_22_04 VM and name one **Cowrie** and one **Attacker**
- Import the following VM into VMWare Workstation: **Cowrie**
- Change the allocated RAM to 4GB
- Rename the VM (optional)
- Ensure the network adapter is set to NAT
- Take a snapshot of the VM, name it "140223" (RightClick > SnapShot > SnapShot Manager > Take SnapShot)
- Power on each VM and when asked select "*I copied it*"
- Login with password: ubuntu2204
- Open a Terminal window and make a note of the IP address using command

  ip address

Install Cowrie [SOC User]

- Install cowrie using the following commands. Note: install commands can be combined using && but are shown here separately for clarity.

  sudo apt-get update
  sudo apt-get upgrade
  sudo apt-get install -y git
  sudo apt-get install -y python3-virtualenv
  sudo apt-get install -y libssl-dev
  sudo apt-get install -y libffi-dev
  sudo apt-get install -y build-essential
  sudo apt-get install -y libpython3-dev
  sudo apt-get install -y python3-minimal
  sudo apt-get install -y authbind
  sudo apt-get install -y virtualenv

sudo adduser --disabled-password cowrie
sudo su – cowrie

- **[SOC User]** Open a second terminal window to download cowrie leaving the above terminal window open/ Issue the following commands:

  git clone https://github.com/cowrie/cowrie/archive/refs/tags/v2.4.0.tar.gz
  tar -xvzf cowrie-2.4.0.tar.gz
  mv cowrie-2.4.0 cowrie
  sudo mv cowrie /home/cowrie
  sudo chown -R cowrie:cowrie /home/cowrie/
  ls -l

- **[Cowrie User]** Return to the original terminal window to continue using cowrie

  cd cowrie
  virtualenv --python=python3 cowrie-env
  source cowrie-env/bin/activate
  pip install --upgrade pip
  pip install --upgrade -r requirements.txt
  deactivate

Configure Cowrie [Cowrie User]

- Configure with minimum settings

  cd /home/cowrie/cowrie/etc
  cp cowrie.cfg.dist cowrie.cfg
  nano cowrie.cfg
  [honeypot] # configure setting below
  hostname = cowrieInstallation

  [SSH] # configure setting below
  listen_endpoints = tcp:22:interface=0.0.0.0

  [telnet] # configure setting below
  enabled = true
  listen_endpoints = tcp:23:interface=0.0.0.0
  exit

Install authbind [SOC User]

- Install authbind and necessary files

sudo apt install authbind
sudo touch /etc/authbind/byport/22
sudo chown cowrie:cowrie /etc/authbind/byport/22
sudo chmod 770 /etc/authbind/byport/22
sudo touch /etc/authbind/byport/23
sudo chown cowrie:cowrie /etc/authbind/byport/23
sudo chmod 770 /etc/authbind/byport/23

Install and configure SSH [SOC User]

- Install and configure SSH and Telnet

sudo apt-get install ssh
sudo nano /etc/ssh/sshd_config
Port 3393
listenAddress 0.0.0.0
exit

sudo systemctl restart sshd

Basic usage of Cowrie [Cowrie User]

- Start/Stop/Status of cowrie

sudo su - cowrie
cd /home/cowrie/cowrie
bin/cowrie start

- View log files

ATTACKER SETUP

Setup VM and import into VMWare Workstation

- Import the following VM into VMWare Workstation: Attacker

- Change the allocated RAM to 4GB
- Rename the VM (Optional)
- Ensure the network adapter is set to NAT
- Take a snapshot of the VM, name it "140223" (RightClick > SnapShot > SnapShot Manager > Take SnapShot)
- Power on each VM and when asked select "*I copied it*"
- Login with password: ubuntu2004
- Open a Terminal window and make a note of the IP address using

  ip address

Install Cowrie Detect and Tools

sudo apt-get update
sudo apt-get upgrade
sudo apt-get install -y git
git clone https://github.com/boscutti939/Cowrie_Detect.git

pip3 install python3-nmap
pip3 install paramiko

TESTING

- Ensure both **Cowrie** and **Attacker** VMs and make a note of each IP address
- **[Cowrie User]** On the **Cowrie** VM start the HoneyPot running and check the status

  sudo su - cowrie
  cd /home/cowrie/cowrie

bin/cowrie start
bin/cowrie status

- On the **Attacker** VM run the cowrie_detect script to evaluate if the remote host is a honeypot, where x.x.x.x is the IP address of your HoneyPot

  cd Cowrie_Detect/
  ./cowrie_detect.py x.x.x.x

Q Was the remote device detected as a HoneyPot?
Q What variables did the script check for?

Change default Cowrie Configuration [Cowrie User]

- On the **Cowrie** VM stop the HoneyPot running

  sudo su - cowrie
  cd /home/cowrie/cowrie
  bin/cowrie stop
  bin/cowrie status

- Change the default configuration of Cowrie and run the cowrie_detect.py script again

  Customisations to the default cowrie configuration can be made to make the honeypot detection by an attacker more difficult. Default values can be modified, new directories added, and system configurations changed as appropriate. This is to ensure that the honeypot system is not easily detected as a honeypot by automated tools such as **cowrie_detect.py**

- Take a snapshot of the cowrie VM before changing anything (just in case). Some examples of changes that can be made to default settings are:

  Configuration File (**./etc/cowrie.cfg**)

  Pickle Filesystem (**./share/cowrie/fs.pickle**): This is a fake virtual filesystem on Cowrie honeypot as the files in this directory do not exist. Attackers could list these files in the directory by issuing the Linux "ls" command

  Filesystem (**./honeyfs**): This is the directory of Cowrie file system where a tree of files/folders could be placed. The files in this directory would be visible to the attackers and the files would be available for viewing.

/home/cowrie/cowrie/honeyfs/etc/motd - post-login banner
/home/cowrie/cowrie/honeyfs/proc/cpuinfo – CPU information
/home/cowrie/cowrie/honeyfs/proc/meminfo – memory information
/home/cowrie/cowrie/honeyfs/proc/mounts – the drive mount
/home/cowrie/cowrie/honeyfs/proc/version – version of the system
/home/cowrie/cowrie/honeyfs/proc/net/arp – address resolution protocol of the server

- For the purpose of this exercise we will make minor changes and rerun the detect script. Change the CPU information to the details below

  cd */home/cowrie/cowrie/honeyfs/proc/*
  nano cpuinfo

  Intel(R) Core(TM)2 Duo CPU E8400 @ 3.70GHz

  nano meminfo

  *** (change some random values)*

- Start the HoneyPot on **Cowrie** VM
  cd /home/cowrie/cowrie
  bin/cowrie start
  bin/cowrie status

- On the **Attacker** VM run the cowrie_detect script to evaluate if the remote host is a honeypot, where x.x.x.x is the IP address of your HoneyPot

  cd Cowrie_Detect/
  ./cowrie_detect.py x.x.x.x

Q Was detection of the remote computer as a HoneyPot any different? (it may take a little time to make sufficient changes

- On the **Attacker** VM ssh into the HoneyPot (use username:password root:password)and issue some commands, where x.x.x.x is the IP address of your HoneyPot

  ssh root@x.x.x.x
  ls
  pwd
  cd /home

ls
cd /phil
ls
exit

- On the **Cowrie** VM check the log file to see if you can see evidence of the remote connection

  cd /cowrie/var/log/cowrie
  ls
  cat cowrie.json

Q Could you see the commands you issued as the attacker?