**CUCKOO INSTALLATION (Malware Analysis)**



Cuckoo Sandbox is an open-source software for automating analysis of suspicious files running in an isolated environment.

* Initially nested VT-x/ AMD V has to be enabled, from bios as well as in virtual box.

sudo apt update -y

sudo apt install net-tools

sudo apt upgrade -y

sudo apt install git –y

sudo apt-get install python python-pip python-dev libffi-dev libssl-dev -y

sudo apt-get install python-virtualenv python-setuptools -y

sudo apt-get install libjpeg-dev zlib1g-dev swig -y

sudo apt-get install mongodb -y

sudo apt-get install postgresql libpq-dev -y

sudo apt install virtualbox –y

virtualbox

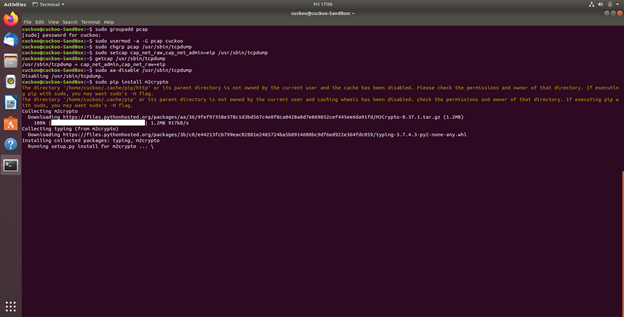
if any troubleshoot occurs while installing virtualbox refer

below link

<https://www.bojankomazec.com/2019/04/how-to-install-virtualbox-on-ubuntu-1804.html>

sudo apt-get install tcpdump apparmor-utils -y

create a seperate user for cuckoo since its not recommended to run cuckoo as root



sudo groupadd pcap  
sudo usermod -a -G pcap cuckoo

You’ll need a network sniffer properly configured to capture the traffic and dump it to a file.

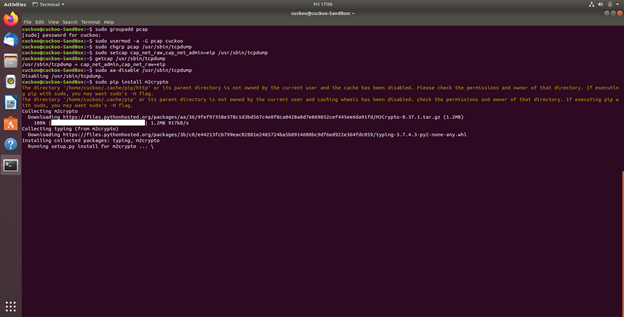
sudo chgrp pcap /usr/sbin/tcpdump

sudo setcap cap\_net\_raw,cap\_net\_admin=eip /usr/sbin/tcpdump

getcap /usr/sbin/tcpdump

sudo aa-disable /usr/sbin/tcpdump

Install the latest version of m2crypto



sudo pip install m2crypto

If you havent created the user as cuckoo at installation, create a new user called cuckoo now

sudo adduser cuckoo

Add the user to vboxusers group

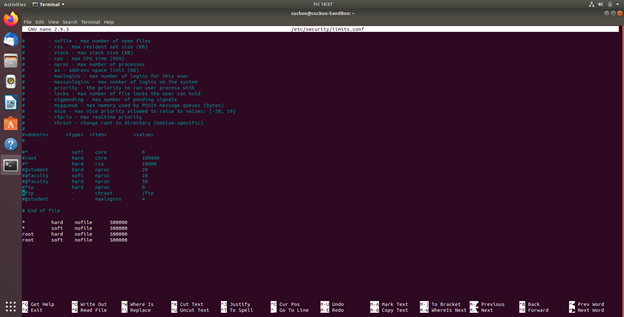
sudo usermod -a -G vboxusers cuckoo

**Increase “Open Files Limit”**

https://miro.medium.com/max/780/1*s8zIOiyo7gHquN84TFU2pA.png

sudo nano /etc/security/limits.conf

Add the following at the end and save



Copy and paste below lines at last in limits.conf

\* hard nofile 500000

\* soft nofile 500000

root hard nofile 500000

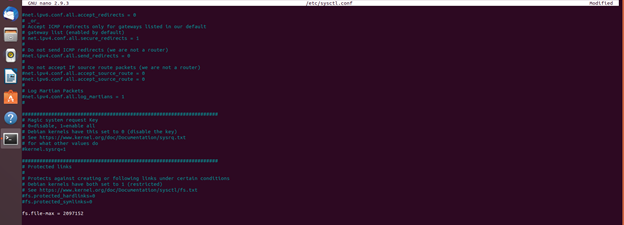
root soft nofile 500000

sudo nano /etc/pam.d/common-session

Add at end and save

session required pam\_limits.so

Set this higher than user-limit set above.

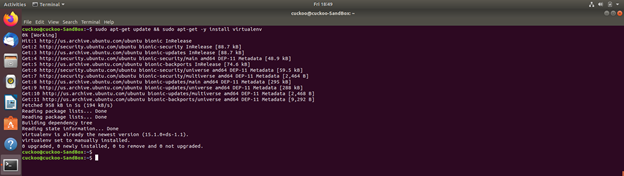


sudo nano /etc/sysctl.conf

fs.file-max = 2097152

**INSTALLING CUCKOO**

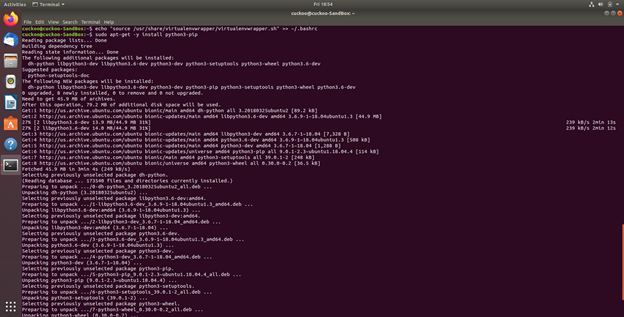
Install virtualenv



sudo apt-get update && sudo apt-get -y install virtualenv

Install virtualenvwrapper

sudo apt-get -y install virtualenvwrapper



Create bash file

#!/usr/bin/env bash

# Author: Josh Stroschein

# Source: https://askubuntu.com/questions/244641/how-to-set-up-and-use-a-virtual-python-environment-in-ubuntu

# NOTES: Run this script as: sudo -u <USERNAME> cuckoo-setup-virtualenv.sh

# Additionally, your environment may not allow the script to source bashrc and you may need to do this manually after the script completes

# install virtualenv

sudo apt-get update && sudo apt-get -y install virtualenv

# install virtualenvwrapper

sudo apt-get -y install virtualenvwrapper

echo "source /usr/share/virtualenvwrapper/virtualenvwrapper.sh" >> ~/.bashrc

# install pip for python3

sudo apt-get -y install python3-pip

# turn on bash auto-complete for pip

pip3 completion --bash >> ~/.bashrc

# avoid installing with root

pip3 install --user virtualenvwrapper

echo "export VIRTUALENVWRAPPER\_PYTHON=/usr/bin/python3" >> ~/.bashrc

echo "source ~/.local/bin/virtualenvwrapper.sh" >> ~/.bashrc

export WORKON\_HOME=~/.virtualenvs

echo "export WORKON\_HOME=~/.virtualenvs" >> ~/.bashrc

echo "export PIP\_VIRTUALENV\_BASE=~/.virtualenvs" >> ~/.bashrc

source ~/.bashrc

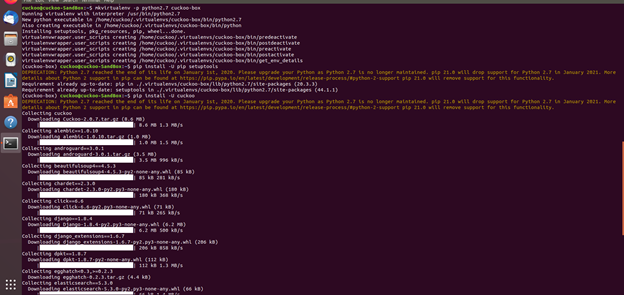
Create bash file(as above) and save it on desktop

chmod 777 <bash filename.sh>

sudo –u <username> ./<bash file name.sh>

source ~/.bashrc

Make python virtual environment for cuckoo



mkvirtualenv -p python2.7 cuckoo-box

**To reactivate virtual environment**

workon cuckoo-box

Now inside the venv

(venv)$ pip install -U pip setuptools

(venv)$ pip install -U cuckoo

**CREATING A WINDOWS 7 VM**

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |

sudo wget https://cuckoo.sh/win7ultimate.iso

sudo mkdir /mnt/win7

sudo chown cuckoo:<user name> /mnt/win7/

sudo mount -o ro,loop win7ultimate.iso /mnt/win7

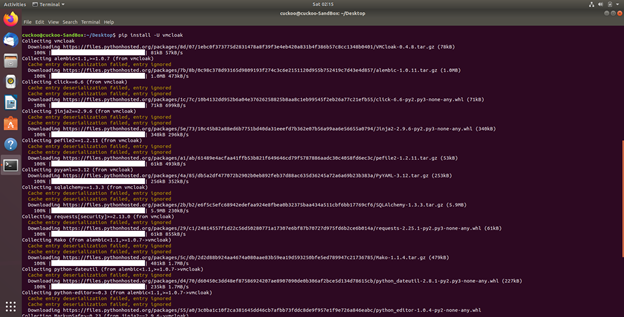
More dependencies, same as initial, just a recheck

sudo apt-get -y install build-essential libssl-dev libffi-dev python-dev genisoimage

sudo apt-get -y install zlib1g-dev libjpeg-dev

sudo apt-get -y install python-pip python-virtualenv python-setuptools swig

Install vmcloak, which automates most of the configuring of guests



pip install vmcloak

https://miro.medium.com/max/780/1*j38_19_B9KtVZ6Ollnfd1A.png

vmcloak-vboxnet0

Most malware check if the host is a vm before executing, so its better to have 2 CPU and decent ram

vmcloak init --verbose --win7x64 win7x64base --cpus 2 --ramsize 2048

Clone the new vm, this will eliminate the need to having to configure a vm from the beginning

https://miro.medium.com/max/780/1*YmeAwidEnA9EnD188eAn8Q.png

vmcloak clone win7x64base win7x64cuckoo

vmcloak list deps (option)

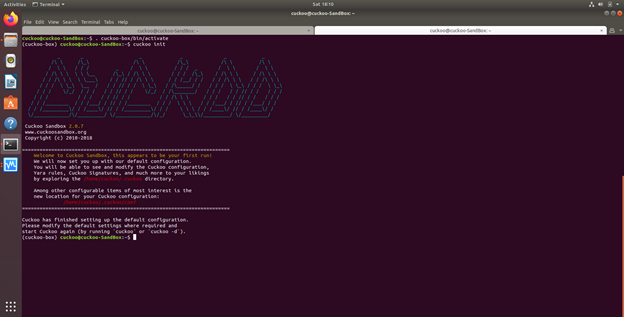
vmcloak install win7x64cuckoo ie11

Using — count multiple snapshots can be created

vmcloak snapshot --count 4 win7x64cuckoo 192.168.56.101

**STARTING CUCKOO**

While still in venv



**cuckoo init or workon cuckoo**

Browse to cuckoo directory

cd ~/.cuckoo

Here we can do some configurations, and the important files are



Install cuckoo community, which has useful signatures that will be useful while in analysis

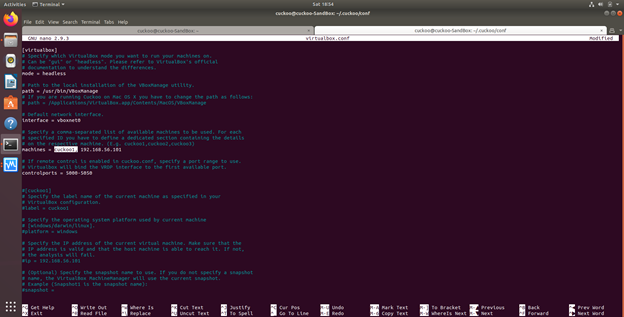
cuckoo community

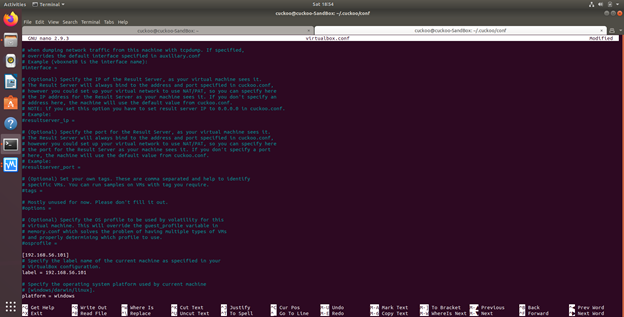
It would be easier to write using vmcloak

https://miro.medium.com/max/780/1*YFmLaPfVgBC-wlih15oN8A.png

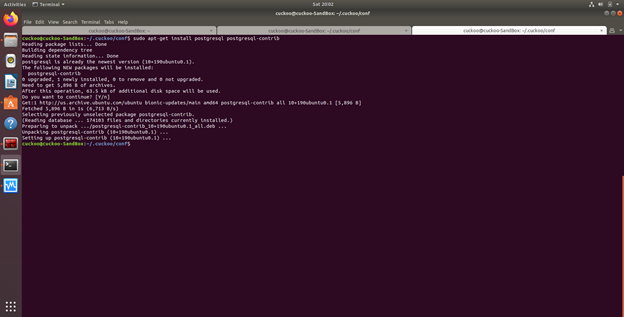
while read -r vm ip; do cuckoo machine --add $vm $ip; done < <(vmcloak list vms)

Remove the entry cuckoo1 from machines and its definitions up to entry “osprofile” where a real VM ip is present





**Postgres as DBMS**



sudo apt-get postgresql postgresql-contrib

Install PostgreSQL database driver inside venv

https://miro.medium.com/max/780/1*g1r8_vyUWXWnT6OP71rDWg.png

(cuckoo-box) cuckoo@cuckoo-SandBox:~$ pip install psycopg2

Create a user and database



sudo -u postgres psql

CREATE DATABASE cuckoo;

CREATE USER cuckoo WITH ENCRYPTED PASSWORD 'password';

GRANT ALL PRIVILEGES ON DATABASE cuckoo TO cuckoo;

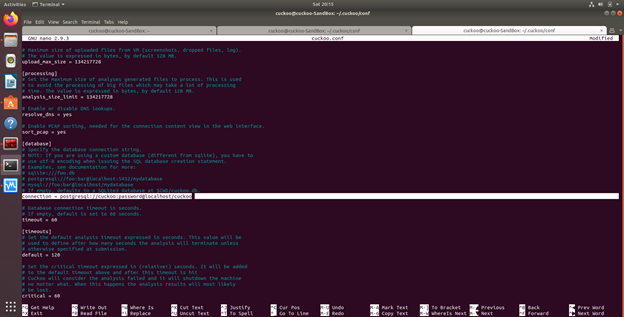
\q

Open the

$CWD/conf/cuckoo.conf

File and find the**[database]** section.

Change the **connection =** line to:

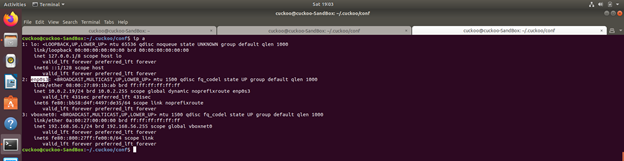


connection = postgresql://cuckoo:password@localhost/cuckoo

**CONFIGURING THE NETWORK**

Use a new terminal

Find your local adapter, run



ip a

Its **enp0s3**this time, and enable forwarding on

https://miro.medium.com/max/780/1*nFWM8sguADmqme7H6vJR2w.png

sudo sysctl -w net.ipv4.conf.vboxnet0.forwarding=1

sudo sysctl -w net.ipv4.conf.enp0s3.forwarding=1

To enable global routing for all VMs connected to vboxnet0

https://miro.medium.com/max/780/1*hCXU5kFOHrXPl2GoH0CISg.png

sudo iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.56.0/24 -j MASQUERADE

sudo iptables -P FORWARD DROP

sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

sudo iptables -A FORWARD -s 192.168.56.0/24 -j ACCEPT

**STARTING CUCKOO**

Start terminator and split(optional) (you might need to install terminator)



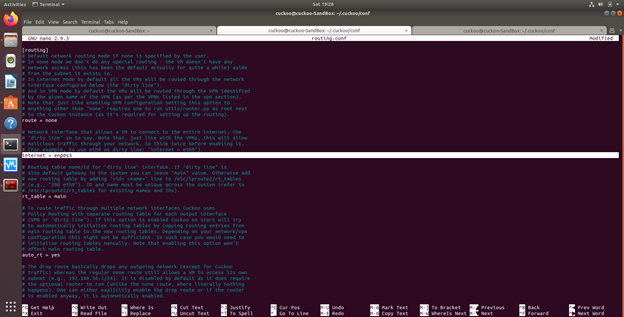
**In Terminal -1:**

workon cuckoo-box

cuckoo rooter --sudo --group <username>

After starting rooter, make some changes in routing.conf

Change from none to adapter name under internet

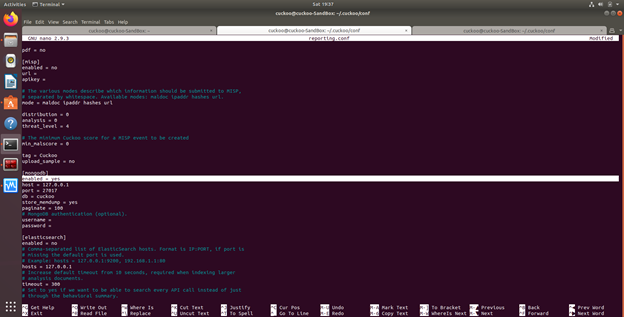


internet = <LAN\_card name>

Save and quit

**STARTING WEB SERVER**

First, edit reporting.conf



[mongodb] enabled = yes

**In terminal - 2**

Run cuckoo now in a new terminal under venv and you should see rooter responding

(cuckoo-box) cuckoo@cuckoo-SandBox:~$ cuckoo

**In Terminal -3**

*run web interface, run*

cuckoo web — host 127.0.0.1 — port 8080

*once you’re done analyzing make sure you clean the sandboxes by*

cuckoo clean