



N00b H@ck3r < https://beginninghacking.net/>

How to Setup Your Own Malware Analysis Box – Cuckoo Sandbox

 [lightkunyagami < https://beginninghacking.net/author/lightkunyagami/>](https://beginninghacking.net/author/lightkunyagami/)

 [November 16, 2022 < https://beginninghacking.net/2022/11/16/how-to-setup-your-own-malware-analysis-box-cuckoo-sandbox/>](https://beginninghacking.net/2022/11/16/how-to-setup-your-own-malware-analysis-box-cuckoo-sandbox/)

 [2 Comments < https://beginninghacking.net/2022/11/16/how-to-setup-your-own-malware-analysis-box-cuckoo-sandbox/#comments>](https://beginninghacking.net/2022/11/16/how-to-setup-your-own-malware-analysis-box-cuckoo-sandbox/#comments)

I am writing this blog entry because I know I was not the only one who had trouble setting up my own malware analysis box – Cuckoo. I have tried many tutorials, both written and video recorded, and I could never make it work. Finally, I decided to work on it until I came up with the process that successfully deployed Cuckoo for me. This is me putting things together from different sources. I hope someone will find this entry helpful.

I am also a visual learner, so in addition to the text commands, I've also included screenshots of each step I took:

Requirements:

- [Ubuntu 18.04 < https://releases.ubuntu.com/18.04/ubuntu-18.04.6-desktop-amd64.iso>](https://releases.ubuntu.com/18.04/ubuntu-18.04.6-desktop-amd64.iso) – This is your Guest OS in VMWare Workstation
- [Windows 7 < http://cuckoo.sh/win7ultimate.iso>](http://cuckoo.sh/win7ultimate.iso) – This is your Guest OS in VirtualBox inside the Ubuntu Guest OS

- Enable virtualization within VMWare in Ubuntu VM:

Virtualization engine

- ☒ Virtualize Intel VT-x/EPT or AMD-V/RVI
- ☒ Virtualize CPU performance counters
- ☒ Virtualize IOMMU (IO memory management unit)

Steps:

1. Update Ubuntu box:

- ***sudo apt-get update***

```
jonald@cuckoo-demo:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Fetched 261 kB in 1s (416 kB/s)
Reading package lists... Done
```

2. Install required packages and apt repositories:

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

```
jonald@cuckoo-demo:~$ sudo apt-get -y install python python-pip python-dev libffi-dev libssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

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

```
jonald@cuckoo-demo:~$ sudo apt-get -y install python-virtualenv python-setuptools
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

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

```
jonald@cuckoo-demo:~$ sudo apt-get -y install libjpeg-dev zlib1g-dev swig
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

3. Install MongoDB:

- ***sudo apt-get -y install mongodb***

```
jonald@cuckoo-demo:~$ sudo apt-get -y install mongodb
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

4. Install PostgreSQL:

- ***sudo apt-get -y install postgresql libpq-dev***

```
jonald@cuckoo-demo:~$ sudo apt-get -y install postgresql libpq-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

5. Install VirtualBox:

- ***sudo apt-get -y install virtualbox***

```
jonald@cuckoo-demo:~$ sudo apt-get -y install virtualbox
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

6. Install tcpdump AppArmor:

- ***sudo apt-get -y install tcpdump apparmor-utils***

```
jonald@cuckoo-demo:~$ sudo apt-get -y install tcpdump apparmor-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

- ***sudo aa-disable /usr/sbin/tcpdump***

```
jonald@cuckoo-demo:~$ sudo aa-disable /usr/sbin/tcpdump
Disabling /usr/sbin/tcpdump.
```

7. Add a new group and add a user so you don't have to run as root:

- ***sudo adduser --disabled-password --gecos "" jonaldtest***

```
jonald@cuckoo-demo:~$ sudo adduser --disabled-password --gecos "" jonaldtest
Adding user `jonalddtest' ...
Adding new group `jonalddtest' (1001) ...
Adding new user `jonalddtest' (1001) with group `jonalddtest' ...
Creating home directory `/home/jonalddtest' ...
Copying files from `/etc/skel' ...
```

- ***sudo groupadd pcap***
- ***sudo usermod -a -G pcap jonaldtest***
- ***sudo chgrp pcap /usr/sbin/tcpdump***
- ***sudo setcap cap_net_raw,cap_net_admin=eip
/usr/sbin/tcpdump***

8. Verify the last command:

- **getcap /usr/sbin/tcpdump**

```
jonald@cuckoo-demo:~$ getcap /usr/sbin/tcpdump
/usr/sbin/tcpdump = cap_net_admin,cap_net_raw+eip
```

9. Install M2Crypto:

- **sudo pip install m2crypto**

```
jonald@cuckoo-demo:~$ sudo pip install m2crypto
```

10. Add the account you created in step 7 to vboxusers group:

- **sudo usermod -a -G vboxusers jonaldtest**

```
jonald@cuckoo-demo:~$ sudo usermod -a -G vboxusers jonaldtest
```

11. Create a virtual environment by using a script. Save the script as **cuckoo-setup-virtualenv.sh** (Shoutout to Josh Stroschein for the code) You can download the script at the end of this post:

12. Change the permission of the script:

- **sudo chmod +x cuckoo-setup-virtualenv.sh**

```
jonald@cuckoo-demo:~$ sudo chmod +x cuckoo-setup-virtualenv.sh
```

13. Run the script using your current logged-in user and not the one you created in step 7:

- **sudo -u jonald ./cuckoo-setup-virtualenv.sh**

```
jonald@cuckoo-demo:~$ sudo -u jonald ./cuckoo-setup-virtualenv.sh
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
```

14. Update your current shell environment:

- **source ~/.bashrc**

```
jonald@cuckoo-demo:~$ source ~/.bashrc
```

15. Create a virtual environment. You can name your virtual environment anything you want, I am using sandbox:

- ***mkvirtualenv -p python2.7 sandbox***

```
jonald@cuckoo-demo:~$ mkvirtualenv -p python2.7 sandbox
Running virtualenv with interpreter /usr/bin/python2.7
New python executable in /home/jonald/.virtualenvs/sandbox/bin/python2.7
Also creating executable in /home/jonald/.virtualenvs/sandbox/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
virtualenvwrapper.user_scripts creating /home/jonald/.virtualenvs/sandbox/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/jonald/.virtualenvs/sandbox/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/jonald/.virtualenvs/sandbox/bin/preactivate
virtualenvwrapper.user_scripts creating /home/jonald/.virtualenvs/sandbox/bin/postactivate
virtualenvwrapper.user_scripts creating /home/jonald/.virtualenvs/sandbox/bin/get_env_details
(sandbox) jonald@cuckoo-demo:~$
```

16. Setup and install cuckoo while you are inside your newly created virtual env (sandbox):

- ***pip install -U pip setuptools***

```
(sandbox) jonald@cuckoo-demo:~$ pip install -U pip setuptools
```

- ***pip install -U cuckoo***

```
(sandbox) jonald@cuckoo-demo:~$ pip install -U cuckoo
```

17. Create a directory to mount Windows 7 iso (open a new terminal):

- ***sudo mkdir /mnt/win7***
- ***sudo chown jonaldtest:jonaldtest /mnt/win7***

```
jonald@cuckoo-demo:~$ sudo mkdir /mnt/win7
[sudo] password for jonald:
jonald@cuckoo-demo:~$ sudo chown jonaldtest:jonaldtest /mnt/win7
```

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

```
jonald@cuckoo-demo:~$ sudo mount -o ro,loop win7ultimate.iso /mnt/win7
```

18. Install packages again just to make sure that there are no missing packages after everything that we have installed so far:

- ***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***


```
jonald@cuckoo-demo:~$ sudo apt-get -y install build-essential libssl-dev libffi-dev python-dev genisoimage
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.4ubuntu1).
build-essential set to manually installed.
genisoimage is already the newest version (9:1.1.11-3ubuntu2).
libffi-dev is already the newest version (3.2.1-8).
python-dev is already the newest version (2.7.15-rc1-1).
libssl-dev is already the newest version (1.1.1-1ubuntu2.1-18.04.20).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
jonald@cuckoo-demo:~$ sudo apt-get -y install zlib1g-dev libjpeg-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libjpeg-dev is already the newest version (8c-2ubuntu8).
zlib1g-dev is already the newest version (1:1.2.11.dfsg-0ubuntu2.2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
jonald@cuckoo-demo:~$ sudo apt-get -y install python-pip python-virtualenv python-setuptools swig
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-setuptools is already the newest version (39.0.1-2).
python-virtualenv is already the newest version (15.1.0+ds-1.1).
swig is already the newest version (3.0.12-1).
python-pip is already the newest version (9.0.1-2.3-ubuntu1.18.04.5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

19. Install vmcloak and run it (inside the virtual environment):

- ***pip install -U vmcloak***

```
(sandbox) jonald@cuckoo-demo:~$ pip install -U vmcloak
```

- ***vmcloak***

20. Create a HOST-ONLY network adapter using vmcloak:

- ***vmcloak-vboxnet0***

```
(sandbox) jonald@cuckoo-demo:~$ vmcloak-vboxnet0
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Interface 'vboxnet0' was successfully created
```

21. Setup Windows VM (this takes between 45 – 60 minutes):

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

22. Clone the Windows VM:

- ***vmcloak clone win7x64base win7x64cuckoo***

```
(sandbox) jonald@cuckoo-demo:~$ vmcloak clone win7x64base win7x64cuckoo
```

23. Install some basic software packages:

- ***vmcloak install win7x64cuckoo adobe-pdf pillow java flash vcredist vcredist.version=2015u3 wallpaper ie11 office office.version=2013 office.isopath=/home/jonald/Office_2013_Plus.iso***

office.serialkey= XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX

```
(sandbox) jonald@cuckoo-demo:~$ vmcloak install win7x64cuckoo adobe.pdf pillow java flash vcredist vcredist.version=2015u3 wallpaper test office office.version=2013
office.isopath=/home/jonald/Office 2013 Plus.iso office.serialkey=
```

24. Create the Windows VMs:

- **vmcloak snapshot --count 4 win7x64cuckoo 192.168.56.101**

```
(sandbox) jonald@cuckoo-demo:~$ vmcloak snapshot --count 4 win7x64cuckoo 192.168.56.101
```

25. View the list of VMs:

- **vmcloak list vms**

```
(sandbox) jonald@cuckoo-demo:~$ vmcloak list vms
```

26. Create the cuckoo directory where all config files get saved (still inside the virtual environment):

- **cuckoo init**

```
(sandbox) jonald@cuckoo-demo:~$ cuckoo init
```

```
Cuckoo Sandbox
no chance for malwares!
```

```
Cuckoo Sandbox 2.0.7
www.cuckoosandbox.org
Copyright (c) 2010-2018
```

27. Update cuckoo to the latest signature (you have to do this regularly so the signature gets updated with the current known threats signature):

- **cd .cuckoo/conf**
- **cuckoo community --force**

```
(sandbox) jonald@cuckoo-demo:~$ cd .cuckoo/conf
(sandbox) jonald@cuckoo-demo:~/.cuckoo/conf$ cuckoo community --force
2022-11-15 15:54:14,392 [cuckoo.apps.apps] INFO: Downloading.. https://
```

28. Open .cuckoo/conf/virtualbox.conf and change the **MODE** to GUI:

- **nano virtualbox.conf**
- **mode = gui (from headless)**

- *save the change*

```
[virtualbox]
# Specify which Virtual
# Can be "gui" or "head
# documentation to unde
mode = gui
```

29. Copy and paste the below command to add the 4 VMs we created to the conf file:

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

```
(sandbox) jonald@cuckoo-demo:~/cuckoo/conf$ while read -r vm ip; do cuckoo machine --add $vm $ip; done < <(vmcloak list vms)
```

30. Open **virtualbox.conf** again, remove **cuckoo1** under machines, delete everything after **controlports** and stop when you see the first IP address that matches the IP under machines:

- *Delete cuckoo1*

```
# Specify a comma-separated list of available machines to be used. For each
# specified ID you have to define a dedicated section containing the details
# on the respective machine. (E.g. cuckoo1,cuckoo2,cuckoo3)
machines = cuckoo1, 192.168.56.1011, 192.168.56.1012, 192.168.56.1013, 192.168.56.1014
```

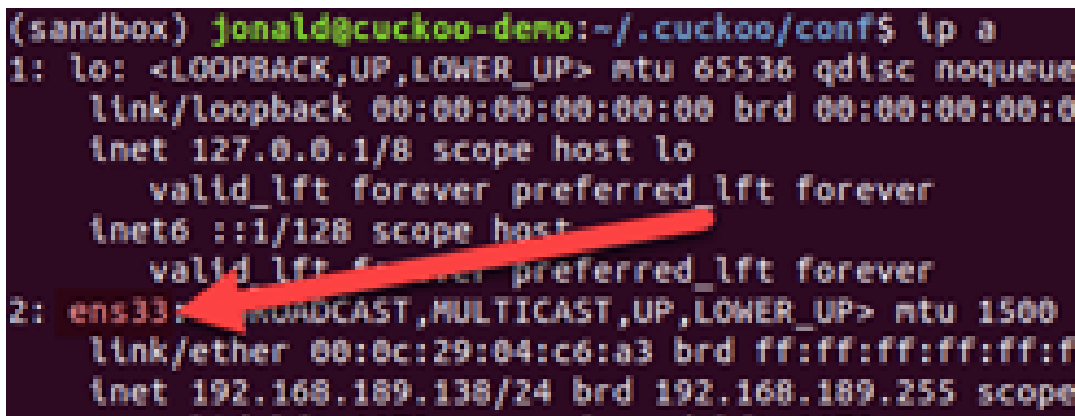
```
# If remote control is enable
# Virtualbox will bind the VR
controlports = 5000-5050
```

```
[192.168.56.1011]
```

```
# Specify the label name of t
# VirtualBox configuration.
```


31. Check your network adapter for the next setup steps:

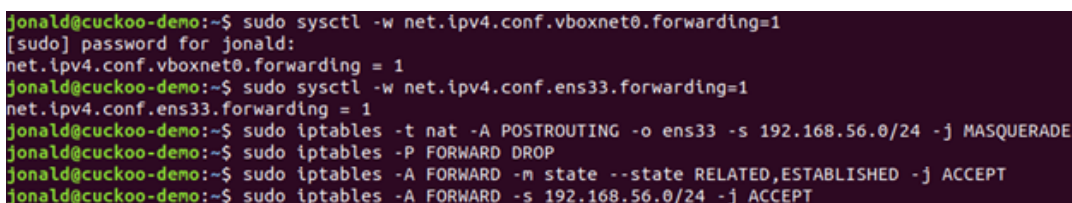
- ***ip a***



```
(sandbox) jonald@cuckoo-demo:~/.cuckoo/conf$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 00:0c:29:04:c6:a3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.189.138/24 brd 192.168.189.255 scope
```

32. Run the following commands inside of the virtual environment:

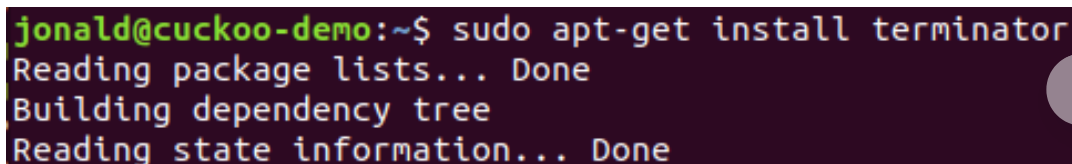
- ***sudo sysctl -w net.ipv4.conf.vboxnet0.forwarding=1***
- ***sudo sysctl -w net.ipv4.conf.ens33.forwarding=1***
- ***sudo iptables -t nat -A POSTROUTING -o ens33 -s 192.168.56.0/24 -j MASQUERADE***
- ***sudo iptables -P FORWARD DROP***
- ***sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT***
- ***sudo iptables -A FORWARD -s 192.168.56.0/24 -j ACCEPT***



```
jonald@cuckoo-demo:~$ sudo sysctl -w net.ipv4.conf.vboxnet0.forwarding=1
[sudo] password for jonald:
net.ipv4.conf.vboxnet0.forwarding = 1
jonald@cuckoo-demo:~$ sudo sysctl -w net.ipv4.conf.ens33.forwarding=1
net.ipv4.conf.ens33.forwarding = 1
jonald@cuckoo-demo:~$ sudo iptables -t nat -A POSTROUTING -o ens33 -s 192.168.56.0/24 -j MASQUERADE
jonald@cuckoo-demo:~$ sudo iptables -P FORWARD DROP
jonald@cuckoo-demo:~$ sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
jonald@cuckoo-demo:~$ sudo iptables -A FORWARD -s 192.168.56.0/24 -j ACCEPT
```

33. OPTIONAL: Install terminator:

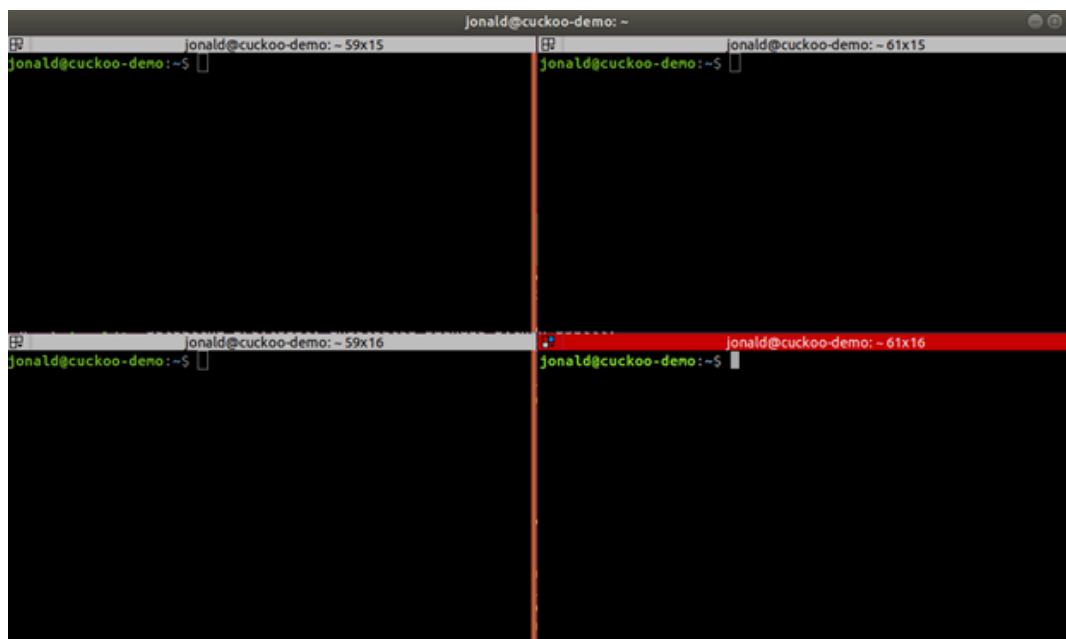
- ***sudo apt-get install terminator***



```
jonald@cuckoo-demo:~$ sudo apt-get install terminator
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

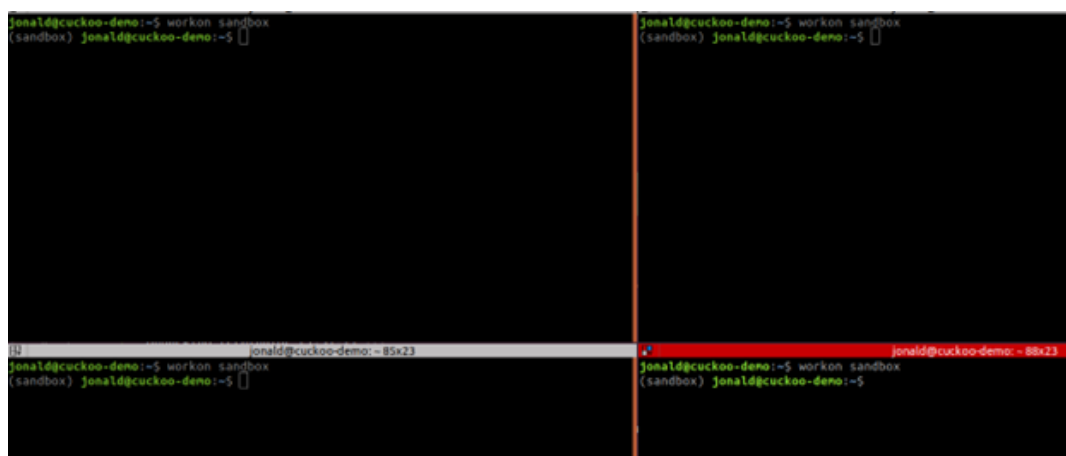
34. Open Terminator and split it to four windows:

- Split to 4 windows



35. Enter the virtual environment in all terminals:

- ***workon sandbox***



36. (Terminator window 1) type:

- ***cuckoo roter --sudo --group jonald***

```
(sandbox) jonald@cuckoo-demo:~$ cuckoo roter --sudo --group jonald
```

- Leave terminal window 1 running

37. Change the routing information. Open **routing.conf** and change the **internet** entry to your network adapter (ens33):

- ***internet = ens33***

```
# Network interface to use for outgoing traffic
# "dirty line" so to not break the network
# malicious traffic to the internet
# (For example, to use the internet)
internet = ens33
```

- save the file

38. Change the reporting information. Open **reporting.conf** and change the **MongoDB** entry to yes:

- ***enabled = yes***

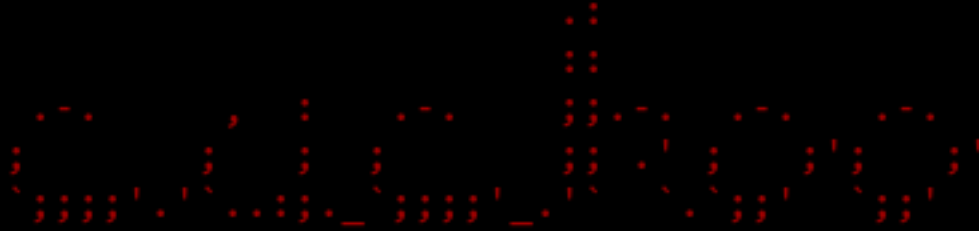
```
[mongodb]
enabled = yes
host = 127.0.0.1
port = 27017
db = cuckoo
store_memdump = yes
paginate = 100
```

- save the file

39. Do not touch the first window with command “cuckoo rooter -sudo -group jonald”. Go to another and start **cuckoo**:

- *cuckoo*

```
(sandbox) jonald@cuckoo-demo:~$ cuckoo
```



Cuckoo Sandbox 2.0.7
www.cuckoosandbox.org
Copyright (c) 2010-2018

Checking for updates...
You're good to go!

- Leave terminal window 2 running

40. In a third terminal, start the cuckoo web server:

- ***cuckoo web -host 127.0.0.1 -port 8080***

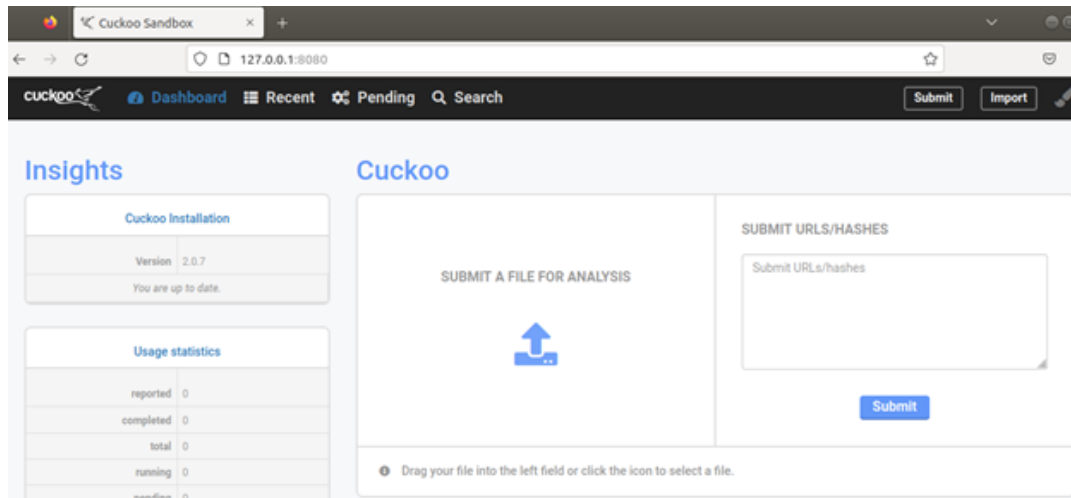
```
(sandbox) jonald@cuckoo-demo:~$ cuckoo web --host 127.0.0.1 --port 8080
Performing system checks...

System check identified no issues (0 silenced).
November 15, 2022 - 16:38:39
Django version 1.8.4, using settings 'cuckoo.web.web.settings'
Starting development server at http://127.0.0.1:8080/
Quit the server with CONTROL-C.
```

- Leave terminal window 3 running

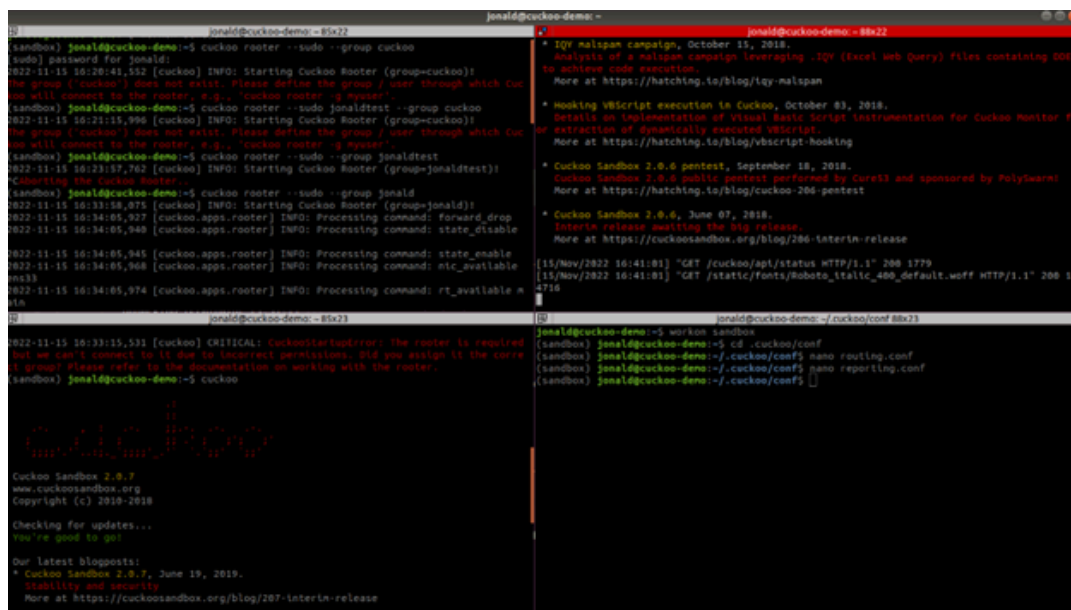
41. Open a web browser to access the cuckoo web interface:

- <http://127.0.0.1:8080> < <http://127.0.0.1:8080>>



42. This is how your Terminator should look like while using cuckoo sandbox:

- Terminator



- Don't forget to take a snapshot of your Ubuntu VM. I would recommend reverting to a snapshot after a verified malicious file analysis. You don't want to take the risk of an o-day VM escape vulnerability. Haha

I hope you find this helpful. If you haven't subscribed here, please do so.

Below is the script I used in step 11.

```
#!/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
```

Published by lightkunyagami <https://tryhackme.com/badge/18276>

[View more posts <](#)

[https://beginninghacking.net/author/lightkunyagami/>](https://beginninghacking.net/author/lightkunyagami/)

Join the Conversation



2 Comments



jdmorto

November 16, 2022 at 4:21 pm < <https://beginninghacking.net/2022/11/16/how-to-setup-your-own-malware-analysis-box-cuckoo-sandbox/#comment-58>>

This will be so helpful and come in handy. Thank you!

★ < https://beginninghacking.net/2022/11/16/how-to-setup-your-own-malware-analysis-box-cuckoo-sandbox/?like_comment=58&_wpnonce=b99dde5ea4 >
Like

Pingback:

[Week 47 – 2022 – This Week In 4n6 < http://thisweekin4n6.com/2022/11/20/week-47-2022/>](http://thisweekin4n6.com/2022/11/20/week-47-2022/)

N00b_H@ck3r < <https://beginninghacking.net/> > , Website Powered by WordPress.com < https://wordpress.com/?ref=footer_custom_powered > .