9/29/2021

Cuckoo Installation Guide

For Dynamic Malware analysis

Afif Hidayatullah
THREAT LAB INDONESIA

What is Cuckoo ?

Cuckoo is open source sandbox used for dynamic analysis of a malware file.

Minimum requirements for Cuckoo Installation

- 2 cores
- 6 GB of RAM
- 250 HD
- KVM as a hypervisor or virtualbox
- Ubuntu OS (**recommend for host os***)
- Windows 7 to analyze windows malware or if you want to analyze Linux and Android malware, you can install Linux OS like Ubuntu and for android you can install Android OS (**for guest os**).

Noted: *in this case I use virtualbox as guest os analyst.*

Installation Cuckoo Packages

1. First, you must update and upgrade os ubuntu.


```
sudo apt update -y && sudo apt upgrade -y
```

2. Please follow this command to install package.

```
sudo apt install python python3-pip python-dev libffi-dev libssl-dev &&
sudo apt install python-setuptools && pip3 install virtualenv && sudo
apt install python3-virtualenv && sudo apt install libjpeg-dev zlib1g-
dev swig && sudo apt install mongodb && sudo apt install postgresql
libpq-dev && sudo apt install virtualbox && sudo apt install tcpdump
apparmor-utils && sudo apt-get install curl && curl https://boot-
strap.pypa.io/pip/2.7/get-pip.py --outpu get-pip.py && sudo python2 get-
pip.py && sudo apt-get install -y libmagic-dev
```

3. If you want to have your own cuckoo user, you can follow the useradd command or you will just use your default user account, you can skip the useradd command and follow the second command.

    - `sudo useradd cuckoo`
    - `sudo usermod -a -G vboxusers cuckoo`
    - `sudo usermod -aG sudo cuckoo`

    noted: *"cuckoo name is your user account name which will be used for group vboxusers".*

4. Install additional packages.

```
sudo apt-get install -y libffi-dev libssl-dev libfuzzy-dev libtool flex autoconf libjansson-dev git
```

5. You need install *psycopg2, distorm3==3.4.4, psycopg, pycrypto, openpyxl and* please follow this command.

```
sudo -H pip install distorm3==3.4.4
sudo -H pip install psycopg2 pycrypto openpyxl
```

6. Next, install *pydeep* and *volatitily*.

```
sudo -H pip install git+https://github.com/kbandla/pydeep.git
sudo -H pip install git+https://github.com/volatilityfoundation/volatility.git
```

7. Install *pyopenssl*.

```
sudo -H pip install pyopenssl -U
```


8. Install *yara*.

```
git clone https://github.com/VirusTotal/yara.git && cd yara* && sudo ./bootstrap.sh && sudo ./configure --enable-cuckoo --enable-magic --enable-dotnet && sudo make && sudo make install
```

note: *"if you yara error like this "yara: error while loading shared libraries: libyara.so.8: cannot open shared object file: No such file or directory", you can use command in bellow"*:

- `sudo echo "/usr/local/lib" >> /etc/ld.so.conf (if this command doesn't work or doesn't solve the problem, you can skip to second command)`
- `sudo ldconfig`

9. Install *ssdeep*.

```
sudo apt-get install -y ssdeep
```

10. Now we will install tcpdump to enable packet capture analysis.

```
sudo groupadd pcap && sudo usermod -a -G pcap cuckoo && sudo chgrp pcap /usr/sbin/tcpdump && sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump && sudo aa-disable /usr/sbin/tcpdump
```

11. Check if every one is correct.

```
getcap /usr/sbin/tcpdump
```

12. Make environment python2.

```
python3 -m virtualenv -p /usr/bin/python2.7 sandbox
```

13. Now, we can install cuckoo sanbox.

```
pip install setuptools
pip install -U cuckoo
cuckoo init
cuckoo community
```

14. Basically in Ubuntu OS, there is no net-tools to use ifconfig. We need to install net-tools to setup the virtualbox.

- `sudo apt install -y net-tools`
- `ifconfig`

15. Add host-only network adapter and IP address on virtualbox.

- `vboxmanage hostonlyif create`
- `vboxmanage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1`
- `ifconfig`

16. We will create a script so that these changes persist on reboot and automatically run system startup. Follow the commands bellow:

Create a folder in systemd with the name vboxhostonly.

- `sudo mkdir /opt/systemd/`
- `sudo nano /opt/systemd/vboxhostonly`

Put this script bash in vboxhostonly.

```
#!/bin/bash

vboxmanage hostonlyif create
vboxmanage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1
```

If you have finished entering the script above. please save with command **ctrl+x** then **y.**

17. Change permission vboxhostonly.

- `cd /opt/systemd/`
- `sudo chmod a+x vboxhostonly`

18. We need to create a service.

- `sudo touch /etc/systemd/system/vboxhostonlynic.service`
- `sudo nano /etc/systemd/system/vboxhostonlynic.service`

Then, you have to put below script to `vboxhostonlynic.service.`

```
Description=Setup VirtualBox Hostonly Adapter
After=vboxdrv.service

[Service]
Type=oneshot
ExecStart=/opt/systemd/vboxhostonly

[Install]
WantedBy=multi-user.target
```

19. Now, after configuring the service, we can load the deamon-reload.

- `systemctl daemon-reload`
- `systemctl enable vboxhostonlynic.service`

Setup Guest OS for Analysis Malware

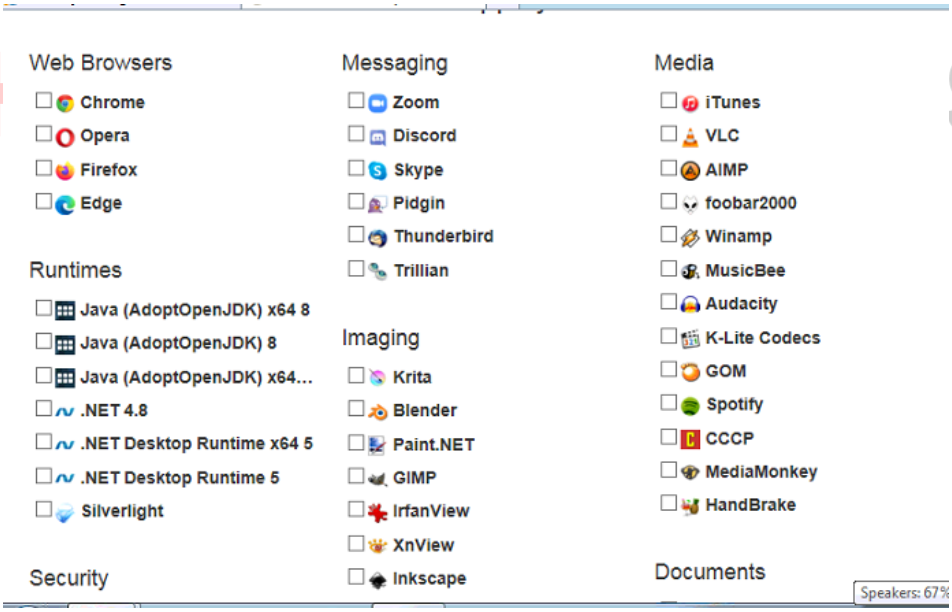
1. For windows 7 iso, you can download it below.

   Url download google drive: https://bit.ly/2WotcYJ

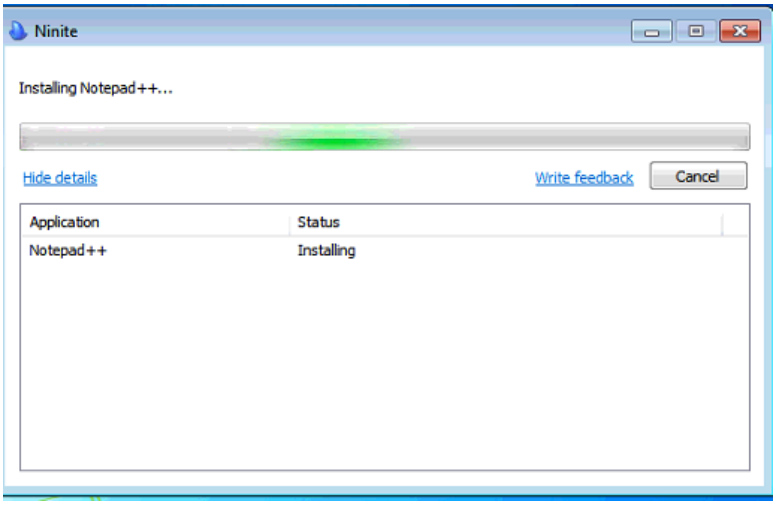
2. After the download is complete you can install the windows os. then if the windows 7 installation is complete, please install the windows dependencies needed to run malware programs and other general needs. In this case I use ninite.

   - Ninite (https://ninite.com/)
   - Chocolatey (https://chocolatey.org/install)

   Noted: "*don't forget to change internet explorer settings so you can download files or make them vulnerable*".



You can choose what you need and if have done download, next install ninite on your windows 7 guest. Example when installing:

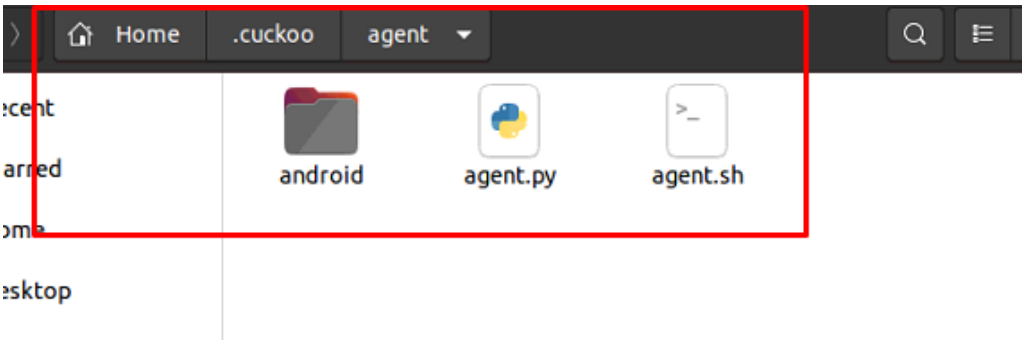


3. Now we need to install Python 2.7 for windows. You can download it from https://www.python.org/ftp/python/2.7.8/python-2.7.8.amd64.msi.

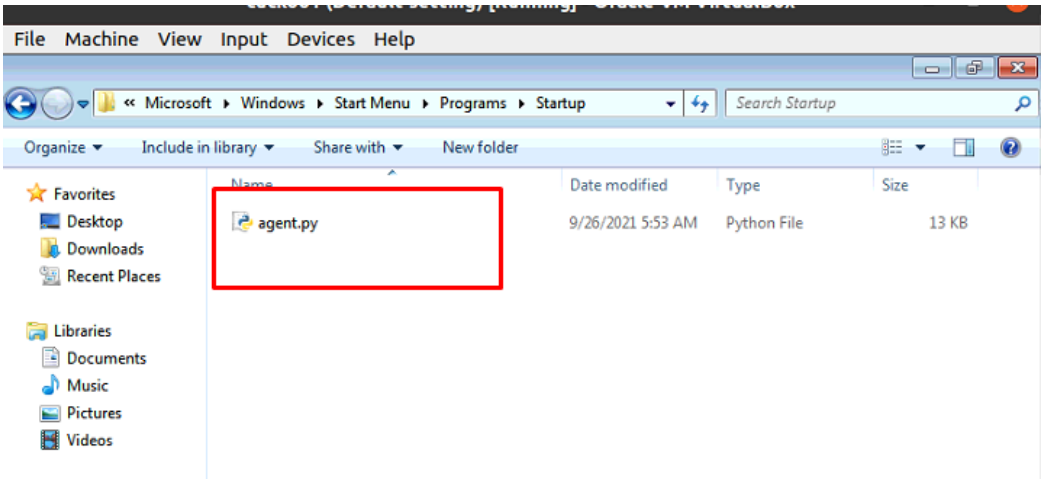
4. Next download Python Pillow from https://pypi.python.org/packages/2.7/P/Pillow/Pillow-2.5.3.win-amd64-py2.7.exe#md5=33c3a581ff1538b4f79b4651084090c8.

5. For the get result analysis in windows to cuckoo, we need place file `agent.py` to windows 7 start program folder.

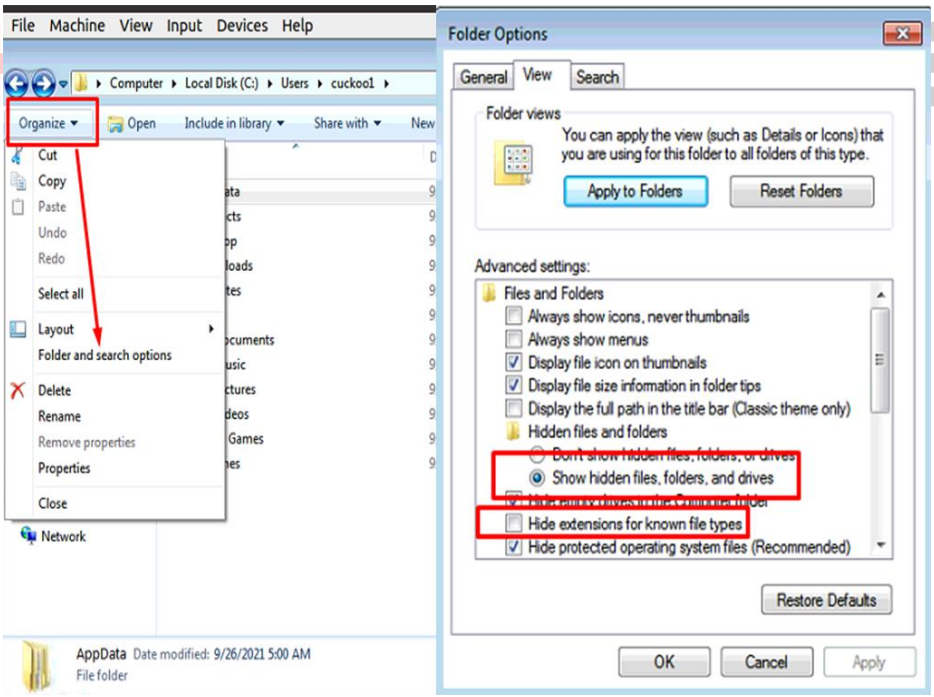
   You can find `agent.py` in `~/.cuckoo/agent` or `$USER/.cuckoo/agent` folder on directory Ubuntu host.



Copy the `agent.py` file and place it in `C:\Users\*USERNAME*\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup` this will then start the `agent.py` on boot up of the Virtual Machine.



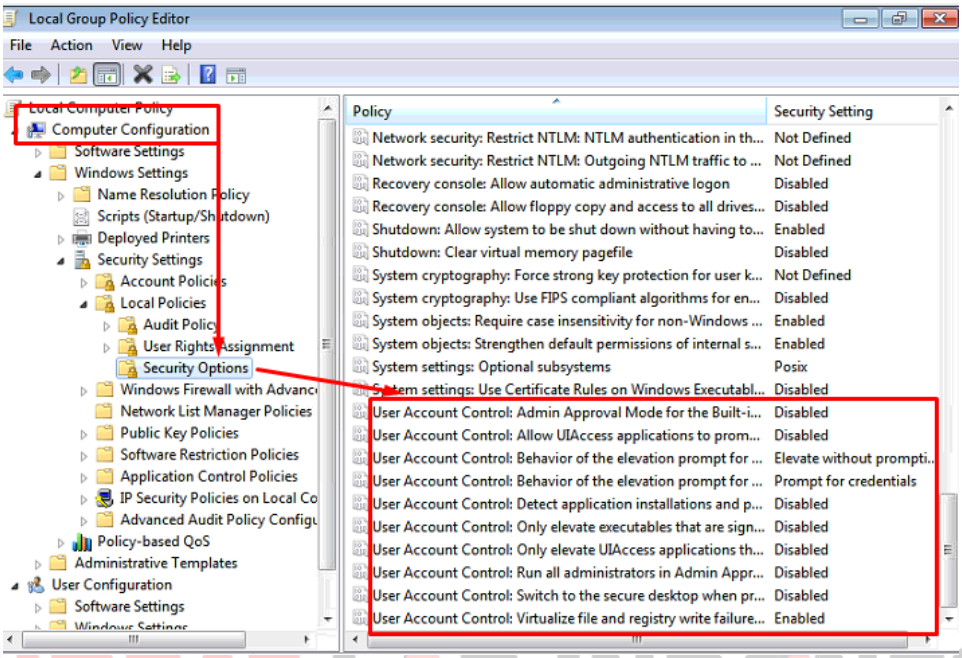
Noted: *"You will need to Show hidden files and folders for the AppData folder to be seen".*



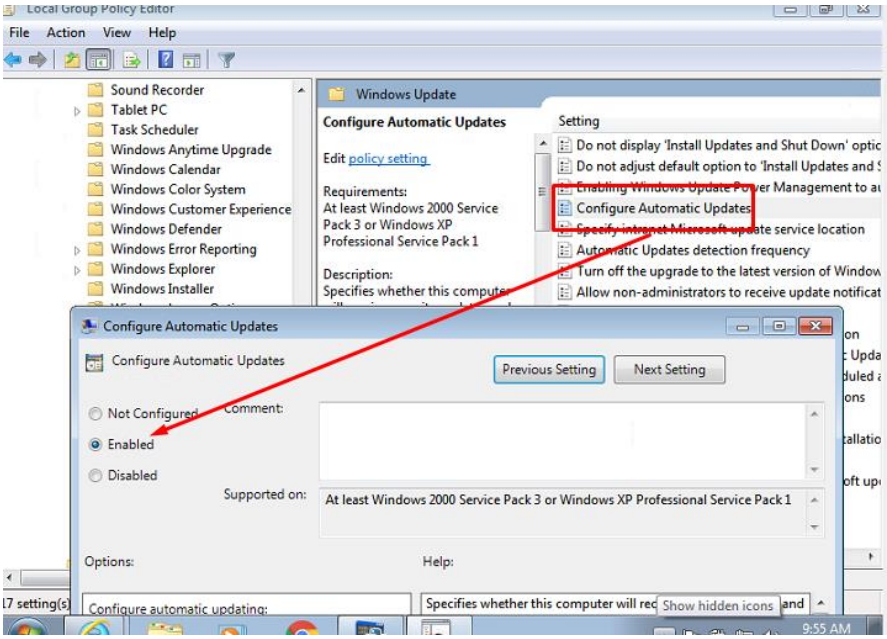
6. In Windows 7, we will change the protection settings to make analysis easier.

   Go to start and type in Group. You should see the Edit Group Policy option.

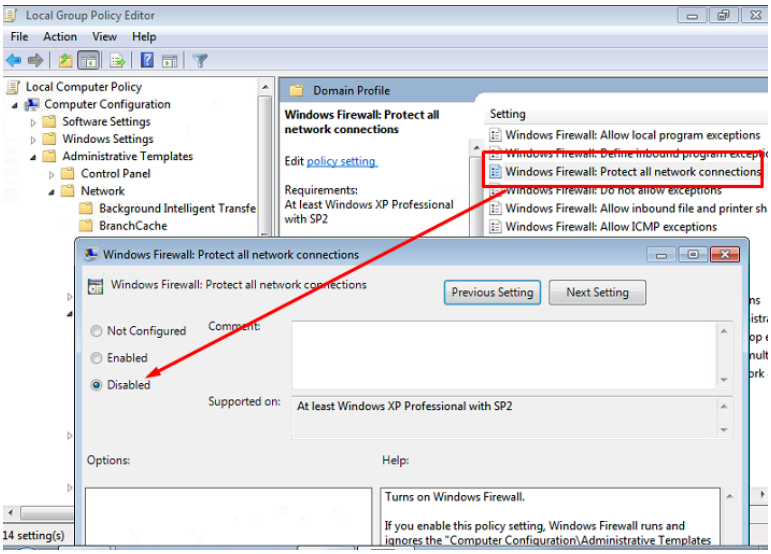
Expand Computer `Configuration` > `Windows Settings` > `Security Settings` > `Local Policies` > `Security Options`. Scroll down to the `User Account Control` options. Please follow this setup.



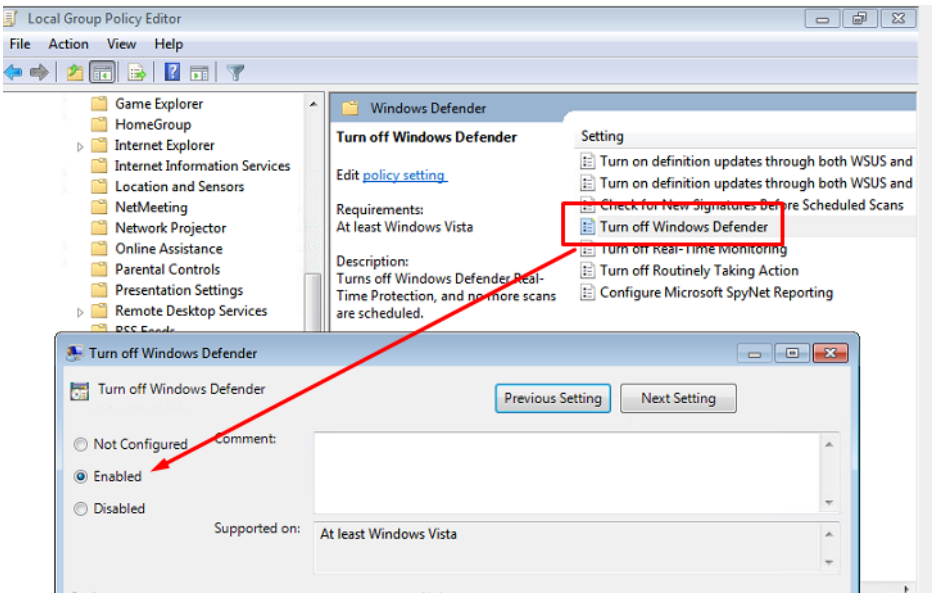
7. Disable the automatic install of Windows Updates. Go to Computer `Configuration` > `Administrative Templates` > `Windows Components` > `Windows Update` and right click `Configure Automatic Updates` and edit to Enabled and `2- Notify for download and notify for install`, then click ok.



8. Remove the Windows protection from the network. Go to `Computer Configuration` > `Administrative Templates` > `Network` > `Network connections` > `Windows Firewall` > `Domain Profile` > `Windows Firewall` and change "`Protect all network connections`" to Disabled.

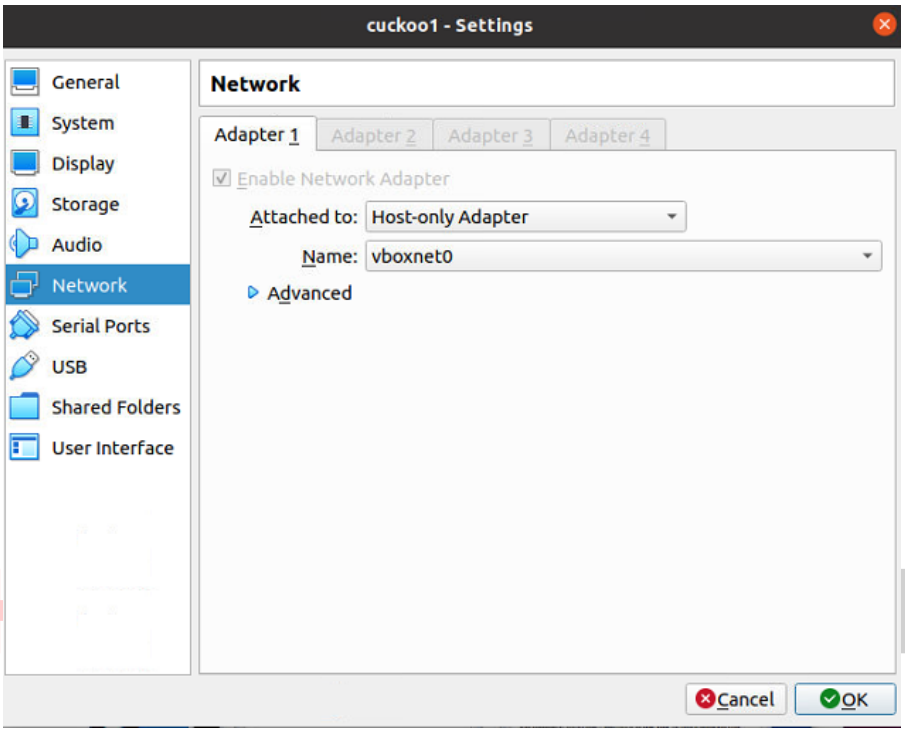


9. Next, disable Windows Defender. you can do with `Computer Configuration` > `Administrative Templates` > `Windows Components` > `Windows Defender Antivirus` then set "`Turn off Windows Defender Antivirus`" to Enabled.

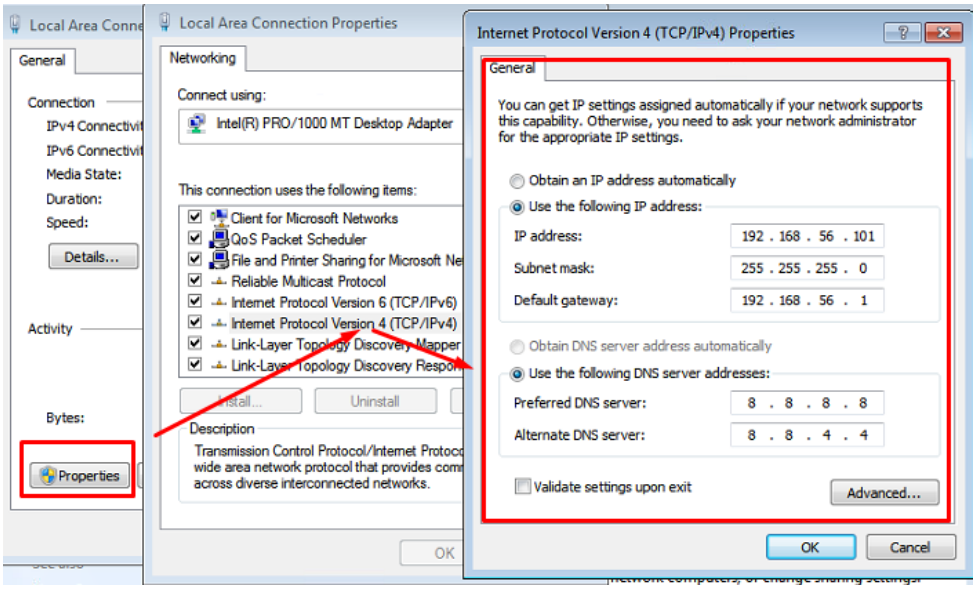


10. Then make sure, if UAC (User Account Control, Windows Firewall, and Windows Defender have been disabled.

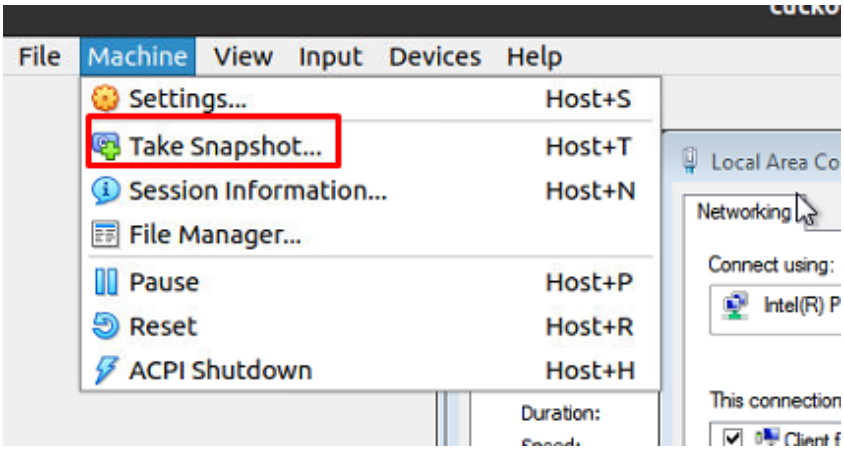
11. Turn off guest windows 7 and change network adapter to hostonly and vboxnet0.



12. Configuration to static ipv4 on windows 7.



13. We need to make a snapshot of the windows 7 machine, so that every malware infection to windows 7, we can return to the initial settings.



Configuration Firewall Ubuntu to NAT

This step so that host-only adapter has connection to internet.

1. Install iptables-persistent.

```
sudo apt-get install -y iptables-persistent
```

2. Check your rules by running.

```
sudo iptables -L
```

3. Follow this command for forward ipv4.

- `echo 1 | sudo tee -a /proc/sys/net/ipv4/ip_forward`
- `sudo sysctl -w net.ipv4.ip_forward=1`

4. Make sure that the IP forwarding starts up after a reboot.

```
sudo nano /etc/sysctl.conf
```

```
remove the # from net.ipv4.ip_forward=1
then add net.ipv4.conf.all.proxy_arp = 1 below it
```

5. Configure iptables to open NAT share connections.

    - `sudo iptables -t nat -A POSTROUTING -o eth0 -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`
    - `sudo iptables -L`

6. Get access root.

    ```
    sudo su
    ```

7. We will save iptables configuration to iptables rules.v4.

    - `iptables-save > /etc/sysconfig/iptables`
    - `iptables-save > /etc/iptables/rules.v4`
    - `service iptables restart`

Configuration Cuckoo

1. Create database in postgresql.

    ```
    sudo -u postgres psql

    CREATE DATABASE cuckoo;
    CREATE USER cuckoo WITH ENCRYPTED PASSWORD 'password';
    GRANT ALL PRIVILEGES ON DATABASE cuckoo TO cuckoo;
    \q
    ```

2. Change directory to cuckoo folder.

    ```
    cd .cuckoo/conf
    ```

3. Edit file `cuckoo.conf`.

    following are set:
    ```
    machinery = virtualbox
    memory_dump = yes
    ```

```
# Specify the name of the machinery module to use, this module will
# define the interaction between Cuckoo and your virtualization software
# of choice.
machinery = virtualbox

# Enable creation of memory dump of the analysis machine before shutting
# down. Even if turned off, this functionality can also be enabled at
# submission. Currently available for: VirtualBox and libvirt modules (KVM).
memory_dump = yes
```

resultserver ip = 192.168.56.1

```
[resultserver]
# The Result Server is used to receive in real time the behavioral logs
# produced by the analyzer.
# Specify the IP address of the host. The analysis machines should be able
# to contact the host through such address, so make sure it's valid.
# NOTE: if you set resultserver IP to 0.0.0.0 you have to set the option
# `resultserver_ip` for all your virtual machines in machinery configuration.
ip = 192.168.56.1

# Specify a port number to bind the result server on. Set to 0 to use a random
# port.
port = 2042

# Maximum size of uploaded files from VM (screenshots, dropped files, log).
# The value is expressed in bytes, by default 128 MB.
upload_max_size = 134217728
```

Change the connection = line to:
connection = postgresql://cuckoo:password@localhost/cuckoo


```
[database]
# Specify the database connection string.
# NOTE: If you are using a custom database (different from sqlite), y
# use utf-8 encoding when issuing the SQL database creation statement
# Examples, see documentation for more:
# sqlite:///foo.db
# postgresql://foo:bar@localhost:5432/mydatabase
# mysql://foo:bar@localhost/mydatabase
# If empty, defaults to a SQLite3 database at $CWD/cuckoo.db.
connection = postgresql://cuckoo:password@localhost/cuckoo
```

4. Edit file auxiliary.conf.

sudo nano auxiliary.conf
ensure that the sniffer and mitm is enabled = yes

```
[sniffer]
# Enable or disable the use of an external sniffer (tcpdump) [yes/no].
enabled = yes

# Specify the path to your local installation of tcpdump. Make sure this
# path is correct.
tcpdump = /usr/sbin/tcpdump

# We used to define the network interface to capture on in auxiliary.conf, but
# this has been moved to the "interface" field of each Virtual Machinery
# configuration.

# Specify a Berkeley packet filter to pass to tcpdump.
# Note: packer filtering is not possible when using "nictrace" functionality
# from VirtualBox (for example dumping inter-VM traffic).
bpf =

[mitm]
# Enable man in the middle proxying (mitmdump) [yes/no].
enabled = yes

# Specify the path to your local installation of mitmdump. Make sure this
# path is correct.
mitmdump = /usr/local/bin/mitmdump
```

5. Edit file `virtualbox.conf`.

   `sudo nano virtualbox.conf`
   ensure that `virtualbox` mode `=` `gui` and machines `=` `cuckoo1` (make sure the created windows virtual machine name matches yours)

```
[virtualbox]
# Specify which VirtualBox mode you want to run your machines on.
# Can be "gui" or "headless". Please refer to VirtualBox's official
# documentation to understand the differences.
mode = gui

# Path to the local installation of the VBoxManage utility.
path = /usr/bin/VBoxManage
# If you are running Cuckoo on Mac OS X you have to change the path as follows
# path = /Applications/VirtualBox.app/Contents/MacOS/VBoxManage

# Default network interface.
interface = vboxnet0

# 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
# of the respective machine. (E.g. cuckoo1,cuckoo2,cuckoo3)
machines = cuckoo1

# If remote control is enabled in cuckoo.conf, specify a port range to use.
# Virtualbox will bind the VRDP interface to the first available port.
controlports = 5000-5050


[cuckoo1]
# Specify the label name of the current machine as specified in your
# VirtualBox configuration.
label = cuckoo1

# Specify the operating system platform used by current machine
# [windows/darwin/linux].
platform = windows
```

Make sure label = `cuckoo1` and platform = `windows` and ip = `192.168.56.101` then Ctrl + X to exit , Y to save and enter to write file.

```
[cuckoo1]
# Specify the label name of the current machine as specified in your
# VirtualBox configuration.
label = cuckoo1

# Specify the operating system platform used by current machine
# [windows/darwin/linux].
platform = windows

# Specify the IP address of the current virtual machine. Make sure that the
# IP address is valid and that the host machine is able to reach it. If not,
# the analysis will fail.
ip = 192.168.56.101
```

6. Edit file `processing.conf`.

   sudo nano processing.conf
   ensure that `memory` enabled = `yes`

```
[memory]
# Create a memory dump of the entire Virtual Machine. This memory dump will
# then be analyzed using Volatility to locate interesting events that can be
# extracted from memory.
enabled = yes
```

7. Edit file `memory.conf`.

   `sudo nano memory.conf`
   ensure that `basic` guest_profile = `Win7SP1x64`

```
# Volatility configuration

# Basic settings
[basic]
# Profile to avoid wasting time identifying it
guest_profile = Win7SP1x64
```

8. Edit file `reporting.conf`.

   `sudo nano reporting.conf`
   ensure that `singlefile` Enable creation of report.html enabled = `yes`, report.html/report.pdf enabled = `yes`, report.pdf enabled = `yes` and mongodb enabled = `yes`

```
[singlefile]
# Enable creation of report.html and/or report.pdf?
enabled = yes
# Enable creation of report.html?
html = yes
# Enable creation of report.pdf?
pdf = yes

[misp]
enabled = no
url =
apikey =

# The various modes describe which information should be submitted to MISP,
# separated by whitespace. Available modes: maldoc ipaddr hashes url.
mode = maldoc ipaddr hashes url

distribution = 0
analysis = 0
threat_level = 4

# The minimum Cuckoo score for a MISP event to be created
min_malscore = 0

tag = Cuckoo
upload_sample = no

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

Configuration Cuckoo

If everything is done, you can run cuckoo with the command below.

```
source sandbox/bin/activate
```

```
cuckoo web
```

```
cuckoo@cuckoo:~$ source sandbox/bin/activate
(sandbox) cuckoo@cuckoo:~$ cuckoo web
Performing system checks...

System check identified no issues (0 silenced).
October 01, 2021 - 11:34:32
Django version 1.8.4, using settings 'cuckoo.web.web.settings'
Starting development server at http://localhost:8000/
```

Then run debug to check if there is any error in cuckoo during analysis.

```
cuckoo -d
```

```
cuckoo@cuckoo:~$ source sandbox/bin/activate
(sandbox) cuckoo@cuckoo:~$ cuckoo -d

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

2021-10-01 11:34:52,260 [cuckoo] DEBUG: Increasing resource limit for number of open files to 1048576
Checking for updates...
You're good to go!
```

Cuckoo web view