**Cuckoo Installation**

**Prerequisites:**

* OS: Ubuntu 20.04 (<https://releases.ubuntu.com/20.04/>)
* RAM: 16 GB recommended
* Storage: 500 GB (will depend on usecase)
* Internet Connection
* Avoid the name “cuckoo” as usernames to avoid conflicts with below commands.
* If Ubuntu is in a VM, make sure BT-x is compatible and enabled.
* Enable VT-x or AMD-V from BIOS (<https://bce.berkeley.edu/enabling-virtualization-in-your-pc-bios.html> [Ignore this point --> Delete any existing VMs])
* Windows 7 64-bit ISO (<https://getintopc.com/softwares/operating-systems/windows-7-ultimate-32-64-bit-updated-aug-2020-download-1747331/?id=000265748703>)

**Cuckoo Installation Steps:**

* Boot into Ubuntu
* Open Terminal (Ctrl+Alt+T)
* Update and upgrade the ***apt***: (Will take time)
  + sudo apt-get update && sudo apt-get upgrade -y
* Create user “cuckoo” and add them to sudo:
  + sudo adduser cuckoo
  + sudo adduser cuckoo sudo
* Installing Applications, Packages and Libraries needed by cuckoo in Ubuntu:
  + sudo apt-get install -y python-dev libffi-dev libssl-dev libfuzzy-dev libtool flex autoconf libjansson-dev git curl python mongodb postgresql libpq-dev python-setuptools libjpeg-dev zlib1g-dev swig ssdeep net-tools
* Download and Install pip 2.7
  + curl [https://bootstrap.pypa.io/pip/2.7/get-pip.py -o pip.py](https://bootstrap.pypa.io/pip/2.7/get-pip.py%20-o%20get-pip.py)
  + sudo python pip.py
* Download and Install Volatility
  + git clone <https://github.com/volatilityfoundation/volatility.git>
  + sudo python setup.py build
  + sudo python setup.py install
* Python Tools
  + sudo -H pip install distorm3==3.4.4
  + sudo -H pip install yara-python==3.6.3
  + sudo -H pip install pydeep
  + sudo -H pip install openpyxl
  + sudo -H pip install ujson
  + sudo -H pip install jupyter (ignore the error [**pip install jupyter[notebook] ipywidgets==7.5**])
* Installing TCP Dump for network packet analysis
  + sudo apt install -y tcpdump libcap2-bin apparmor-utils
  + sudo setcap cap\_net\_raw,cap\_net\_admin=eip /usr/sbin/tcpdump
  + getcap /usr/sbin/tcpdump (above line should be printed at this command’s execution)
  + sudo aa-disable /usr/sbin/tcpdump
* Installing Cuckoo
  + sudo -H pip install -U cuckoo
  + cuckoo
* Installation and Configuration of VirtualBox
  + curl https://www.virtualbox.org/download/oracle\_vbox\_2016.asc | gpg --dearmor > oracle\_vbox\_2016.gpg
  + curl https://www.virtualbox.org/download/oracle\_vbox.asc | gpg --dearmor > oracle\_vbox.gpg
  + sudo install -o root -g root -m 644 oracle\_vbox\_2016.gpg /etc/apt/trusted.gpg.d/
  + sudo install -o root -g root -m 644 oracle\_vbox.gpg /etc/apt/trusted.gpg.d/
  + echo "deb [arch=amd64] http://download.virtualbox.org/virtualbox/debian $(lsb\_release -sc) contrib" | sudo tee /etc/apt/sources.list.d/virtualbox.list
  + sudo apt update
  + sudo apt install -y linux-headers-$(uname -r) dkms
  + sudo apt install virtualbox-6.1 -y
  + Host Only Network Adapter Configuration
  + vboxmanage hostonlyif create
  + vboxmanage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1 (Change in IP may result in issues)

For reboot persistence

* + - sudo mkdir /opt/systemd/
    - sudo nano /opt/systemd/vboxhostonly (write the below code and save the file)

!/bin/bash  
hostonlyif create  
vboxmanage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1

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

Creating and Executing Service

* + - sudo nano /etc/systemd/system/vboxhostonlynic.service (write the below code and save the file)

Description=Setup VirtualBox Hostonly Adapter

After=vboxdrv.service  
  
[Service]  
Type=oneshot  
ExecStart=/opt/systemd/vboxhostonly  
  
[Install]  
WantedBy=multi-user.target

* + - sudo systemctl daemon-reload
    - sudo systemctl enable vboxhostonlynic.service
    - sudo systemctl start vboxhostonlynic.service
* VM Windows Installation and Configuration
  + Install Windows 7 on the Virtual Machine with the following specs:
    - VM Label: win7x64
    - OS: Windows 7 x64
    - RAM: 2 GB (memory dump will be of 2 GB for each analysis)
    - NETWORK: NAT (Switch to Host Only Adapter while cuckoo is running)
    - CPU: 4 (i5 8300H has 8 logical cores; Use half of your cpu’s core)
    - Storage Size: 32 GB [use fixed type for better performance]
  + Boot to Windows 7
  + To disable forced signed driver installation in x64 windows, launch command prompt as admin and execute:
    - bcdedit -set TESTSIGNING ON
  + In VM toolbar, go to, Devices --> Insert Guest Additions CD Image
  + Go to Windows 7 and install VirtualBox Guest Additions and Reboot
  + Open run dialog (Win+R) and type ‘gpedit.msc’:
    - Expand Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options. Scroll down to the User Account Control options:
      * Right click the “User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode” and select Properties. Using the drop-down menu select Elevate without prompting and click ok.
      * Right click the “User Account Control: Detect application installations and prompt for elevation” and select Properties. Check the Disabled option and click ok.
      * Right click the “User Account Control: Run all administrators in Admin Approval Mode” and select Properties. Check the Disabled option and click ok.
    - Disable the automatic install of Windows Updates: Computer Configuration > Administrative Templates > Windows Components > Windows Update and right click Configure Automatic Updates and edit. Select Enabled, 2- Notify for download and notify for install, then click ok.
    - Disable Firewall: Computer Configuration > Administrative Templates > Network > Network connections > Windows Firewall > Domain Profile > Windows Firewall and change “Protect all network connections” to Disabled.
    - Disable Windows Defender: Computer Configuration > Administrative Templates > Windows Components > Windows Defender Antivirus then set “Turn off Windows Defender Antivirus” to Enabled.

* + In VM window, go to Settings > Shared Folders, and create a shared folder (move files from Ubuntu to Windows).
  + Place the following files in the Shared Folder in Ubuntu:
    - Python 2.7: <https://www.python.org/ftp/python/2.7.8/python-2.7.8.amd64.msi>
    - Pillow 2.5.3: <https://pypi.python.org/packages/2.7/P/Pillow/Pillow-2.5.3.win-amd64-py2.7.exe>
    - Agent.py: /home/<username>/.cuckoo/agent/agent.py
    - Any other files to make the VM look like an authentic target. (MS Office 2013, Java, Flash, Acrobat Reader)
  + Access the folder in Windows 7 by going to \\vboxsvr\ in the network and:
    - Install all programs
    - Move agent.py to “C:\Users\***\*USERNAME\****\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup”

(Show Hidden Files: **Organize ->Folder Options->View->Show Hidden files, folders and drives**)

(Moving is optional as snapshot will have agent.py running)

* + Activate Windows (Optional) (<https://github.com/DEAM0/Windows-7-Activator> : Run the .bat file)
  + Reboot Windows and if network popup appears, click allow.
  + Internet Setup for vboxnet0 (Host Only Network Adapter)
    - sudo apt install -y iptables-persistent (choose yes when prompts to save current config)
    - Find adapter name with internet. In terminal, execute “ifconfig” and copy the name of the adapter. In my case it is ‘ens33’

A computer screen shot of a program

Description automatically generated

* + - sudo iptables -A FORWARD -o ens33 -i vboxnet0 -s 192.168.56.0/24 -m conntrack --ctstate NEW -j ACCEPT
    - sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
    - sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
    - Check rules with: sudo iptables -L

IP Forwarding in Kernel

* + - echo 1 | sudo tee -a /proc/sys/net/ipv4/ip\_forward
    - sudo sysctl -w net.ipv4.ip\_forward=1

IP Forwarding after rebootsudo nano /etc/sysctl.conf (remove # from net.ipv4.ip\_forward=1)

* + - sudo su
    - iptables-save > /etc/iptables/rules.v4
    - exit
  + In VirtualBox, change Windows 7 VM Network Adapter to Host Only Adapter and select vboxnet0 as interface
  + In Windows 7, go to Adapter Settings (Start > Control Panel > Network and Sharing Center > Change adapter settings > Properties > Properties > Internet Protocol Version 4 (TCP/IPv4) Properties), set the following values after selecting “Use the following IP address:”:
    - IP Address: 192.168.56.101
    - Subnet Mask: 255.255.255.0
    - Default Gateway: 192.168.56.1
    - Preferred DNS Server: 8.8.8.8

* + Disable and then Enable the Network Adapter
  + Perform ‘ping google.com’ to confirm internet access
  + Make sure ‘agent.py’ is running (blank command prompt)
  + Ensure output when ‘curl 192.168.56.101:8000’ is executed in Ubuntu
  + Start 2 or 3 applications (to appear like a normal victim in the middle of her/his work)
  + Take a snapshot, VM Window: Machines > Snapshot, name it machine\_ready\_snapshot
  + Shut Down Windows VM.
* Configuring Cuckoo
  + Go to /home/<username>/.cuckoo/conf
    - sudo nano cuckoo.conf

set: machinery = virtualbox

memory\_dump = yes

[resultserver] ip = 192.168.56.1

* + - sudo nano virtualbox.conf

set: mode = gui

label = win7x64

platform = windows

ip = 192.168.56.101

snapshot = machine\_ready\_snapshot

* + - sudo nano processing.conf

set: [memory] enabled = yes

* + - sudo nano memory.conf

set: guest\_profile = Win7SP1x64 (If other OS, find profile by: vol.py --info |grep Profiles -A48)

delete\_memdump = yes (To delete memory dumps of VM)

* + - sudo nano reporting.conf

set: [mongodb] enabled = yes

report.html enabled = yes

report.pdf enabled = yes

* Reboot Ubuntu
* Starting Cuckoo
  + cuckoo community (Optional to run everytime, run once in a while)
  + vboxmanage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1 (Everytime after reboot)
  + cuckoo -d (in one terminal)
  + cuckoo run webserver 0.0.0.0:8000 (in another terminal)
* Open Browser, head to [0.0.0.0:8000](http://0.0.0.0:8000) (or [127.0.0.1:8000](http://127.0.0.1:8000)) for Web UI of Cuckoo

**Scanning using Cuckoo:**

* Starting Cuckoo (leave this command to execute in one terminal)
  + cuckoo -d
* Single File Submission
  + cuckoo submit path/to/binary
* Bulk Submission
  + cuckoo submit --timeout 60 <path to malware binary folder>
* URL Submission
  + cuckoo submit --url <http://www.example.com>
* Higher Priority Submission
  + cuckoo submit --priority 5 /path/to/binary
* Timeout
  + cuckoo submit --timeout 60 /path/to/binary
* Custom Time VM Submission (Format is %m-%d-%Y %H:%M:%S)
  + cuckoo submit --clock "01-24-2001 14:41:20" /path/to/binary

**Resources:**

**WindowsError: [Error 5] Access is denied:** https://github.com/cuckoosandbox/cuckoo/issues/2150

**Cuckoo Installation:** https://utopianknight.com/malware/cuckoo-installation-on-ubuntu-20/

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated