OT Lab 4 (Malware Analysis) - AKOR JACOB TERUNGWA

Created	@May 10, 2025 11:30 PM
Attendance Required	☐

Task 1: Set up your environment.

1. Use any virtualization environment, better to use the latest version.

2. Prepare and secure malware analysis environment, e.g. FlareVM or Remnux, etc. Make sure that VM uses a **HOST ONLY** network adapter.

3. Or you can create a Virtual Machine and set it up as a malware analysis environment

Preparation:

(Setting Up Cuckoo Sandbox)

Setting up the host machine

My host machine is Ubuntu 18.04 with 16GB of RAM. Before installing Cuckoo on my host machine, it

is required to install some Python libraries and software packages.

I updated the package information and downloaded the available updates.

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

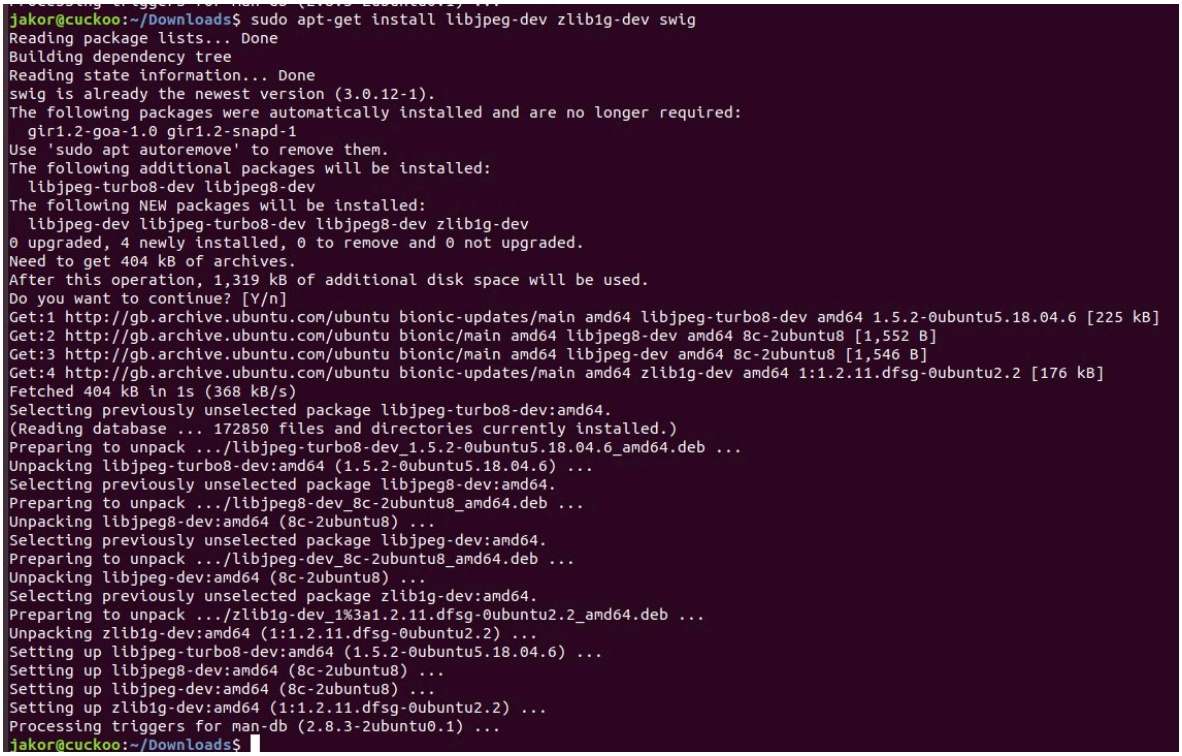
Next, the following software packages from the apt repositories are required to get Cuckoo to install and run correctly:

1. **Installed Dependencies**

- sudo apt-get install python python-pip python-dev libffi-dev libssl-dev
- sudo apt-get install python-virtualenv python-setuptools

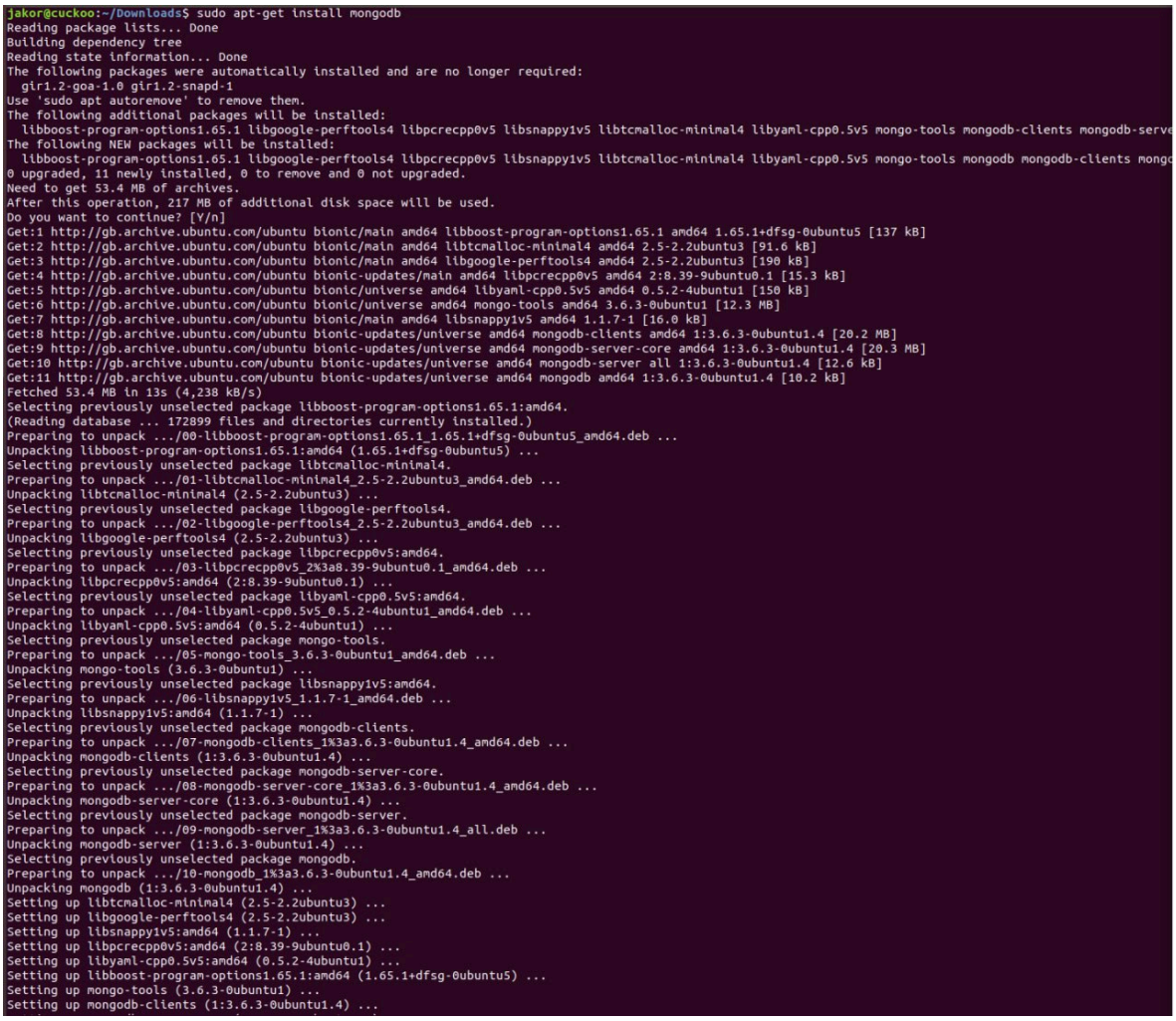
```
jakor@cuckoo:~/Downloads$ sudo apt-get install python python-pip python-dev libffi-dev libssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libffi-dev is already the newest version (3.2.1-8).
python is already the newest version (2.7.15~rc1-1).
python-dev is already the newest version (2.7.15~rc1-1).
libssl-dev is already the newest version (1.1.1-1ubuntu2.1~18.04.23).
python-pip is already the newest version (9.0.1-2.3~ubuntu1.18.04.8).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
jakor@cuckoo:~/Downloads$ sudo apt-get install python-virtualenv python-setuptools
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-setuptools is already the newest version (39.0.1-2ubuntu0.1).
python-setuptools set to manually installed.
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-distutils python3-lib2to3 python3-virtualenv virtualenv
The following NEW packages will be installed:
  python-virtualenv python3-distutils python3-lib2to3 python3-virtualenv virtualenv
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 316 kB of archives.
After this operation, 3,459 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://gb.archive.ubuntu.com/ubuntu bionic/universe amd64 python-virtualenv all 15.1.0+ds-1.1 [46.8 kB]
Get:2 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-lib2to3 all 3.6.9-1~18.04 [77.4 kB]
Get:3 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-distutils all 3.6.9-1~18.04 [144 kB]
Get:4 http://gb.archive.ubuntu.com/ubuntu bionic/universe amd64 python3-virtualenv all 15.1.0+ds-1.1 [43.4 kB]
Get:5 http://gb.archive.ubuntu.com/ubuntu bionic/universe amd64 virtualenv all 15.1.0+ds-1.1 [4,476 B]
Fetched 316 kB in 2s (197 kB/s)
Selecting previously unselected package python-virtualenv.
(Reading database ... 172422 files and directories currently installed.)
Preparing to unpack .../python-virtualenv_15.1.0+ds-1.1_all.deb ...
Unpacking python-virtualenv (15.1.0+ds-1.1) ...
Selecting previously unselected package python3-lib2to3.
Preparing to unpack .../python3-lib2to3_3.6.9-1~18.04_all.deb ...
Unpacking python3-lib2to3 (3.6.9-1~18.04) ...
Selecting previously unselected package python3-distutils.
Preparing to unpack .../python3-distutils_3.6.9-1~18.04_all.deb ...
Unpacking python3-distutils (3.6.9-1~18.04) ...
Selecting previously unselected package python3-virtualenv.
Preparing to unpack .../python3-virtualenv_15.1.0+ds-1.1_all.deb ...
Unpacking python3-virtualenv (15.1.0+ds-1.1) ...
Selecting previously unselected package virtualenv.
Preparing to unpack .../virtualenv_15.1.0+ds-1.1_all.deb ...
Unpacking virtualenv (15.1.0+ds-1.1) ...
Setting up python-virtualenv (15.1.0+ds-1.1) ...
Setting up python3-lib2to3 (3.6.9-1~18.04) ...
Setting up python3-distutils (3.6.9-1~18.04) ...
Setting up python3-virtualenv (15.1.0+ds-1.1) ...
Setting up virtualenv (15.1.0+ds-1.1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

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



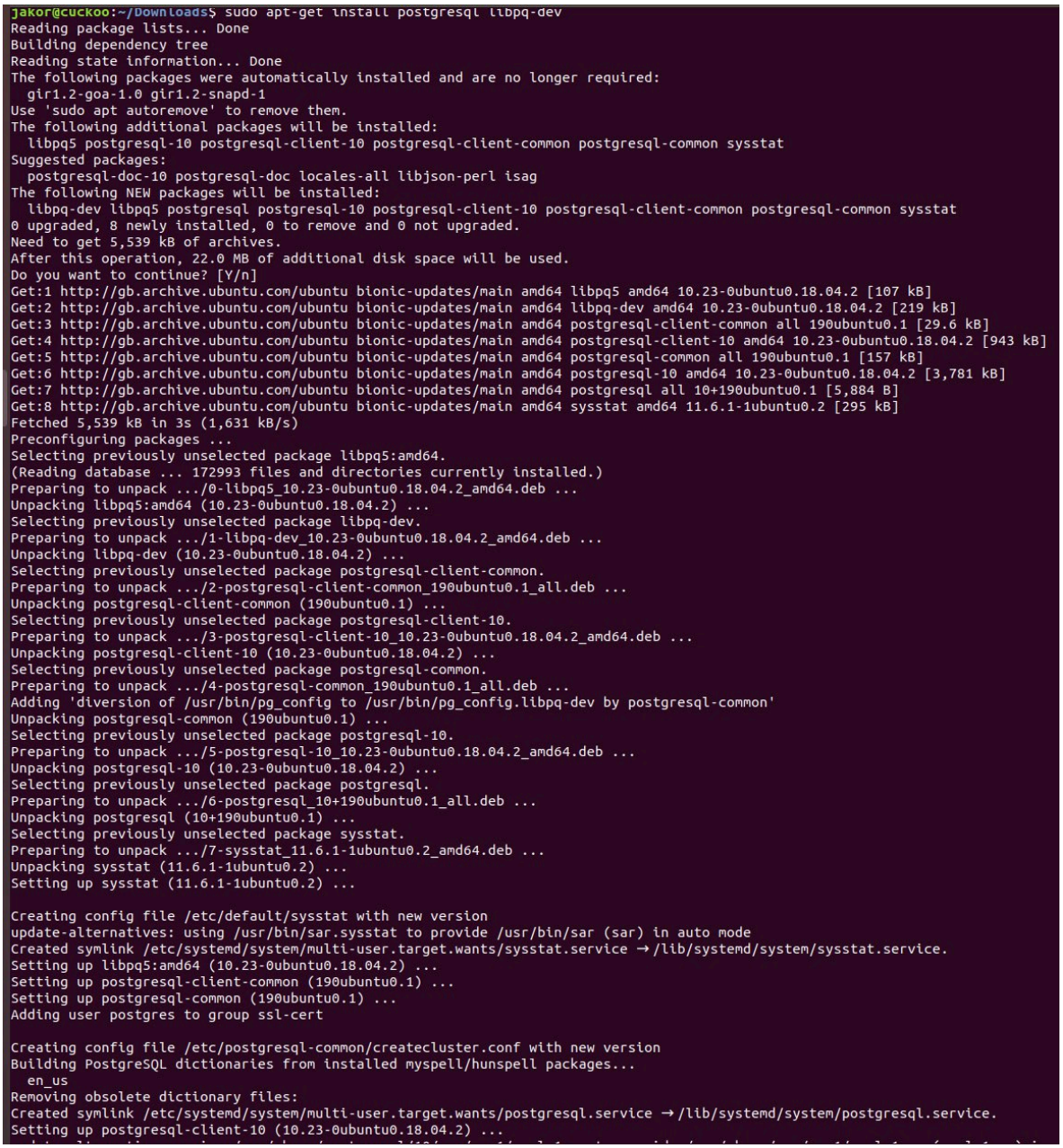
To use the Django-based Web Interface, MongoDB is required:

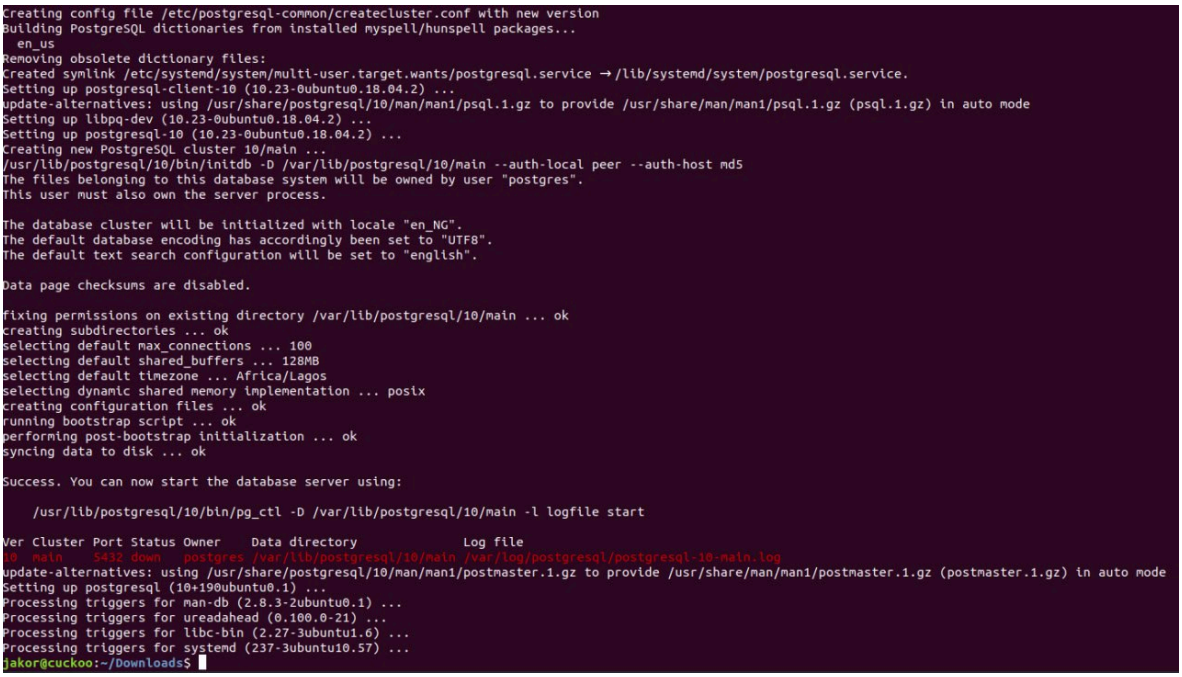
- sudo apt-get install MongoDB



To use PostgreSQL as a database, PostgreSQL will have to be installed as well:

- sudo apt-get install postgresql libpq-dev


jakor@cuckoo:~/Downloads$ sudo apt-get install postgresql libpq-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
libpq5 postgresql-10 postgresql-client-10 postgresql-client-common postgresql-common sysstat
Suggested packages:
postgresql-doc-10 postgresql-doc locales-all libjson-perl isag
The following NEW packages will be installed:
libpq-dev libpq5 postgresql postgresql-10 postgresql-client-10 postgresql-client-common postgresql-common sysstat
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 5,539 kB of archives.
After this operation, 22.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpq5 amd64 10.23-0ubuntu0.18.04.2 [107 kB]
Get:2 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpq-dev amd64 10.23-0ubuntu0.18.04.2 [219 kB]
Get:3 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 postgresql-client-common all 190ubuntu0.1 [29.6 kB]
Get:4 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 postgresql-client-10 amd64 10.23-0ubuntu0.18.04.2 [943 kB]
Get:5 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 postgresql-common all 190ubuntu0.1 [157 kB]
Get:6 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 postgresql-10 amd64 10.23-0ubuntu0.18.04.2 [3,781 kB]
Get:7 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 postgresql all 10+190ubuntu0.1 [5,884 B]
Get:8 http://gb.archive.ubuntu.com/ubuntu bionic-updates/main amd64 sysstat amd64 11.6.1-1ubuntu0.2 [295 kB]
Fetched 5,539 kB in 3s (1,631 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libpq5:amd64.
(Reading database ... 172993 files and directories currently installed.)
Preparing to unpack .../0-libpq5_10.23-0ubuntu0.18.04.2_amd64.deb ...
Unpacking libpq5:amd64 (10.23-0ubuntu0.18.04.2) ...
Selecting previously unselected package libpq-dev.
Preparing to unpack .../1-libpq-dev_10.23-0ubuntu0.18.04.2_amd64.deb ...
Unpacking libpq-dev (10.23-0ubuntu0.18.04.2) ...
Selecting previously unselected package postgresql-client-common.
Preparing to unpack .../2-postgresql-client-common_190ubuntu0.1_all.deb ...
Unpacking postgresql-client-common (190ubuntu0.1) ...
Selecting previously unselected package postgresql-client-10.
Preparing to unpack .../3-postgresql-client-10_10.23-0ubuntu0.18.04.2_amd64.deb ...
Unpacking postgresql-client-10 (10.23-0ubuntu0.18.04.2) ...
Selecting previously unselected package postgresql-common.
Preparing to unpack .../4-postgresql-common_190ubuntu0.1_all.deb ...
Adding 'diversion of /usr/bin/pg_config to /usr/bin/pg_config.libpq-dev by postgresql-common'
Unpacking postgresql-common (190ubuntu0.1) ...
Selecting previously unselected package postgresql-10.
Preparing to unpack .../5-postgresql-10_10.23-0ubuntu0.18.04.2_amd64.deb ...
Unpacking postgresql-10 (10.23-0ubuntu0.18.04.2) ...
Selecting previously unselected package postgresql.
Preparing to unpack .../6-postgresql_10+190ubuntu0.1_all.deb ...
Unpacking postgresql (10+190ubuntu0.1) ...
Selecting previously unselected package sysstat.
Preparing to unpack .../7-sysstat_11.6.1-1ubuntu0.2_amd64.deb ...
Unpacking sysstat (11.6.1-1ubuntu0.2) ...
Setting up sysstat (11.6.1-1ubuntu0.2) ...
Creating config file /etc/default/sysstat with new version
update-alternatives: using /usr/bin/sar.sysstat to provide /usr/bin/sar (sar) in auto mode
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service → /lib/systemd/system/sysstat.service.
Setting up libpq5:amd64 (10.23-0ubuntu0.18.04.2) ...
Setting up postgresql-client-common (190ubuntu0.1) ...
Setting up postgresql-common (190ubuntu0.1) ...
Adding user postgres to group ssl-cert
Creating config file /etc/postgresql-common/createcluster.conf with new version
Building PostgreSQL dictionaries from installed myspell/hunspell packages...
en_us
Removing obsolete dictionary files:
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql.service → /lib/systemd/system/postgresql.service.
Setting up postgresql-client-10 (10.23-0ubuntu0.18.04.2) ...



Below is the command to install VirtualBox 6.0 on the Ubuntu LTS machine.

- sudo dpkg -i virtualbox-6.0_6.0.24-139119~Ubuntu~bionic_amd64.deb

Now I have a working instance of VirtualBox v6.0

Installing tcpdump

To dump the network activity performed by the malware during execution, you'll need a network Sniffer is adequately configured to capture the traffic and dump it into a file.

By default, Cuckoo adopts tcpdump, the prominent open-source solution.

- **sudo apt-get install tcpdump**

```
jakor@cuckoo:~/Downloads$ sudo apt-get install tcpdump
Reading package lists... Done
Building dependency tree
Reading state information... Done
tcpdump is already the newest version (4.9.3-0ubuntu0.18.04.3).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

- sudo apt-get install swig
- sudo pip install m2crypto==0.27.0

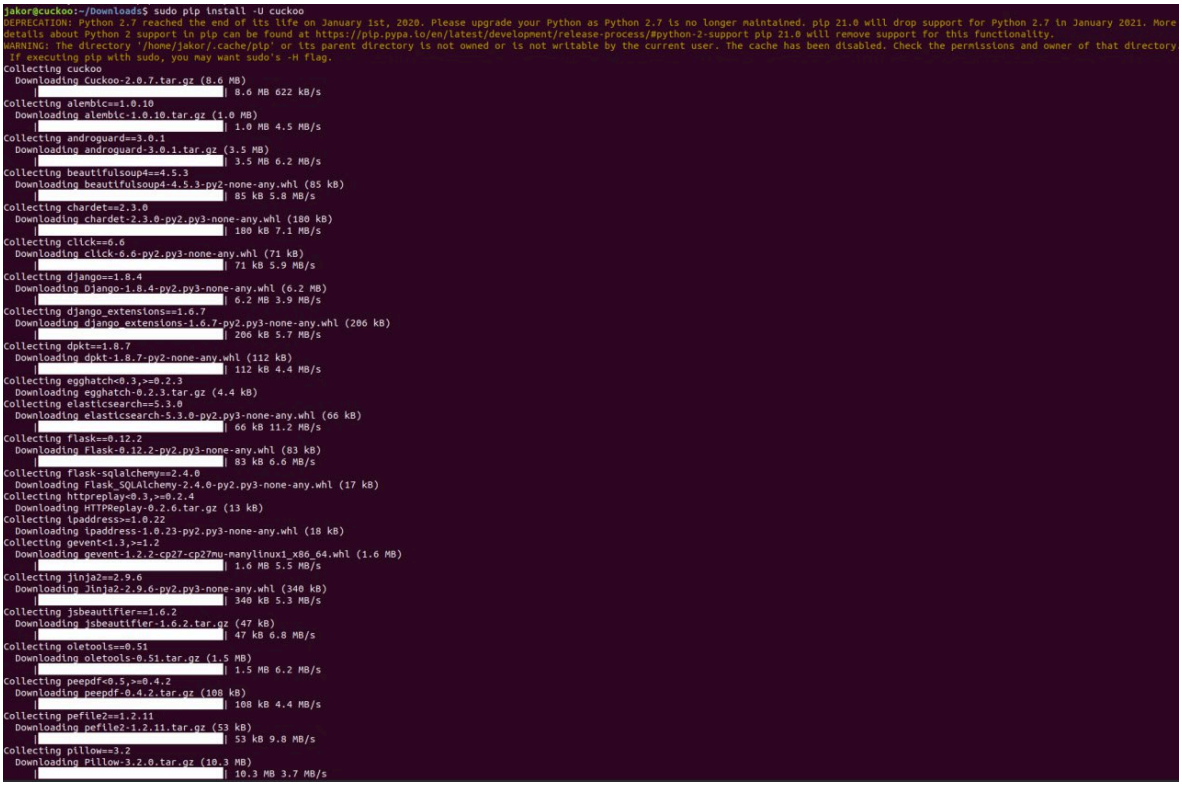
```
jakor@cuckoo:~/Downloads$ sudo apt-get install swig
Reading package lists... Done
Building dependency tree
Reading state information... Done
swig is already the newest version (3.0.12-1).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
jakor@cuckoo:~/Downloads$ sudo pip install m2crypto==0.27.0
The directory '/home/jakor/.cache/pip/http' or its parent directory is not owned by the current user and the cache has been disabled. Please check the permissi
 pip with sudo, you may want sudo's -H flag.
The directory '/home/jakor/.cache/pip' or its parent directory is not owned by the current user and caching wheels has been disabled. check the permissions and
th sudo, you may want sudo's -H flag.
Collecting m2crypto==0.27.0
  Downloading https://files.pythonhosted.org/packages/01/bd/a41491718f9e2bebab015c42b5be7071c6695acfa301e3fc0480bfd6a15b/M2Crypto-0.27.0.tar.gz (1.1MB)
    100% |████████████████████████████████| 1.1MB 1.0MB/s
Collecting typing (from m2crypto==0.27.0)
  Downloading https://files.pythonhosted.org/packages/0b/cb/da856e81731833b94da70a08712f658416266a5fb2a9d9e426c8061becef/typing-3.10.0.0-py2-none-any.whl
Installing collected packages: typing, m2crypto
  Running setup.py install for m2crypto ... done
Successfully installed m2crypto-0.27.0 typing-3.10.0.0
jakor@cuckoo:~/Downloads$
```

After installing these packages, I can now install Cuckoo on my system. To install, i run the following commands. Alternatively, I can simply download the zip file.

- sudo pip install -U pip setuptools

```
jakor@cuckoo:~/Downloads$ sudo pip install -U pip setuptools
The directory '/home/jakor/.cache/pip/http' or its parent directory is not owned by the current user and the cache has been disabled. Please check the permissi
 pip with sudo, you may want sudo's -H flag.
The directory '/home/jakor/.cache/pip' or its parent directory is not owned by the current user and caching wheels has been disabled. check the permissions and
th sudo, you may want sudo's -H flag.
Collecting pip
  Downloading https://files.pythonhosted.org/packages/27/79/8a850fe3496446ff0d584327ae44e7500daf6764ca1a382d2d02789accf7/pip-20.3.4-py2.py3-none-any.whl (1.5MB)
    100% |████████████████████████████████| 1.5MB 791kB/s
Collecting setuptools
  Downloading https://files.pythonhosted.org/packages/e1/b7/182161210a13158cd3ccc41ee19aadef54496b74f2817cc147006ec932b4/setuptools-44.1.1-py2.py3-none-any.whl
    100% |████████████████████████████████| 583kB 1.3MB/s
Installing collected packages: pip, setuptools
  Found existing installation: pip 9.0.1
    Not uninstalling pip at /usr/lib/python2.7/dist-packages, outside environment /usr
  Found existing installation: setuptools 39.0.1
    Not uninstalling setuptools at /usr/lib/python2.7/dist-packages, outside environment /usr
Successfully installed pip-20.3.4 setuptools-44.1.1
jakor@cuckoo:~/Downloads$
```

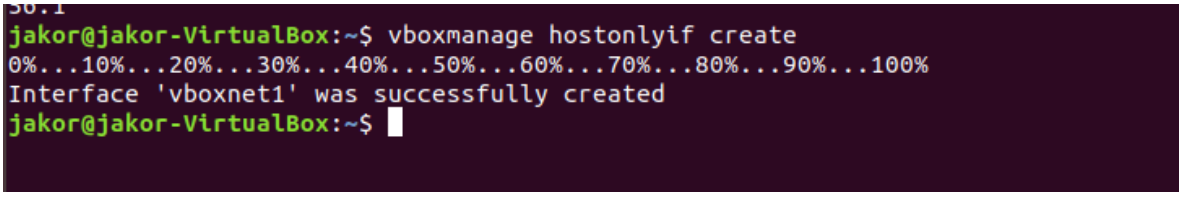
- sudo pip install -U cuckoo



After installing Cuckoo, I correctly set up the VirtualBox and its networking.

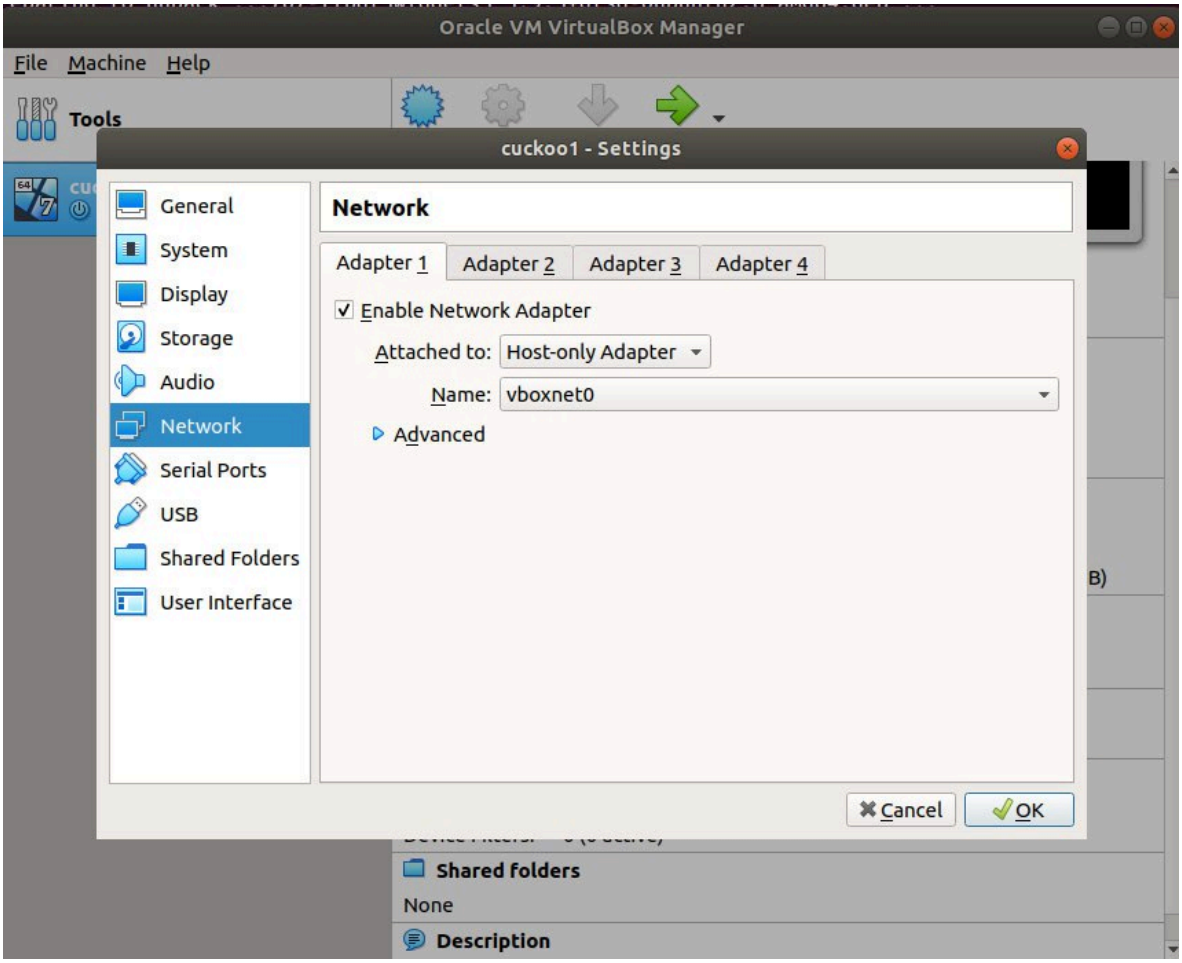
I created "*Host-Only Adapter*" by running the following command:

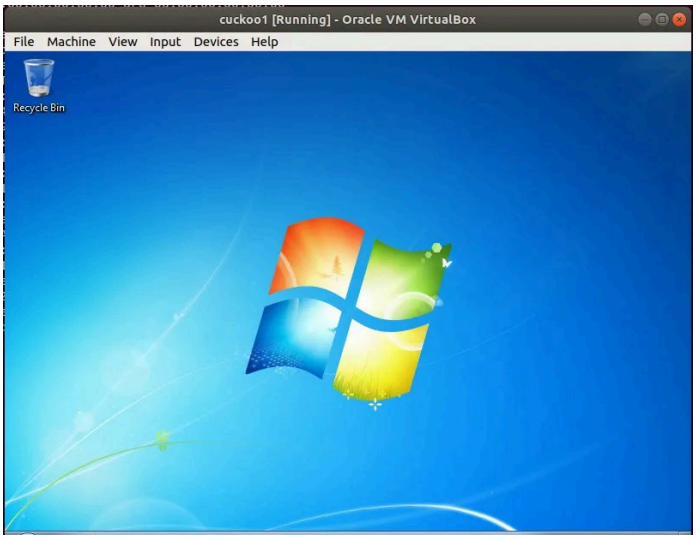
vboxmanage hostonlyif create



I set the IP address for the vboxnet0 interface that I created before.
vboxmanage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1

Next, I created my virtual machine in VirtualBox and installed Windows 7 OS. After installing the OS, I configured the VM network adapter to “Host Only Adapter”.





Configuration:

After that, I need to configure IP forwarding so that an internet connection gets routed from the host machine to the guest VM. Here, the interface assigned to my VM is enp0s3. The VM's IP address is 192.168.56.101, which is on the subnet of 192.168.56.0/24. The outgoing interface connected to the internet is eth0.

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 -A FORWARD -s 192.168.56.0/24 -d 192.168.56.0/24 -j ACCEPT

sudo iptables -A FORWARD -j LOG

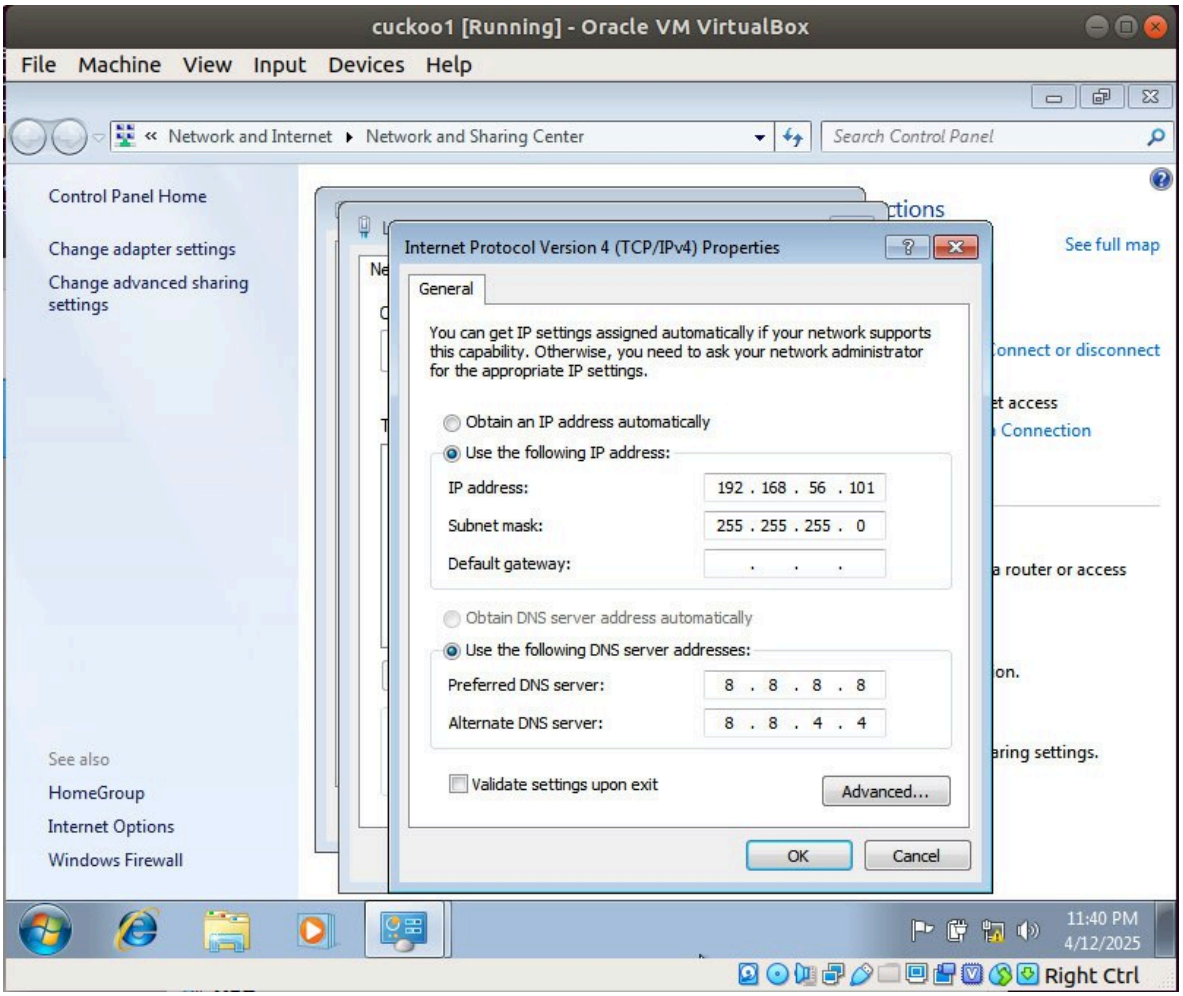
After executing the above commands i had to enable IP forwarding in the kernel. To do that, I executed the following commands:

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

```
jakor@cuckoo:~/Downloads$ echo 1 | sudo tee -a /proc/sys/net/ipv4/ip_forward
1
jakor@cuckoo:~/Downloads$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
jakor@cuckoo:~/Downloads$
```

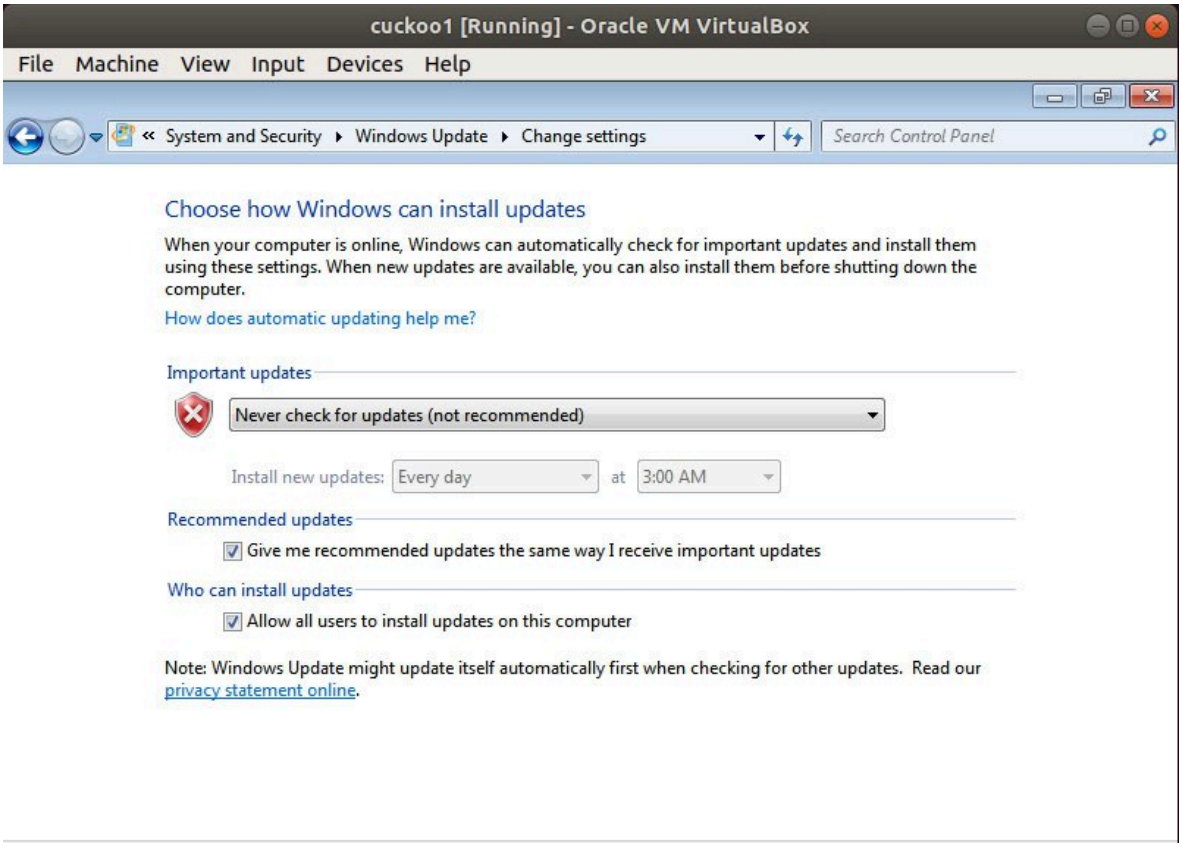
I checked whether I had set up the rules correctly. I Run this command:
sudo iptables -L

Setting up the Guest machine:
Now I started setting up the guest machine, which has Windows installed. First, I configured the Network Adapter settings are as follows,
IP Address — 192.168.56.101 (VM IP address)
Subnet Mask — 255.255.255.0
Default Gateway — 192.168.56.1 (Internet accessing interface)
DNS Servers — 8.8.8.8/8.8.4.4

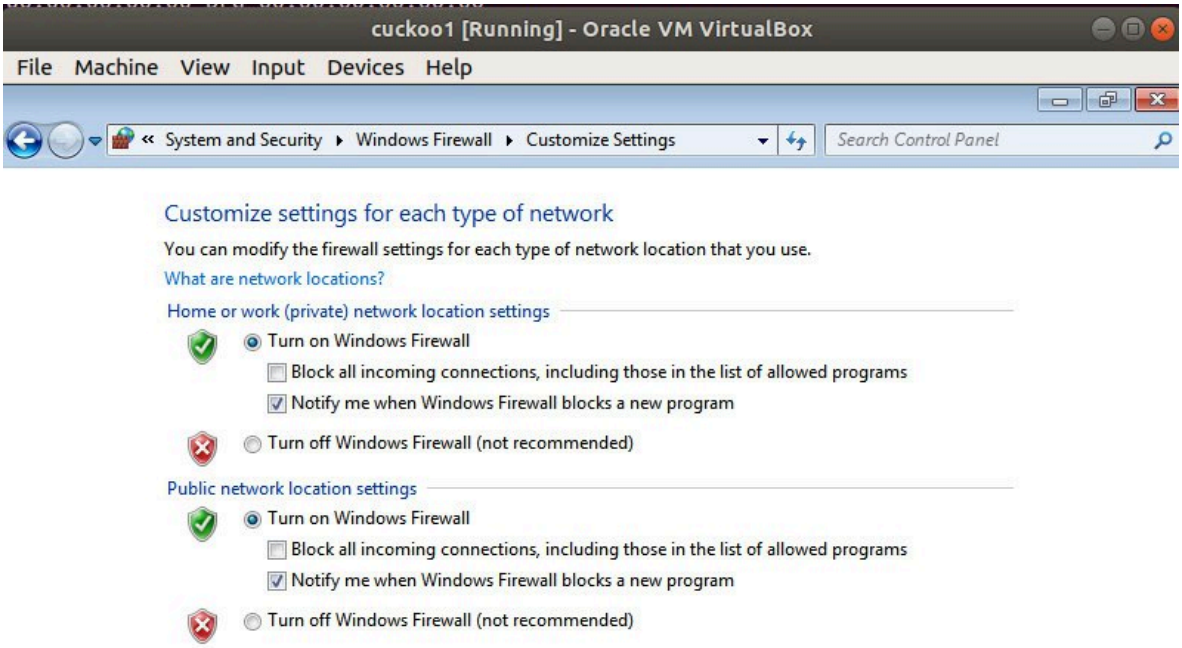


After changing the network configurations, i had to do the following customizations to the VM.

1. I disabled Windows Update and Windows Firewall.

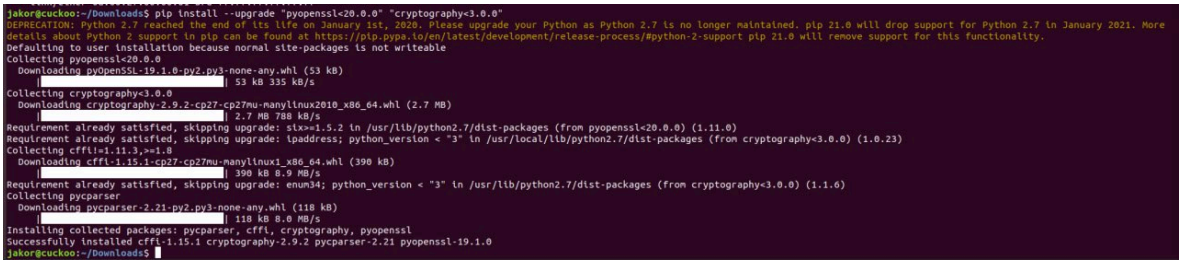


2. Changed User Account Control Settings.



1. Installed Python 2.7 for Windows
2. Uploaded the agent.py file from your host machine, which can be found in the ~/.cuckoo/agent directory. Put it in the Windows startup folder located in “C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Startup”. After rebooting the VM, you will be able to see a terminal open in the VM. (You can enable drag and drop in VirtualBox settings. Enable only drag and drop from host to guest.

pip install --upgrade "pyopenssl<20.0.0" "cryptography<3.0.0”



The cuckoo configuration files are located in the ~/.cuckoo/conf directory. I can open those files in gedit using this command:

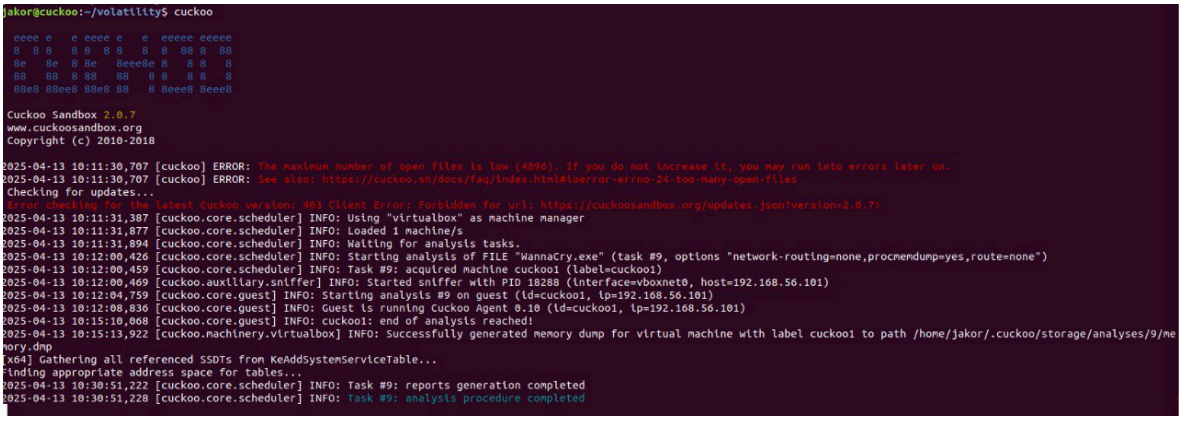
- sudo gedit cuckoo.conf

After finishing configuring, I started Cuckoo.

Analyzing using Cuckoo:

I run the following commands to start Cuckoo and the Cuckoo web interface. I run those in two separate terminal windows.

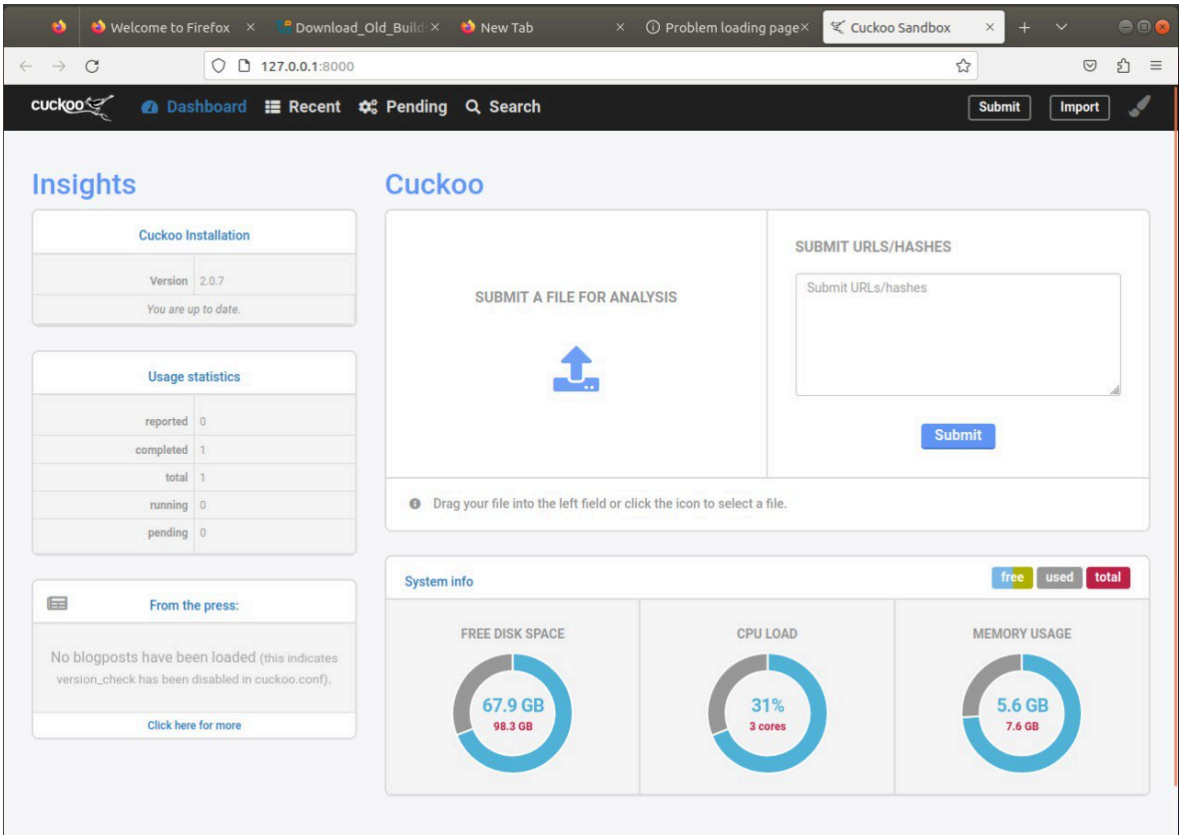
Terminal #1: **cuckoo**



Terminal #2: **cuckoo web runserver**

Then I can now access the web interface by going to this address in the web browser:

localhost:8000



Task 2 - Let's get some malware

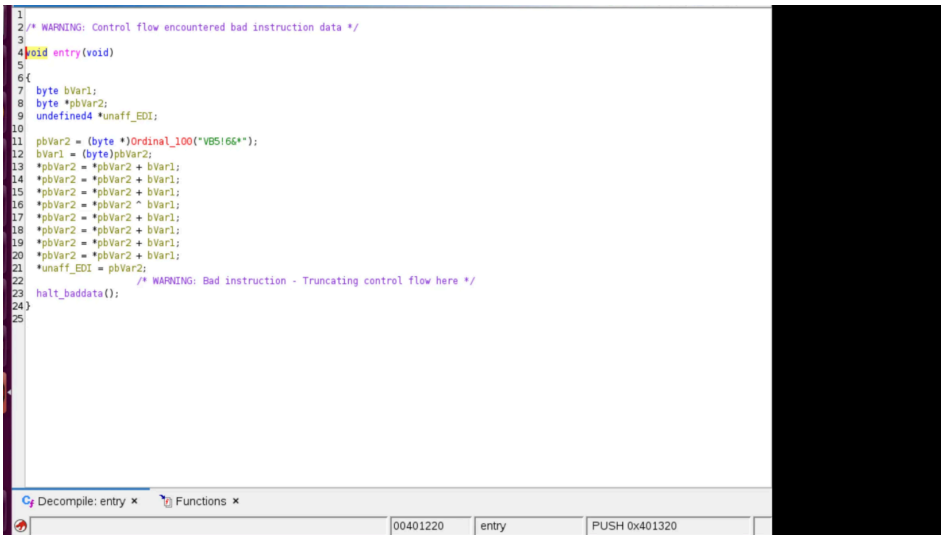
I downloaded the following malware from the internet https://github.com/Da2dalus/The-MALWARERepo

1. WannaCry ransomware
2. CrytoLocker

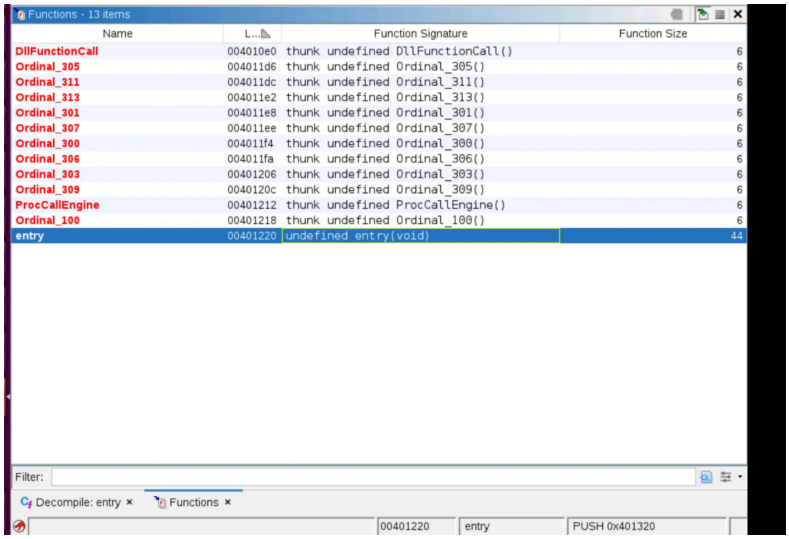
Task 3 - Static Analysis

First, I installed Ghidra using snap

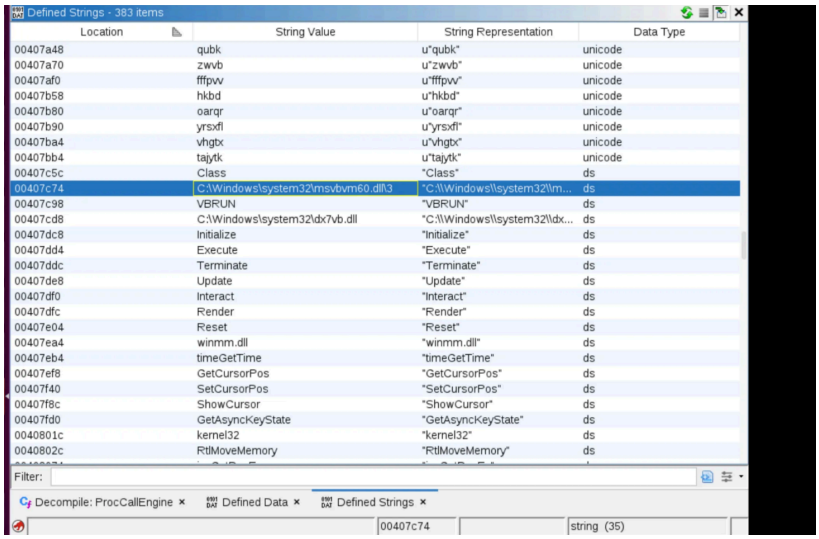
Then I launched Ghidra, created a new project, imported the malware, and started analysis



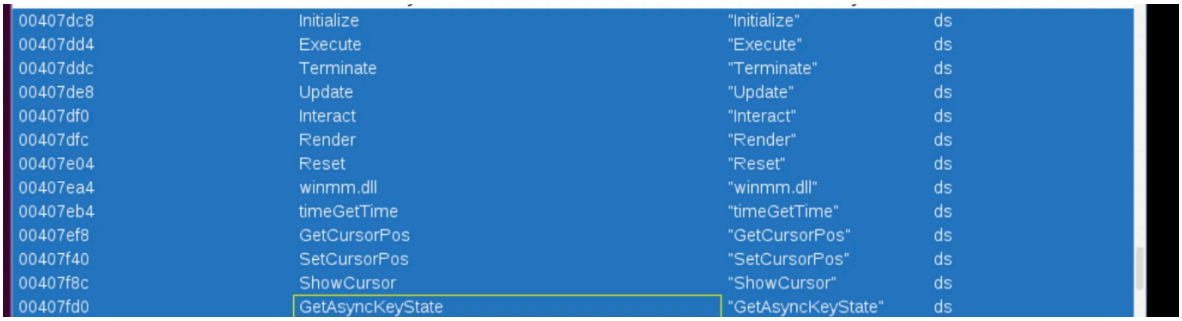
Attached below are the functions found



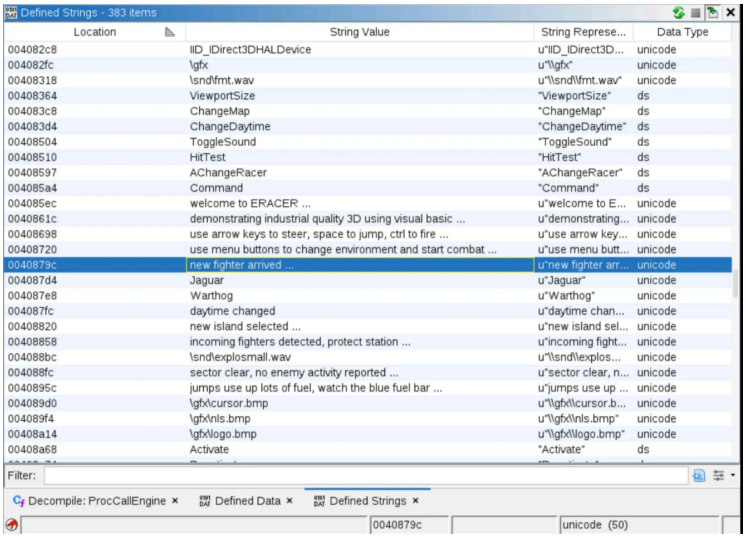
Then I checked defined strings, which contained a lot, but I found the name of a DLL



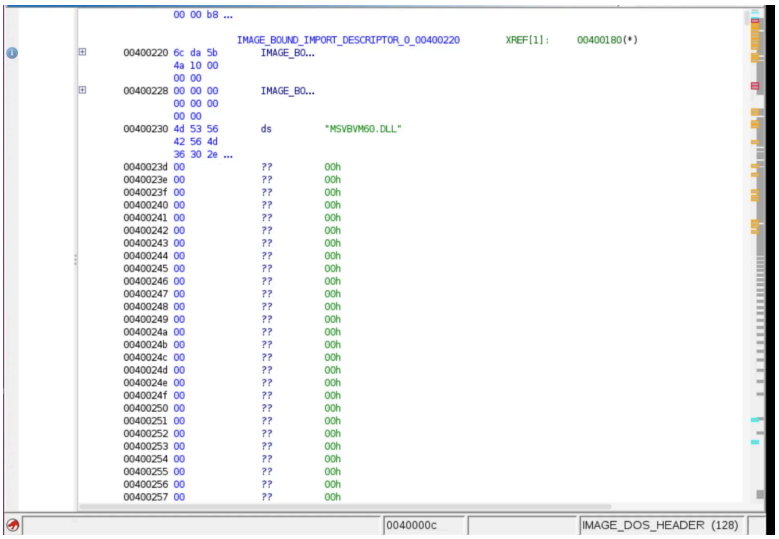
Also, some commands could be found:



And other strings that give an idea of some executables:



Then I opened the disassembler and discovered that the malware contains a lot of zeros at the head, which may be done to confuse a reader



1. Compare the findings of both methods, and see if there are some artifacts that online solution did not manage to find, or vice versa. For example, a piece of code or information that helps you
   in your analysis

WannaCry Analysis

Aspect	Online Analysis (e.g., Hybrid Analysis)	Offline Manual Analysis (e.g., Cuckoo + IDA)
Static Signature	Detects it as `WanaCrypt0r` or `WannaCry`; flags SMB exploitation	Confirms exploit via `EternalBlue` and includes specific SMB payload code
IOC Extraction	IPs, file hashes, mutex names	Deeper discovery of kill switch domain hardcoded in binary (`www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com`)
Payload Behavior	Ransom note dropped and tor site contacted	Observes local encryption flow using AES + RSA and process injection (e.g., `taskdl.exe`)
Artifacts Missed by Online Tools	Limited system call tracing; some anti-VM behavior not triggered	Able to catch fake checks for `disk space`, `VM artifacts`, and obfuscation tricks

CryptoLocker Analysis

Aspect	Online Analysis	Offline Manual Analysis

Malware Family Identification	Identified as `CryptoLocker` variant via static signature	Confirms RSA-2048 key generation, registry persistence
Network Indicators	Detects communication with C2 servers over HTTPS	Observes domain generation algorithm (DGA) with time-based TLD changes
Ransom Behavior	Confirms ransom note file and encryption start	Tracks detailed registry keys used for persistence (`HKCU\Software\CryptoLocker`)
Artifacts Missed by Online Tools	Can't fully trace DGA logic or encryption functions	Able to extract embedded PE resources (e.g., ransom HTML templates) and decode communication headers

4. Try to describe which method is better (Sandboxing vs Static analysis) is better, and which one is more useful in which case.

Sandboxing vs Static Analysis – Comparison and Use Cases

Static Analysis (e.g., using Ghidra)

- **Advantages:**
  - Reveals the internal logic, algorithms, and hardcoded data (e.g., encryption keys, DGA routines).
  - Doesn’t require executing the malware, so it's safer.
  - Useful for understanding obfuscation, code reuse, and stealth techniques.
- **Disadvantages:**
  - Time-consuming and requires reverse-engineering skills.
  - Can be defeated by strong packing or obfuscation.
- **Use Case:**
  - Deep understanding of malware logic.
  - Detecting **zero-day behavior**, obfuscated payloads, and **persistence mechanisms**.
  - Forensic-grade evidence collection.

Sandboxing (e.g., Cuckoo)

- **Advantages:**
  - Fast and automated — reveals high-level behavior, like dropped files, registry changes, and C2 traffic.
  - Good at identifying **Indicators of Compromise (IOCs)** quickly.
  - Useful for non-technical analysts and triage.
- **Disadvantages:**

- Limited by anti-VM/anti-sandbox techniques.
- May miss behaviors that are time-triggered or environment-dependent.

- **Use Case:**
  - Rapid behavioral assessment of malware.
  - IOC generation for SIEM or threat intel.
  - First-stage triage in incident response.

A Better One?

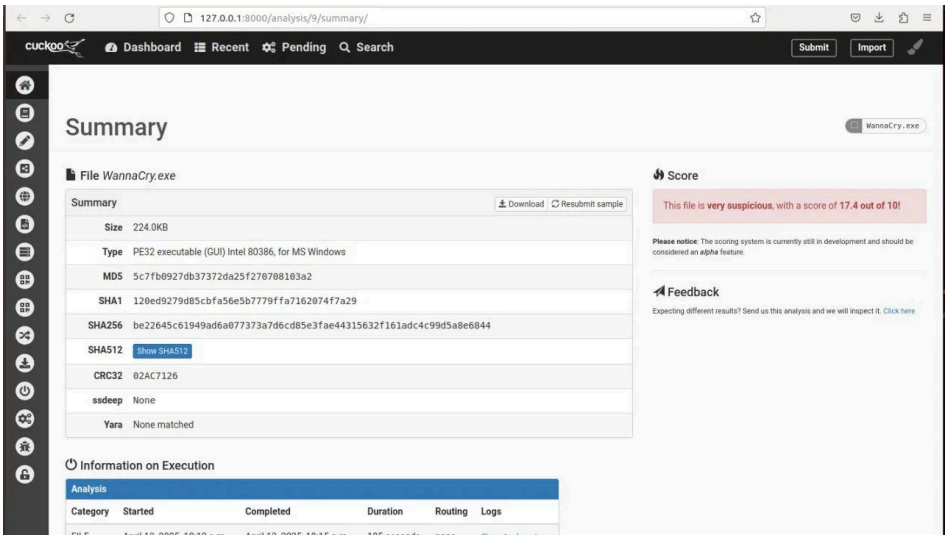
- To me, they are complementary, not competing.
  - Used sandboxing for quick insights and IOC extraction.
  - Used static analysis when sandbox output is inconclusive or when you need to understand the malware's true intent and structure.

Task 5 - Dynamic Analysis (Bonus)

Sandbox Analysis

Step 1: I submited WannaCry Malware to Cuckoo

Analysis Summary of WannaCry Malware



The log from the analysis confirms the execution of WannaCry ransomware (or a variant) with clear behavioral patterns matching its known malicious activities. Below is a detailed breakdown:

Some important Observations

1. Initial Execution & Payload Dropping

    Sample Executed: C:\Users\jakor\AppData\Local\Temp\WannaCry.exe (PID 1208)

    Dropped Files:

    Encryption components: c.wry, m.wry, r.wry, t.wry, u.wry

    Batch script: 205981744552341.bat (likely for persistence or cleanup)

    Ransom note: !Please Read Me!.txt

    Encryption keys: 00000000.pky, 00000000.eky, 00000000.res

2. Document Encryption (Ransomware Behavior)

    Appends .WCRYT extension to files (WannaCry's signature):

    C:\Users\jakor\Documents\BZMcQrIAvBqslJ.ppt.WCRYT

    C:\Users\jakor\Documents\lOFOjaEJOC.docx.WCRYT

    C:\Users\Public\Pictures\Sample Pictures\Chrysanthemum.jpg.WCRYT

    Targets multiple file types:

    Office docs (.docx, .ppt, .docm)

    Images (.jpg, .png, .bmp)

    Media (.mp3, .wma, .wmv)

    System files (thumbnails, databases, logs)

3. Persistence & Propagation

    Creates a shortcut: !WannaDecryptor!.exe.lnk (likely for autorun)

Injects into explorer.exe (PID 1844) to maintain persistence.

Uses cmd.exe & cscript.exe to execute secondary scripts.

4. Anti-Recovery Measures

Executes vssadmin.exe (PID 2696) to delete shadow copies (prevents file recovery).

Uses WMIC.exe (PID 2648) for system interrogation (possibly checking for defenses).

5. Process Injection & Evasion

Injects into multiple processes:

cmd.exe (multiple instances)

!WannaDecryptor!.exe (malicious child processes)

explorer.exe (stealth)

Attempts to kill processes via taskkill.exe (likely disabling security tools).

6. Network Activity (Suspicious)

Queries

dns.msftncsi.com (Microsoft NCSI, checks for internet connectivity).

Connects to 8.8.8.8 (Google DNS)—possibly for C2 communication.

Sandbox Evasion Techniques Detected

The malware employs multiple anti-sandbox checks (as seen in earlier logs):

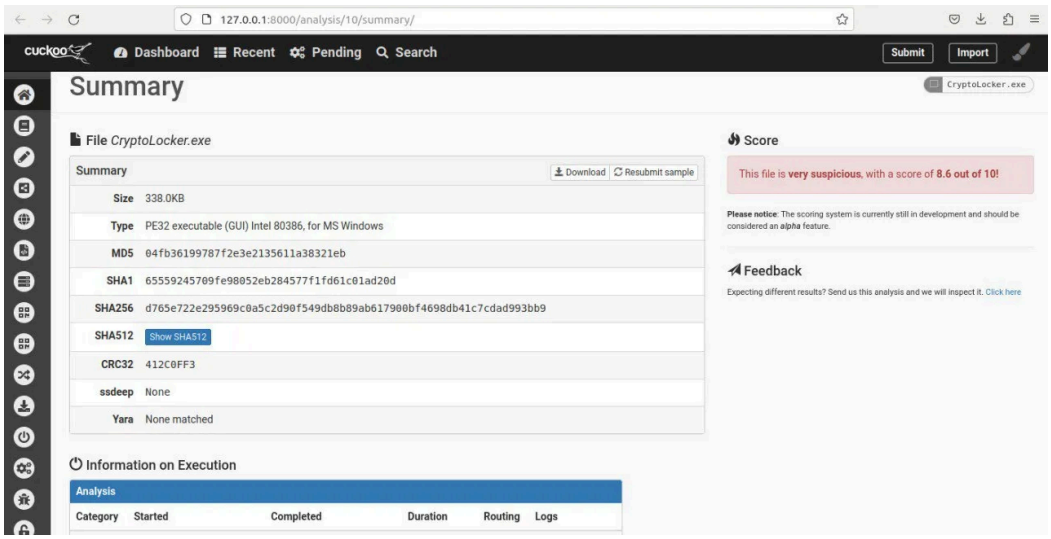
Debugger Detection (IsDebuggerPresent checks)

Foreground Window Monitoring (checks for human activity)

Memory & Disk Queries (detects low-resource VMs)

Delayed Execution (avoids automated analysis timeouts)

Analysis Summary of CrytoLocker:



Some important Observations

1. Malware Persistence & Evasion
   Execution Path: C:\Users\jakor\AppData\Local\Temp\CryptoLocker.exe
   Typical of ransomware using Temp folders to avoid detection.
   Randomized Child Process Names: {34184A33-0407-212E-330B-0C020115E2C2}.exe
   Avoids signature-based detection (UUID-style naming).
2. Process Injection & Encryption
   Two child processes (PIDs 2908 and 2800) spawned and monitored.
   These are likely responsible for file encryption (CryptoLocker uses multi-threaded encryption).
   Memory dumps taken before termination (useful for forensic analysis).
3. Anti-Analysis Tactics
   Short-lived parent process (CryptoLocker.exe exits after 11 seconds).
   No screenshots (Pillow is not installed, but malware may also block screenshots).
   No network activity logged (may use delayed C2 or encrypted channels).
   Indicators of Compromise (IOCs)
   Files

   Main Payload:
   C:\Users\jakor\AppData\Local\Temp\CryptoLocker.exe
   Child Processes:
   {34184A33-0407-212E-330B-0C020115E2C2}.exe (multiple instances)
   Processes
   Parent: CryptoLocker.exe (PID 2864)
   Children: {34184A33...}.exe (PIDs 2908, 2800)
   Behavioral Signatures
   Ransomware-like process injection
   Temp folder execution (common in CryptoLocker)
   Randomized process names (evasion)

Rapid process termination (anti-debugging)
Comparison to Known CryptoLocker Variants
Feature This Sample Classic CryptoLocker
Execution Path %LocalAppData%\Temp %AppData% or %LocalAppData%
Process Names UUID-style ({...}.exe) Random names (e.g., uihf.exe)
Persistence Not yet observed Registry Run keys
Encryption Threads Multiple (PID 2908, 2800) Single-threaded in early versions
Conclusion: This is likely a newer variant with improved evasion tactics.

What kind of benefit does this method have?

- **Automated Dynamic Analysis**

  Cuckoo runs malware in a controlled virtual environment, automatically capturing its behavior, such as file changes, network connections, and process activity.

- **Comprehensive Reports**

  It generates detailed JSON/HTML reports with:

  - API calls
  - Dropped files
  - Registry changes
  - Memory dumps
  - Screenshots

- **Network Traffic Capture**

  Captures and analyzes full PCAPs, helping detect command-and-control (C2) communication, DNS requests, and data exfiltration attempts.

- **Customizable and Extensible**

  You can add custom signatures, integrate YARA rules, or hook into other forensic tools (e.g., Volatility for memory analysis).

- **Supports Multiple Platforms**

  Cuckoo can analyze Windows, Linux, macOS, and Android malware by configuring different virtual machines.

- **Safe Environment**

  Malware is executed in isolated virtual machines, protecting the host system from infection.

- **Useful for Threat Intelligence and IOC Extraction**

  It helps generate Indicators of Compromise (IOCs) that can be shared with SOC teams or used in SIEM systems like Wazuh.