Experimental Evaluation of Malware in Sandboxing Environments

Radhika Pai, Vignesh Rajan, Nishant Rodrigues

Project Report

December 14, 2016

Introduction

The most widespread way of discovering malware attacks is sandboxing. This involves isolation of the unknown suspected files in a virtual environment which emulates an actual host system. The suspected files are then monitored in the virtual environment, to check if they do in fact possess viral strains. If they do, they are labelled as malicious and the information acquired through monitoring is used to prevent further attacks from the new malware. By acting upon the virtual environment, the malware exposes its true self.

However, the art of sandboxing has not seen a lot of new features in the recent year, and that has resulted in attackers trying to detect if their programs are running in virtual environments. While not all the malware includes this additional layer of self-protection, they are beginning to appear more frequently.

Our project involves creation and analysis of how such environment sensing malware evade sandboxes and invade systems.

We use a sandbox called Cuckoo Sandbox to do our malware analysis. We intended to use one more sandbox as per our project proposal, but we decided against using it since a free version of the software was not available. However, cuckoo sandbox is sufficient for the scope of this project and we provide further details about installing, using and analyzing malware using it.

Installation

The installation for Cuckoo requires installing three main components:

- **Host Virtual Machine**

We needed a host system to run the cuckoo application on. We chose to use Ubuntu 15.10 since Linux is freeware and provides nice features for simulating server like behavior. Linux also provides good support for virtualization.

We use VMWare Fusion (license obtained through UMD) for running Ubuntu 15.10. For setting up the VM, download .iso file for the VM from https://www.ubuntu.com/download/desktop, and install inside VMWare. For our VM we use a configuration with 8Gb of memory and 40 Gb of storage.

- ## Cuckoo and Dependencies

The Cuckoo Sandbox is available for download at https://cuckoosandbox.org/. The developers of the project also maintain a development release on Github. Cuckoo is open source and free to use. Cuckoo doesn't run out of the box, but rather needs a lot of configuration and customized tuning depending on what you intend to use it for. We wanted to use it for malware analysis primarily, so we outline the steps required to install cuckoo:

1. Obtain cuckoo:
$ git clone https://github.com/cuckoosandbox/cuckoo.git

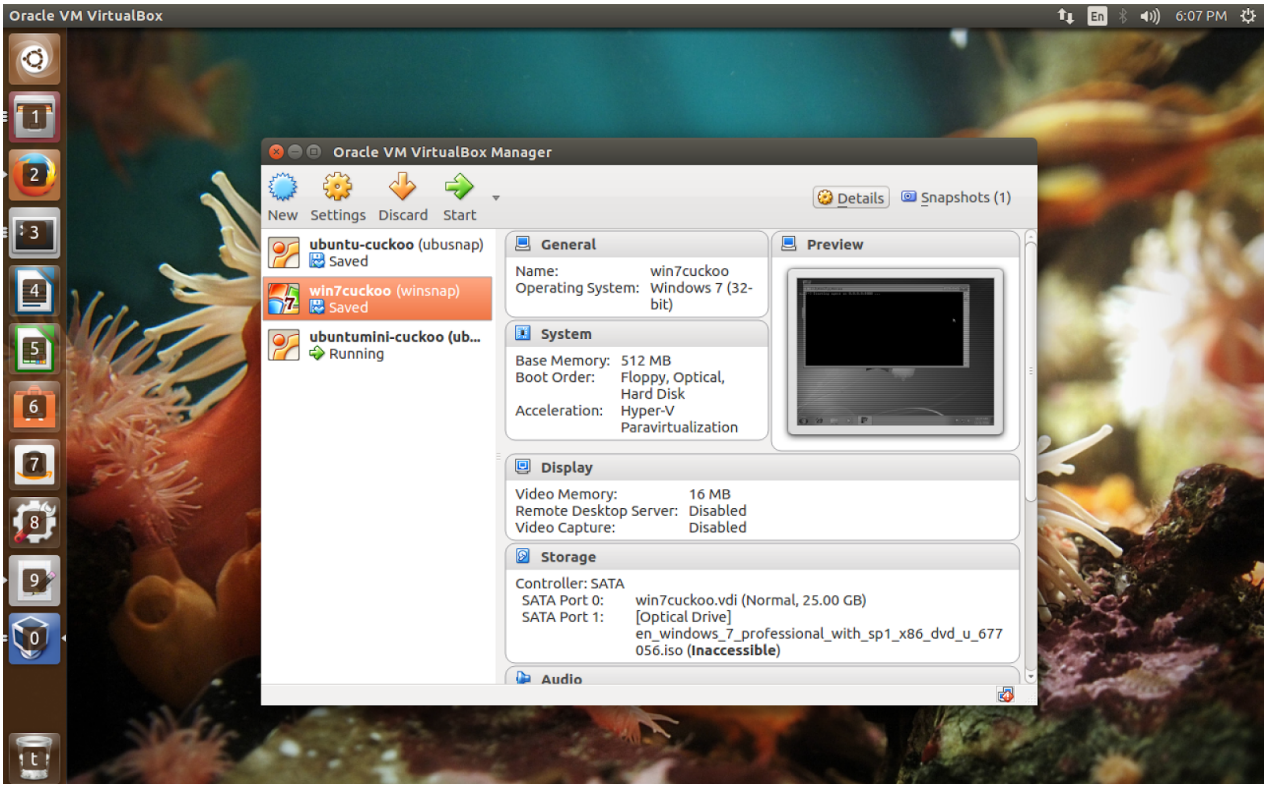
2. Install dependencies:
$ sudo apt-get install python python-pip python-dev libffi-dev libssl-dev
$ sudo apt-get install mongodb
$ sudo pip install -r cuckoo/requirements.txt

3. TcpDump
$ sudo apt-get install tcpdump
$ sudo setcap cap_net_raw,cap_net=eip /usr/sbin/tcpdump

4. Installing cuckoo:
$ sudo adduser cuckoo
$ sudo usermod -a -G vboxusers cuckoo

5. Postgresql (database)
$ sudo apt-get install postgresql postgresql-server

- **Test VMs (for sandboxing)**

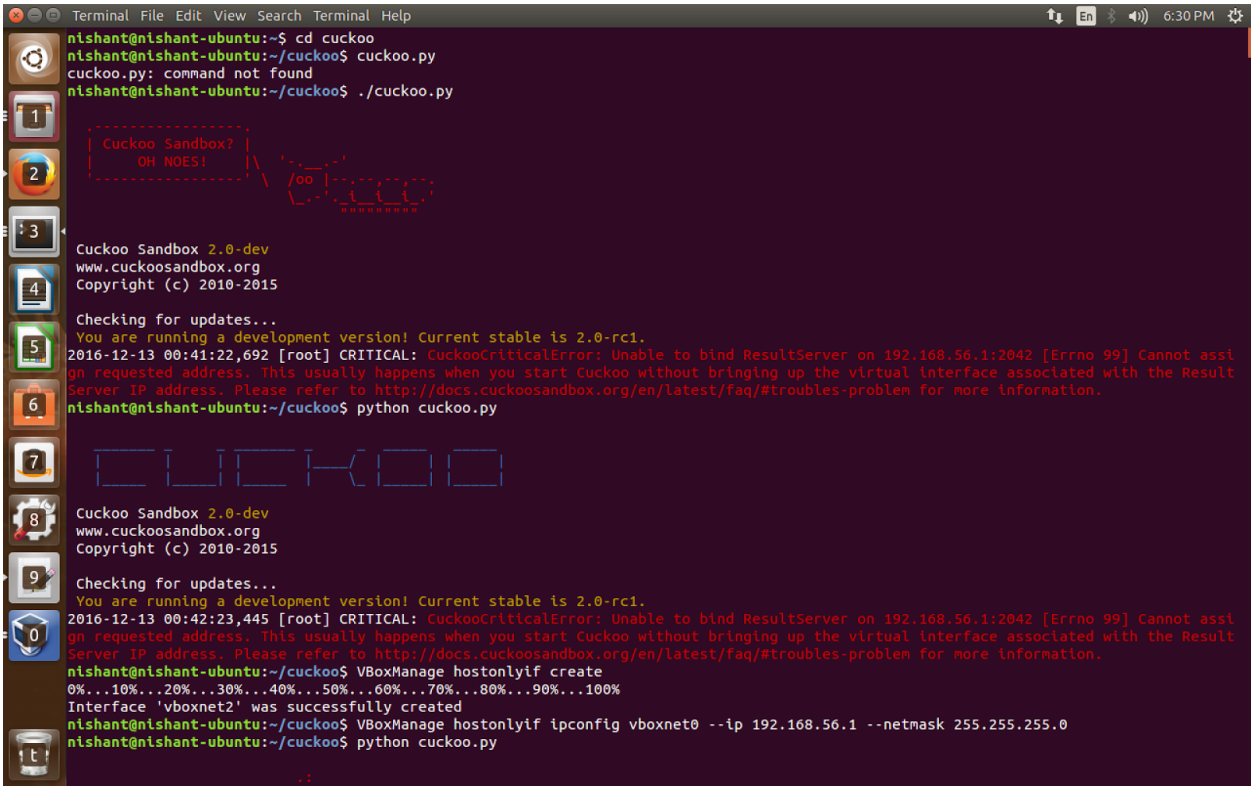


We use three virtual machines inside our host virtual machine. All three VMs are run inside VirtualBox which is free to use. We describe the three VMs (also shown in screenshot above):

1. Ubuntu 14.04 (minimal): This is a minimal installation of Ubuntu. It does not contain any packages except for the linux kernel itself and few essential packages like gcc and other core utilities.
2. Windows 7: This is a full installation of Windows 7 Professional. It does not have any extra installed applications other than the ones that come with the installation. We did install Python 2.7 to be able to run a script used by Cuckoo to communicate with the VM.
3. Ubuntu 14.04: This is a full installation of Ubuntu 14.04. We use this to see if there is a difference in the behavior of malware when it runs on the minimal VM versus this VM.

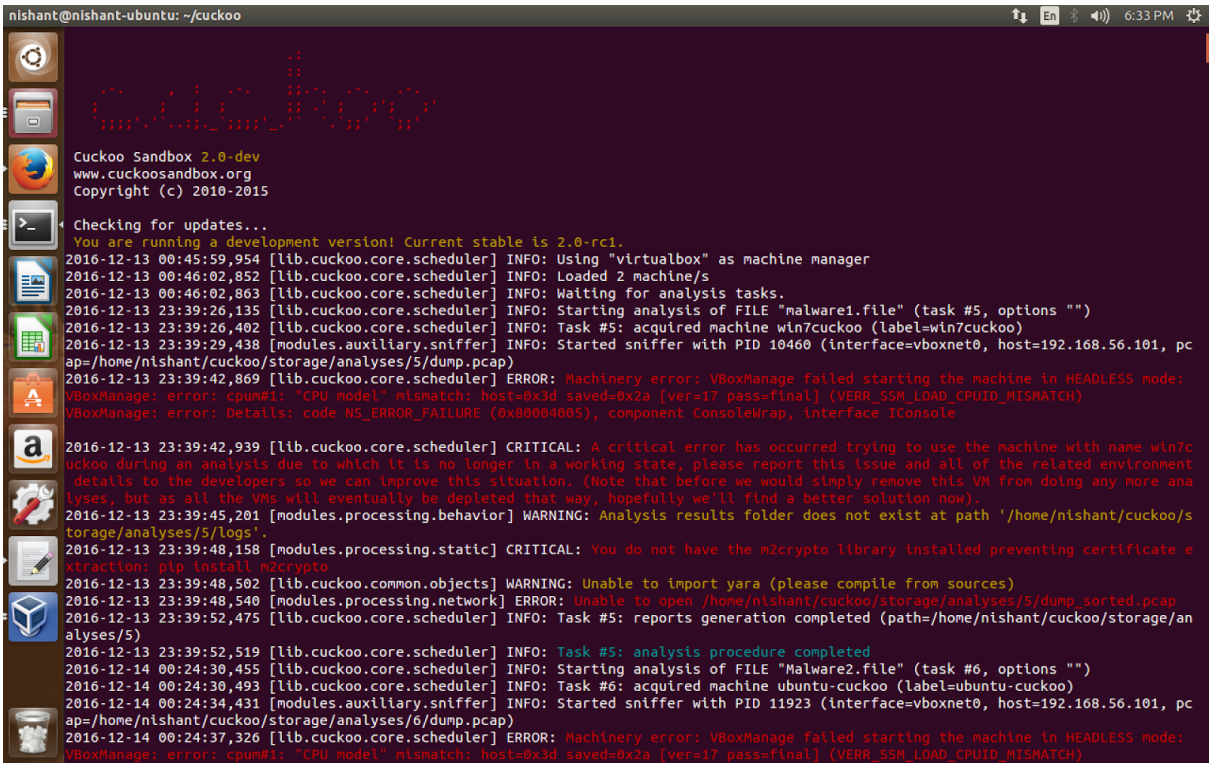
Getting Started with Cuckoo

To get started with cuckoo on your virtual machine (in our case: Ubuntu), navigate to the Cuckoo folder and run the cuckoo.py file.

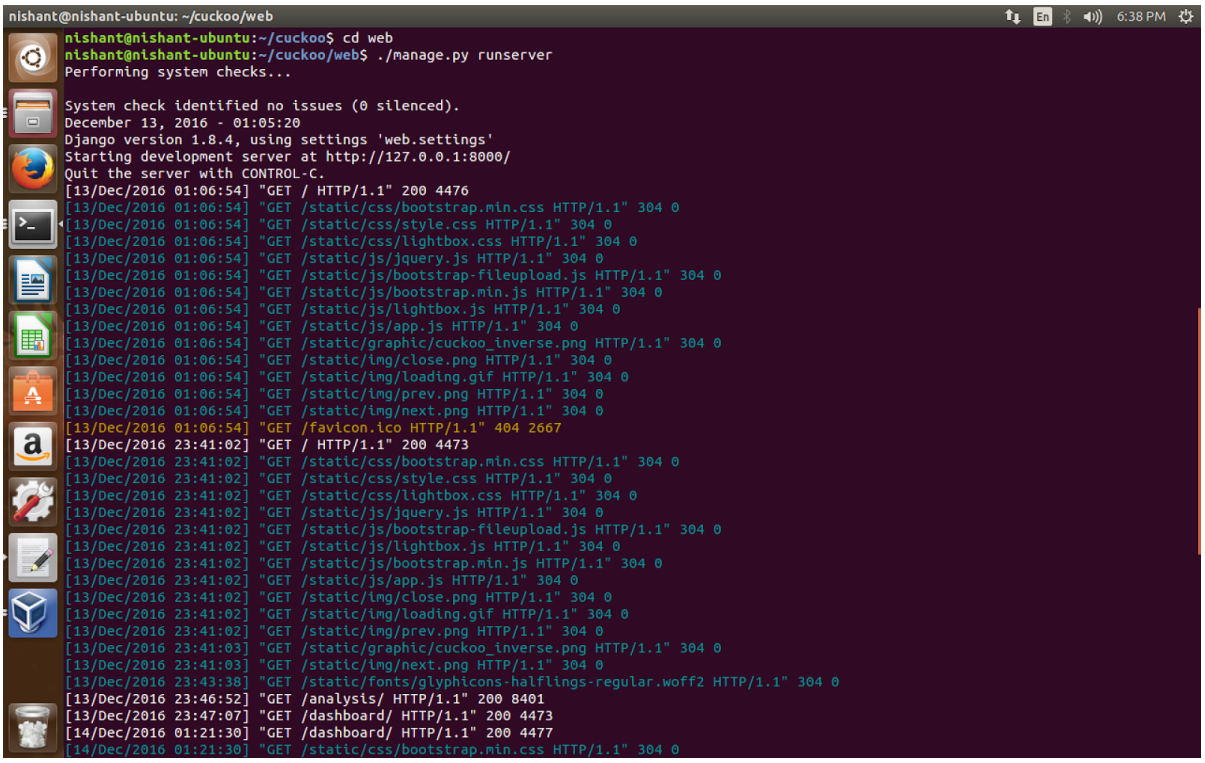


At times, an error occurs which can be resolved by running the below commands as shown in the above screen shot:

$VBoxManage hostonlyif create
$VBoxManage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1 --netmask 255.255.255.0



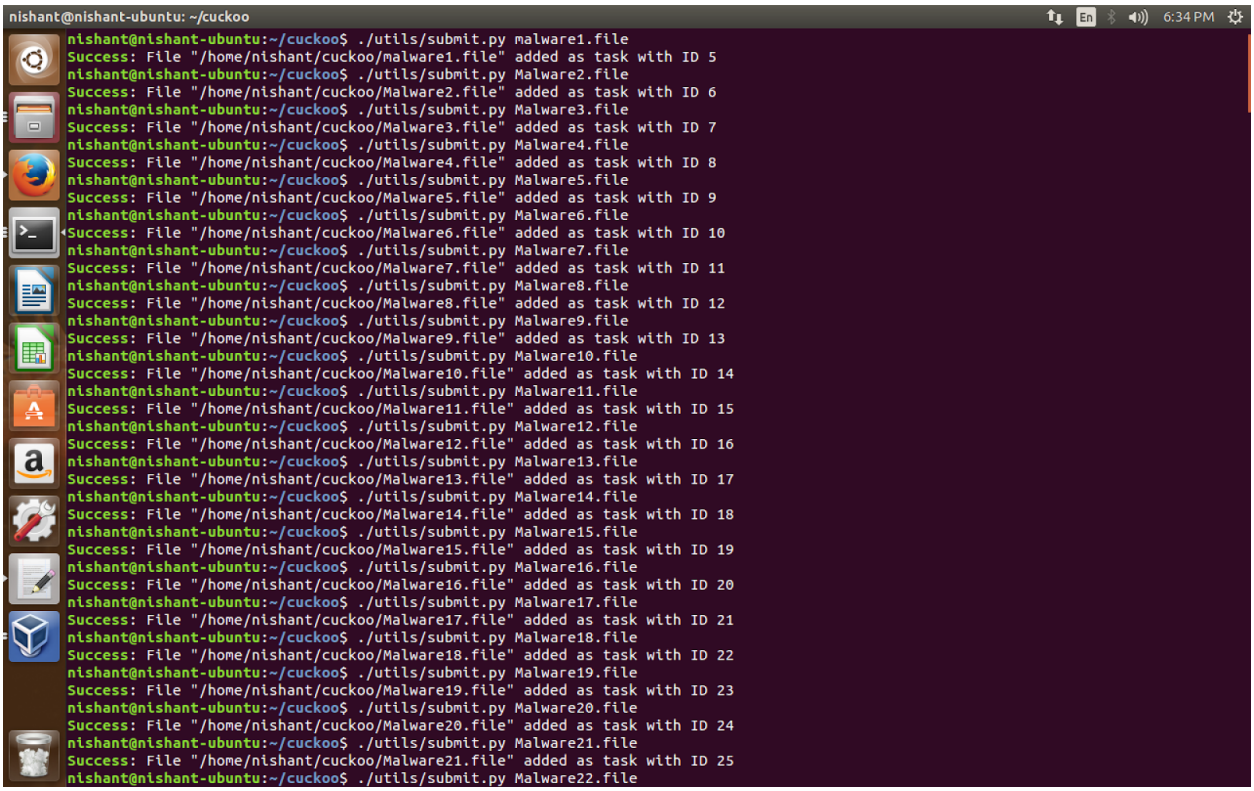
Next open another terminal, navigate to the web folder within cuckoo and run the manage.py file to run the server.



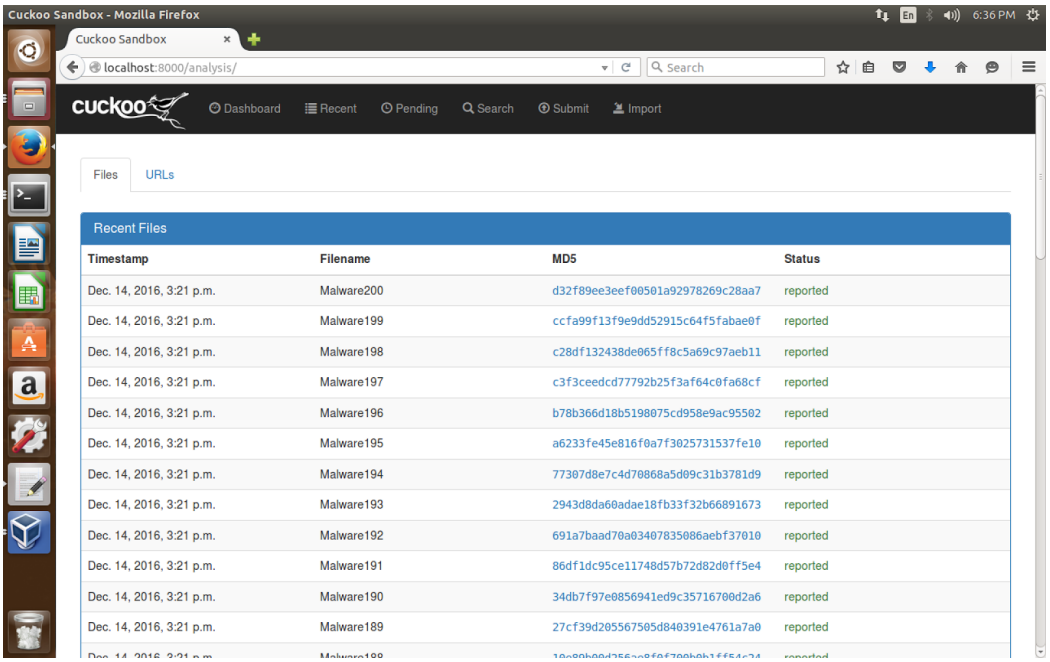
To upload the malware sample for testing on the virtual machine via cuckoo, navigate to cuckoo folder and use the below command.

$ ./utils/submit.py [malware file path]
$ ./utils/submit.py Malware21.file

Malware sample named “Malware21” is located in Cuckoo folder.

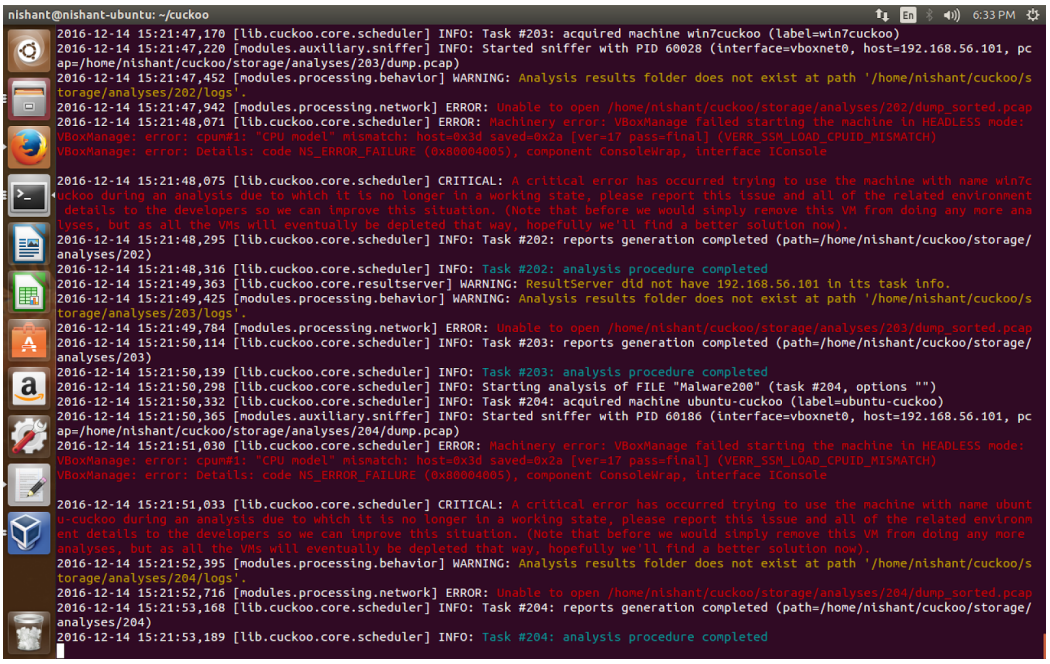


Next, open Firefox and go to localhost:8000 which is the web application for cuckoo.



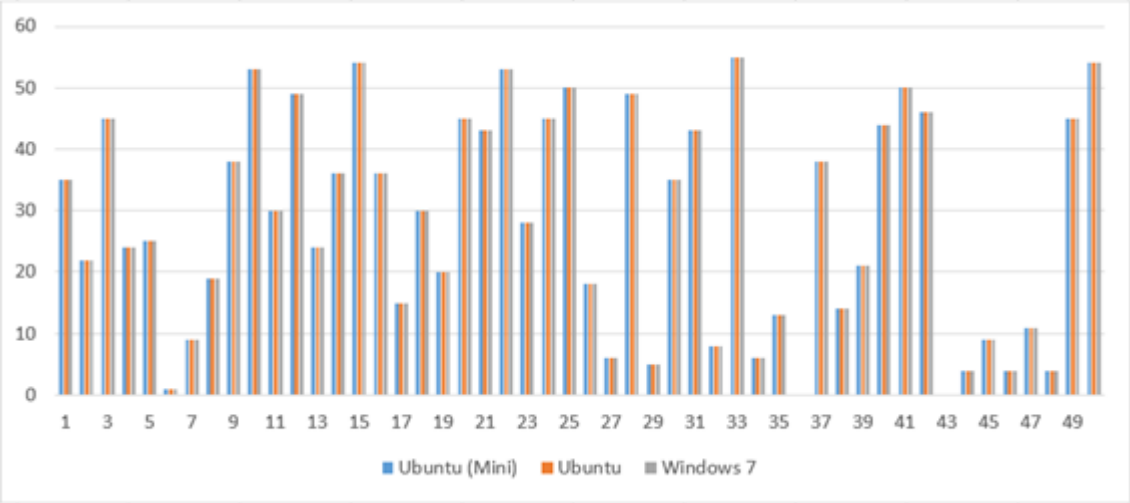
This application provides information regarding the number of malware samples that have been submitted in Cuckoo. It lets us know if any of the submitted malwares have failed, are pending, in progress, reported, etc.

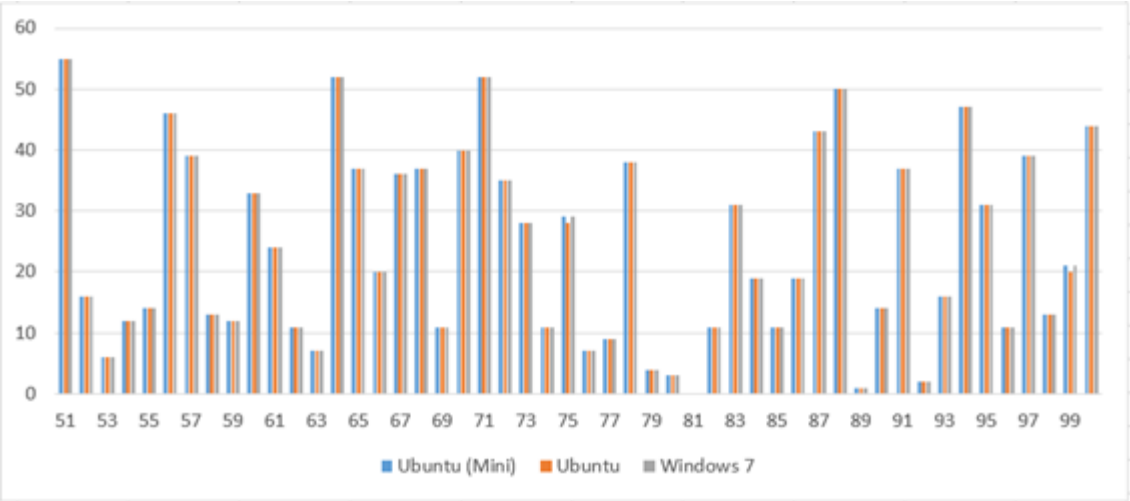
The terminal in below screen shot confirms that the last malware sample run has been successfully completed.

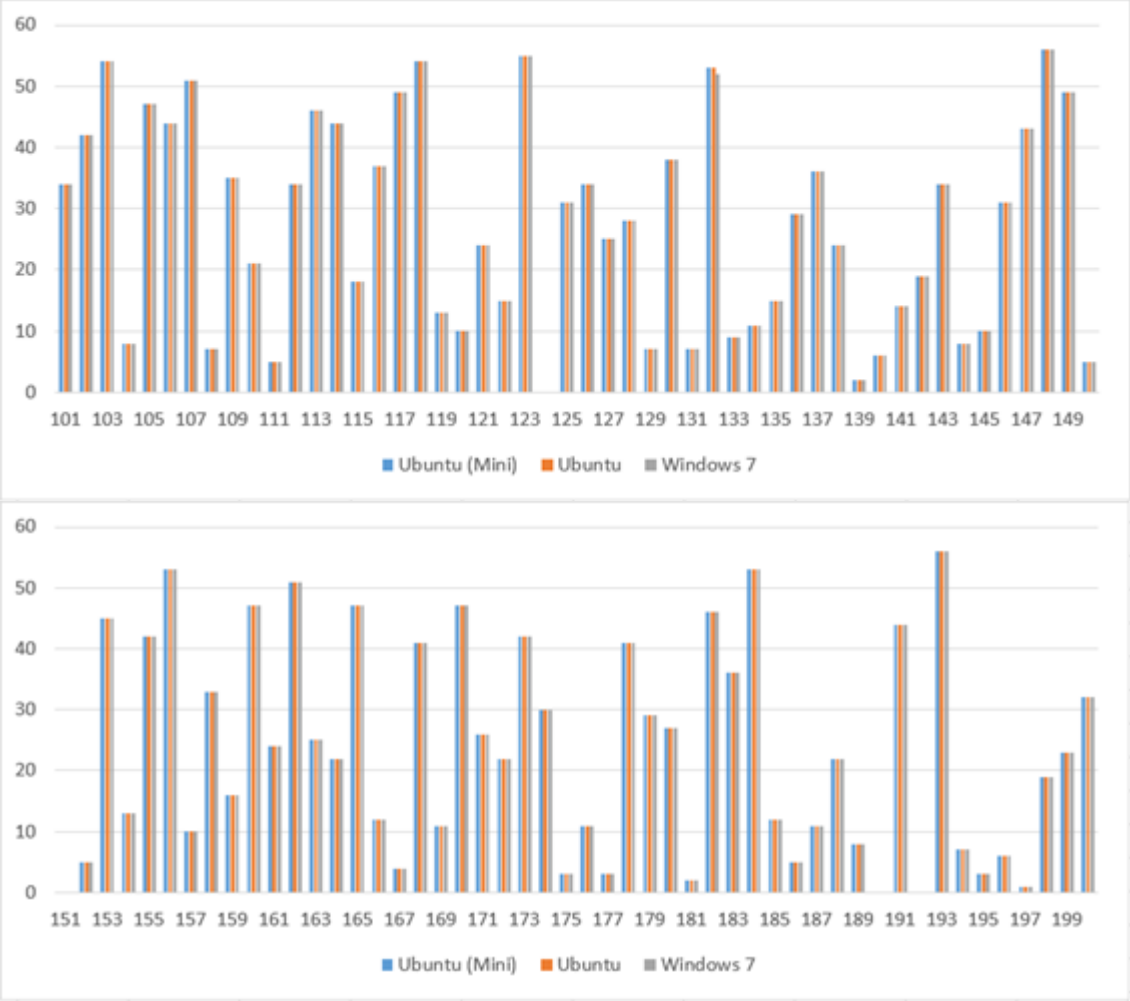


Analysis of Malwares using Cuckoo in Ubuntu (Mini), Ubuntu and Windows 7

200 malware samples were uploaded onto the Virtual environment and their outputs were analyzed. Cuckoo generates JSON reports for each malware uploaded. This report contains information as to whether an antivirus running on that OS would detect the malware file or not. 56 such antiviruses were considered and the information from the JSON reports of the malwares submitted via Cuckoo were parsed to plot a graph, to determine if the same number of antiviruses were detected by the malware on multiple OSes.







After plotting the graph, it was evident that malware 75, 99, 132 were environment sensitive. They were detected with different values in different VMs. This suggested that they were checking the VMs for various parameters and if they were satisfied, they would lay dormant and lie undetected. We labelled these malware as Environment Sensing Malware.

By not providing generic values for such parameters in the sandboxing environments, we could greatly improve the detection rate of malware.

Future Work

We think it is a good idea to port the cuckoo sandbox we currently run on our personal laptops to a server dedicated to running cuckoo. As such, it will provide a platform to continuously check and monitor new threats, and provide real time analysis of unknown samples that need to be labeled. The dedicated server needs to have sufficient amount of memory to be able to spawn multiple VMs for malware analysis and have a good amount of storage to store the analysis logs

to do further statistical analysis on them. These logs can also be used for academic as well as industry purposes. The server can also be used as part of an Intrusion Prevention System.

Another scope for future work is contributing to the open source project itself. Cuckoo sandbox is hosted on Github and it would be nice to add some features to it or increase code coverage for the code by writing test cases for it. This helps to be involved in the community and contribute towards making computing accessible and secure for people and the community as a whole.

Contributions

- Nishant Rodrigues - Installation of cuckoo sandbox, creation of virtual environment for malware analysis.
- Radhika Pai - Testing first 100 samples of malware on the virtual environment and documenting the test results.
- Vignesh Rajan - Testing next 100 samples of malware on the virtual environment and documenting the test results.

All three team members equally contributed in the analysis of the malware samples and in documenting the test results.

References

1. Rieck, Konrad, et al. "Automatic analysis of malware behavior using machine learning." *Journal of Computer Security* 19.4 (2011): 639-668.
2. Balzarotti, Davide, et al. "Efficient Detection of Split Personalities in Malware." *NDSS*. 2010.
3. Lindorfer, Martina, Clemens Kolbitsch, and Paolo Milani Comparetti. "Detecting environment-sensitive malware." International Workshop on Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2011.
4. Inoue, Daisuke, et al. "Automated malware analysis system and its sandbox for revealing malware's internal and external activities." IEICE transactions on information and systems 92.5 (2009): 945-954.
5. Jacob, Grégoire, Hervé Debar, and Eric Filiol. "Behavioral detection of malware: from a survey towards an established taxonomy." *Journal in computer Virology* 4.3 (2008): 251-266.

Appendix

Deliverables:
Our VM is hosted on Google Drive and can be found here: https://drive.google.com/drive/folders/0B36t7HEJk5sdcHFhOU5JMUZxOUU?usp=sharing

The username for Ubuntu 15.10 is 'nishant' and password is 'abcd123'
The password for the virtual machines inside the host VM is also set to 'abcd123'.

Our malware samples were obtained from and can be found at:
http://dasmalwerk.eu/
http://malwaredb.malekal.com/