Dynamic Malware Analysis Using ML

Team : Binary Beasts

Presented By: Tushar (20111071)
Harsika (20111021)
Shubham (20111063)
Sumesh (20111066)
Muskan (20111036)
Abhishek (20111401)

Introduction

We have performed dynamic malware analysis using different machine learning algorithms.

Project Outline:

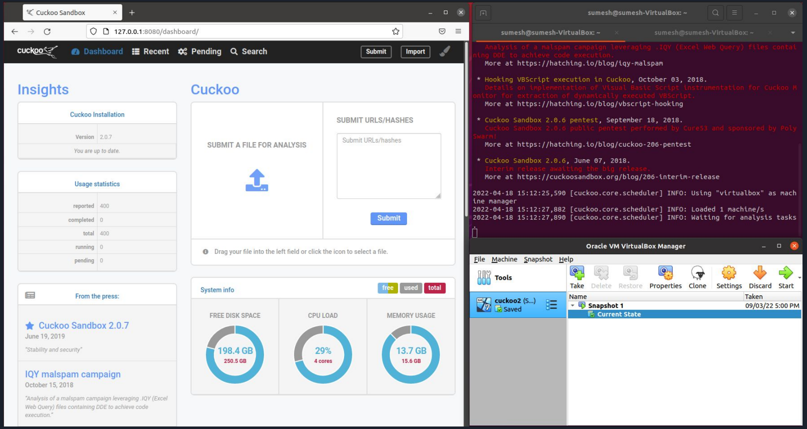
- Collected Portable Executable (PE) files from C3i center, IIT Kanpur.
- Automated installation and configuration of Cuckoo sandbox.
- Then we obtained an analytics report of each PE file using cuckoo sandbox in JSON format.
- Parsed JSON files to collect significant features.
- Pre-processed the data and performed dimensionality reduction
- Applied various ML models and compared their accuracies.

Cuckoo Sandbox Installation

- We had configured Cuckoo Sandbox in a virtual environment.
- Cuckoo host is in Ubuntu 20.04 & Cuckoo guest is in Windows 7 Ultimate.
- As Cuckoo is still based on Python2.7 which is outdated as of now, we had faced errors several times during the setup.
- To overcome various errors, we had used various scripts available on the internet but none of them make installation without any errors.
- So, we have corrected those collection of scripts and used these scripts to install cuckoo on virtual machine and exported that virtual machine in .OVA format.
- Now anyone can install cuckoo by using corrected script or can use cuckoo without installing it by just importing our virtual machine image in their virtualbox.

Cuckoo Sandbox Configuration

- The Cuckoo we had used for our analysis is of version 2.0.6
- We had used Cuckoo guest in “Host-only” network environment.
- For our analysis we configured *cuckoo.conf* for both static & behavioural analysis.
- We had disabled *memory dump* option as it was not needed for our analysis and also it was taking very huge memory space.
- Also, we enabled IP forwarding. So, the internet connection gets routed from host machine to cuckoo guest VM.



Insights

Cuckoo Installation

Version	2.0.7

You are up to date.

Usage statistics

reported	400
completed	0
total	400
running	0
pending	0

From the press:

Cuckoo Sandbox 2.0.7
June 19, 2019
"Stability and security"

IQY malspam campaign
October 15, 2018
"Analysis of a malspam campaign leveraging .IQY (Excel Web Query) files containing DDE to achieve code execution."

Cuckoo

Drag your file into the left field or click the icon to select a file.

System info

FREE DISK SPACE	CPU LOAD	MEMORY USAGE
198.4 GB 250.5 GB	29% 4 cores	13.7 GB 15.6 GB

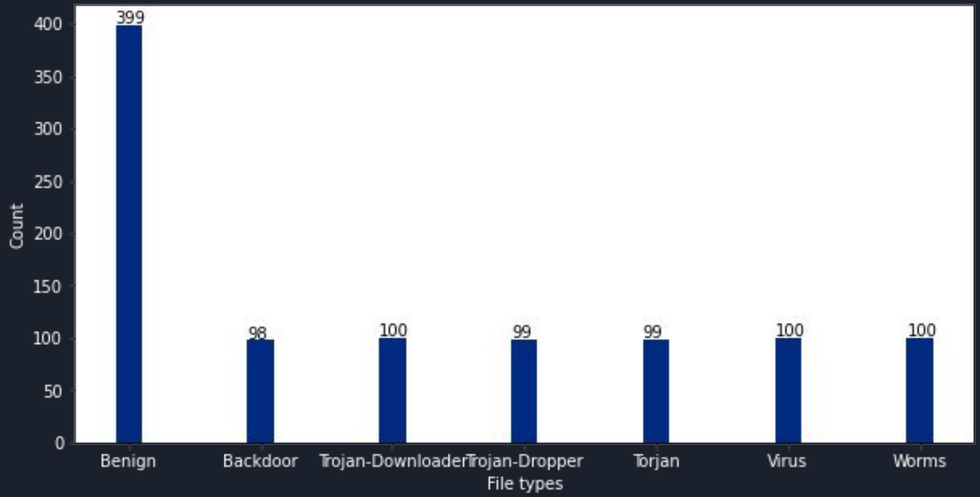
sumesh@sumesh-VirtualBox: ~
sumesh@sumesh-VirtualBox: ~
sumesh@sumesh-VirtualBox: ~

```
  Analysis of a malspam campaign leveraging .IQY (Excel Web Query) files contai
ning DDE to achieve code execution.
  More at https://hatching.io/blog/iqy-malspam

 * Hooking VBScript execution in Cuckoo, October 03, 2018.
  Details on implementation of Visual Basic Script instrumentation for Cuckoo M
onitor for extraction of dynamically executed VBScript.
  More at https://hatching.io/blog/vbscript-hooking

 * Cuckoo Sandbox 2.0.6 pentest, September 18, 2018.
  Cuckoo Sandbox 2.0.6 public pentest performed by Cure53 and sponsored by Poly
Swarm!
  More at https://hatching.io/blog/cuckoo-206-pentest

 * Cuckoo Sandbox 2.0.6, June 07, 2018.
  Interim release awaiting the big release.
  More at https://cuckoosandbox.org/blog/206-interim-release

2022-04-18 15:12:25,590 [cuckoo.core.scheduler] INFO: Using "virtualbox" as mach
ine manager
2022-04-18 15:12:27,882 [cuckoo.core.scheduler] INFO: Loaded 1 machine/s
2022-04-18 15:12:27,890 [cuckoo.core.scheduler] INFO: Waiting for analysis tasks
.
```

Name	Taken
Snapshot 1	09/03/22 5:00 PM
Current State	

Dataset Creation

We have collected JSON files for 995 classified PE files.



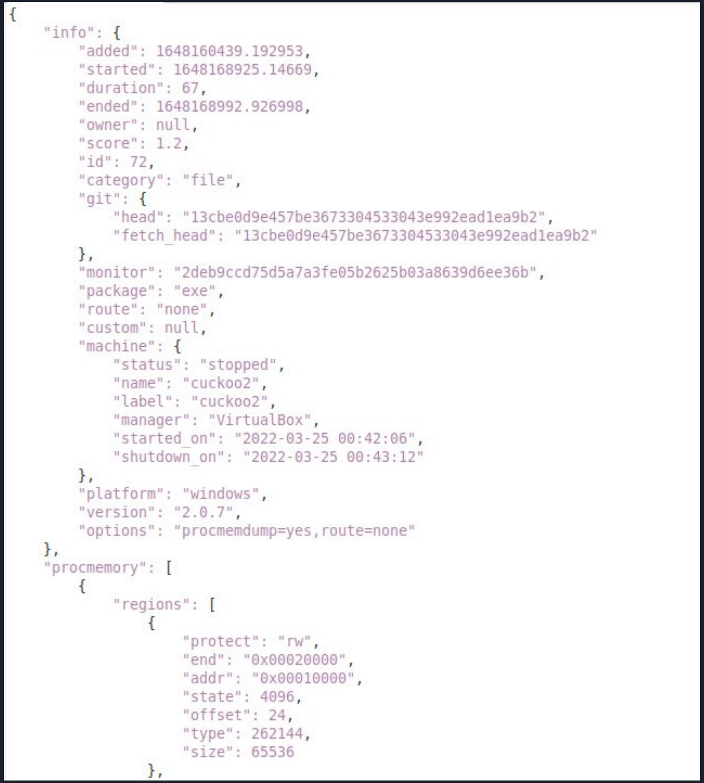
Cuckoo generated Analytic Report for these above file.

JSON file created by Cuckoo sandbox contains lots of information like PE information, signatures, process information, etc.

Features Extracted

Collected the following information as features:

- API Calls
- Files written
- Files deleted
- Regkey written
- Regkey deleted
- DLL loaded
- Duration



```
{
    "info": {
        "added": 1648160439.192953,
        "started": 1648168925.14669,
        "duration": 67,
        "ended": 1648168992.926998,
        "owner": null,
        "score": 1.2,
        "id": 72,
        "category": "file",
        "git": {
            "head": "13cbe0d9e457be3673304533043e992ead1ea9b2",
            "fetch_head": "13cbe0d9e457be3673304533043e992ead1ea9b2"
        },
        "monitor": "2deb9ccd75d5a7a3fe05b2625b03a8639d6ee36b",
        "package": "exe",
        "route": "none",
        "custom": null,
        "machine": {
            "status": "stopped",
            "name": "cuckoo2",
            "label": "cuckoo2",
            "manager": "VirtualBox",
            "started_on": "2022-03-25 00:42:06",
            "shutdown_on": "2022-03-25 00:43:12"
        },
        "platform": "windows",
        "version": "2.0.7",
        "options": "procmemdump=yes,route=none"
    },
    "procmemory": [
        {
            "regions": [
                {
                    "protect": "rw",
                    "end": "0x00020000",
                    "addr": "0x00010000",
                    "state": 4096,
                    "offset": 24,
                    "type": 262144,
                    "size": 65536
                },
```

Dataset Details

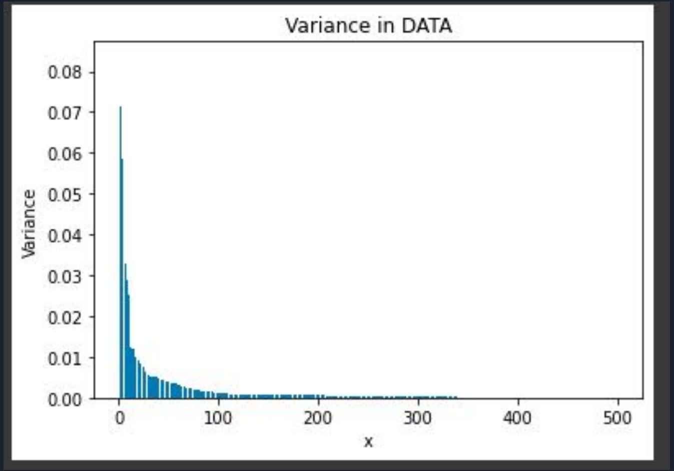
- We treated each api calls, file names and regkey names as features.
- For api calls each cell stores the number of time that call is made.
- For DLLs, files and regkey, each cell contains binary data.
- Duration represents the time duration for which PE is executed.
- Score provided by Cuckoo sandbox is not taken as a feature.

Dataset details after pre-processing

- Dataset details:
  14820 columns; 995 rows
- Classification Labels:
  - Normal
  - Trojan-Dropper
  - Worm
  - Trojan-Downloader
  - Virus
  - Backdoor
  - Trojan

Stages in Machine Learning:

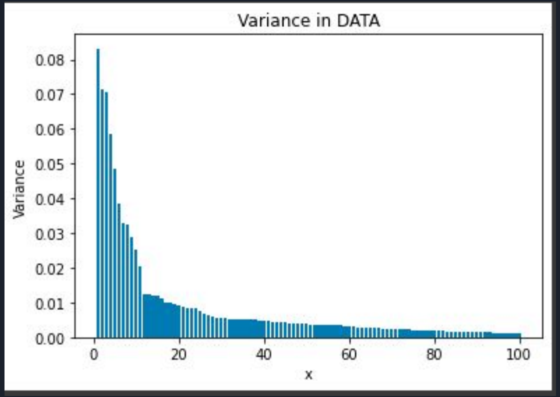
- **Feature Scaling**: Normalised the data before implementing our model.
- **Feature Reduction**: Implemented PCA to reduce from 14k+ features (original) to top K features.



- Implementing and comparing various ML models proven to perform well in classification problems

Stages in Machine Learning:

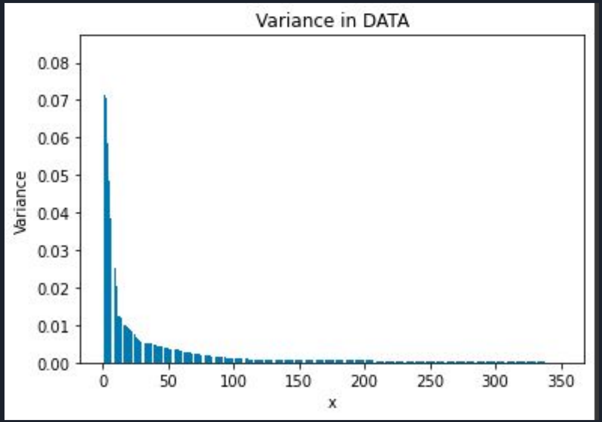
- By Seeing, Variance plot, We took only Top 100 Features :



Model (With 100 features)	Accuracy
SVM	72.36
Decision Tree	89.44
Random Forest	**93.46**
K-Nearest neighbour	89.94
Deep Neural Network	79.30

Stages in Machine Learning:

- By Seeing, Variance plot, We took only Top 350 Features :



Model (With 350 features)	Accuracy
SVM	80.40
Decision Tree	91.95
Random Forest	**93.46**
K-Nearest neighbour	80.40
Deep Neural Network	87.84

ML Classification Models (with Top-500 features):

We tested various ML models with various parameter combinations. We experimented with the parameters of each model and presenting the best performing models as follows:

1. **Multiclass SVM** (best accuracy: 91.46),

Kernel	Accuracy
Poly	41.21
RBF	67.34
Linear	**91.46**

ML Classification Models :

2. **Decision Tree** (best accuracy: 91.45)

Criteria	Accuracy
Gini	88.00
Entropy	91.45

3. **Random Forest** (best accuracy: 93.46)

Estimators	Accuracy
1000	93.46
500	92.46
100	92.46

ML Classification Models :

4. **KNN** (best accuracy: 78.89)

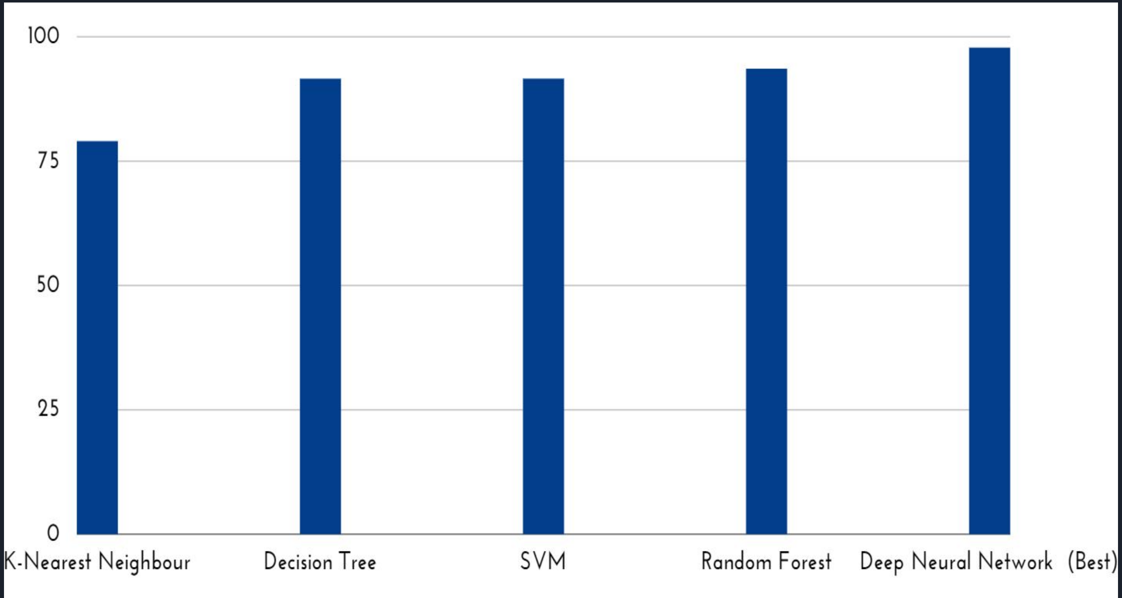
N-Neighbours	Accuracy
15	74.87
7	73.86
5	76.88
3	78.9

5. **Deep Neural Network:**

Layers	Accuracy
3	94.87
4	97.69

Results:

Models	**Best Accuracy**
K-Nearest Neighbour	78.89 %
Decision Tree	91.45 %
SVM	91.46 %
Random Forest	93.46 %
Deep Neural Network (Best)	**97.69 %**



Limitations

- **Small dataset :** Extraction of Analytics Report from Cuckoo took lots of time.
- **Limited significant features :** Taking all fields from JSON files created large feature set.
- **Windows Malware :** Our analysis only covers windows malwares.

Future work

- Extend our analysis for larger dataset.
- Integrate all the pieces to create final malware detection tool.
- Deploy our tool for detecting malicious PE files.

Thank you!