

## **Configuration Manual**

MSc Research Project Cyber Security

Kolarikkal Norman Student ID: x19226365

School of Computing National College of Ireland

Supervisor:

Imran Khan

#### National College of Ireland



#### **MSc Project Submission Sheet**

School of Computing

	Norman Kolarikkal	r8	
Student Name:			
	X19226365		
Student ID:			
	Cyber Security	2020-2	2021
<b>Programme:</b>		Year:	
	Msc Research Project		
Module:	-		
	Imran Khan		
Supervisor:			
Submission	16-August-2021		
Due Date:	~		
	Packer detection using visualisation		
<b>Project Title:</b>			
	4886		
Word Counts	Page Count	18	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

 Norman Kolarikkal

 Signature:

 16-08-2021

 Date:

#### PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)			
Attach a Moodle submission receipt of the online project submission, to each			
project (including multiple copies).			
You must ensure that you retain a HARD COPY of the project, both for your own			
reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on			
computer.			

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

## **Configuration Manual**

# Norman Kolarikkal X19226365:

## **1** Introduction

Our research focuses on detecting and classifying packed samples. We also try to check if we can classify malicious packed files without unpacking the file. In the configuration model, we discuss how to setup the environment and run the program.

## 2 Environmental Setup and prerequisites

## 2.1 Hardware Specification

We have done the implementation with the following system setup:

- OS: Windows 10
- Processor: 10<sup>th</sup> Generation i5 Processor
- RAM: 8GB

We would also recommend having at least 80GB free for downloading the datasets.

### 2.2 Software Specification

Following are the software used for this research method:

- 1. <u>Python:</u> We have coded our implementation in python. In our research, we use python version **3.8.10**.
- 2. <u>Anaconda:</u> For python package management. We used version **4.10.1**.
- 3. <u>Jupyter:</u> We ran our code using jupyter. We used version **7.22.0** in our research
- 4. <u>Python packages:</u> We use multiple available python packages including numpy, tensorflow and PIL. The versions are as below:

Package	Version
Numpy	1.19.5
PIL	8.2.0
Tensorflow	2.5.0
Matplot	3.3.4
Math	1.2.1
pathlib	2.3.5

Cabla	1.	Dackage	versions
able	1:	Package	versions

## 3 Implementation

This section goes through the procedure for setting up the proposed model

#### 3.1 Software Installation

- 1. Install Anaconda by downloading the installer from <u>https://www.anaconda.com/products/individual</u>. It will also install jupyter notebook and python.
- 2. Install required packages. In order to install packages, we open anaconda navigator > Environments > choose base as the environment and search for the above-mentioned packages.

	<b>DA</b> NAVIGATOR		<i>i</i> Upgrade Now Sign in to Anaconda.org
Homstep 1     Homstep	Search Environmen Step 2	Q	Installed   Channels Update index Search
The Environments	base (root)		Name T Description Step 4
Learning	bookEnv		✓ _anaconda_depends ○ 202
Community	myDjangoEnv	]	<ul> <li>_ipyw_jlab_nb_ext</li> <li>A configuration metapackage for enabling anaconda-bundled jupyter extensions</li> <li>0.1.0</li> </ul>
	rstudio	<	■ absl-py ◆ Abseil python common libraries, see https://github.com/abseil/abseil-py. 0.13
	►F		☑ alabaster O Configurable, python 2+3 compati 0.7.
Documentation	u		■ anaconda O Simplifies package management and deployment of anaconda > cust
Developer Blog			■ anaconda-client O Anaconda.org command line client > 1.7.
You 💭	8 8		anaconda-project O Tool for encapsulating, running, a > 0.10
	Create Clone Import Re	move	402 packages available

Figure 1: Anaconda installing packages

3. For converting a file to an image, open "final.ipynb" using jupyter. To open a .ipynb file, open jupyter notebook from anaconda. Once the notebook is open, change the path in the third cell to the file which needs to be converted (if a single file is to be converted) or put the directory path in fourth cell if a batch of files needs to be converted.



Figure 2: Changing path for the file conversion

The converted file would be saved in the root directory which would be defined when anaconda was installed.

- 4. To train the model, run cells sequentially from the first cell till before the "Running Evaluation dataset" section. It is necessary that the correct path in input in the data\_dir variable which is the directory where the samples are stored. The steps followed are as follows:
  - a. Read the images from the path in data\_dir.
  - b. Preprocessing the image. We downsample the images so that the image size is 128x128. We also set the batch size as 32. The batch size depends on the number of samples and the computing power available. From our tests, 32 is a good choice for batch\_number.
  - c. Split the dataset 9:1 for training:validation.
  - d. Configure the layers in CNN model.
  - e. Run the model against the split dataset. The metrics for the model is accuracy, so it will provide the accuracy for each epoch run. We have currently set the epoch to 100.
  - f. The final step would print the accuracy and loss of the model.
- 5. Once the file has been converted, load the predictions matrix to run the file against the trained CNN model. To do this, run the second last cell to load the json file (model.json) and the corresponding weights (model.h5). Once the wights are loaded, the model can be run against a file by running the last cell. To select the file, the filename along with the path should be put in the file variable.

```
Loading model predictions
]: json file = open('model.json', 'r'
   loaded_model_json = json_file.read()
   json file.close()
   loaded_model = model_from_json(loaded_model_json)
   # load weights into new model
   loaded_model.load_weights("model.h5")
   print("Loaded model from disk")
]: file="path to file"
   img = keras.preprocessing.image.load_img(
           file, target_size=(img_height, img_width)
   img array = keras.preprocessing.image.img to array(img)
   img_array = tf.expand_dims(img_array, 0) # Create a batch
   predictions = model.predict(img array)
   score = tf.nn.softmax(predictions[0])
   f.write("{}, {}\n".format(dir,class_names[np.argmax(score)]))
print("{}, {}\n".format(dir,class_names[np.argmax(score)]))
```

Figure 3: Running trained CNN model against a converted image