Lab -1 Objective:

This Lab session includes two main parts. In the first part Operational Codes (OpCode) from a binary collection of Benign and Malware files will be extracted. And the second part, this extracted information will be parsed and pre-processed to convert them into vectorized format to feed into machine learning models.

Materials:

There is a folder in your ML virtual machine namely **"CIS6530-Day-1/Lab-1".** This file includes a "data" folder that includes two subfolders **"malware"** and **"benign".** Each folder includes a collection of about 100+ binary files. Binary files are IoT executable malware and benign files compiled for ARM machines. *Please don't try to run this file on this VM or other machines*. Also**, "data (copy)"** is a copy of data to have access in case of unintentional file remove and etc.

Also, there is a sample of decompiled file namely **"SampleDissassembledFile.opcode".** You can open and see the structure of decompiled file.

Part-1 (Estimated time :10 minutes):

“Objdump” is a simple tool that you can use it to decompile binary codes. In this part, you should write a simple Linux bash script that decompiles files of each malware and benign folders. The name of extracted files should be “X.opcode” where “X” is the name of original binary file. To complete the first part of lab please take following steps:

- Write a bash script in **“CIS6530-Day-1/Lab-1”** that includes two main loop. The top loop is over “malware” or “benign” folders and the second loop over files within each folder.
- For each file in folder use “objdump” command to decompile binary file.
    - You can use this command to decompile ARM-based binaries:
        - **objdump -d $filename -m i386**
        - or write the output into a .opcode file **objdump -d $filename -m i386 >> $filename'.opcode'**

- after about 5 minutes of starting the lab session you can run following script within the **“CIS6530-Day-1/Lab-1”** which download the solution from GitHub and you can see it:
    - run ./download_part_1_soloution.sh
    - see & run solution: “./extract_opcode.sh”

Part-2 (Estimated time :20 minutes):

In the previous part, each binary file was decompiled into an. opcode file. However, the. opcode file is a textual file that includes too many texts and characters that we can not feed this file directly to ML models. To convert it into a suitable format for machine learning models, we should write a parsing python script that loads. opcode files and parses them and convert it into matrix format that includes information about samples we want to train an ML model with them.

- Run "jupyter-lab" and create a new python notebook.
- Write and test your script that reads all .opcode files and create a single Numpy matrix call it **"final_list"** somehow:
  - Each row represents information related to a sample
  - Columns represent information about occurrence of opcodes
  - For example, **final_list[i][j]** shows the count of j'th opcode in i'th .opcode file.
  - So,
    - Number of final_list's rows = number of .opcode files
    - Number of final_list's columns= number of unique opcodes.
    - ***Tip: based on the samples in the data folders, Number of final_list's rows=264 and Number of final_list's columns=532***
- after about 15 minutes of starting the lab session you can run following script within the **"CIS6530-Day-1/Lab-1"** which download the solution from GitHub and you can see it:
  - run ./download_part_2_soloution.sh
  - open & run notebook solution: "Lab-1.ipynb" by jupyter
  - try to understand the solution script logic by adding some prints for different variables.
  - **-> final_list** is almost ready to use for training an ML model. So, in the next labs we are looking for similar vectorized format to train an ML model.