**Transfer Learning Plan (as of October 13, 2021)**

The first part of the research comprised of completing a GAP analysis via literature reviews, investigations into INL and MSU systems, and brainstorming new research for applying ML/TL to the field of cybersecurity and code Quality Assurance.

The driving force of the research for the second part is a belief that, currently, two factors that most likely impact accuracy in ML are:

* ***Data.*** The better the quality and the more the data, the higher the accuracy in general.
* ***Input Representation***. How you represent the problem for a particular task

Data is often limited. Extra data can be found from related domains with Transfer Learning.

We look at three input representations:

1. Translating code into 2D images.
2. Counting machine-code instructions
3. Translating graph representations of code into vectors.

**Data sets**

TL pretrained models:

* Benign versus Malware. Hope to get 800,000 malware and 200,000 benign
* Pretrained image classification models from the community

Transfer to:

* Vulnerabilities measurements matching MSU PIQUE nodes or INL CWE codes
* Other types of Malware.

**Steps to completion:**

We need to:

1. obtain a directory of binary files for Malware, benign, and vulnerable code,
2. create code to make datasets for the various representations,
3. Run experiments on the datasets and compare it with transferring knowledge.
4. Analyze the results. Make alterations. Run experiments. Repeat until satisfied.
5. Write up results.
6. Deliver software and results.

* **Obtaining binaries.** For Malware, we desire on the order of a million code samples from Hoplite, VirusShare, etc. We desire around 200,000 benign code samples, perhaps from Linux or Windows system files.
* **Create ML datasets from the binaries:**
  + *Instruction counts*: We generated one using Pyvex, but the results are not the same as what is shown in @Disco. Either need to improve our code, or preferably get access to @Disco.
  + *Images*: Have code to go from binaries to a single-band grayscale image.
  + *Graph representation*:
    - Problem: the resulting vector depends on the whole set of graphs and not just one graph.
    - Problem: Running out of memory during execution. Takes long time to run.
* **Experiments:**
  + Currently have scripts for TensorFlow. Can use many other ML algorithms.
  + Need to collect previously trained publicly available image classification models.
  + Once datasets are completed:
    - Need to set hyperparameters (exact input representation, layers, learning rate, connectivity, activation functions, etc etc). Lots of CPU time required.
    - Need to create transfer models.
    - Need to test efficacy of transfer, including setting many hyperparameters.
    - Many iterations of looking at results, refining learning and perhaps datasets, rerunning results.
* **Current Hypotheses to be tested**
  + Which representations work best for identifying vulnerabilities/malware in code? Is graph representation the best?
  + How well can we identify vulnerabilities of interest?
  + Can we transfer knowledge from related domains to more quickly identify vulnerabilities?

**Resources:**

* Need personnel and machines at MSU to transfer the binaries into the ML datasets. The plan is to create a directory of the binaries. We then run the code for converting the binaries into various ML datasets (as indicated above)
* Once datasets are created, the experiments can be run anywhere since the datasets are merely descriptors of the binaries and not functioning code. The experiments will be time consuming and need good number crunching machines with good GPUs.
* Could use one student inside MSU creating datasets who doesn’t know ML (Reese) and another student concentrating on ML and TL which can be done and run anywhere.
* Could use a good number crunching machine (5K or so) to run experiments.

**Deliverables:**

* Results and writeup will be like any standard ML paper.
* Software will consist of scripts to automate generating a model from data. Previously generated ML models will also be delivered to be used as related knowledge to jump-start the learning process.