Red Teaming Artificial Intelligence

Breaking CUDA and Nvidia Jetson Nanos^{TM}… For Science!

Summer 2022

Team Members

Taylor McCampbell
- Senior, COSC Major
- INL Intern

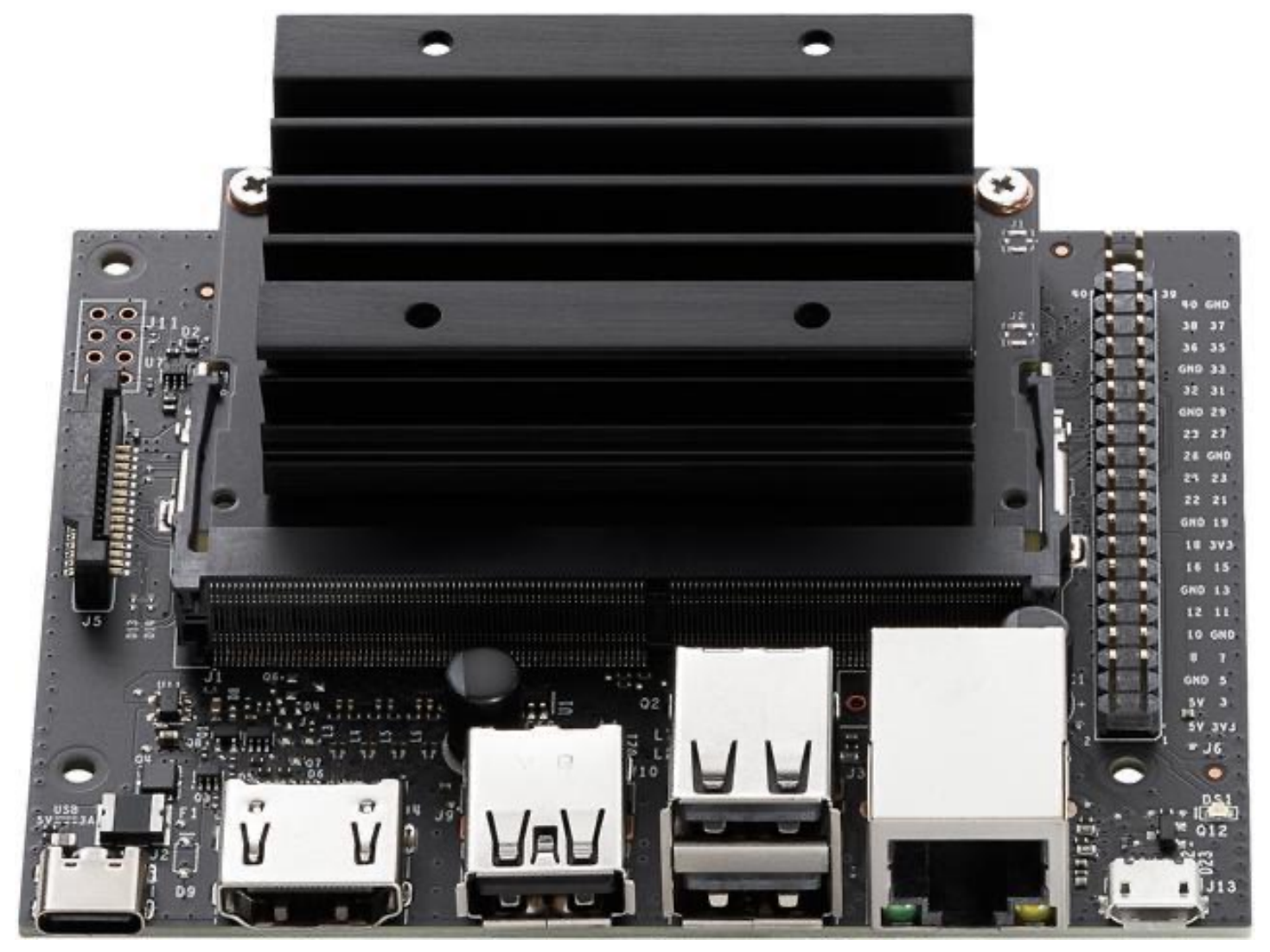
William Brant
- Senior, COSC Major
- INL Intern

Background

- From water treatment facilities to self driving cars artificial intelligence is getting deployed in more and more critical systems. If these deployment environments are found vulnerable from any attack vector the results could be catastrophic.
- Compute Unified Device Architecture (CUDA) is a parallel computing platform only compatible with Nvidia GPUs
- Nvidia GPUs and CUDA are seeing widespread use across critical ecosystems in the private and public sector.
- During the lifespan of CUDA there has been one reported CVE that involved a buffer overflow error in the CUDA Toolkit.

Problem Statement

- Can we utilize input fuzzing and other common attack techniques to uncover more vulnerabilities that lead to privilege escalation or code execution within CUDA and Nvidia Jetson Nanos^{TM}?



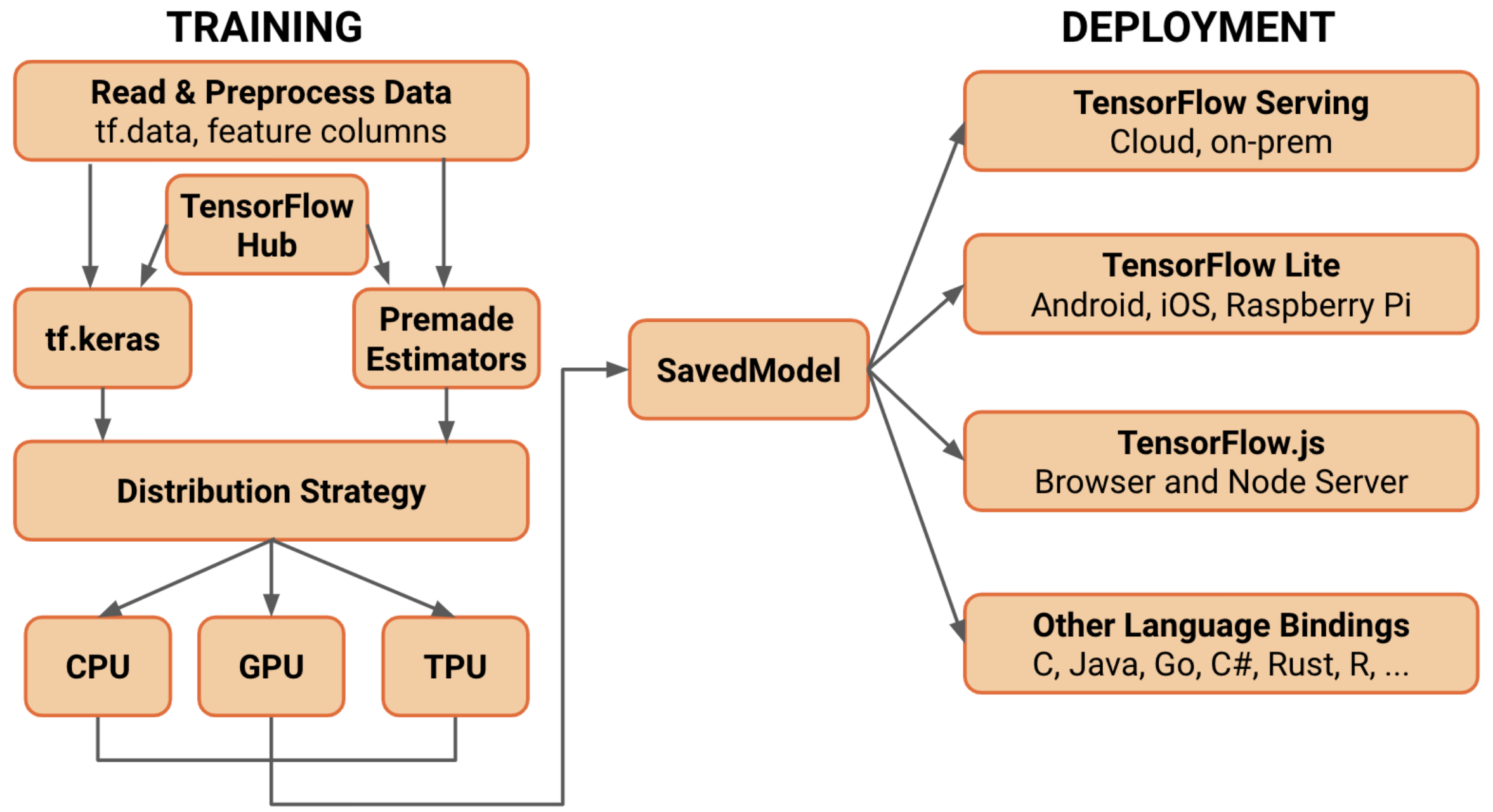
https://c1.neweggimages.com/ProductImage/13-190-013-V01.jpg

NVIDIA CUDA

https://developer-blogs.nvidia.com/wp-content/uploads/2020/07/NVIDIA-A100-GPU-1.png

Methods

- Utilizing other common attack techniques on CUDA.
- Attacking common open-source machine learning pipelines like Hugging Face and Tensor Flow looking for any opportunity of exploitation.

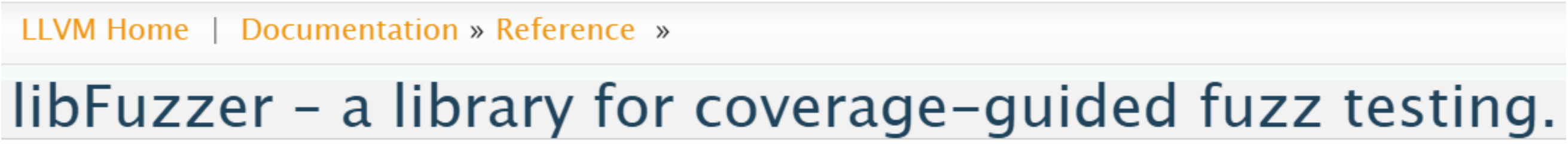


https://1.bp.blogspot.com/-4C_bx62kOI4/XfE3XQT422I/AAAAAAAABmY/AbMfOO8yzjctmg30IcgOBaU5UmcZNpAtwCLcBGAsYHQ/s1600/model.png

Results

- Current results center around open-sources fuzzers, but with severe limitations

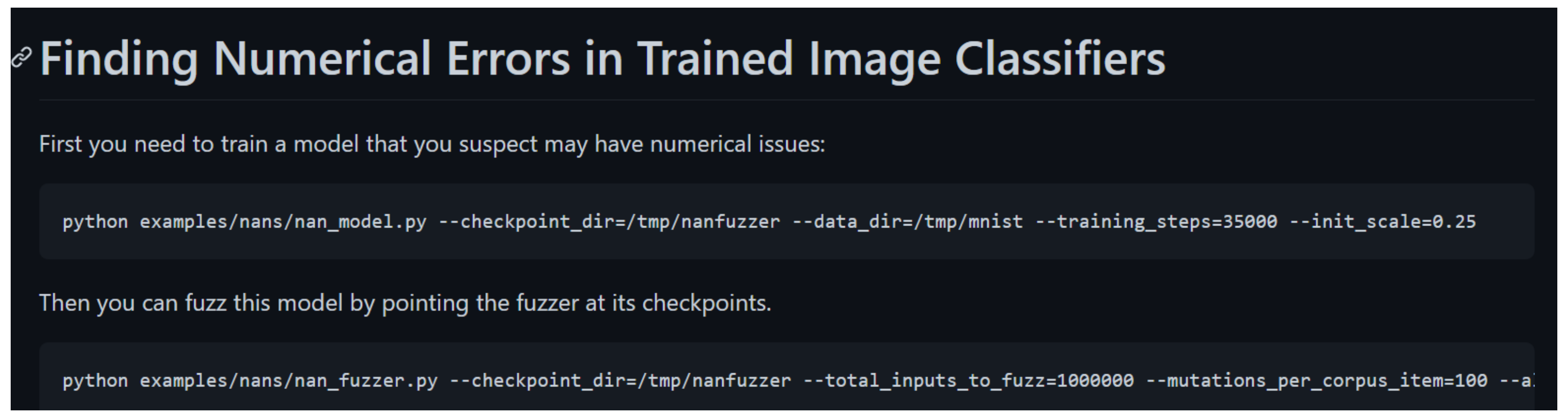
LLVM COMPILER INFRASTRUCTURE



LLVM Home | Documentation » Reference »

libFuzzer – a library for coverage-guided fuzz testing.

https://llvm.org/docs/LibFuzzer.html



Finding Numerical Errors in Trained Image Classifiers

First you need to train a model that you suspect may have numerical issues:

```
python examples/nans/nan_model.py --checkpoint_dir=/tmp/nanfuzzer --data_dir=/tmp/mnist --training_steps=35000 --init_scale=0.25
```

Then you can fuzz this model by pointing the fuzzer at its checkpoints.

```
python examples/nans/nan_fuzzer.py --checkpoint_dir=/tmp/nanfuzzer --total_inputs_to_fuzz=1000000 --mutations_per_corpus_item=100 --a
```

https://github.com/brain-research/tensorfuzz

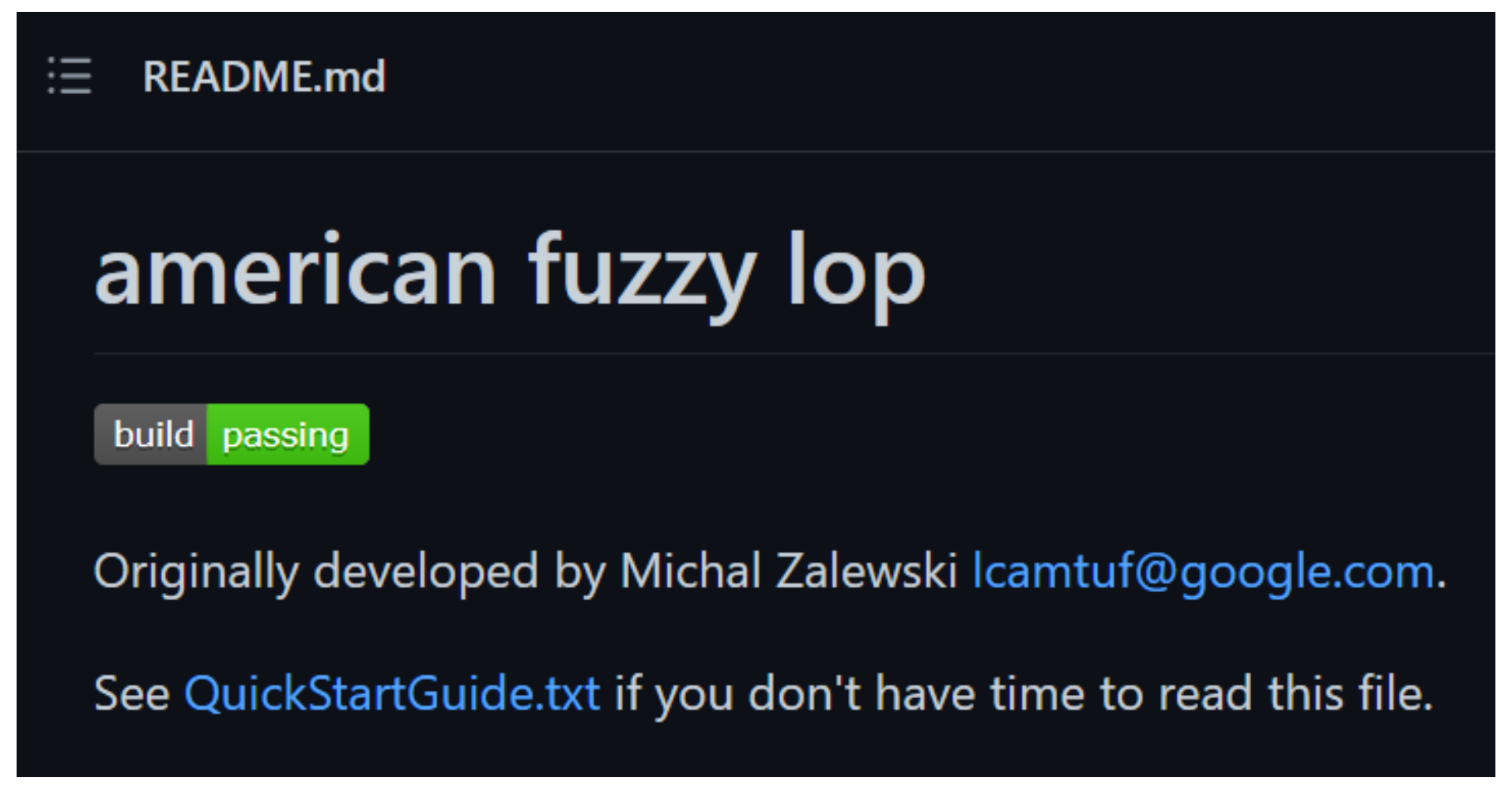
Challenges & Future Work

- Difficult area for fuzzing and other attack techniques.
- Create reports on reproducible attacks that lead to privilege escalation or execution of arbitrary code within any machine learning pipeline.

AFL

american fuzzy lop

https://afl-1.readthedocs.io/en/latest/index.html



american fuzzy lop

Originally developed by Michal Zalewski lcamtuf@google.com.

See QuickStartGuide.txt if you don't have time to read this file.

https://github.com/google/AFL

Advisor: Dr. Mike Borowczak

Group Members:
- William Brant (wbrant@uwyo.edu)
- Taylor McCampbell (tmccampb@uwyo.edu)

University of Wyoming

Idaho National Laboratory

UW College of Engineering and Physical Sciences

UW Cybersecurity Education and Research Center

UW School of Computing

UW College of Engineering and Physical Sciences Electrical Engineering and Computer Science

CEDAR Cybersecurity Education And Research Center