
EE/CPRE/SE 492 BI-WEEKLY REPORT 02

February 18 – March 4

Group number: 16

Project title: Robustness of Microarchitecture Attacks/Malware Detection Tools against Adversarial Artificial Intelligence Attacks

Client &/Advisor: Berk Gulmezoglu

Team Members:

Shi Yong Goh

Connor McLoud

Felipe Bautista Salamanca

Kevin Lin

Liam Anderson

- **Bi-Weekly Summary:**

Since our last bi-weekly report, we have met with our advisor twice and discussed our current progress as well as what we'll be working on next. We've each continued working on our individual tasks as discussed with our advisor. These include further development and testing of the GUI, error handling /reporting, experimenting with x86 instructions in the attack code, testing the instructions with C-code, data collection, and comparing the results against our ML model. We've also begun clean up and refactoring of the GUI as well as begun testing vector x86 instructions. Lastly, we've continued to build upon our success with fooling the model and adversarial examples.

- **Previous Week's Accomplishments:**
  - Shi Yong Goh:
    - Through experimentation with the sleep and floating-point instructions, including fadds and fld, able to determine the minimum number of instructions required to affect the results of the ML model. Comparing the execution time of the source code with the original source code.
  - Connor Mcloud:
    - Designed an implementation & source code for basic warning notifications & errors in the GUI using Qdialouge boxes from python Qt6. Continued researching a way to handle checking errors during the attack codes runtime.
  - Felipe Bautista:
    - I went refactored a lot of parts of the code to have more meaningful variable names that make the program easier to understand. I tried to optimize some section of code that were not very efficient. Began implementing the widgets that will be utilized to handle errors when while using the GUI.
  - Kevin Lin:

        - Focused on profiling various floating point x86 instructions, namely floating-point loading, multiplication, division, and square root. Testing of these attack codes on machine learning model.
    - Eduardo Robles:
        - Tested with logical x86 instructions. Inserted these instructions into the attack code and ran it against the ML model.
    - Liam Anderson:
        - Experimented with x86 instructions. Created more c and attack codes to test these instructions and better understand the situation.

- **Pending Issues:**
    - Shi Yong Goh:
        - N/A
    - Connor Mcloud:
        - Unable to process warnings for attack code during runtime. Unable to push/pull from git on my laptop, researching t
    - Felipe Bautista:
        - N/A
    - Kevin Lin:
        - Unable to compile AVX/AVX2 vector instructions – running into errors with needing a specific flag.
    - Eduardo Robles:
        - Ran into many errors when compiling some x86 code
    - Liam Anderson:
        - Get vector instructions working
        - Systematically profile x86 instructions and collect data

- **Individual Contributions:**

Team Member Names	**Individual Contributions**	**Hours** (this week)	**HOURS** (cumulative)
Shi Yong Goh	Experimented with x86 instructions in c source code.	7	42
Connor Mcloud	GUI research, development, & debugging	5	33
Felipe Bautista	GUI code refactoring & implement error handling	6	38
Kevin Lin	Testing various forms of inserting x86 instructions; repeated x86 instructions, small code snippets/programs, and a combination of the two. Presentation of power signatures and inference against ML model.	8	43
Eduardo Robles	Continued testing on different x86 instructions and checking the accuracy against the ML model	7	40
Liam Anderson	Created a variety of C codes that test out x86 performance and power consumption. Collected data and presented results to the team.	12	48

- **Plans for the Upcoming Week:**
  - Shi Yong Goh:
    - Will try vectors instruction on the source code.
  - Connor Mcloud:
    - Finish Implementing error handling for the GUI. Fix issues with connecting to git.
  - Felipe Bautista:
    - Create comments on the code to document the functions of the program. Fix any issues that my occur when the error handling code gets pushed to the main branch.
  - Kevin Lin:
    - Testing of vector instructions in the attack code and check for any instructions that cause a statistically significant change in machine learning model confidence.
  - Eduardo Robles:
    - Move onto more instructions to test on the attack code.
  - Liam Anderson:
    - Create addition scripts and C programs so we can systemically collect data on x86 instructions performance and power consumption

- **Summary of Weekly Advisor Meeting:** After speaking with our advisor, we determined that the team should focus on refining the GUI by adding in error detection, and putting more of a focus on instruction profiling, specifically for the Spectre attack. Testing of vector instructions will be important, as they have the potential to cause a spike in power consumption, which would likely trick the machine learning model. There should be a focus on finding the minimum number of instructions needed to trick the model.