ENS 491 – Graduation Project (Design)

Progress Report I

Project Title: Reproducing and Analyzing Side Channel Attacks

Group Members:

Ali Parlakçı (28114), Doğukan Yıldırım (28364), Rana İşlek (27836)

Supervisor(s): Cemal Yılmaz

Date: 05.01.2023

1. PROJECT SUMMARY

The project is about side-channel attacks which are security exploits that try to gather information from the medium. To make it easier to understand, that is an attack where an attacker steals someone's credit card PIN via the fingerprints on the POS machine is a side-channel attack. The vulnerability is caused by the operation's overall nature and basic architecture, not by a fault in the procedure itself. In a similar vein, attackers take advantage of specific features and operational aspects of computers. Computers can only do one task at a time, despite the appearance that they can multitask and complete activities simultaneously. However, they switch between tasks so fast that it creates the illusion of parallelism. One of the things that will cause computers to switch is interruptions. When an interrupt happens, the operating system enters kernel mode, suspending all other processes so that the interrupt can be handled. Typically, interruptions are brought on by an external force, like a newly received network packet. Context switches are used to describe this task switching.

A prior study indirectly monitored computer interrupts. They keep a straightforward loop-counter running in a browser and occasionally read the value. In the meantime, they loaded other websites in a different browser tab. As the browser loads the website, interruptions take place because computers carry out operations in a linear fashion. The counter in the other will be different in each period as a result of how the interruptions are processed. According to a study, using the increment values in the browser, a machine learning model can predict which websites are open in the browser when the counter data is provided to it (Cook et al., 2022).

As we stated in the proposal, the project has some critical objectives and tasks which can be worth recalling. Before moving forward with the project, it was essential to ensure that everyone engaged has acquired a fundamental understanding of the existing study because this project primarily focuses on a literature review about side channel attacks. Our main source of information based on two papers which are There's Always a Bigger Fish: A Clarifying Analysis of a Machine-Learning-Assisted Side-Channel Attack (Cook et al., 2022), and On the Effectiveness of Using Graphics Interrupt as a Side Channel for User Behavior Snooping (Ma et al., 2021). Every team member cloned the There's Always a Bigger Fish (Cook, 2022) GitHub repository mentioned

in the research paper, ran and tested the source code, visualized the data obtained by running the code with Python data visualization libraries, and trained the default machine learning models provided in the repository and tested their accuracy. So that, each project member has become familiar with the source code of There's Always a Bigger Fish, and has verified that the source code works as intended on their personal or virtual computers, and if not working as intended, has taken notes of the problems in code and/or data.

Since our project is a software development/computer engineering project, time, scope, and cost (economics) are the three most important constraints that are dependent on each other and also affect the quality of our final product. Our aim is providing the best possible working environment for every team member to have the greatest outcome from the project.

2. SCIENTIFIC/TECHNICAL DEVELOPMENTS

As proposed earlier, we started with analyzing methods in the prior work: There's *always a bigger fish: a clarifying analysis of a machine-learning-assisted side-channel attack* (Cook, et al., 2022). We first imitated the experiments which were about trying to determine the target user's internet browsing history by collecting data during browsing. Each group member executed the experiments on their behalf and gathered results. Our findings were consistent with stated results in the article.

We, then, used the same methods but developed different experiments. The current method was to inject a spy process to the user's browser and collect data as computer executed tasks. As for the machine learning model, a 10-fold Random Forest Classifier was used. Instead of browsing the internet, we played short videos for fifteen seconds in each trial. The videos had four different versions which were encoded with different codecs. We also used three different video players. When the video player was the control variable, the method was able to guess the video codec correctly nearly half the time. However, when the codec was fixed, the model was more successful. It correctly predicted which video player the video was played on with an accuracy of ~80%.

While carrying out the experiments, we also found out that the model was more accurate on a Linux system throughout different experiments. The collected data were always less relevant (included noise) in Windows systems than its counterparts.

We improved the testing source code to support different browsers for collecting data. Microsoft Edge, Mozilla Firefox and Apple Safari were chosen in addition to Google Chrome. However, we have not executed new experiments for this new independent variable, yet.

3. ENCOUNTERED PROBLEMS

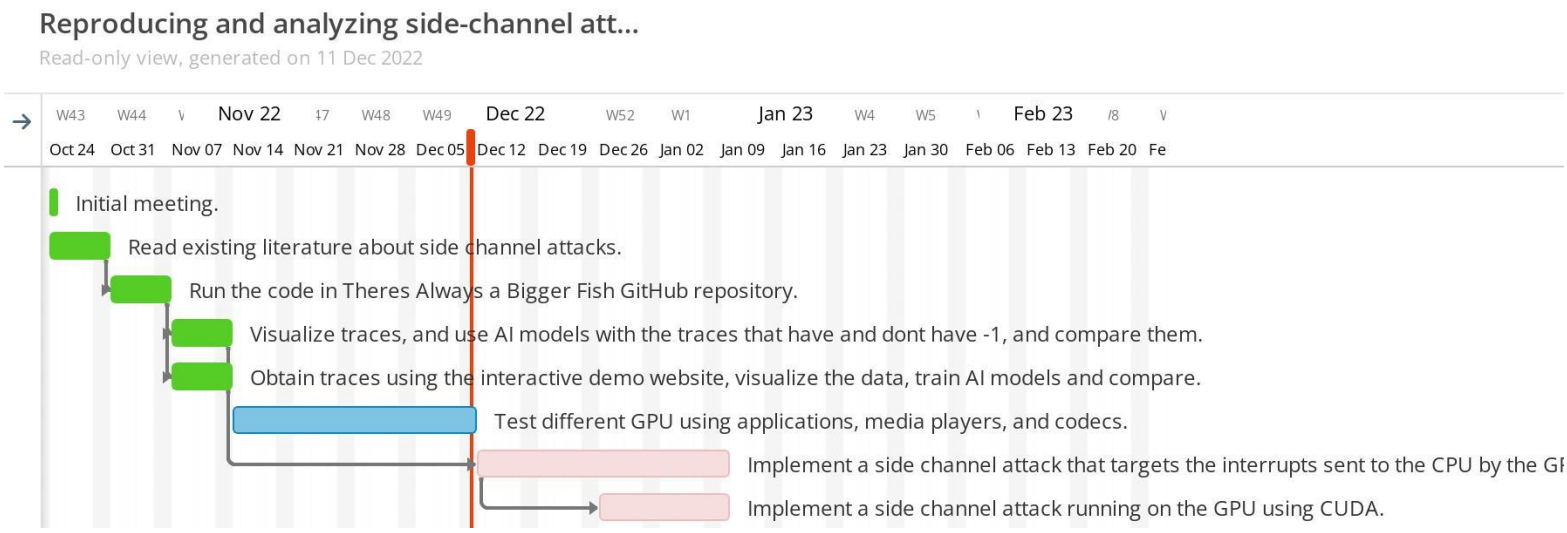

Figure 1: Gantt Chart for the first term

Figure 1, above, represents the time interval starting from the beginning of the term until the end which we also mentioned in the proposal. In the time table, the similar tasks are colored as the same with respect to the task definitions to categorize them together. As a team we could complete green tasks on time, but the blue one lasted longer than we expected. The reason behind this is the technical issues about the testing procedure.

In testing it is so important to look from different perspectives and test different variables each time, so that there are many things to consider about. That causes the other crucial point in testing which also created the problem: the number of runs we executed in each runtime. Some of the cases take around fifteen hours to complete which is a lot to do in daytime if you have one computer. That's why we came up with the solution that the people who have more than one computer would execute those tests and the others would test the rest. But still the members who did longer tests had some challenges such as setting alarms in the middle of the night to wake up and check the computer since fifteen hours sometimes ended at nighttime. So, this process taught us the importance of time management and task distribution in the project team.

The other challenge that we faced throughout the project was the logic behind some parts of the code. The data, which represent the counter values in a single run, obtained by running the source code of There's Always a Bigger Fish (Cook, 2022) has erroneous values in it. Counter values should be starting from "0", however, due to an error, some of the counter values are being recorded as "-1". In order to find the source of the error, we made several analyses and used different methods to execute the program. Finding the root cause of the issue and fixing it took some amount of time which was more than expected but held well with a great task distribution which was led by our team member Ali Parlakçı.

As mentioned earlier, starting from next week, the tasks above the pink one in Figure 1 will be done and we will start working on implementing a side channel attack that targets the interrupts sent to the CPU by the GPU. So that we will catch the schedule before the term ends and we will start the second term with a new time table.

4. TASKS TO BE COMPLETED UNTIL PROGRESS REPORT II

The Gantt Chart for the next semester can be found below, it is divided into two parts to have a better look at the writings on it. As mentioned in the Figure 2 for the remaining part of

the project, we will collect more information about the system we test on and introduce more independent variables such as computer architecture, operating system, user, etc. The goal is to enlarge the data set to increase the reliability of the results. The next step is to replace the spy process executed by CPU with a spy process executed by GPU (Graphical Processing Unit). It will allow us to compare CPU sniffing with GPU sniffing and decide which method is more successful at understanding user behavior, in other words, which unit is more vulnerable to such attacks.

After the request from our supervisor, with the beginning of the second term we are going to start working on writing an academic paper about this project. Steps of writing the paper are also mentioned in the time tables below. Since we have not created a strict calendar for the second semester yet, new tasks would be added as we proceed with the project.

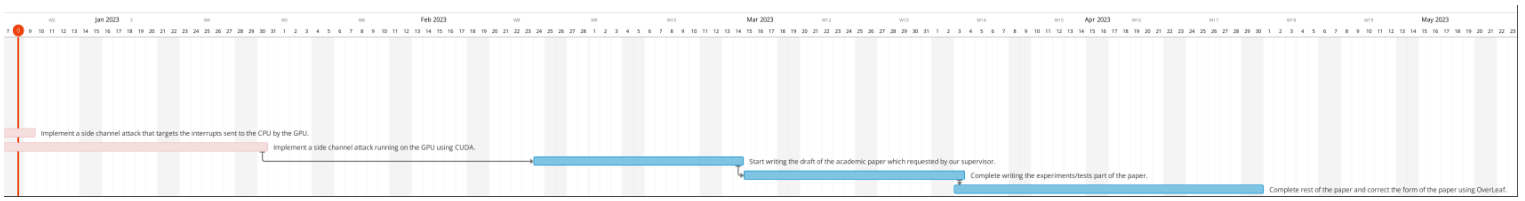

Figure 2: Gantt Chart for the second term

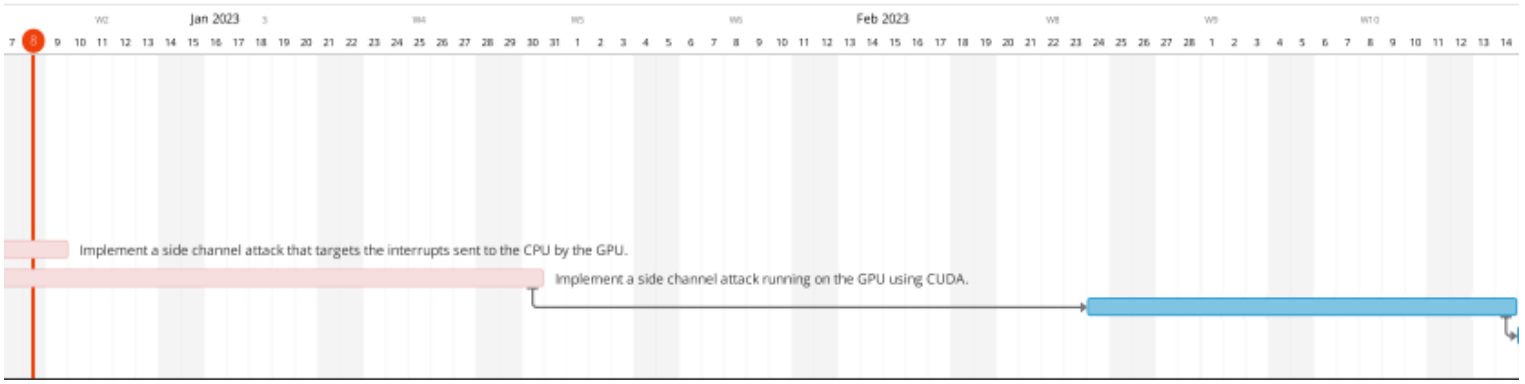

Figure 3: The first half of the Gantt Chart for the second term

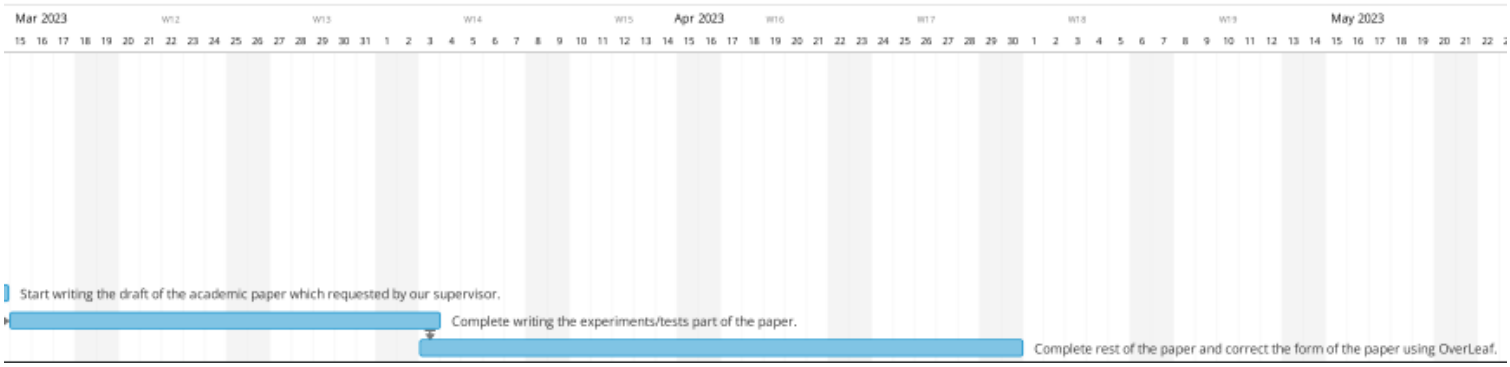

Figure 4: The second half of the Gantt Chart for the second term

5. REFERENCES

Cook, J., Drean, J., Behrens, J., & Yan, M. (2022). There's always a bigger fish: a clarifying analysis of a machine-learning-assisted side-channel attack. *Proceedings of the 49th Annual International Symposium on Computer Architecture*., 204-217. https://doi.org/10.1145/3470496.3527416

Cook, J. (2022). There's Always a Bigger Fish. *GitHub Repository*. Retrieved from https://github.com/jackcook/bigger-fish

Cook, J., Drean, J., Behrens, J., & Yan, M. (2022). There's Always a Bigger Fish Interactive Demo. Retrieved from https://jackcook.github.io/bigger-fish

Ma, H., Tian, J., Gao, D., & Jia, C. (2021). On the effectiveness of using graphics interrupt as a side channel for user behavior snooping. *IEEE Transactions on Dependable and Secure Computing*, 1-14.

Software quality standards – ISO 5055. CISQ. (2022, October 4). Retrieved November 13, 2022, from https://www.it-cisq.org/standards/code-quality-standards/