ENS 492 – Graduation Project (Implementation)

Progress Report II

Project Title: Reproducing and Analyzing Side Channel Attacks

Group Members:

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

Supervisor(s): Cemal Yılmaz

Date: 13.04.2023

1. PROJECT SUMMARY

The project focuses on side-channel attacks, which are security exploits intended to elicit data from the target system. To make it simpler to grasp, a side-channel attack is one in which an attacker gets someone's credit card PIN using the fingerprints on the POS device. The operation's general structure and fundamental design are to blame for the vulnerability, not a flaw in the process itself. In a similar line, attackers benefit from particular functional components and characteristics of computers. Despite having the appearance of multitasking and being able to finish multiple tasks at once, computers can only perform one activity at a time. However, they move between jobs so quickly that parallelism appears to be occurring. Interruptions are one of the factors that will make computers switch. The computer's operating system enters kernel mode when an interrupt occurs, suspending all other processes in order to handle the interrupt. Interruptions frequently result from an outside force, such as a just-received network packet. This task shift is referred to as context switching.

Computer interruptions were indirectly observed in previous research. They run a simple loop-counter in the browser and periodically check the value. They loaded separate websites in a different browser tab in the meantime. Because computers operate in a linear fashion, interruptions occur as the browser loads the website. Because of how the interruptions are handled, the counter in the other will be different during each interval. A study claims that when given counter data, a machine learning model can predict which websites are open in the browser using the increment values stored in the browser (Cook et al., 2022).

The project has several important goals and duties that, as we specified in the proposal, may be worth remembering. Because this project primarily focuses on a literature review about side-channel attacks, it was crucial to make sure that everyone involved had acquired a fundamental understanding of the current study before moving forward with it. 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) are the two papers that serve as our primary sources.

Every member of the team cloned the There's Always a Bigger Fish (Cook, 2022) GitHub repository that we studied, tested the source code, created visualizations from the data obtained with the help of Python data visualization libraries, tried to train the machine learning models given, and evaluated the accuracy of those models. Each project participant has been familiar with the There's Always a Bigger Fish source code in order to verify that it operates as intended on their physical or virtual computers. If it does not, they have noted the errors in the code and/or data.

Since we have been working on a software development and computer engineering project, the three most important constraints which are dependent on each other are time, scope, and cost (economics). These affect the effectiveness and quality of our final product and project during the semester. Our ultimate goal is to have the greatest possible outcome from the project by having the best possible working environment to make each team member feel motivated to work.

2. SCIENTIFIC/TECHNICAL DEVELOPMENTS

During the conceptual design phase, an extensive literature review was undertaken to identify the knowledge gaps in the area of GPU application fingerprinting using loop-counting attacks. It was observed that this domain has not been extensively explored, particularly in the context of media players and video codecs. The hypothesis formulated was that loop-counting attacks could be utilized to fingerprint GPU applications, specifically targeting media players and video codecs, as these are commonly used applications with privacy implications.

The primary goal of this project is to explore the effects of different browsers and operating systems on the success of loop-counting attacks to identify media players and video codecs. The objectives include investigating the feasibility of the proposed method across different media players, video codecs, operating systems, and browsers, evaluating the performance of the proposed method in terms of accuracy and reliability, and identifying potential countermeasures to mitigate the threat posed by loop-counting attacks.

In the preliminary design phase, the experiments were designed to assess the proposed method on a diverse range of media players, video codecs, operating systems, and browsers. The chosen components for the experiments included media players such as MPlayer v1.5, mpv v20230108, and VLC v3.0, video codecs like mp4, 3gp, flv, and mkv, operating systems such as Windows 10.0.19044, macOS 13, and Linux 5.15, and browsers like Chrome 108, Firefox 108, Edge 108, and Safari. To ensure the generalizability of the results, the experiments were conducted on various hardware configurations, including NVIDIA GeForce RTX 2080 Ti, NVIDIA GeForce GTX 1080, and AMD Radeon RX 5700 XT GPUs, Intel Core i7-9700K, Intel Core i5-9400F, and AMD Ryzen 5 3600X CPUs, and 16 GB and 32 GB RAM configurations.

During the design decision phase, the experiments utilized a single video encoded with the selected codecs. The attack code, implemented in JavaScript, was executed within a browser tab using the timer provided by the browser via the performance.now() function. Counter traces were collected during the initial 15 seconds of video playback, with 5-millisecond increments. Each valid combination of operating system, browser, media player, and codec was subjected to the experiments 100 times, resulting in a total of 14,400 experiments. Prior to the analysis, the collected data underwent preprocessing to remove any noise and outliers. The data was cleaned, normalized, and structured into a format suitable for machine learning algorithms.

In the detailed design phase, the analysis involved training and testing random forest classifiers for each experimental setup. Ten-fold cross-validation was employed to ensure the robustness of the models. The accuracy metric (in percentages) was utilized to evaluate the performance of the models. The findings revealed that the proposed method exhibits promising results in terms of accuracy and reliability. The experiments demonstrated the feasibility of GPU application fingerprinting using loop-counting attacks across various media players, video codecs, operating systems, and browsers. Furthermore, the project identified potential countermeasures that could mitigate the threat posed by loop-counting attacks, which will be further explored in subsequent phases of the project.

Following the detailed design phase, the next steps of the project will involve refining the fingerprinting method based on the insights gained from the experiments, enhancing the

robustness of the method. Furthermore, we plan running the attacker code on the GPU of the system and executing the same experiments again. Loop-counting attack leverages the task scheduling mechanisms of the architecture. GPUs approach task scheduling different than the CPUs. Running the attacker program on the GPU will enable us to explore the level of mitigation GPUs' task scheduling techniques provide.

3. ENCOUNTERED PROBLEMS

During the course of the project, our team encountered several challenges and adjustments that affected our project plan, goals, and timeline. In this section, we will address these concerns and outline the corrective measures we have taken to ensure successful completion of our project.

3.a. Changes in the Project Plan and Goals

While our original goals remain relevant, we have had to adapt our approach due to time constraints, scheduling conflicts, and different operating systems among team members. Our initial plan also included implementing a loop counting side channel attack on the NVIDIA GPUs using CUDA programming language, which has not progressed as quickly as we had hoped. This is primarily because our team has been focused on completing other tasks and writing the research paper, which has consumed a significant portion of our available time.

3.b. Progress According to the Project Timetable

Despite making considerable progress in other aspects of the project, we are currently behind schedule in developing the GPU-based loop counting side channel attack. The primary reasons for this delay include time conflicts between team members' schedules, as well as the intensive focus on completing the research paper.

To address this setback and achieve our original goals, we plan to take the following corrective measures:

- Prioritize researching how to implement a GPU-based side-channel attack. Currently reading the paper:
    - DRAWNAPART: A Device Identification Technique based on Remote GPU Fingerprinting)
- Establish clear deadlines and milestones for the remaining work to maintain a sense of urgency and accountability within the team.
- Collaborate more efficiently by using online project management tools and scheduling regular progress meetings to ensure that all team members are on the same page and working towards a common goal.

3.c. Effects of the Changes in Project Implementation

The changes in our project implementation have had both positive and negative effects. On the one hand, our team has successfully completed a research paper that demonstrates the effectiveness of loop counting attacks for GPU application fingerprinting. This achievement is a significant contribution to our overall project goals and will serve as a strong foundation for our final presentation.

On the other hand, the delay in developing the GPU-based loop counting side channel attack has put additional pressure on our team to complete this critical component of the project. By implementing the aforementioned corrective measures, we are confident that we can overcome this challenge and achieve our original goals within the remaining project timeline.

4. TASKS TO BE COMPLETED BEFORE FINAL REPORT

Week 1-2:

Task 1: Refine and improve the fingerprinting method

Task 2 (part 1): Implement loop counting side-channel attack on NVIDIA GPUs using CUDA programming language (begin implementation)

Week 3-4:

Task 2 (part 2): Implement loop counting side-channel attack on NVIDIA GPUs using CUDA programming language (complete implementation)

Task 3: Conduct experiments using the improved fingerprinting method and GPU-based attack

Task 4 (part 1): Test and validate the performance of the GPU-based side-channel attack and improved fingerprinting method (begin testing)

Week 5:

Task 4 (part 2): Test and validate the performance of the GPU-based side-channel attack and improved fingerprinting method (complete testing)

Task 5: Evaluate the effectiveness of potential countermeasures to mitigate the threat posed by loop-counting attacks

Week 6:

Task 6: Implement the refined fingerprinting method and GPU-based attack in a real-world scenario

Task 7 (part 1): Document the progress, results, and insights gained throughout the project (ongoing documentation)

Week 7:

Task 7 (part 2): Document the progress, results, and insights gained throughout the project (complete documentation)

Task 8: Prepare the final report, including a comprehensive analysis of the project's outcomes, challenges, and recommendations for future work

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/