University of British Columbia

CPSC 538M

Assignment

Department of Computer Science
Instructor: Aastha Mehta
TA: Amir Sabzi

Due Date:
08 Oct 2021

Submission instructions

Clone the git repo: `https://github.com/aasthakm/cpsc538m`. The repo contains a directory with library code that will be useful for this assignment. For problems 2-4, submit your code in respective solution directories. Create a pdf report answering the questions listed in the problems below and add it to the report directory. Submit a copy of the repo with your solutions and report in `.tgz` format to the instructor.

Problem 1

(2 Points)

Profile the cache hit and miss latencies on your board using flush+reload style operations. Submit a plot of the observed latency distributions on your board.

Problem 2

(2 Points)

Create two different processes and use flush+reload as a cache covert channel for communication between them. You are free to decide how you set up the shared memory between the processes and start the transmission initially. For example, you may even write to the shared memory. However, once the transmission is established, there should be no more direct communication between the two processes. Write a brief summary of your approach, explaining the communication mechanism and the details of the flush+reload covert channel.

Problem 3

(2 Points)

Implement one of the artifical applications from Section 4 of the Cache Template Attacks paper and reproduce the cache template matrix for the application. You may use the tool in the `cache_template_attacks` repo: `https://github.com/aasthakm/cpsc538m`. Write a brief summary of your approach, explaining the implementation and the generated cache template matrix.

Problem 4

(4 Points)

Use the `cache_template_attacks` tool to leak keystroke timings in a text editor of your choice. One suitable editor is `gedit`, although you might need to attack `libgedit.so`. It suffices to find some address in the editor where the observed cache timings are correlated to keystrokes. We do not expect you to be able to distinguish between different keystrokes. Write a brief summary of your approach, explaining the method for searching the addresses and the observations generated.

(Hint: start by finding the start offset of the .text segment in your binary.)