Side-Channel Security 2021
Exercise 1

In this exercise, you will get hands-on experience with a small selection of microarchitectural attacks.

1. This exercise is worth **up to 15 points**.
2. Only task 1 is mandatory: if you do not submit it, you will not get a grade.
3. Put all source code, tools, intermediate files, and result files for this submission into folder `ex1` in your repository.
4. Tag your submission with the tag `ex1-1` before the first deadline, and `ex1-2` before the second.
5. For all tasks, you can find more information at `https://www.iaik.tugraz.at/course/side-channel-security-705048-sommersemester-2021/`

1 Introduction to Cache Attacks – 1 Point

Mandatory
Deadline: Wednesday, March 17 2021, 8:00am

Produce a Flush+Reload cache hit/miss histogram on at least one machine per team member, and **submit a plot image** into your repository. Use your histogram to decide on a *good* threshold for each machine. Both team members must be able to reproduce their histogram in the exercise interview and explain their choice of threshold.
You may use the histogram demo in the upstream repository, **no programming is necessary** for this task!

2 Cache Covert Channel – 6 Points

Optional
Deadline: Monday, May 3 2021, 8:00am

Implement a *fast* (see Table 1) and reliable cross-core covert channel based on a cache attack (Flush+Reload, Prime+Probe or Flush+Flush). Transmit a file large enough that the transmission takes 60 seconds, with random (`/dev/rand`) or meaningful non-uniform content (`png`, `jpeg`, `mp3`, etc.). Compute the true channel capacity T as

$$T = C \cdot (1 + ((1-p) \cdot \log_2(1-p) + p \cdot \log_2(p))),$$

where C is the *raw capacity* and p is the *bit error ratio*.
You do not need to perform any error correction.

Capacity T (kB/s)	1.5	7.5	13	20	27	35	45	**55**	67	80	110	150	400
Points	1.5	2	2.5	3	3.5	4	4.5	**5**	5.5	6	6.5	7	7.5

Table 1: True channel capacity to points.

2.1 Bonus Points

- +1 – Your group has the fastest channel[1].
- +1 – your channel uses Prime+Probe or Flush+Flush.
- +1 – your Prime+Probe channel needs no access to physical addresses.
- +1 – your Prime+Probe channel works across virtual machines.

A scaling factor of 2/3 will be applied to the capacity requirement for Prime+Probe channels.

3 Cache Template Attack – 5 Points

Optional
Deadline: Monday, May 3 2021, 8:00am

For this task, you will re-implement the tools shown in the lecture, you can find the binaries in your repository for reference. Implement **both** phases of the attack, profiling and exploitation. Your implementation does not need to behave exactly like the reference (which you can find in your repository) but the profiling should stay universally applicable. Demonstrate your implementation on a program of your choice (e.g. editors, IDEs etc.). Register for this task by sending an email naming your target program to `lukas.giner@iaik.tugraz.at` (please include *SCS* in the subject line). Every group has to attack a different program, the exercise page `https://www.iaik.tugraz.at/teaching/materials/scs/exercises/ex1/` shows a list of the programs that were already chosen.

Fully functional profiling and exploitation phases are worth 2.5 points each.

3.1 Bonus Points

- +1 – extraordinary accuracy, you have virtually no false positives or false negatives.
- +1 – you can distinguish groups of keys from each other.
- +1 – you can identify at least 5 individual keys (includes point for groups).

4 Intel, AMD or GME: speculation is dangerous! – 4 Points

Optional
Deadline: Monday, May 3 2021, 8:00am

We have found a library that we think is susceptible to a Spectre-PHT attack[2]. We have provided it to you in your repository. You can access anything defined in the library's header to extract the secret string, defined in the library's source. You may not change the library or read the secret directly from memory. Most modern processors are affected by Spectre, but we recommend you use Intel or AMD. During the exercise interview, you will be given a new shared object file with a different secret, so make sure to test if your solution works for any string (though you can assume ASCII 63-95). Points will be awarded according to Table 2/Table 3 (Intel/AMD), though for this task your hardware and your explanation during the exercise interviews will also have an influence.

time to secret (s)	<300	<220	<160	<100	<70	<50	**<35**	<10	<1
Points	1	1.5	2	2.5	3	3.5	**4**	4.5	5

Table 2: Time to recovery of the (correct) secret to points on Intel.

4.1 Bonus Points

- +3 – you can mistrain the target branch from a different thread or process[3], without calling the library function in that thread or process.

time to secret (s)	<300	<220	<160	<100	<70	<50	**<35**	<15	<5
Points	1	1.5	2	2.5	3	3.5	**4**	4.5	5

Table 3: Time to recovery of the (correct) secret to points on AMD.

Notes

[1]Send an email to `lukas.giner@iaik.tugraz.at` (or a discord message) with *raw capacity*, *bit error ratio*, and any extra details you want included to update the speed record page `https://www.iaik.tugraz.at/teaching/materials/scs/exercises/ex1/`. Final rankings are based on the exercise interviews.

[2]`http://spectreattack.com/`

[3]On newer hardware, you may need to deactivate STIBP, ask in Discord if you're unsure. `https://software.intel.com/security-software-guidance/deep-dives/deep-dive-single-thread-indirect-branch-predictors`