This is a take home exam. Please complete all the questions in a separate WORD or PDF document to be submitted via Webcourses. Make sure that your name is on each page of the document and that each answer has the appropriate question number (i.e. 1.a, 1.b, 2.a, 2.b, etc) to the left of the answer. Documentation matters. Do not hesitate to provide additional information in the event of any questions.

This take home exam is *NOT* a group project. There will be a separate affadavit as part of your exam submission that specifically confirms your submission is your work and only your work.

Question:	1	2	3	4	5	Total
Points:	25	25	25	25	0	100
Bonus Points:	0	0	0	0	2	2
Score:						

1. Create a *Virtual Guest* on your multi-core machine. Use the *Virtual Host* software of your choice. All the questions shown below have been tested on a *Debian 10 (buster) with 13.1 GB memory running Linux Mint 20-xfce 64 bit guest on Virtual Machine Manager 2.0.0 (https://virt-manager.org) and Mac OSX (10.11.6) with 8 GB of memory running Linux Mint 20-xfce 64 bit guest on VirtualBox (6.1.6)(https://www.virtualbox.org/) Both guests were configured with 1 CPU, 1 GB of memory, and 20 GB of virtual disk on SSD drive on each host.*
   The *Guest* will be the *Rowhammer Simulation Platform*. Software will be installed on the *Guest* to accomplish the simulation.

15 (a) Install a *Guest* running Linux Mint 20-xfce (64 bit) configured with 1 GB of memory on a 20 GB virtual disk. Provide a screen shot of an open terminal window on the desktop running the command **uname -r** to document and confirm the installed kernel version. Useful *Guest* config data:

   1. 1 GB memory
   2. 20 GB Virtual Hard Dive

10 (b) Install the required software[1] as defined in the Hammertime GIT Project. Specifically confirm the following:

   1. POSIX compatible OS (Linux recommended)
   2. Python >= 3.2 — used by tools
   3. RAMSES (included as a `git submodule`; make sure to clone recursively or manually initialize and update before building)

[1] Linux Mint 20-xfce (64 bit) was specifically chosen as a *Guest* because it met the software requirements enumerated on the Hammertime GIT Project page. In the event you choose another Linux Distro take **great** care.

- Clone or download the RAMSES memory address translation library from the Github ramses project.
- *The hammertime directory structure contains an empty folder named **ramses**. The goal is to provide the hammertime repository with access to the **ramses** code library.* Move the **`cloned ramses`** directory to the **`hammertime`** `directory also named` **`ramses`**.

Once this has been confirmed the following software will need to be installed by executing the following commands:

- **sudo apt install gcc g++**
- **sudo apt install build-essential git**
- **sudo apt install make**
- **sudo apt install make-guile**

NB
Make sure to do the installs shown above in the order shown. (There are some sequence issues otherwise.)

sudo make
Now that all the tools & libraries have been installed, it is appropriate to install the *Hammertime GIT Project* per the instructions on the *Hammertime GIT Project* page. Specifically make sure to do the following set up:

1. **make** (in the *hammertime* home directory)
   (a) In the event that the **make** is unsucessful, use the **make clean** command.

Once the **make** is successful, proceed to the next step, memory configuration analysis.

2. The *hammertime* toolset has three major parts:

- Memory configuration
- *Rowhammer* vulnerability testing
- *Rowhammer* simulations

10 (a) There are several python tools & scripts that perform *rowhammer* configuration and testing. Configure the memory using the following commands:

1. **sudo ramses/tools/msys_detect.py** which asks for the memory controller configuration. (Use the **tee** command to log the outputs to a file named *msysConfig.log* for submission as part of the assignment.) The questions are as follows:
   - Memory controller selection for *ddr3*, *ddr4*, *intel:sandy*, and *intel:haswell*. Start with the *ddr3* selection (**0**).
   - Enable address pin mirroring for second rank, either a yes or no question, with **N** being the preferred default.
   - Select additional on-DIMM remap choices of *none*, *rasxor:bit=3:mask=6*, and *custom rasxor*. *None* is the preferred default. (**0**)

- The last memory controller configuration question is to set the output file name, which has a default name of **./mem.msys**.

Create two memory configuration files named **m0N0.msys** (configured as *ddr3*, *no pin mirroring*, & *no additional on-DIMM remap*) and **m0P0.msys** (configured as *ddr3*, *PIN mirroring*, & *no additional on-DIMM remap*).

(All commands are executed in the *hammertime* home directory, unless *explicitly* stated otherwise.)

7 ½ (b) Test for the *rowhammer* vulnerability using the following commands:

- **sudo profile/profile 256m m0N0.msys** (tee the output to **profile256m0N0.log**)
- **sudo profile/profile 256m m0P0.msys** (tee the output to **profile256m0P0.log**)
- These log files will be submitted for grading.

These profiles may take a while, depending upon CPU speed, and more.

7 ½ (c) Simulate three well known *rowhammer* attacks using the following scripts. Note that the logs are captured as part of the deliverables to be submitted for grading.

- Dedup Est Machina (S&P'16)
  Run the command:**sudo py/dem_exploit.py profile256m0N0.log m0N0.msys |tee dem-m0N0.log**
  Run the command:**sudo py/dem_exploit.py profile256m0P0.log m0P0.msys |tee dem-m0P0.log**
- Flip Fen Shui (Black Hat Europe '16)
  Run the command:**sudo py/ffs_exploit.py profile256m0N0.log m0N0.msys |tee ffs-m0N0.log**
  Run the command:**sudo py/ffs_exploit.py profile256m0P0.log m0P0.msys |tee ffs-m0P0.log**
- Exploits targeting parts of an x86(_64) page table entry (PTE)
  Run the command:**sudo py/x86pte_exploits.py profile256m0N0.log m0N0.msys |tee x86-m0N0.log**
  Run the command:**sudo py/x86pte_exploits.py profile256m0P0.log m0P0.msys |tee x86-m0P0.log**

Remember to submit these logfiles as part of your exam.

3. Given the alledgedly ubiquituous nature of the *rowhammer* defect in DDR3 memory and the hardware nature of the problem, describe the *software mitigation* strategies that minimize the likelihood of a successful *rowhammer* attack. The deliverable is outlined as follows:

10 (a) Discuss the impact of the mitigation strategies on software design and testing.

7 ½ (b) Discuss the impact of outside software selection and potential vulnerabilities.

7 ½ (c) Discuss potential validation strategies to prove meaningful *rowhammer* defenses.

Make sure to address each topic completely, at the minimum, with two substantive paragraphs. (Some answers may require more than two paragraphs.)

4. Consider is a *rowhammer* attack possible via *Javascript*?

10 (a) If so, describe the attack vectors and the potential impact of those vectors or threats of a *rowhammer* attack in a browser. If it is not possible to execute a successful *rowhammer* attack in a browser explain why not and if there are any anticipated defenses against it. The deliverables are shown below. Make sure to address each topic completely, at the minimum, with two substantive paragraphs. (Some answers may require more than two paragraphs.)
Be specific.

5 (b) What defenses or mitigation strategies are possible to defend against such vectors?
Be specific.

5 (c) Identify the top 3 *Javascript* engines and their *rowhammer* status.
Be specific.

5 (d) Are there any *Javascript* engines that have meaningful *rowhammer* defenses built-in, configured, or anticipated?
Be specific.

2 (bonus) 5. Configuring the *memory controller selections* has the following options: *ddr3*, *ddr4*, *intel:sandy*, and *intel:haswell*. Why would *intel:sandy* or *intel:haswell* make a difference? A short answer or explanation is acceptable.

Document Requirements

- Make sure your submitted document answers each question and those answers are identified by the question number and subcategory, i.e. 1.a, 1.b, 2.a, 2.b, etc.
- It is *required* that any proposed mitigation strategies or other ideas have meaningful and clear basis of fact substantiated by citations. These citations, in either *APA* or *IEEE* format, can be of either websites or any of the readings provided in this course or discovered by you, *as long as they are primary sources*.
- Make sure to include the following at the end your document file: *Academic Integrity* statement (substitute your name and NID) - *"I [name] ([NID]) affirm that this document is entirely my own work and that I have neither developed my results and product with any another person, nor copied any results from any other person, nor permitted my results to be copied or otherwise used by any other person, nor have I copied, modified, or otherwise used results created by others. I acknowledge that any violation of the above terms will be treated as academic dishonesty."*