LeftoverLocals

Listening to LLM responses through leaked GPU local memory

Tyler Sorensen
Researcher@Trail of Bits
Assistant Professor@UC Santa Cruz

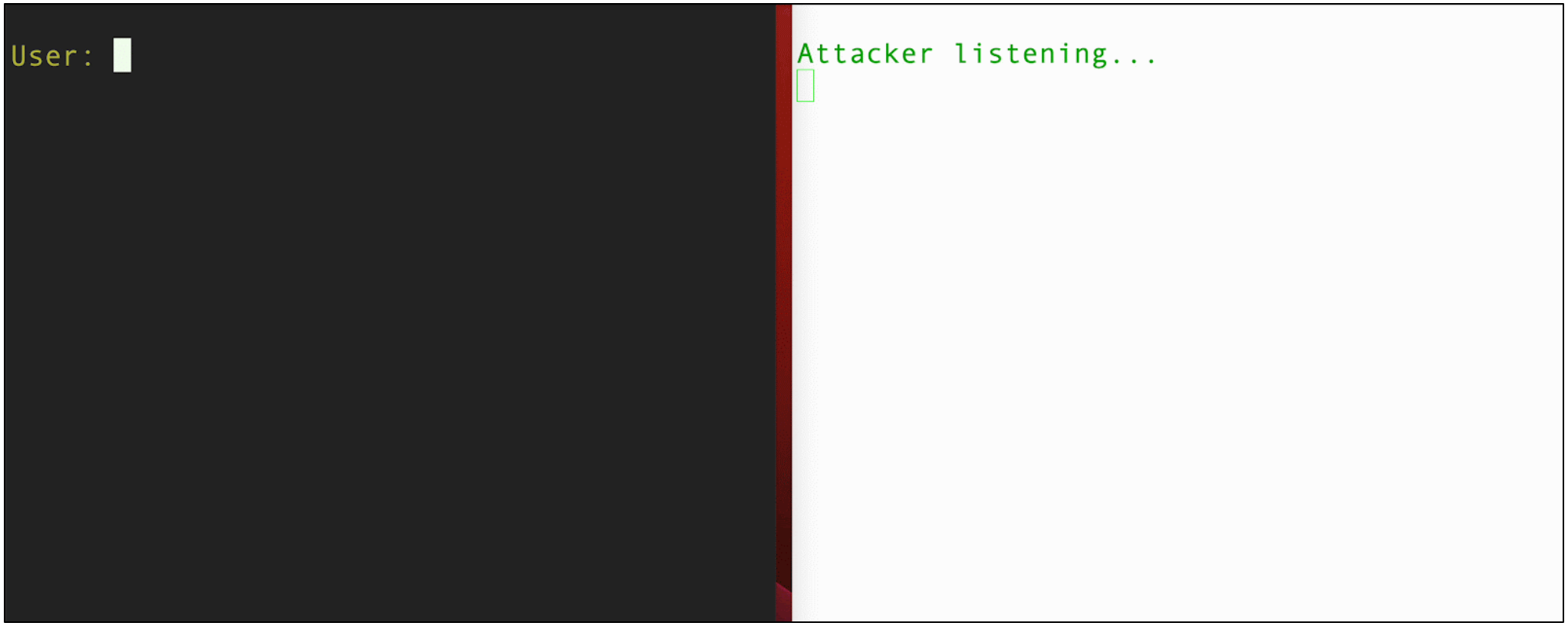
UNIVERSITY OF CALIFORNIA
SANTA CRUZ

Bsides SF
2024

TRAIL OF BITS

LeftoverLocals Demo

LeftoverLocals Demo backup



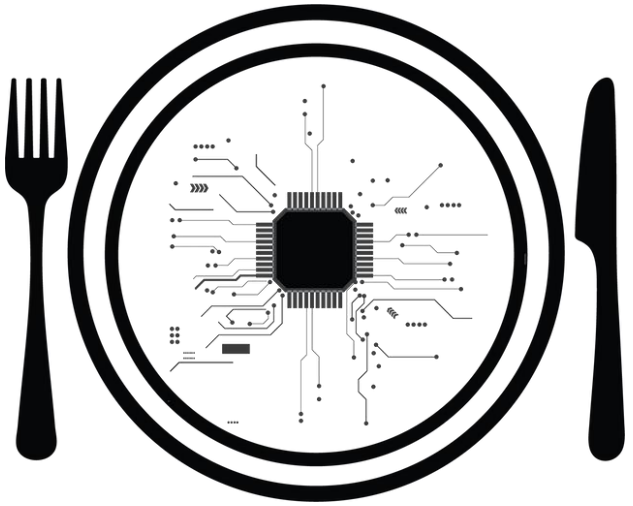
LeftoverLocals

Discovered in Summer of 2023 to impact GPUs from AMD, Apple, Qualcomm, and Imagination

Reported widely, e.g., by WIRED Magizine and Forbes

This talk will overview the technical details of the vulnerability

By
Tyler Sorensen
Heidy Khlaaf



WIRED

A Flaw in Millions of Apple, AMD, and Qualcomm GPUs Could Expose AI Data

Tyler Sorensen

- **Faculty** at UC Santa Cruz since summer 2020
  - Please ask questions during the talk!
- **Security researcher** in AI/ML at Trail of Bits since summer 2023
- Invited member of the **Khronos Group** since 2019
- Previously
  - Post doc at Princeton
  - PhD Student at Imperial College London
  - BS/MS at University of Utah

https://users.soe.ucsc.edu/~tsorensen/

My group at UCSC: Heterogeneous Programming Lab

- Working on GPU/heterogeneous programming models
  - Memory models
  - Semantics
  - Performance modeling
  - Compilers
  - Security
- 5 PhD students
  - Reese Levine, Yanwen Xu, Devon McKee, Jessica Dagostini, Rithik Sharma
- many MS and undergrads

AI/ML Security Researcher

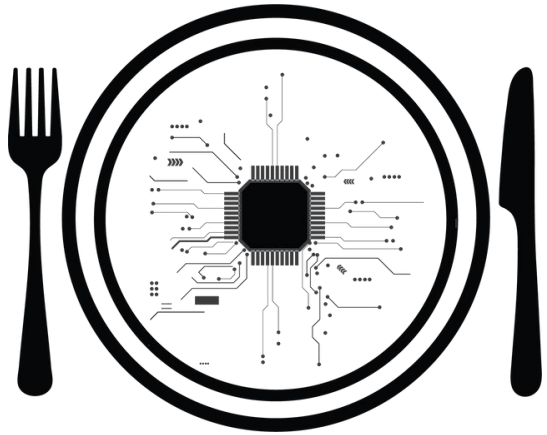
- Supply chain view of AI/ML security
  - Data formats
  - Adversarial models
  - Deployment
  - Computation stacks
    - GPUs

- Some collaborators:
  - Adelin Travers
  - Suha Hussain

LeftoverLocals

Two users co-resident on a machine with a GPU:

- Victim
- Attacker



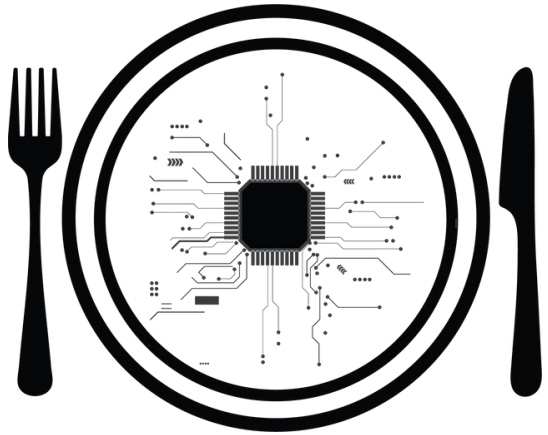
The victim is running an LLM using an open-source model, e.g., from hugging face, accelerated on the GPU

The attacker can “listen in”, and see the responses from the victims LLM on AMD, Apple, Qualcomm and Imagination GPUs

LeftoverLocals

To understand how this works, we need small ingredients from:

- The GPU architecture and execution model
- How DNN operations are computed on GPUs
- LLM architecture and models



https://github.com/trailofbits/LeftoverLocalsRelease

LeftoverLocals

Can we really cover all this in 1 hour?

Absolutely not, but we will look at a simplified model, hopefully enough to understand the issue!

GPU architecture

Core - execute a thread of computation

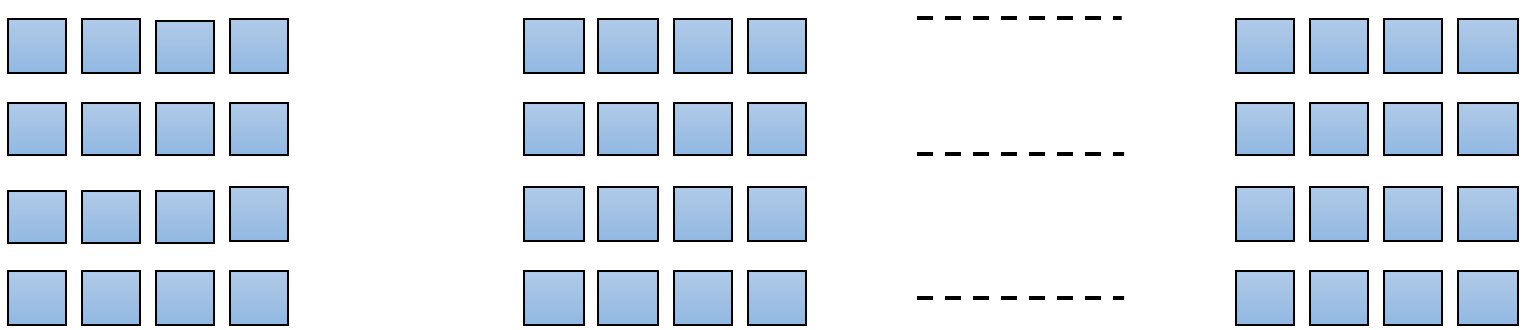
Main Memory

Main memory: VRAM for discrete GPUs. DRAM for integrated GPUs

GPU architecture

Core - execute a thread of computation

GPUs have lots of them! (almost 7K for the Nvidia A100)

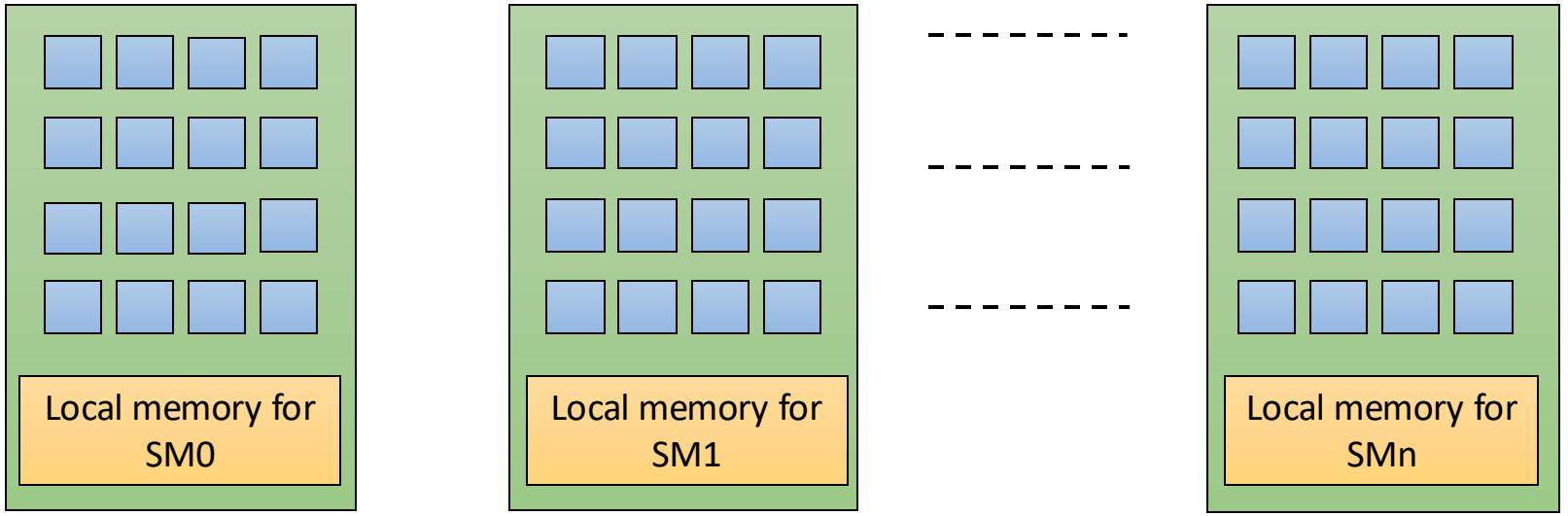


Main Memory

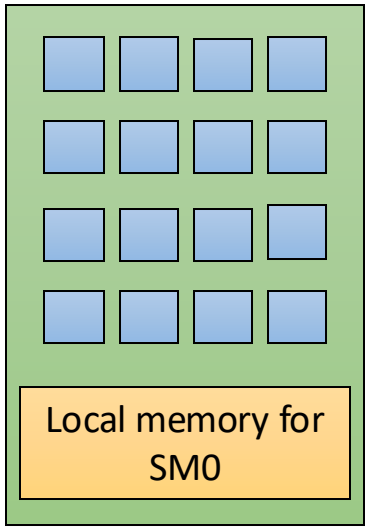
Main memory: VRAM for discrete GPUs. DRAM for integrated GPUs

GPU architecture

Cores are partitioned into ***streaming multiprocessors*** (SMs). Each SM has a software managed cache that is only shared between the cores on an SM. Called ***Shared*** or ***Local memory***.



GPU architecture



Local memory: Also known as a software managed cache. Important properties:

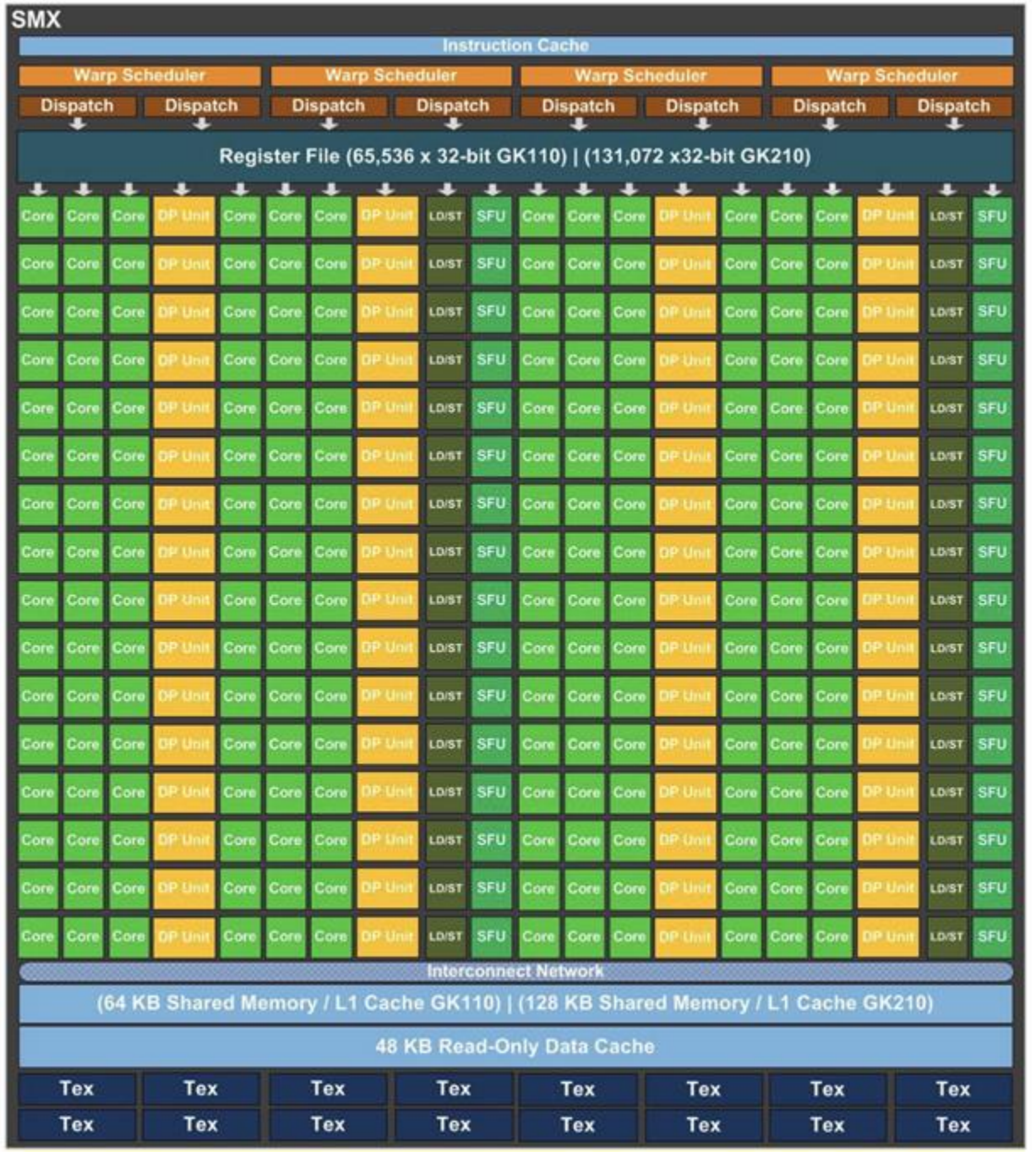
- Not specified to be persistent. Anything you want to save, you need to save it back to main memory!
- Limited in size (~65K per SM on many GPUs)
- Fast! GPU main memory is long latency. You don’t want to access it very often! Often taught in GPU programming 101

Much more…

More memory regions:

- Texture memory (used for graphics)
- Read-only memory
- Huge register files
- Shared instruction caches

From the white paper for Nvidia Kepler architecture



GPU programming model

What code does does a GPU execute?

GPU kernel - entry point for GPU execution. An application typically executes many GPU kernels. Each kernel typically does not execute for very long

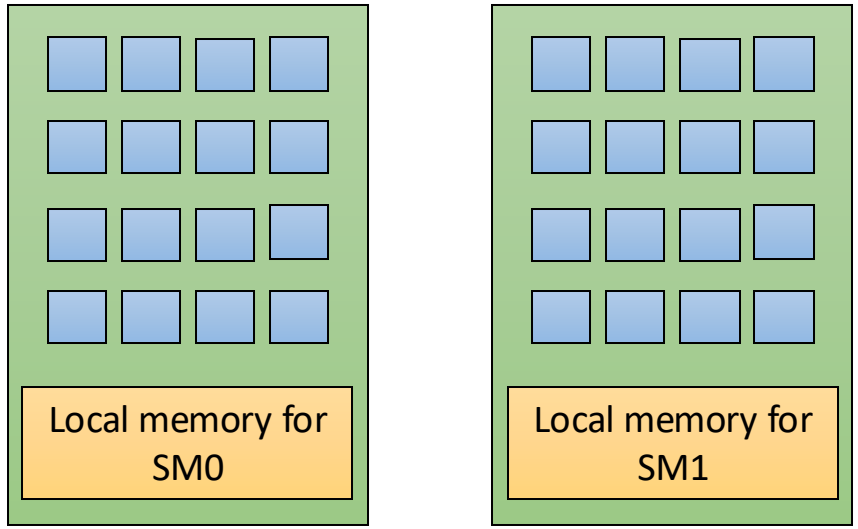
Examples: matrix multiplication, convolution, etc.

CPU staging - the CPU code (i.e. the *host*) copies memory to GPU main memory and *launches* the GPU kernel, specifying how many threads, and the size of the workgroups

GPU multiprocessing

process 0	process 1
`Kernel_00`	`Kernel_10`
`Kernel_01`	`Kernel_11`
`Kernel_02`	`Kernel_12`
…	…

Example:
GPU with 2 SMs



GPU multiprocessing

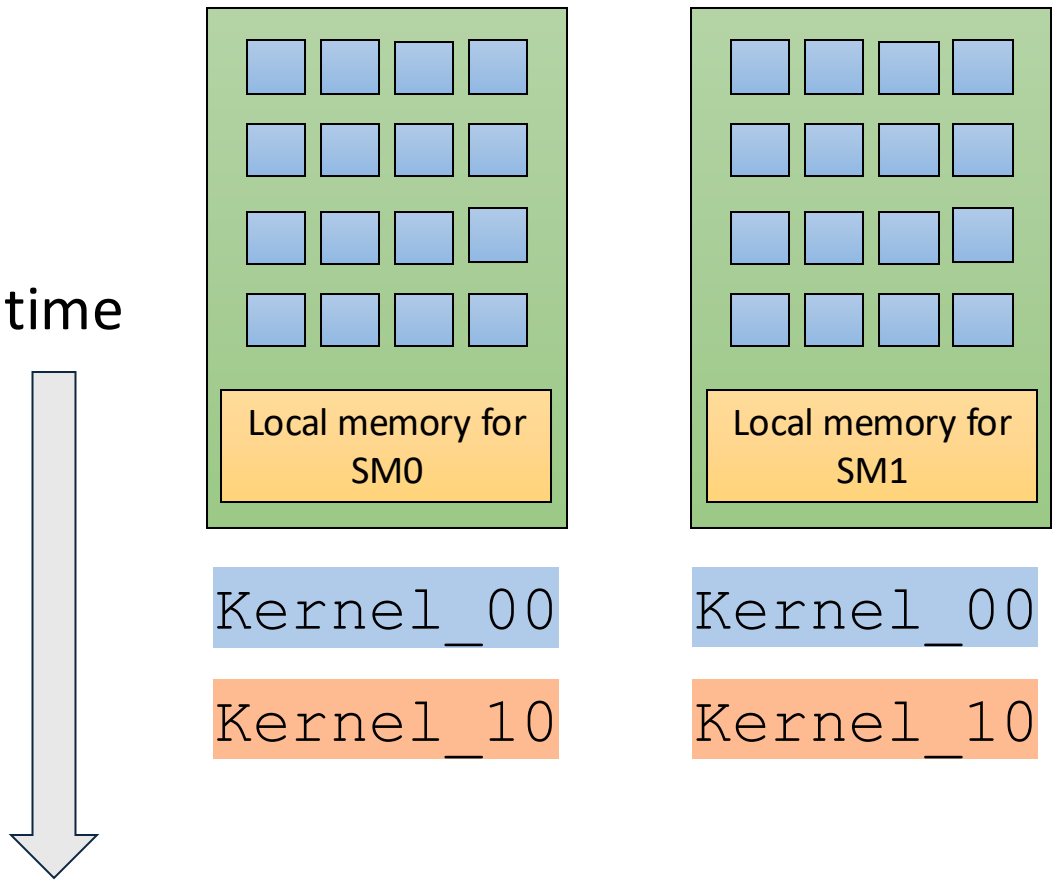
process 0

```
Kernel_00
Kernel_01
Kernel_02
…
```

process 1

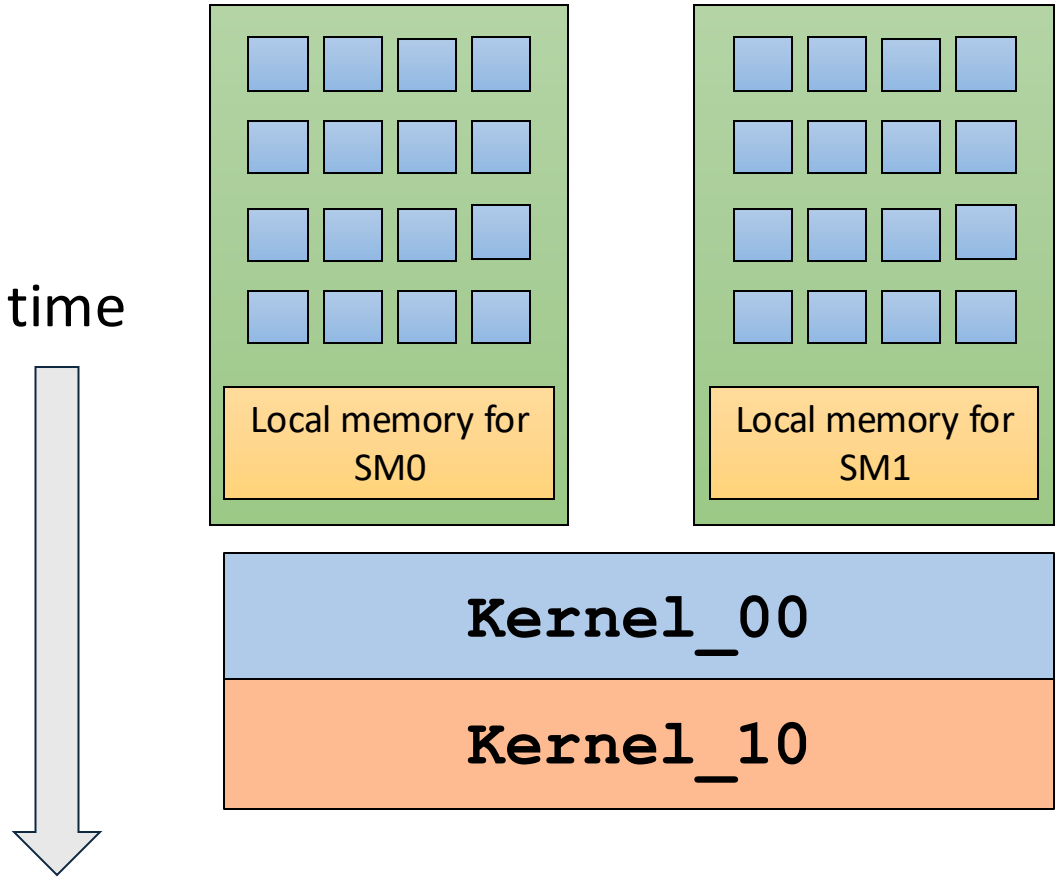
```
Kernel_10
Kernel_11
Kernel_12
…
```

Processes get the entire GPU and they can switch at kernel boundaries



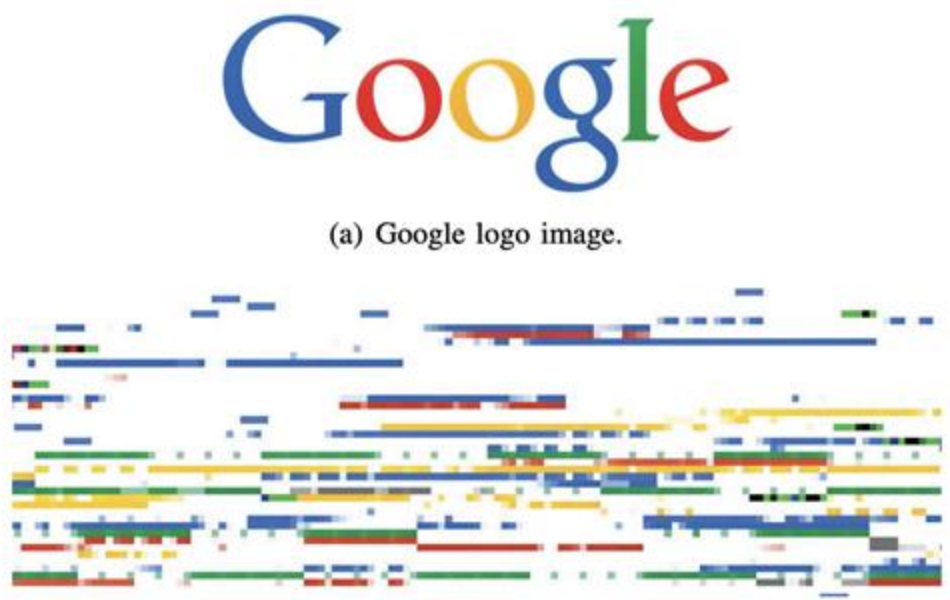
Let’s think about potential vulnerabilities

Memory leak - Can `Kernel_10` read leftover state from `Kernel_00`?



Let’s think about potential vulnerabilities

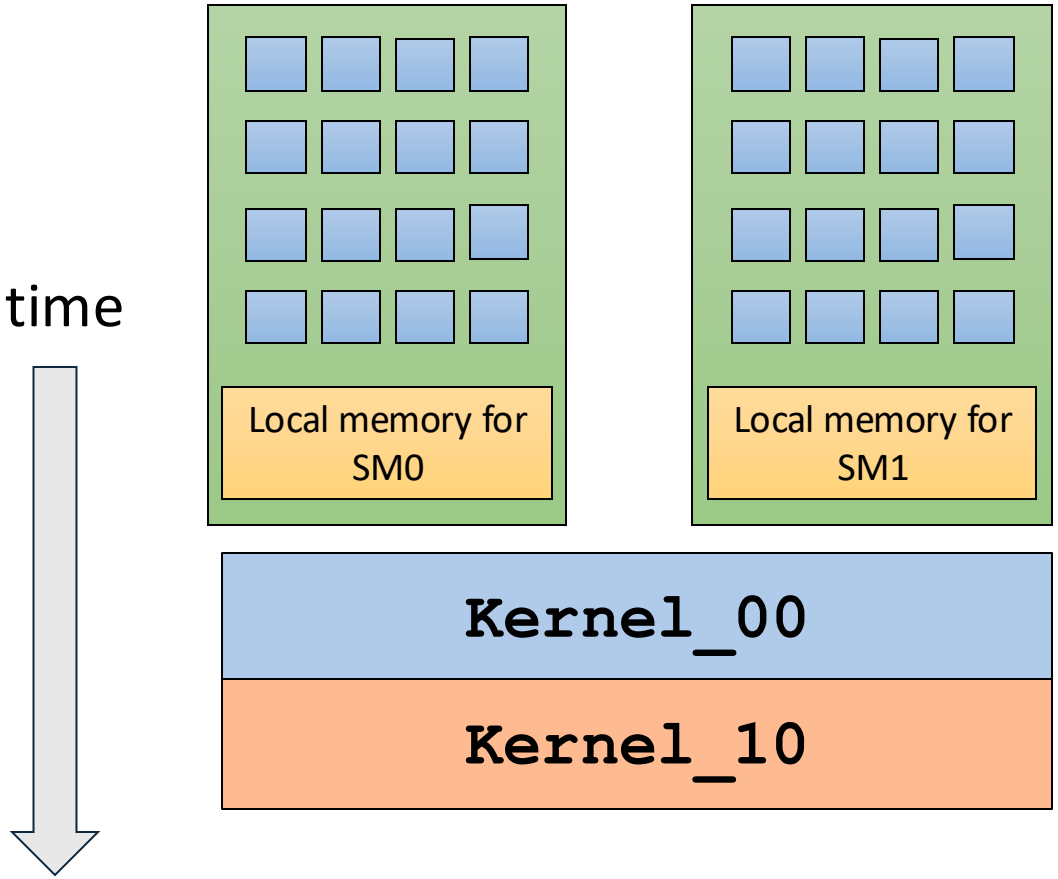
Memory leak - Can `Kernel_10` read leftover state from `Kernel_00`?



(a) Google logo image.

(b) Partial dump of Google webpage textures.

Lee et al. “Stealing Webpages Rendered on Your Browser by Exploiting GPU Vulnerabilities” 2014



But they probably shouldn’t

https://registry.khronos.org/vulkan/specs/1.3-extensions/html/vkspec.html#fundamentals-validusage

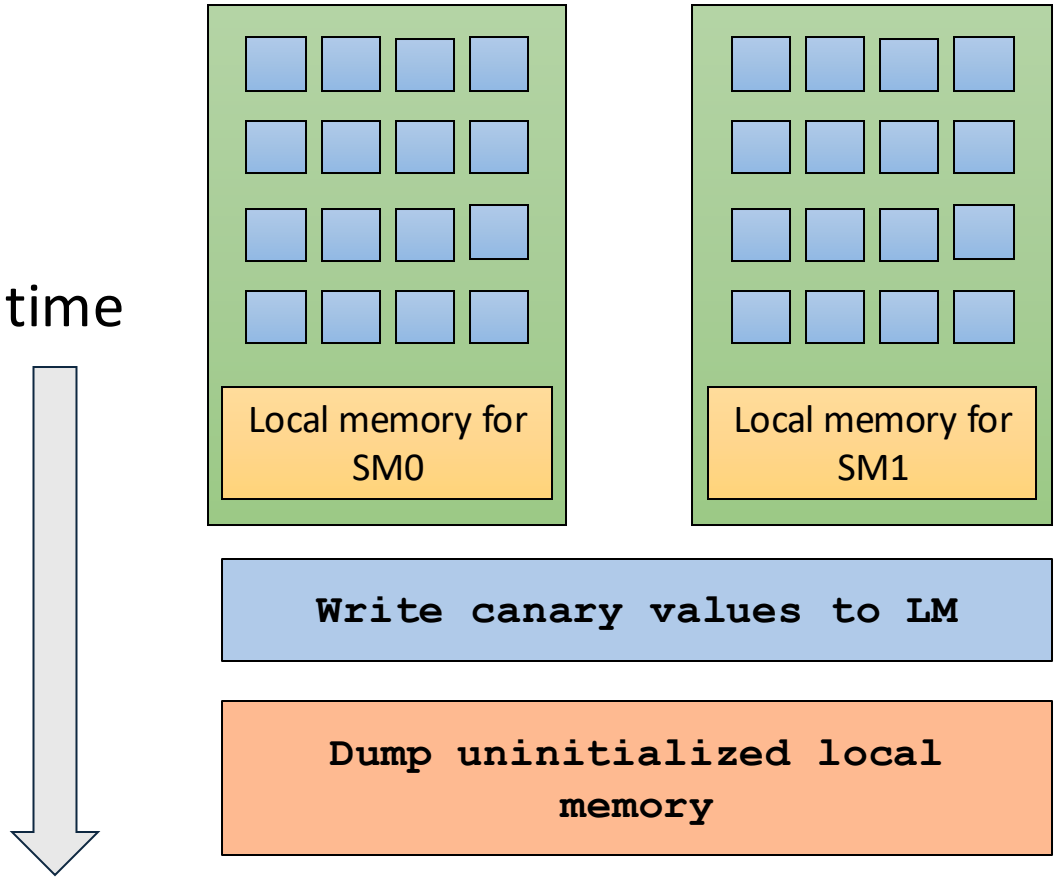
*In particular, any guarantees made by an operating system about whether memory from one process **can** be visible to another process or not **must** not be violated by a Vulkan implementation for **any memory allocation**.*

We investigated local memory:

Memory leak - Can process 1 read leftover local memory from process 0?

process 0	process 1
`Writer`	`Reader`
`Writer`	`Reader`
`Writer`	`Reader`
…	…

Can the reader ever read values from the writer?

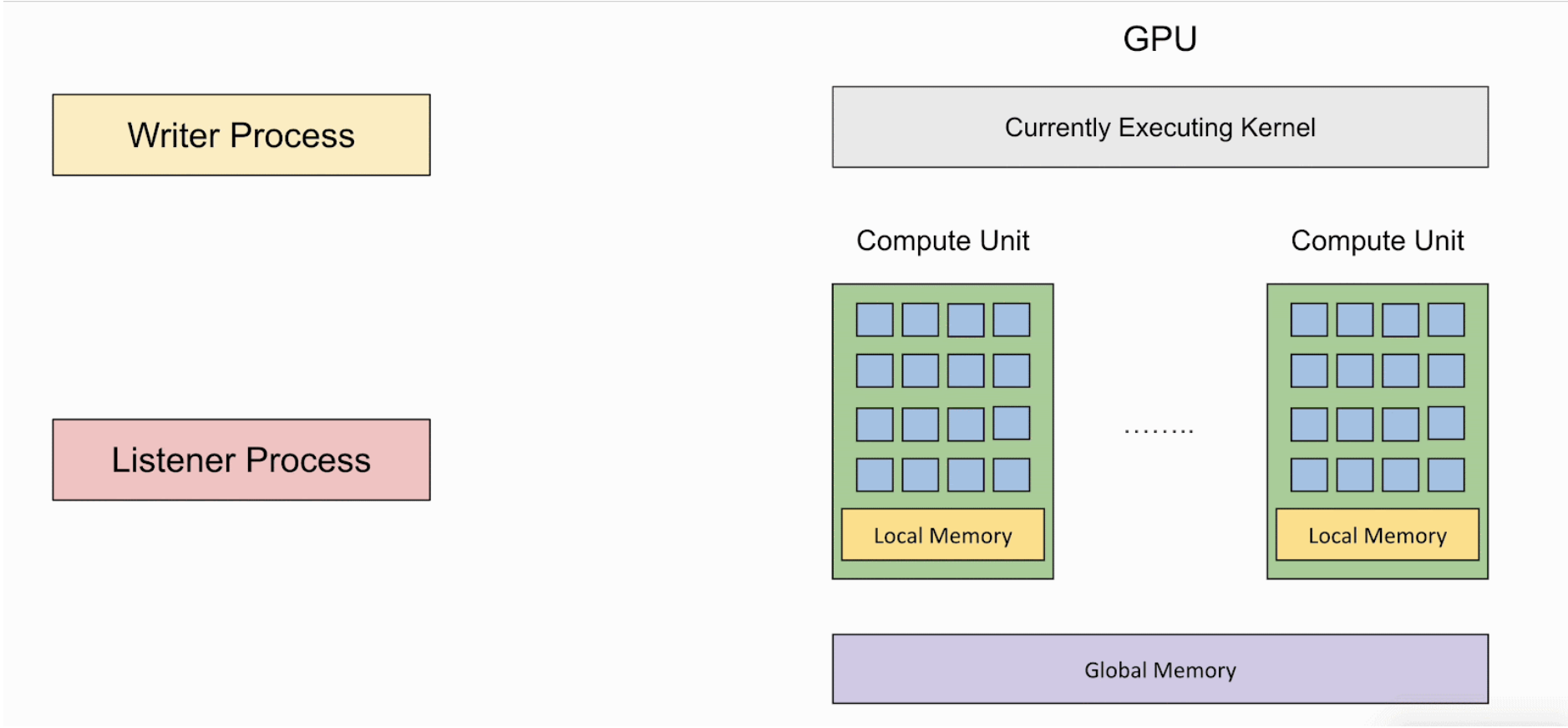


Testing for LeftoverLocals

Performed a large testing campaign.

Designed an:

- OpenCL command line tester
- Vulkan command line tester
- Android app using Vulkan
- Apple App (desktop and mobile)


GPU
Writer Process
Currently Executing Kernel
Compute Unit
Compute Unit
Listener Process
.....
Local Memory
Local Memory
Global Memory

Testing for LeftoverLocals

Observed violations:

Device (GPU)	GPU Framework	OS/Driver/Build system
Apple iPhone 12 Pro (A14)	Metal	iOS 16.6, Xcode 14.3.1 (14E300c)
Apple iPad Air 3rd G (A12)	Metal	iOS 16.5.1, Xcode 14.3.1 (14E300c)
Apple MacBook Air (M2)	Metal	MacOS 13.4.1, Xcode 14.3.1 (14E300c)
AMD Radeon RX 7900 XT	Vulkan	Arch Linux, Mesa 23.1.4
AMD Radeon RX 7900 XT	OpenCL	Arch Linux, OpenCL 2.1 AMD-APP.dbg (3570.0)
AMD Ryzen 7 5700G (int. GPU)	Vulkan	Arch Linux, Mesa 23.1.4
AMD RX 6700 XT	Vulkan	Windows 11 Pro 22H2, AMD Vulkan 2.0.270
HTC 1+ 11 (QC Snapdragon 8 g2)	Vulkan	Android 13, Android Studio (2022.3.1)
HTC 1+ 5T (QC Snapdragon 835)	Vulkan	Android 13, Android Studio (2022.3.1)

Testing for LeftoverLocals

Did not observe violations

Device (GPU)	GPU Framework	OS/Driver/Build system
Nvidia GeForce RTX 4070	Vulkan	Arch Linux, Mesa 23.1.4
Nvidia GeForce RTX 4070	OpenCL	Arch Linux, OpenCL 3.0 CUDA 12.2.128
Intel NUC (NUC10I5FNK)	Vulkan	Ubuntu 22.04, Mesa 20.3.2
Intel NUC (NUC10I5FNK)	OpenCL	Ubuntu 22.04, OpenCL 3.0 NEO (22.31.23852)
Galaxy Tab A (Arm Mali G78)	Vulkan	Android 13, Android Studio (2022.3.1)
Google Pixel 6 (Arm Mali G71)	Vulkan	Android 11, Android Studio (2022.3.1)
Google Pixel 7 (Arm Mali G710)	Vulkan	Android 13, Android Studio (2022.3.1)
Motorola M.G (IM PowerVR GE8320)	Vulkan	Android 11, Android Studio (2022.3.1)

Testing for LeftoverLocals

Did not observe violations

Device (GPU)	GPU Framework	OS/Driver/Build system
Nvidia GeForce RTX 4070	Vulkan	Arch Linux, Mesa 23.1.4
Nvidia GeForce RTX 4070	OpenCL	Arch Linux, OpenCL 3.0 CUDA 12.2.128
Intel NUC (NUC10I5FNK)	Vulkan	Ubuntu 22.04, Mesa 20.3.2
Intel NUC (NUC10I5FNK)	OpenCL	Ubuntu 22.04, OpenCL 3.0 NEO (22.31.23852)
Galaxy Tab A (Arm Mali G78)	Vulkan	Android 13, Android Studio (2022.3.1)
Google Pixel 6 (Arm Mali G71)	Vulkan	Android 11, Android Studio (2022.3.1)
Google Pixel 7 (Arm Mali G710)	Vulkan	Android 13, Android Studio (2022.3.1)
Motorola M.G (IM PowerVR GE8320)	Vulkan	Android 11, Android Studio (2022.3.1)

Later Imagination confirmed that they were impacted and issued a patch

What applications can this impact?

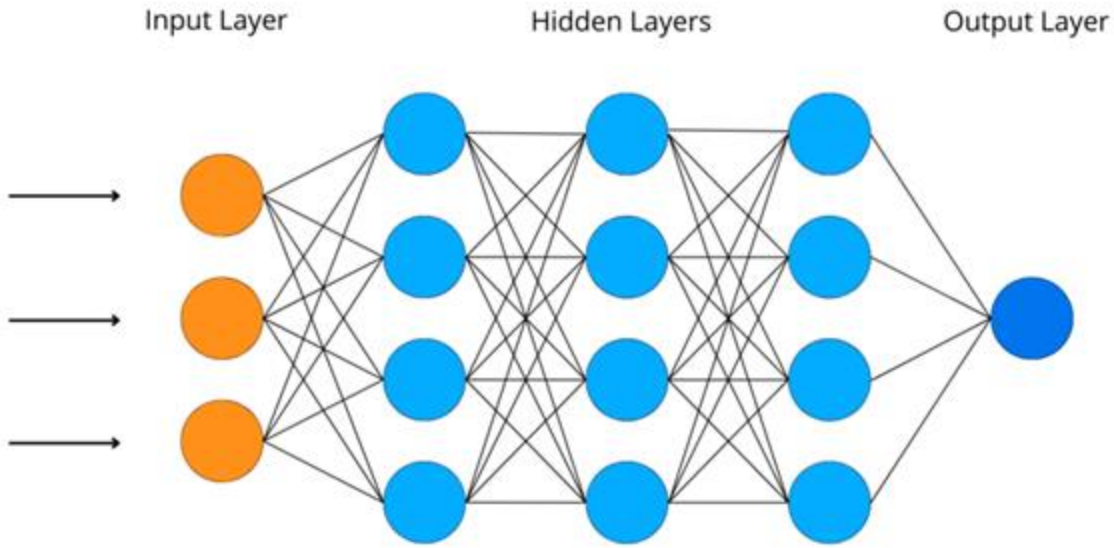
Probably a lot of them! Lots of GPU compute applications use local memory

Large Language Models

Looking at Llama.cpp

Properties:

- Open source
- Straightforward code base
- Supports GPUs from many different vendors
  - CUDA for Nvidia
  - Metal for Apple
  - OpenCL for others



DNN architecture:
* Many hidden layers (my example has 33)
* Each layer is a matrix multiplication
* The matrix multiplication is accelerated by the GPU

Image From: https://www.scalablepath.com/data-science/chatgpt-architecture-explained

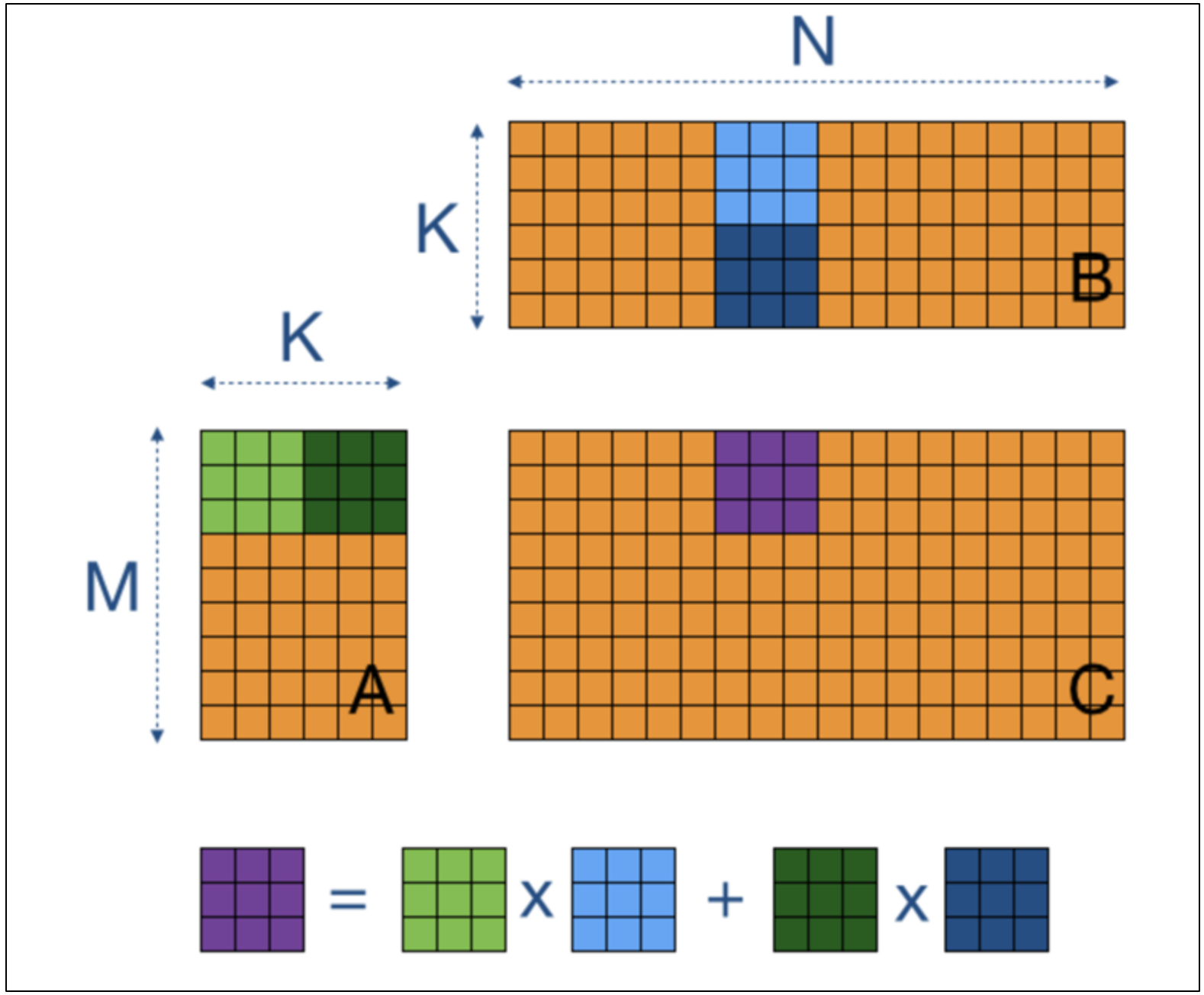
Optimized GPU Matrix Multiplication

Compute:

C = A x B

Cache tiles in local memory

Leaky Llamas can then grab the cached tiles!



From: https://cnugteren.github.io/tutorial/pages/page4.html

Using LeftoverLocals

process 0

```
Llama_layer0
Llama_layer1
Llama_layer2
…
Llama_layer33
```

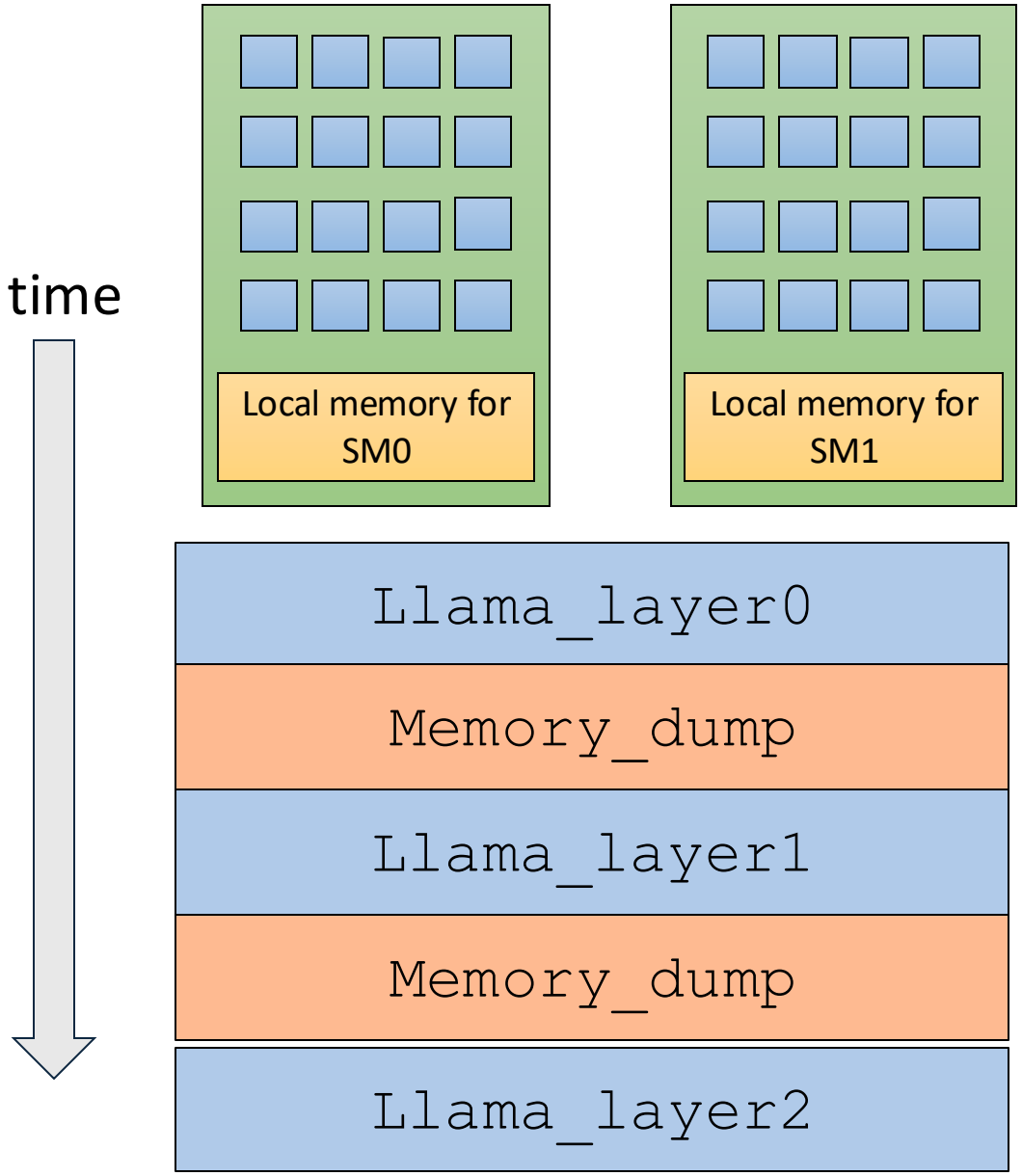
process 1

```
MemoryDump
MemoryDump
MemoryDump
…
```

We can steal a decent amount each layer:

~3 MB of weights per layer - We can use it to fingerprint the model

~3 MB of intermediate computation

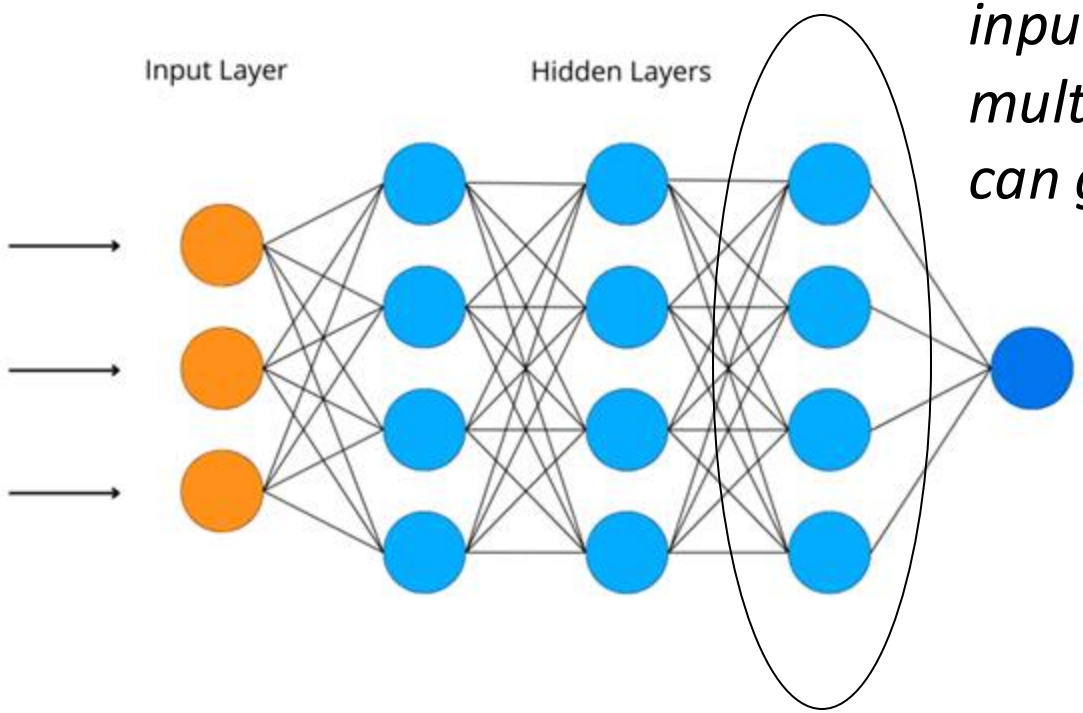


DNN Layers

But one layer is more interesting than the others:

DNN Layers

But one layer is more interesting than the others:



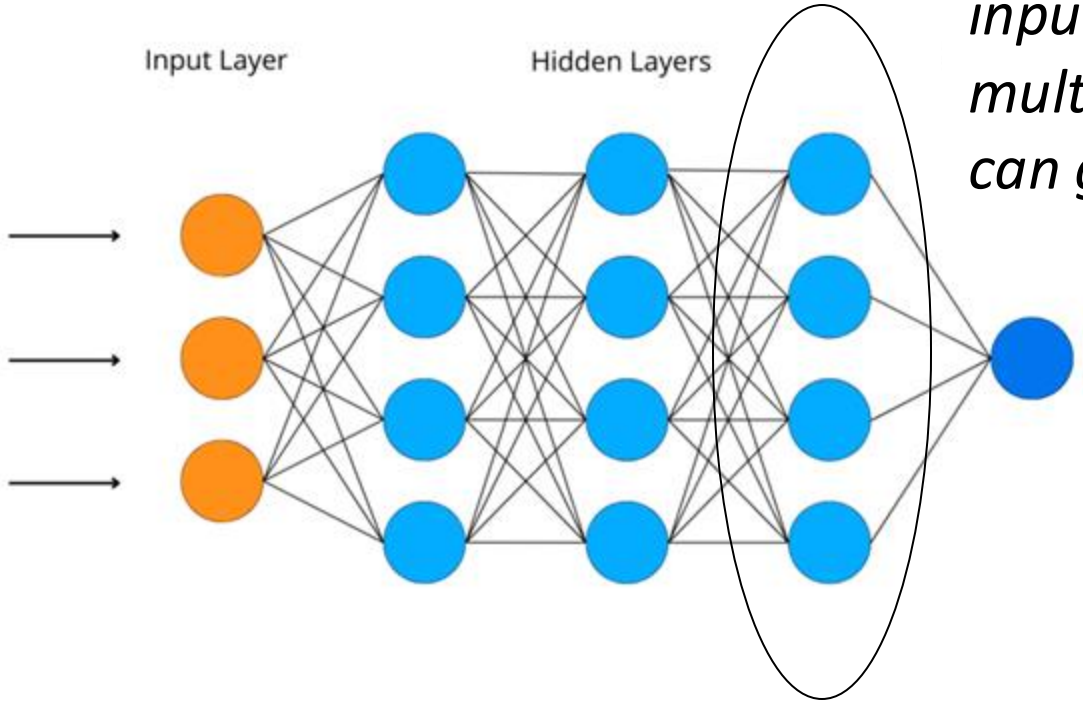
If the attacker can get the inputs to this matrix multiplication then we can get the output!

For the model I'm using:

Weights: 32001 x 4096
Input: 4096 x 1

DNN Layers

But one layer is more interesting than the others:



If the attacker can get the inputs to this matrix multiplication then we can get the output!

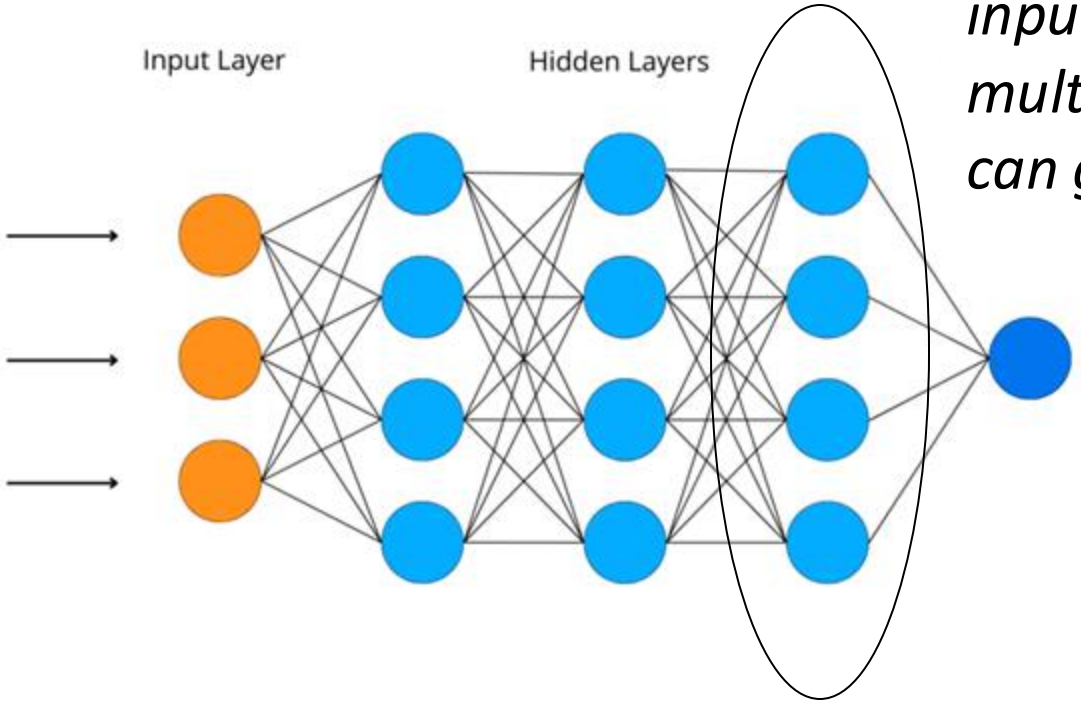
For the model I'm using:

Weights: 32001 x 4096
Input: 4096 x 1

Weights too big to steal :(

DNN Layers

But one layer is more interesting than the others:



If the attacker can get the inputs to this matrix multiplication then we can get the output!

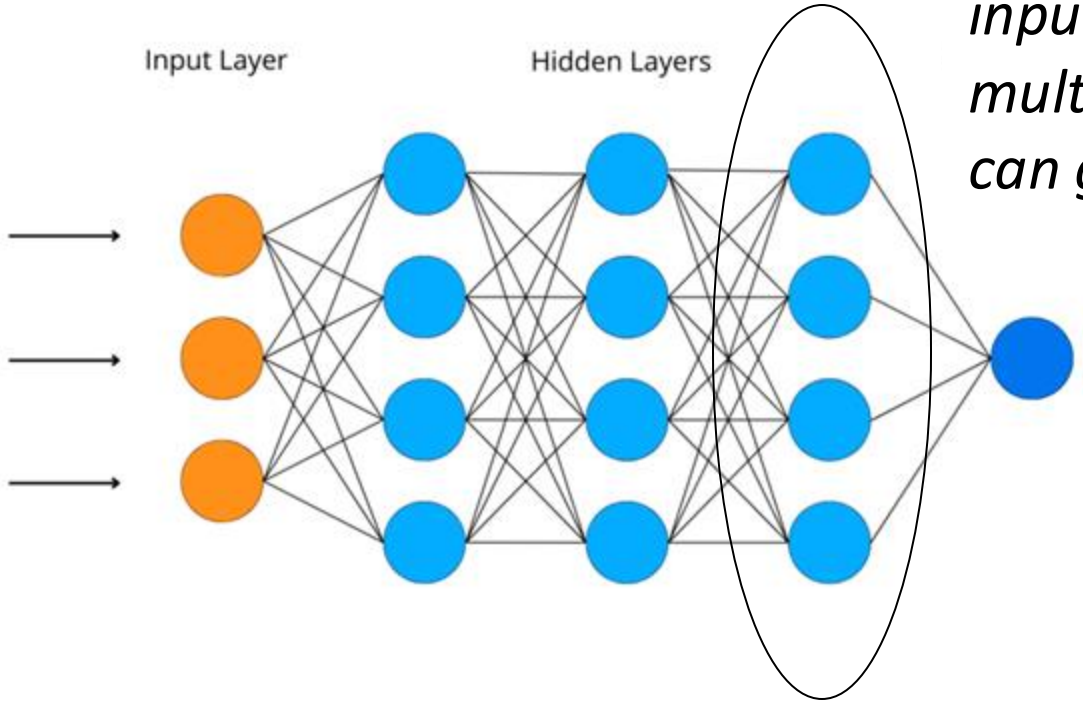
But we can finger print the model and get all weights from hugging face

For the model I'm using:

Weights: 32001 x 4096
Input: 4096 x 1

DNN Layers

But one layer is more interesting than the others:



If the attacker can get the inputs to this matrix multiplication then we can get the output!

For the model I'm using:

Weights: 32001 x 4096
Input: 4096 x 1

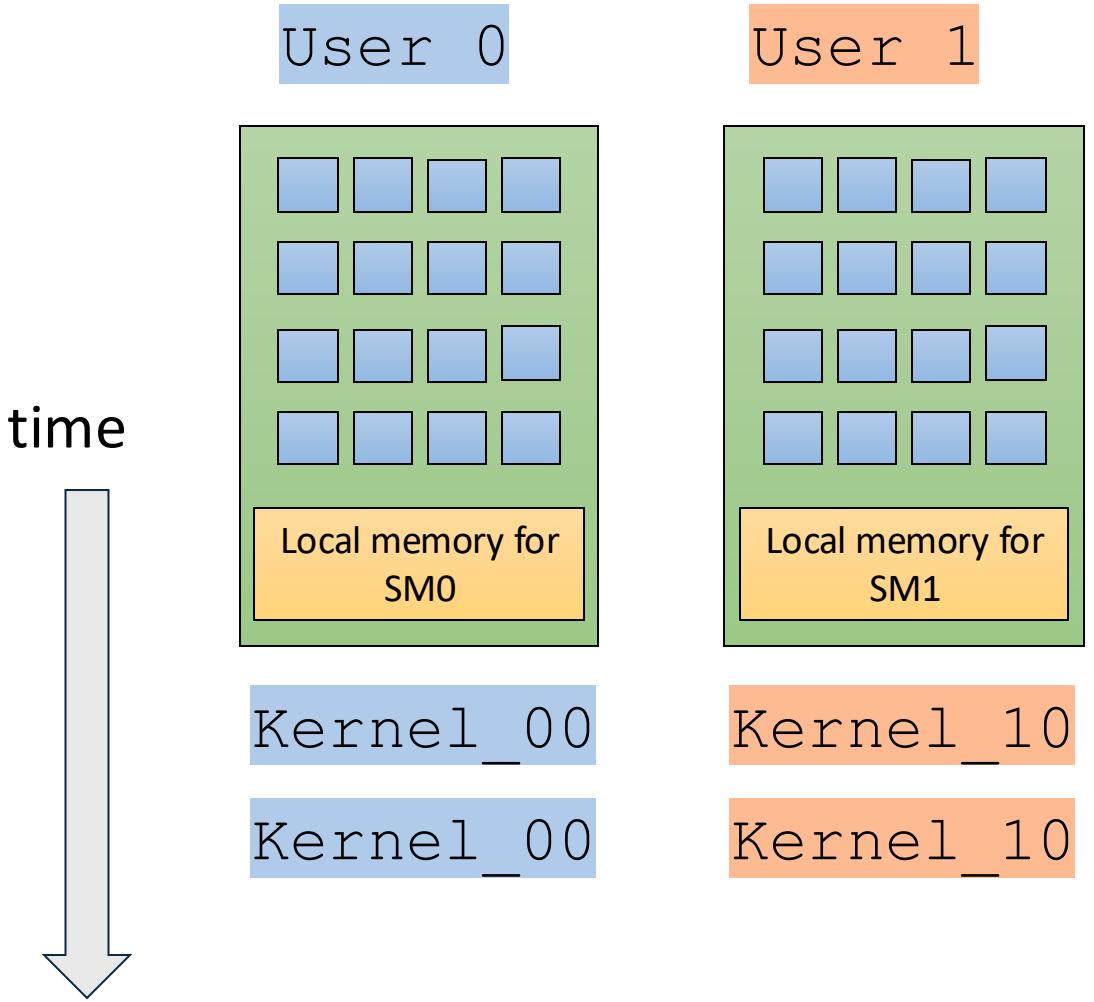
Input is small enough to fit in local memory

Impact of leak

Physically partitioned GPU, available in high-end Nvidia and AMD models

Are you safe??

- AMD and Nvidia physically partition GPUs for virtualization
- AWS and Azure do not allow multi-tenant access to their GPUs
- WebGPU enforces memory safety through their compiler
- Other use cases on your phone/computer/cloud? Potentially vulnerable



Creative people might run into trouble...

omkaar
@omkizzy

GPU Poor? **Friends** is all you need.

i built a distributed training module from scratch that performs model training over a cluster of M-series macs connected over the same network.

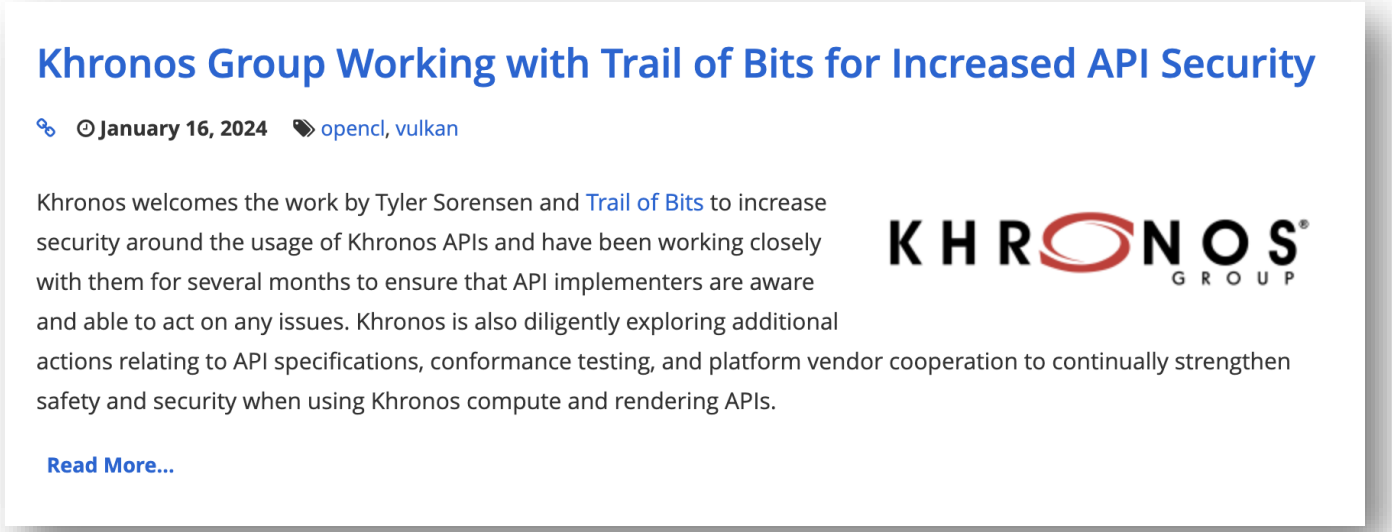
this was inspired by @anyscalecompute's Ray, @hnasr's work, @ShopifyEng's merlin article and @maximelabonne's spirit.

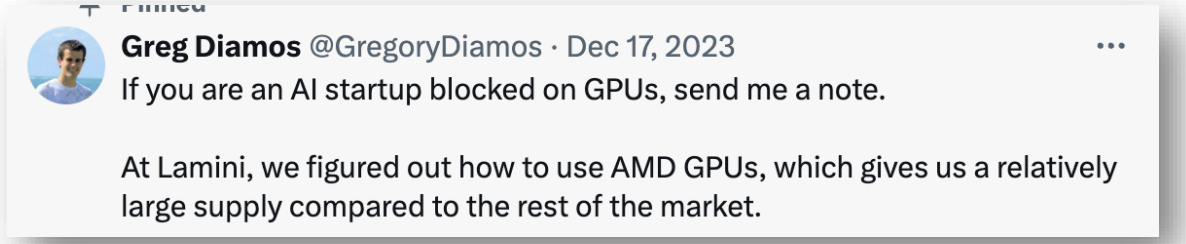
Reporting and current state

- Reporting
  - Worked with CERT on a coordinated disclosure
  - Worked with vendors on a slightly longer deadline (125 days) than the usual 90 day deadline
  - Lots of vendors involved, but went smoothly all things considered
- Qualcomm, Imagination issued a patch
- Apple has patched some devices
  - A17 and M3 are patched
  - Others are not (including the M2 as the demo shows)
- AMD has promised a patch
  - Mentioned that they were aiming for a patch end of April

Future?

- Lots of research to do!
  - Testing
  - Verification
  - Threat modeling
- Lots of outreach to do
  - Where do we specify/test this?
- Lots of domains to consider
  - WebGPU
  - Local models
  - Federated Learning
  - Cloud with different GPU types



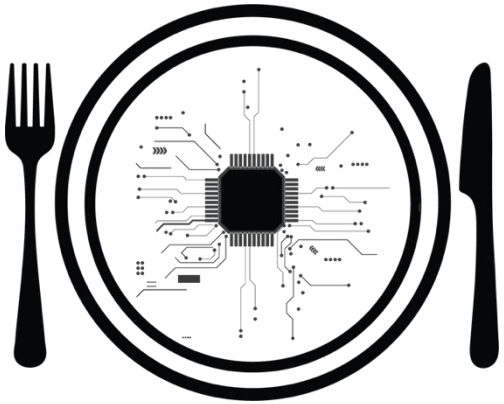


LeftoverLocals Conclusion

- Very simple approach was able to identify significant data leakage
  - Surely many more exploits to be found

- Exploit required reasoning about unique GPU memory regions and execution model
  - Also took advantage of open source models

- Lots of interesting work to do in GPU + Security!



https://users.soe.ucsc.edu/~tsorensen/
https://twitter.com/Tyler_UCSC

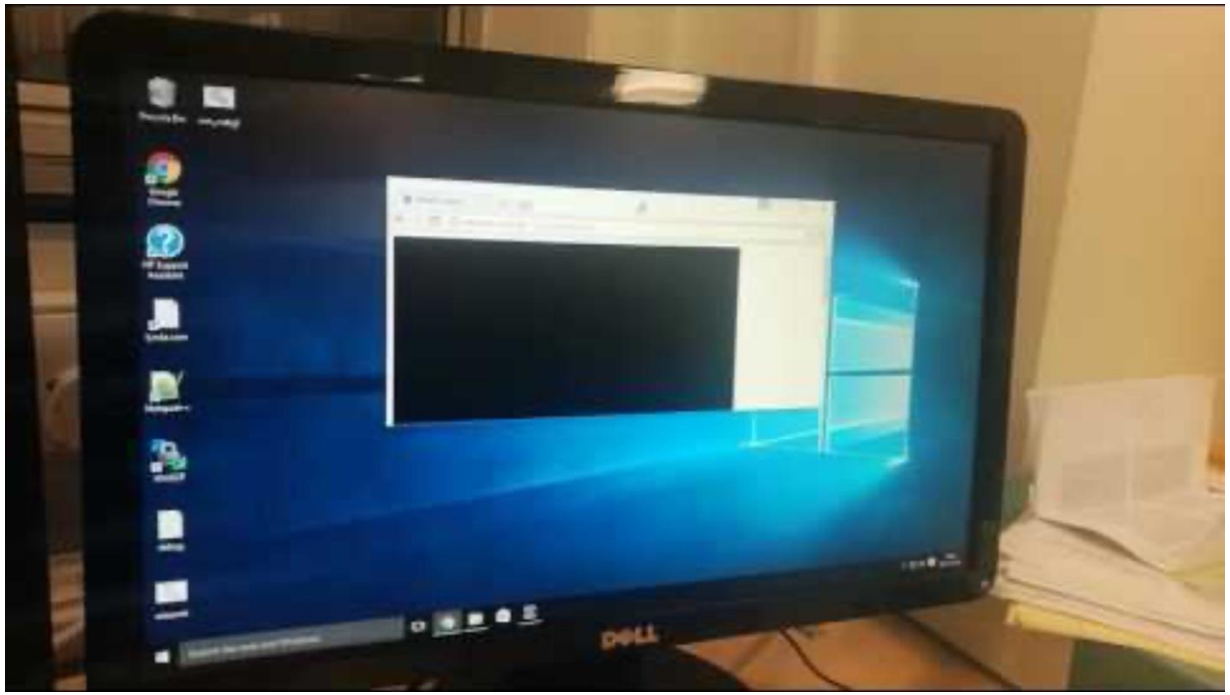
COVERT CHANNEL DEMO

Let’s think about potential vulnerabilities

DoS - What if `Kernel_00` spins forever?

CVE-2017-6259

NVIDIA GPU Display Driver contains a vulnerability in the kernel mode layer handler where an incorrect detection and recovery from an invalid state produced by specific user actions may lead to a denial of service.



https://nvidia.custhelp.com/app/answers/detail/a_id/4525/~/security-bulletin

https://medium.com/@afd_icl/no-more-amd-webgl-bluescreens-f6da1df19f4d