Stealing Webpages Rendered on Your Browser by Exploiting GPU Vulnerabilities

2014 IEEE Symposium on Security and Privacy

Sangho Lee, Youngsok Kim, Jangwoo Kim, Jong Kim
POSTECH, Korea

2014-5-19

Graphics Processing Unit (GPU)

- An essential component of modern computers
  - Graphics APIs (DirectX, OpenGL)
  - Computing APIs (CUDA, OpenCL)
- Various applications
  - Game/graphics applications
  - Data analysis tools
  - Security applications
    - Cryptographic engines
    - Intrusion detection systems

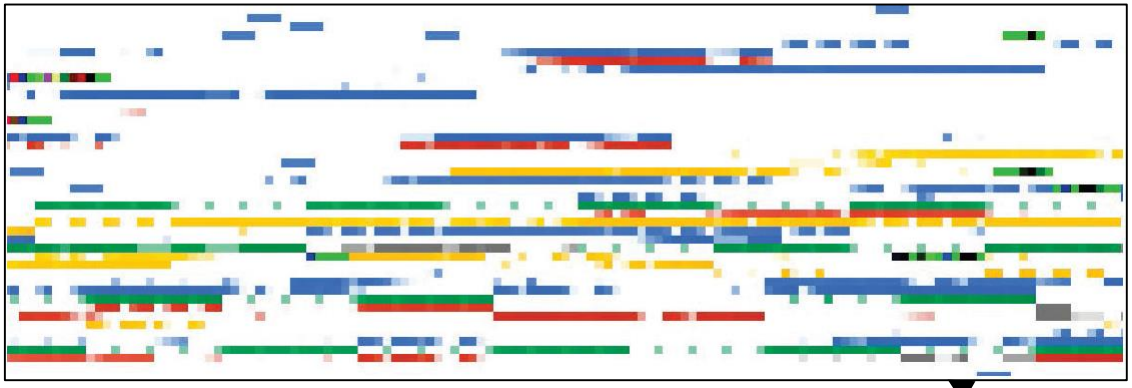
Is GPU secure?

GPU Can Be Shared

- GPU cluster
  - Multiple users can use GPUs to perform computing tasks in a time-sharing fashion.
- Personal computer/workstation
  - A foreground user solely uses a GPU for graphics tasks.
  - Background users can use the GPU for computing tasks.

Can attacker access other's data remaining in GPU memory?

Isn't It Familiar?



A GPU memory dump obtained after visiting google.com

Google

It's me!

Sensitive data are leaked!

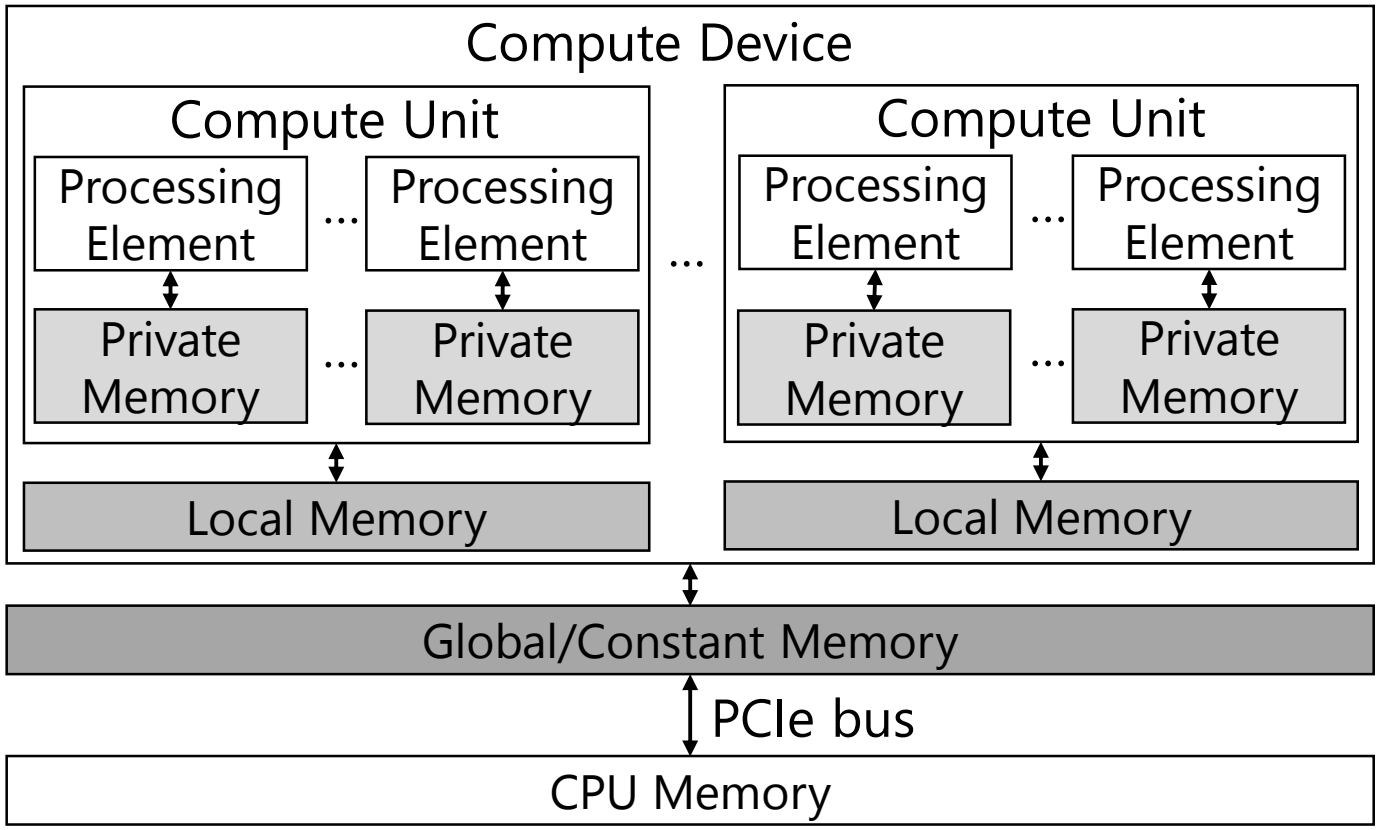
Motivation and Research Goal

- Motivation
  - Lack of an in-depth study of GPU security problems
  - Sensitive data leakage through a disclosed GPU memory
- Research goal
  - Establish and emphasize GPU security problems
    - Discover and specify the security problems of GPUs
    - Develop widely-applicable security attacks on GPUs

Contents

- Introduction
- **GPU Basics and Concerns**
- Disclosing GPU Memory
- Inferring Browsing History from GPUs
- Discussion
- Conclusion

High-level Architecture of GPU



Security Concerns about GPU

- Uninitialized memory
  - GPUs do not initialize the contents of newly allocated memory pages.
- Unerasable memory
  - Some memory types cannot be deleted even manually.
    - Constant memory, codes, call-by-value arguments
- Manually-managed memory
  - No isolation mechanism exists for the private and local memories of GPUs.

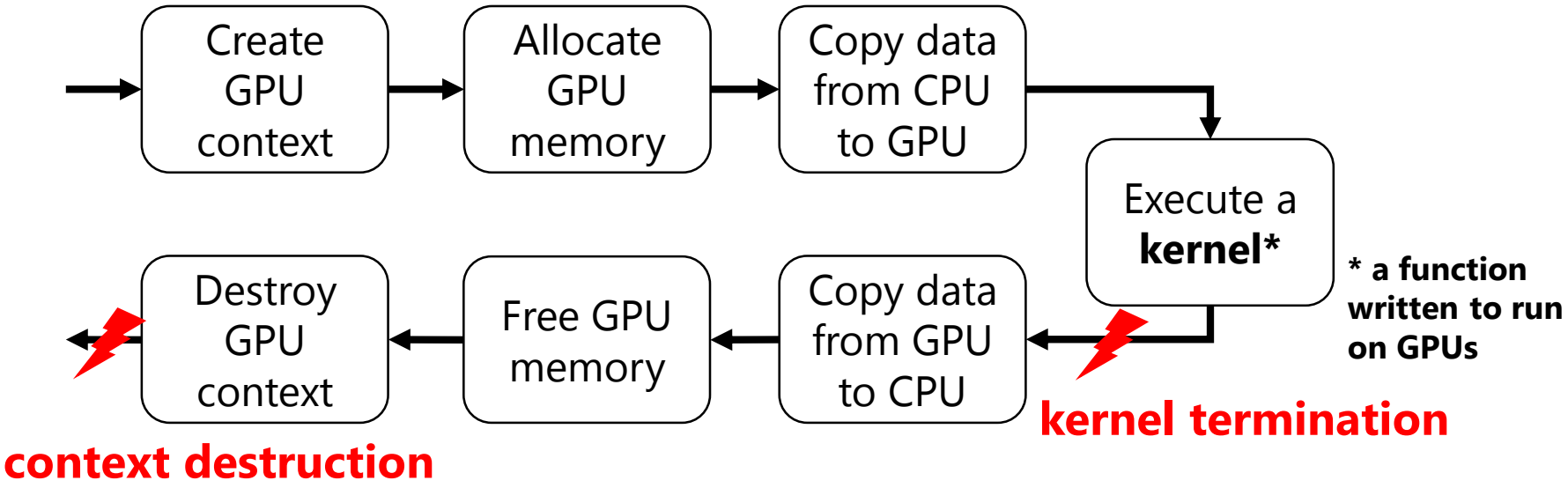
Contents

- Introduction
- GPU Basics and Concerns
- **Disclosing GPU Memory**
- Inferring Browsing History from GPUs
- Discussion
- Conclusion

System and Attack Models

- Target GPU system
  - Run both graphics and computing tasks
  - Support multiple users
- Victim
  - Use GPU-accelerated programs
  - Occupy the system's screen to use graphics APIs
- Attacker
  - Have no root privilege (another normal user)
  - Can access the GPU to use computing APIs
  - Attempt to attack the victim by using the GPU

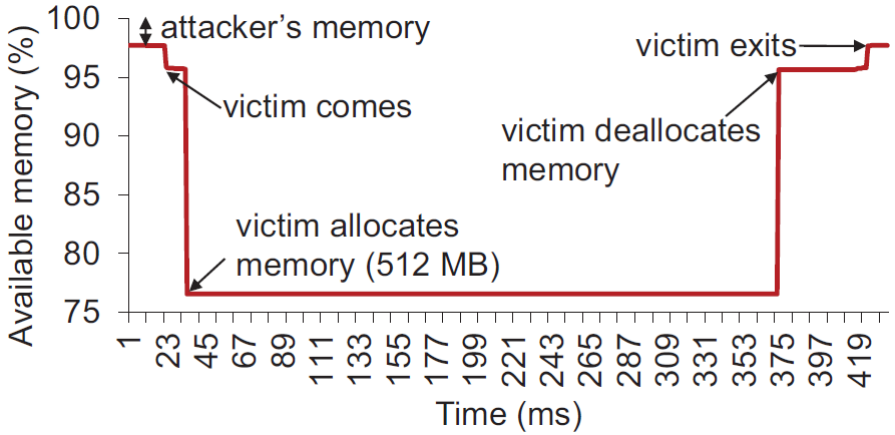
GPU Execution Flow and Attack Points



- After kernel termination
  - Disclose private/local memories to obtain the results of a kernel function
- After context destruction
  - Disclose global memory to obtain the final results, kernel codes, arguments, …

Recognizing Victim's Activities

- Recognizing kernel termination
  - GPU queues kernels and executes them one at a time.
  - By measuring the delay of kernel execution, attackers can identify whether a victim executes a kernel.
- Recognizing context destruction
  - By monitoring the available memory size, attackers can identify the activities of a victim.

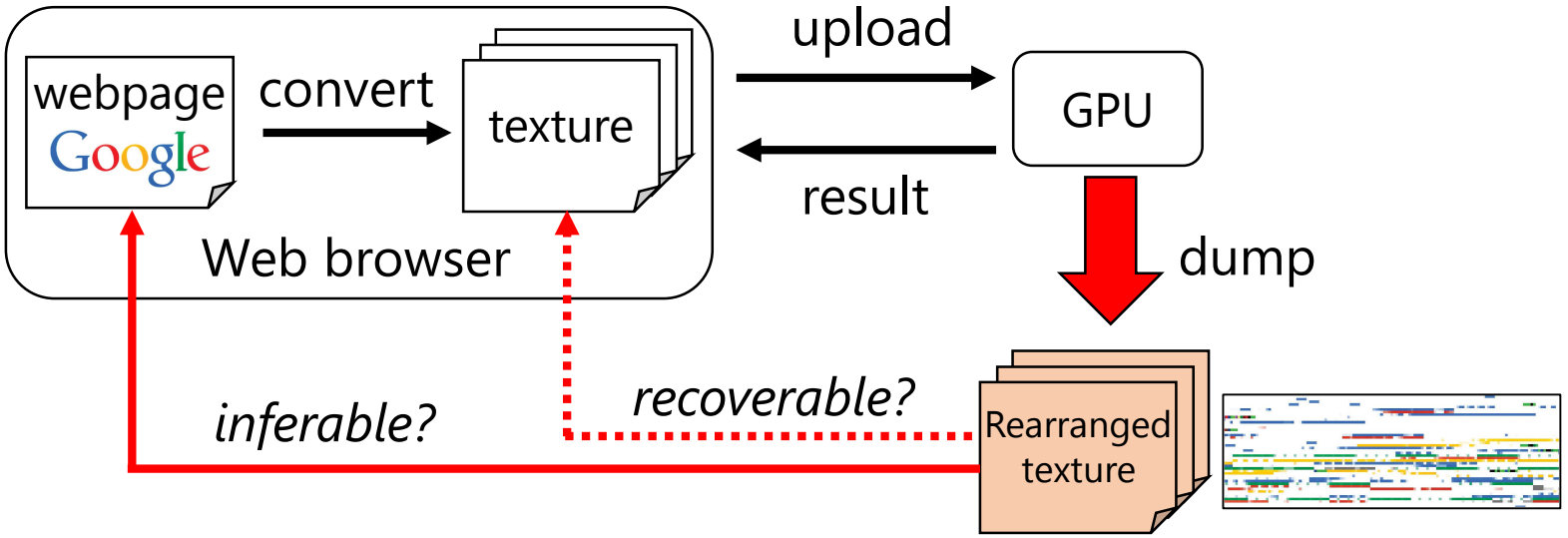


* 2687 MB in total

Contents

- Introduction
- GPU Basics and Concerns
- Disclosing GPU Memory
- **Inferring Browsing History from GPUs**
- Discussion
- Conclusion

GPU-accelerated Webpage Rendering



- Textures uploaded to GPU memory are rearranged.
  - Recovering is almost impossible.
- Can we infer the original webpage from rearranged textures?
  - **We compare the similarity between the victim's dump with some data of popular webpages.**

Attack Scenarios

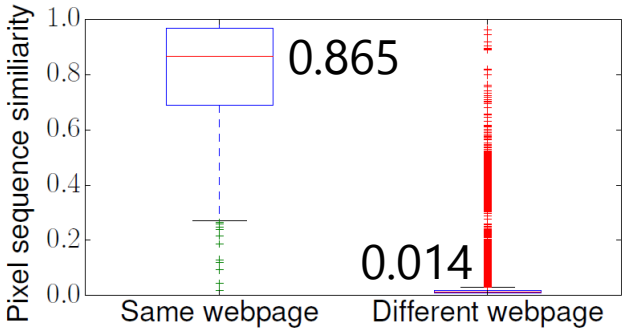
- Attack using GPU memory dumps
  - Prepare the dumps of popular webpages by using the same GPU that a victim uses
  - Compare a victim's dump with the prepared dumps
- Attack using webpage snapshots
  - Prepare the image snapshots of popular webpages by using any systems
  - Compare a victim's dump with the prepared snapshots

Pixel Sequence Matching

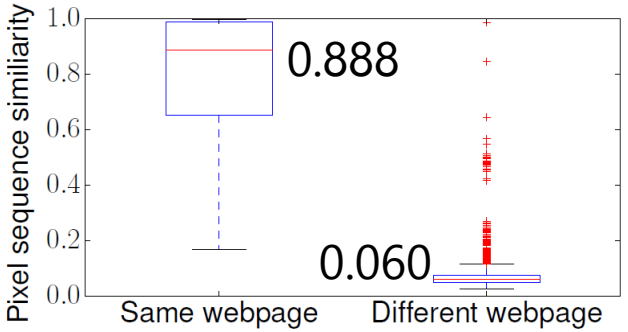
- Extract non-black and non-white contiguous pixel sequences from GPU memory dumps
  - Black: cannot be distinguished with zero
  - White: the default background color of webpages
- Compute Jaccard Index between pixel sequence sets

Jaccard Index: $\frac{|set_1 \cap set_2|}{|set_1 \cup set_2|} = \frac{2}{7}$

Pixel Sequence Similarity



Chrome/NVIDIA GPU



Firefox/AMD GPU

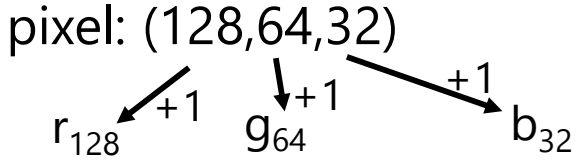
The similarity between the same pages is higher than that of different pages.

- The front pages of Alexa Top 1000 domains
- Visiting each page 10 times

RGB Histogram Matching

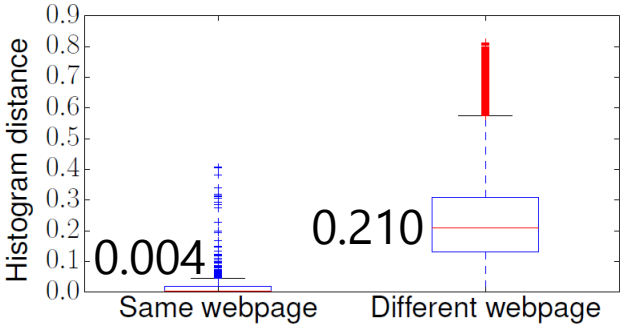
- Derive non-black and non-white RGB histograms from GPU memory dumps
  - RGB histogram: a tuple of red, green, and blue channels (256 values for each)

$$H = (r_0, r_1, \ldots, r_{255}, g_0, g_1, \ldots, g_{255}, b_0, b_1, \ldots, b_{255})$$

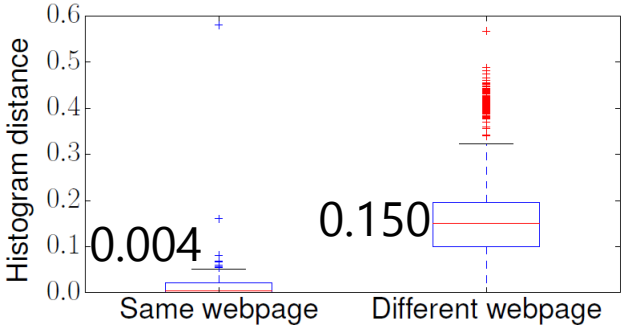


- Compute Euclidean distance between histograms
  - Divide each value by the sum of all 768 values (normalization)
  - Use a random projection for dimensionality reduction

RGB Histogram Distance



Chrome/NVIDIA GPU



Firefox/AMD GPU

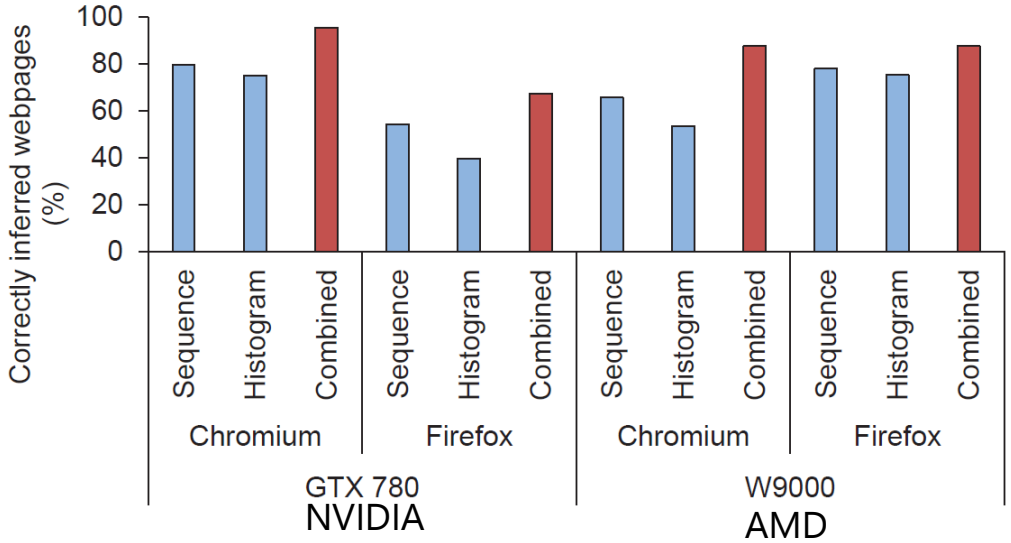
The distance between the same pages is shorter than that of different pages.

Pixel Sequence vs. RGB Histogram

- Pixel sequence matching
  - Do not work when two dumps originate from different generation/vendor GPUs
  - Have slow matching speed
    - Take ~0.451s to compare two dumps
- RGB histogram matching
  - Work even when two dumps originate from different generation/vendor GPUs
  - Have fast matching speed
    - take ~0.002s to compare two dumps

* Measured at Intel Core i7-2600, 8 GB

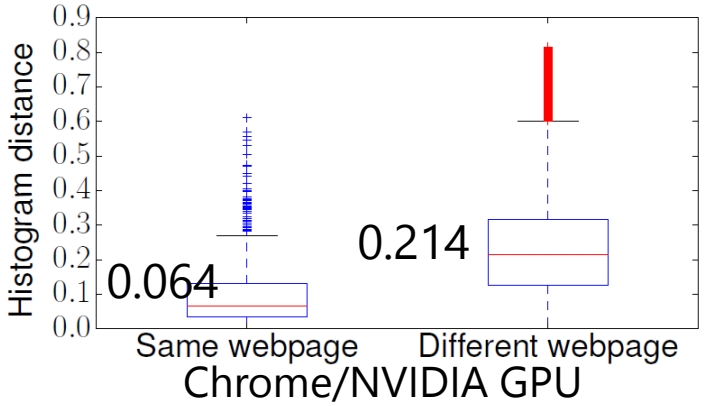
Inference Accuracy

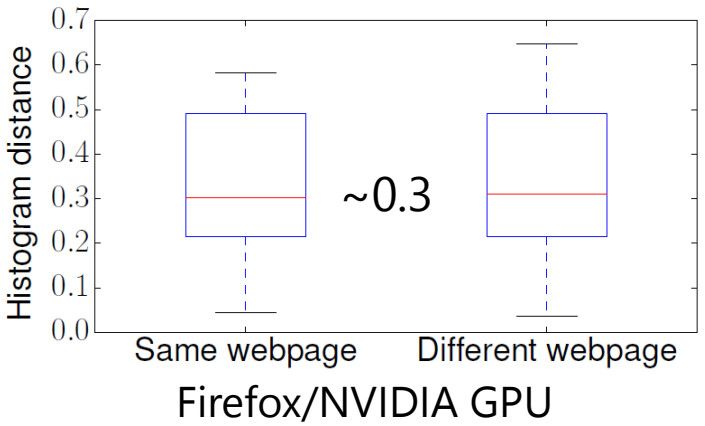


- Combined matching
  - Use pixel sequence matching to select top-k similar dumps
  - Apply RGB histogram matching on the selected dumps
- Inferring random 100 pages of Top 1000 domain
  - Best accuracy: **95.4%** of Chromium with NVIDIA GPU

Attack using Webpage Snapshot

- Preparing webpage snapshots
  - Use PhantomJS to take image snapshots
- Histogram distance b/w dumps and snapshots





- Inference accuracy
  - Chrome: **~50%**, Firefox: **~22%** ← **Dumps have many non-texture data.**

Contents

- Introduction
- GPU Basics and Concerns
- Disclosing GPU Memory
- Inferring Browsing History from GPUs
- **Discussion**
- **Conclusion**

Discussion

- Mitigation
  - Initialize newly allocated memory pages
  - Delete the entire private and local memories at each GPU context switch
- Unavoidable performance degradation
  - GPU context switch cost: **<25µs**
  - Clearing cost of the private/local memories: **~80µs**

We need efficient solutions.

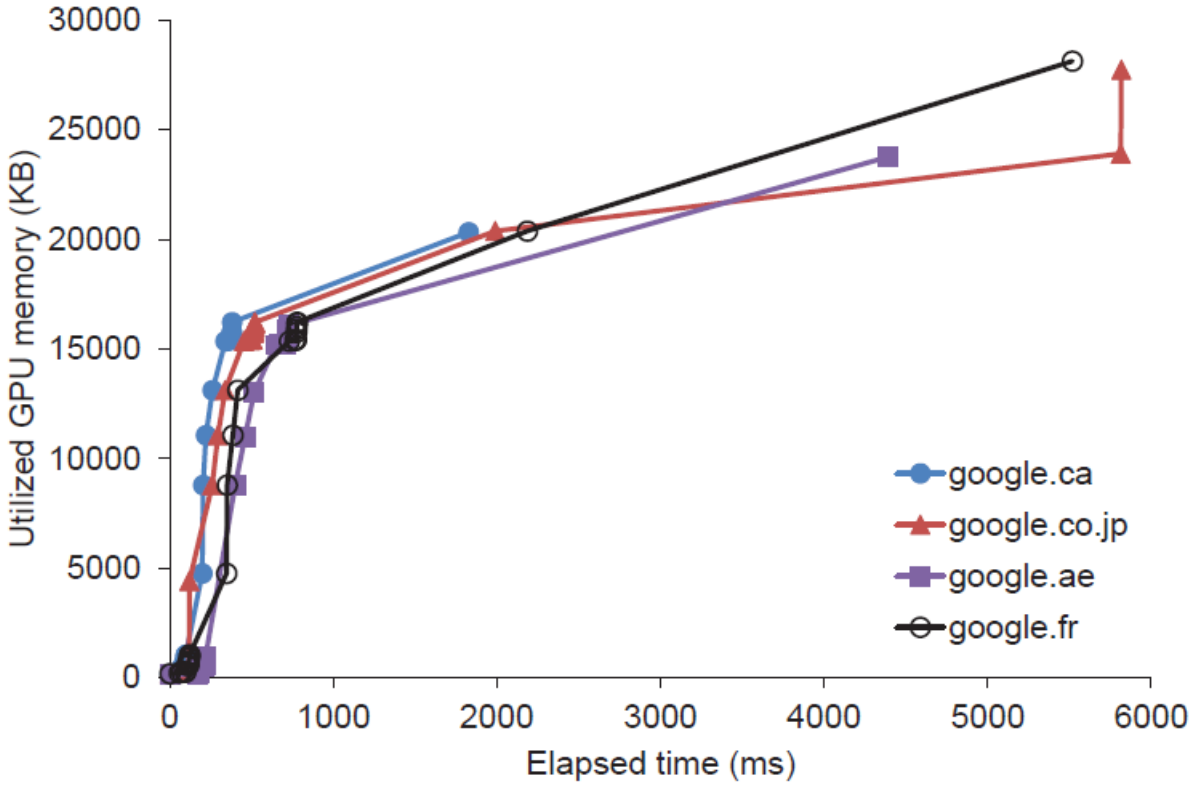
* Measured at NVIDIA GeForce GTX 780

Conclusion

- We analyze and emphasize the security problems of GPUs.
  - Conduct an in-depth study of GPU security problems
  - Describe attacks to reveal sensitive data kept in GPU memory
  - Apply the attacks on popular web browsers
- Effective countermeasures need to be developed.

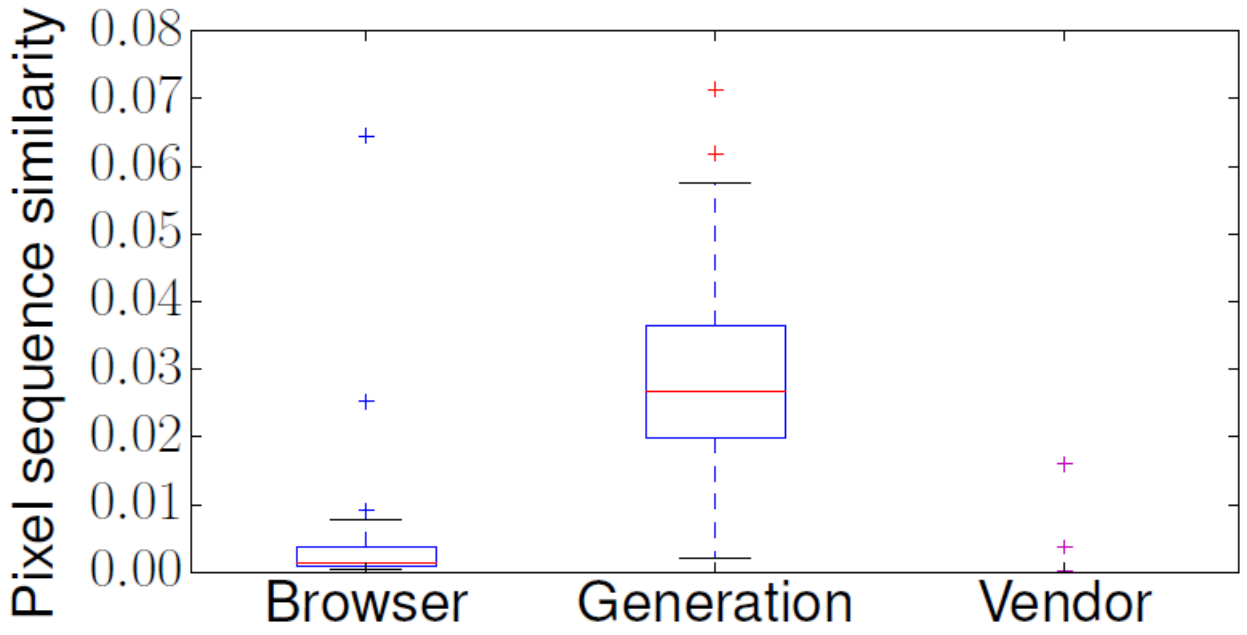
Backup Slides

Regional Google's



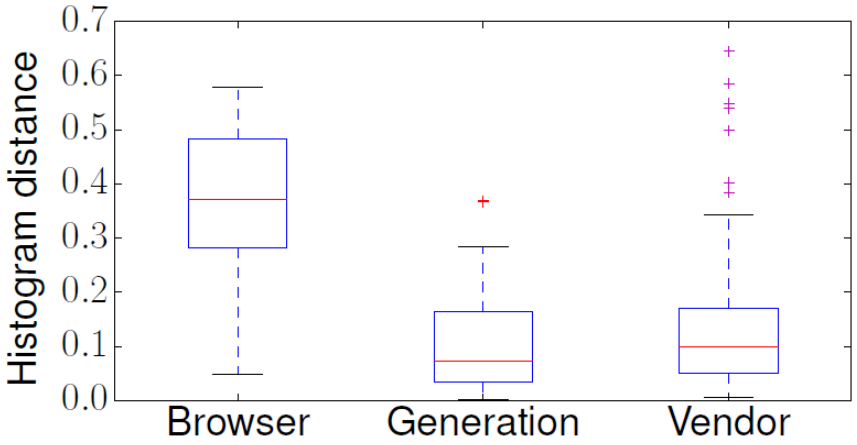
Have different time vs. memory usage patterns

Pixel Sequence Matching between Same Page Dumps from Different GPUs

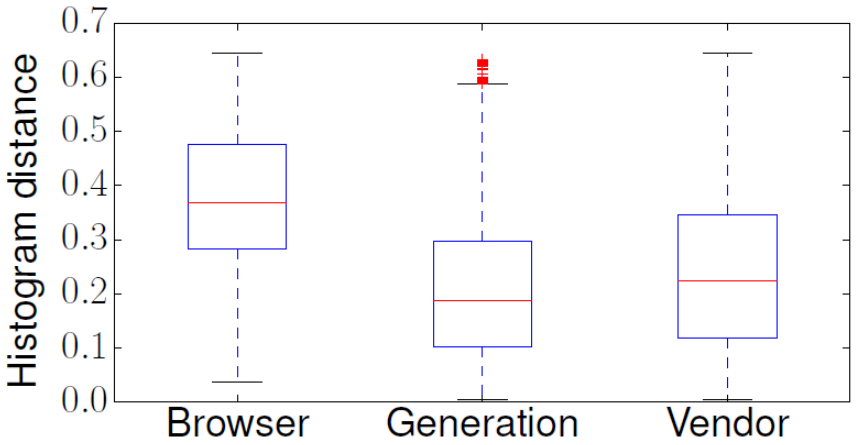


The same webpage dumps from different GPUs are treated different.

Histogram Matching between Different GPUs



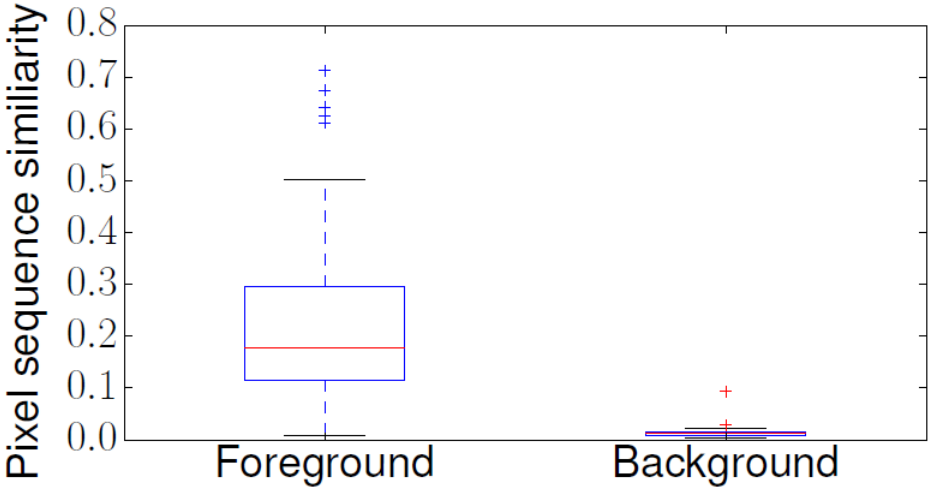
(a) Same webpage.



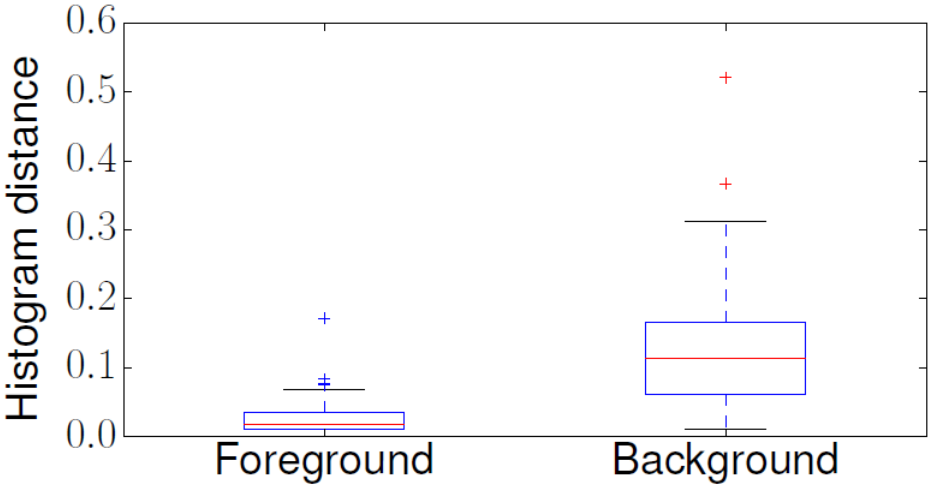
(b) Different webpage.

Dumps from different generation/vendor GPUs can be distinguished.

Two Tabs



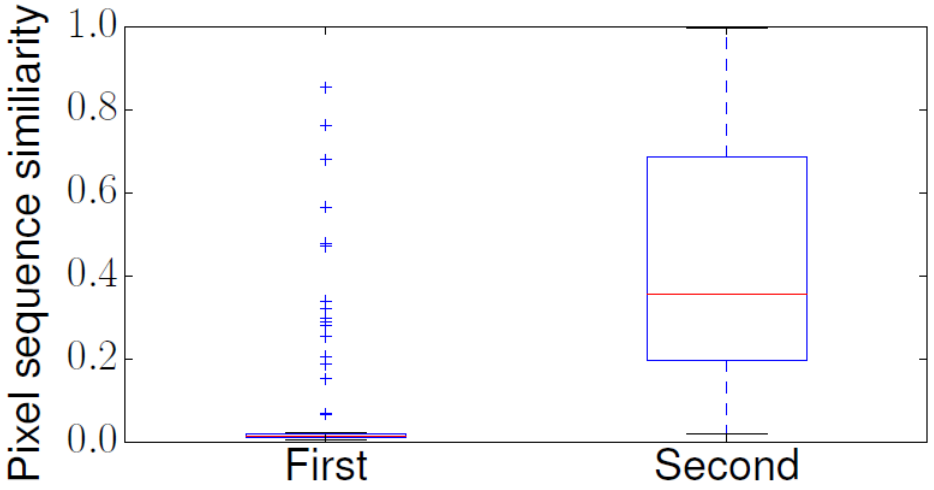
(a) Pixel sequence.



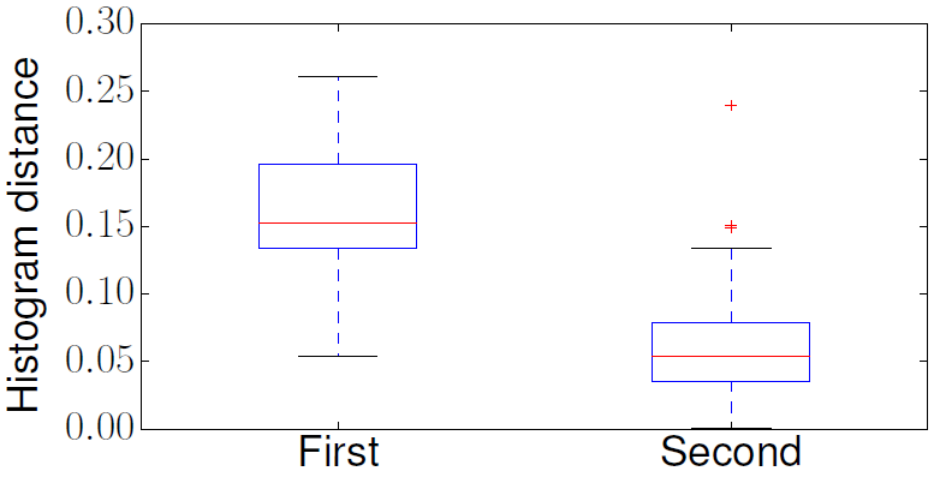
(b) RGB histogram.

Textures of the foreground tab remain.

Two Windows



(a) Pixel sequence.



(b) RGB histogram.

Textures of the lastly opened window remain.