RAEAX

Rowhammer Amplification by Execution of Additional X86 instructions

Martin Heckel

June 29, 2021

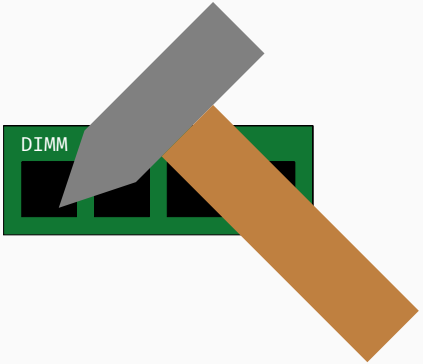
Hof University, University of Applied Sciences

Outline

Introduction

What is Rowhammer?

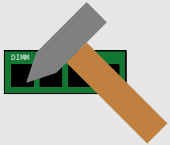
- Flipping Bits in Memory Without Accessing Them [3]
- Current DRAM has a very high integration density
- Frequent access to memory locations leads to bit flips in nearby memory locations
- Not the intended behaviour (this should not happen)



What is Rowhammer?

- Flipping Bits in Memory Without Accessing Them [2]
- Curre
densi
- Frequ
to bit
- Not th
happen)

So what?



> *Space [...] is* ***big****. Really big. You just won't believe how vastly hugely mindbogglingly big it is.*
>
> — Douglas Adams

Mostly harmless

- Local Denial of Service (DoS)

Mostly harmless … or …

- Local Denial of Service (DoS)
- Escalate privileges locally on desktop computers [5]

Mostly harmless … or …

- Local Denial of Service (DoS)
- Escalate privileges locally on desktop computers [5]
- Escalate privileges locally on mobile devices [7]

Mostly harmless … or … not as harmless.

- Local Denial of Service (DoS)
- Escalate privileges locally on desktop computers [5]
- Escalate privileges locally on mobile devices [7]
- Using browsers (rowhammer JavaScript) [2]

Mostly harmless … or … not as harmless.

- Local Denial of Service (DoS)
- Escalate privileges locally on desktop computers [5]
- Escalate privileges locally on mobile devices [7]
- Using browsers (rowhammer JavaScript) [2]
- Over the network (by sending network packages) [6]

Mostly harmless … or … not as harmless.

- Local Denial of Service (DoS)
- Escalate privileges locally on desktop computers [5]
- Escalate privileges locally on mobile devices [7]
- Using browsers (rowhammer JavaScript) [2]
- Over the network (by sending network packages) [6]
- Break VMs isolation and get access to other VMs on the same host [4]

Mostly harmless … or … not as harmless.

- Local Denial of Service (DoS)
- Escalate privileges locally on desktop computers [5]
- Escalate privileges locally on mobile devices [7]
- Using browsers (rowhammer JavaScript) [2]
- Over the network (by sending network packages) [6]
- Break VMs isolation and get access to other VMs on the same host [4]
- …

Mostly harmless … or … not as harmless.

- Local Denial of Service (DoS)
- Esc
- Esc
- Usi
- Ove
- Bre
- …

Problem detected!

Let’s **mitigate** it!

Is this still a problem?

- DDR2 systems are typically not affected due to lower integration density
- DDR3 and DDR4 systems got BIOS updates with mitigations starting in 2015
- DDR4 memory modules often contain mitigations additionally.
- Rowhammer is not a problem anymore

Is this still a problem?

- DDR2 systems are typically not affected due to lower integration density
- DDR
- DDR

Of course, the problem is not solved!

2015

¯_(ツ)_/¯

STILL FLIPPING ANYWAY

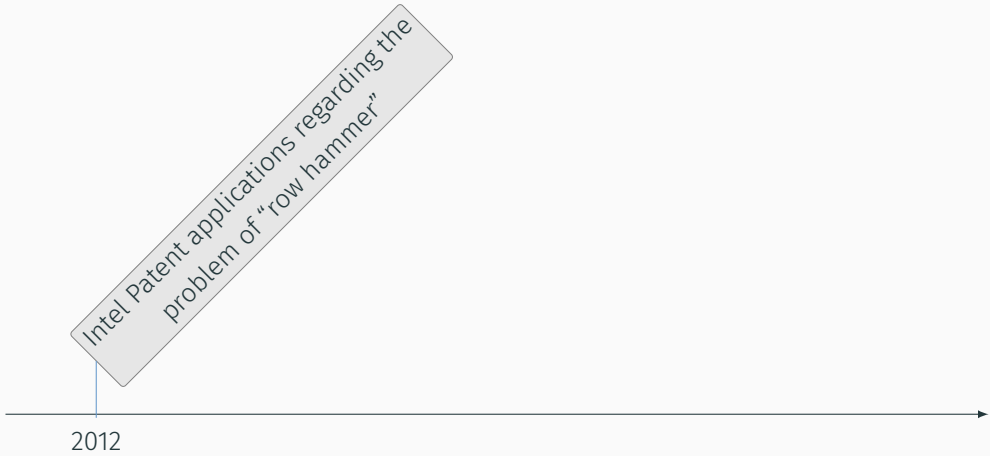
Is this still a problem?

- DDR2 systems are typically not affected due to lower integration density
- DDR3 and DDR4 systems got BIOS updates with mitigations starting in 2015
- DDR4 memory modules often contain mitigations additionally.

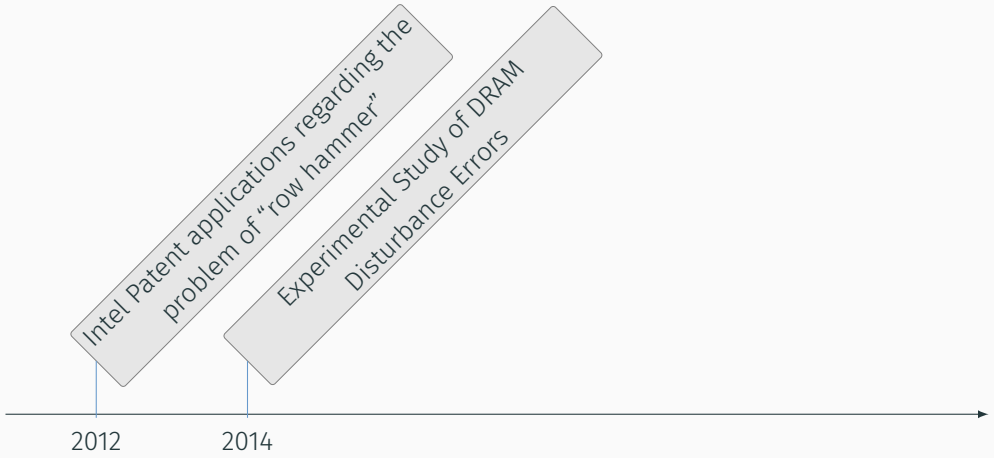
- The mitigations on DDR3 systems are not as effective as assumed
- $\frac{1}{4}$ of DDR4 modules are still vulnerable [1]

History of rowhammer

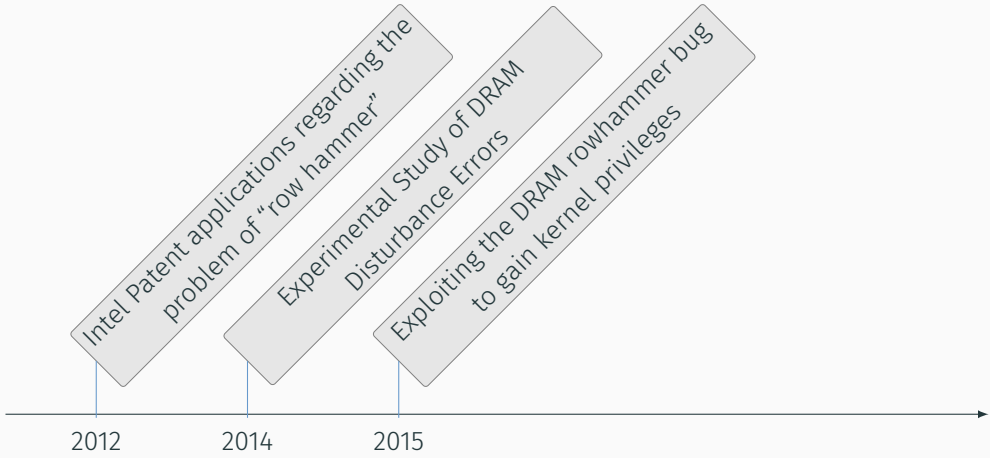
History of rowhammer



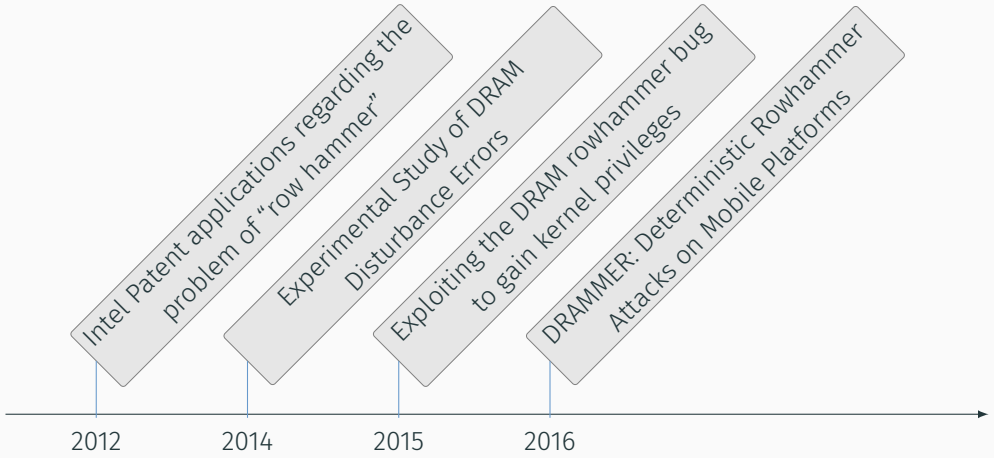
History of rowhammer



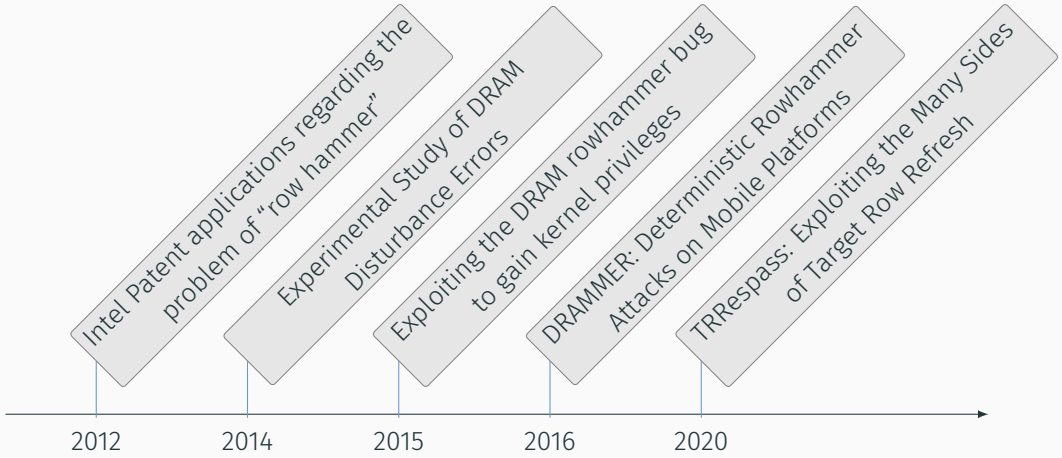
History of rowhammer



History of rowhammer



History of rowhammer



Background

Reminder: Virtual Memory Management

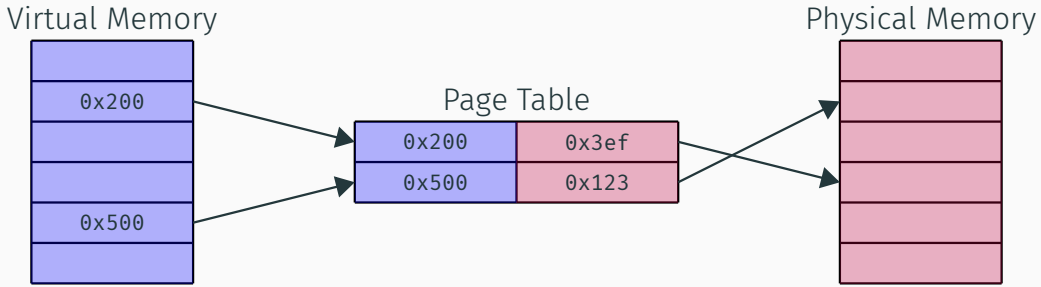
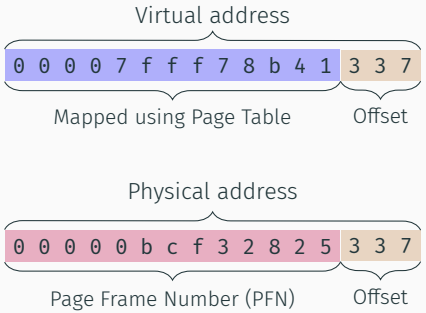


Image from the slides “0x0D Low-Level Fundamentals”

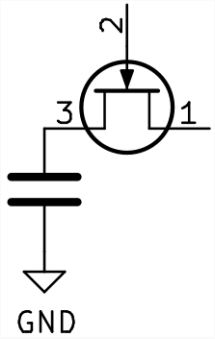
Virtual Memory Management Part 1 — Pages

- Memory is managed in Pages
- On x86 Linux, one page has a size of 4 KiB (12 bit required for addressing)
- An address in virtual memory can be considered as a concatenation of “page address” and “offset”.
- First 52 bits are mapped, last 12 bits are “copied”



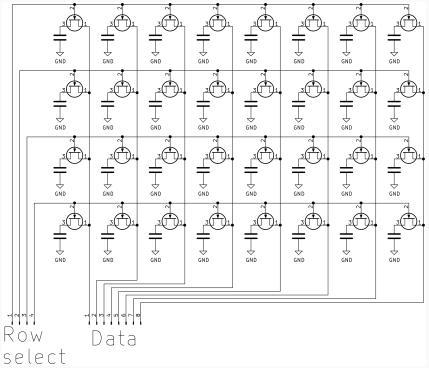
DRAM basics Part 1 — Cells

- A single cell consists of a capacitor storing the actual data and a transistor controlling the access
- In this example: *control* at pin 2 and *access* at pin 1
- Reading procedure: Enable the control pin and read the voltage at the access pin
- Writing procedure: Apply the level that should be written to the access pin and enable the control pin



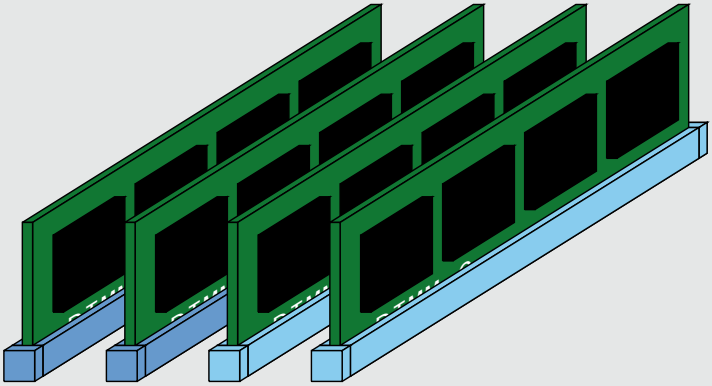
DRAM basics Part 2 — Array

- Multiple of these cells are organized in an array
- Control pins of the cells are connected in rows (only entire rows can be enabled)
- Access pins of the cells are conneted in columns
- Capacitors loose chage over time, so it is required to refresh the cells periodically



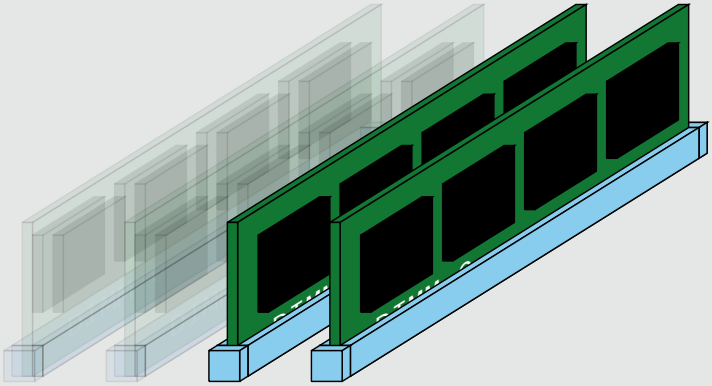
Physical DRAM architecture

System DRAM



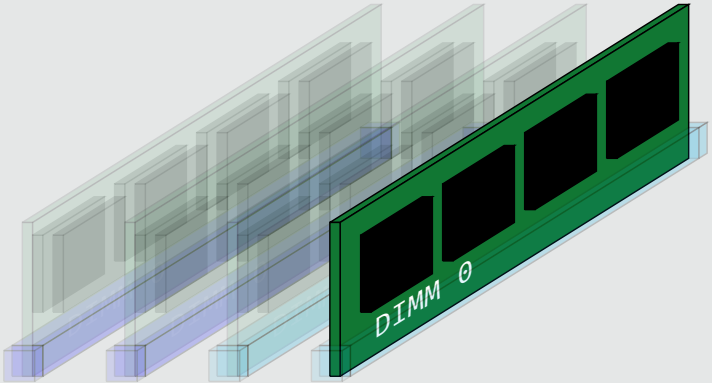
Physical DRAM architecture

Channel



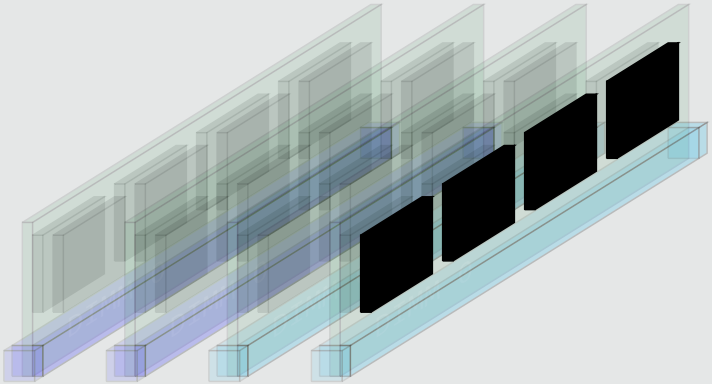
Physical DRAM architecture

DIMM



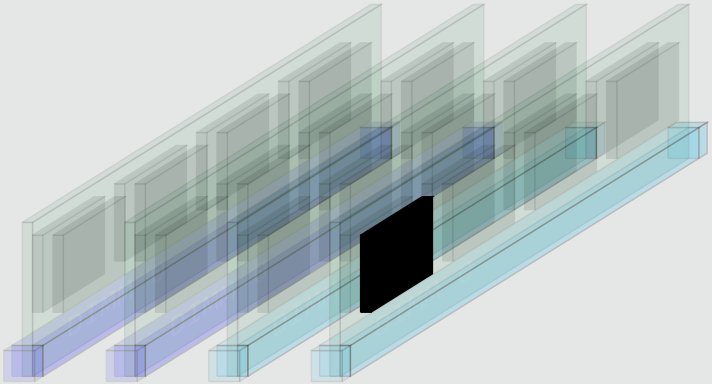
Physical DRAM architecture

Rank



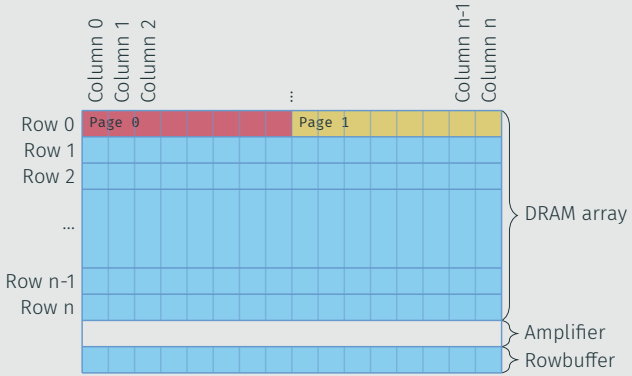
Physical DRAM architecture

Bank



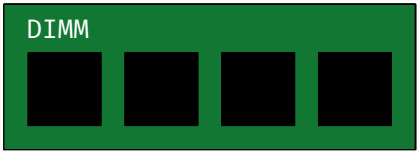
Physical DRAM architecture

Inside a bank



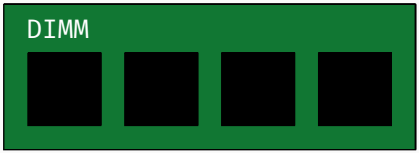
Virtual Memory Management Part 2 — DRAM addressing

- Data is stored in physical memory:
  - Channel
  - DIMM
  - Rank
  - Bank
  - Row
  - Column
- The Memory Management Unit (MMU) translates physical addresses to memory locations



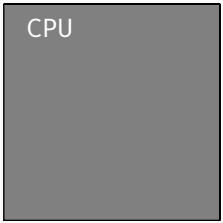
Virtual Memory Management Part 2 — DRAM addressing

- Data is stored in physical memory:
  - Channel
  - DIMM
  - Rank
  - Bank
  - Row
  - Column
- The Memory Management Unit (MMU) translates physical addresses to memory locations



Rowhammer from the perspective of the CPU

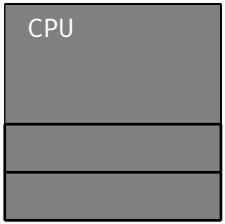
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of the CPU

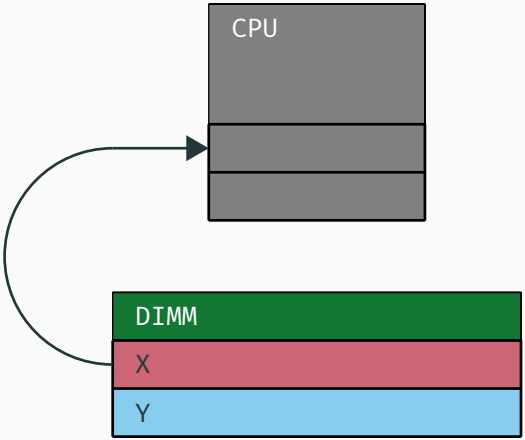
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of the CPU

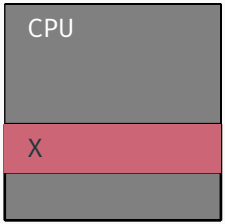
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of the CPU

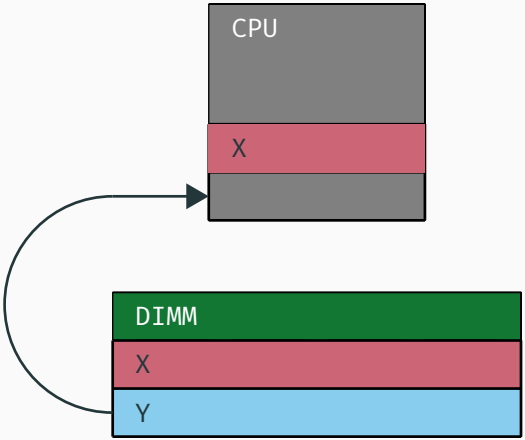
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of the CPU

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```

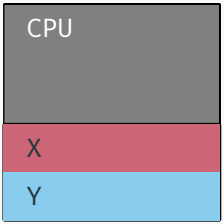


Source code from Wikipedia

Rowhammer from the perspective of the CPU

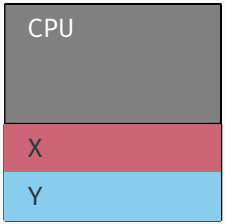
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```

Source code from Wikipedia



Rowhammer from the perspective of the CPU

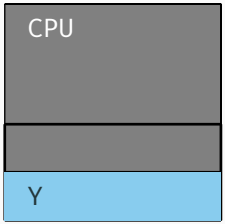
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of the CPU

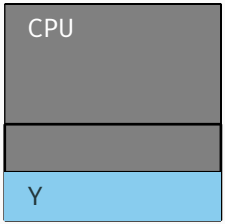
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of the CPU

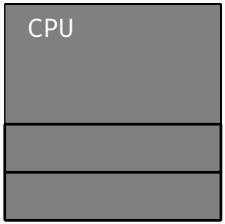
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of the CPU

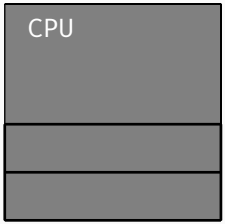
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of the CPU

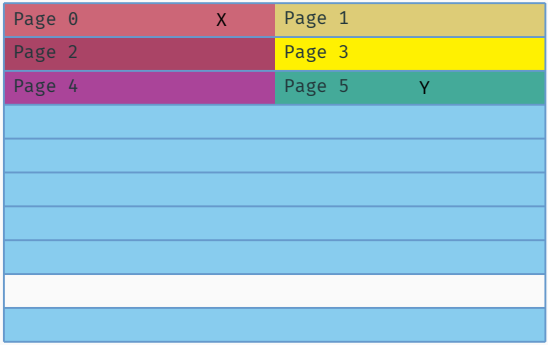
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

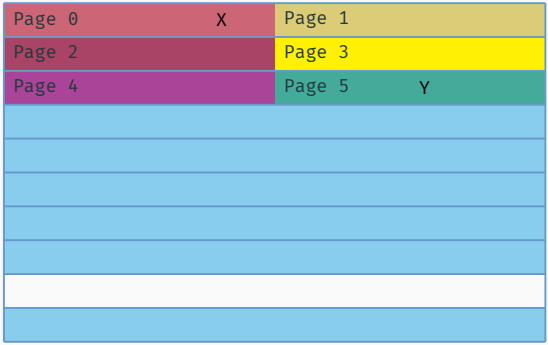
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

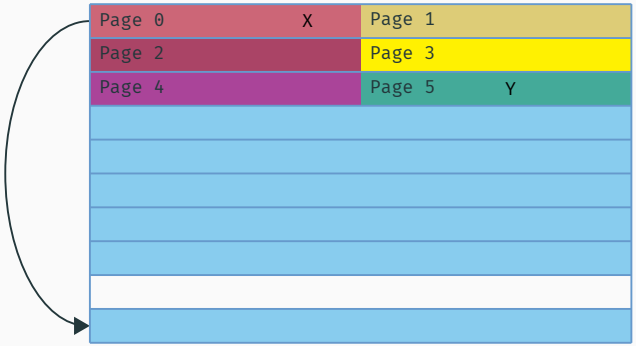
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```

Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```

Source code from Wikipedia

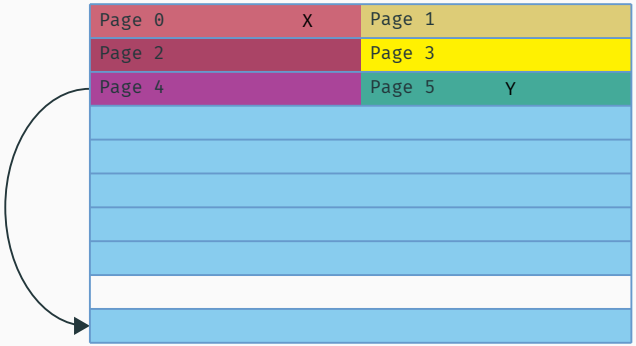
Rowhammer from the perspective of a DRAM bank

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```

Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

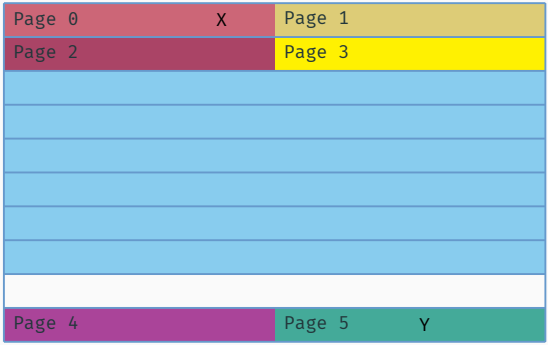
```
1   hammer:
2     mov eax, X
3     mov ebx, Y
4     clflush X
5     clflush Y
6     jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

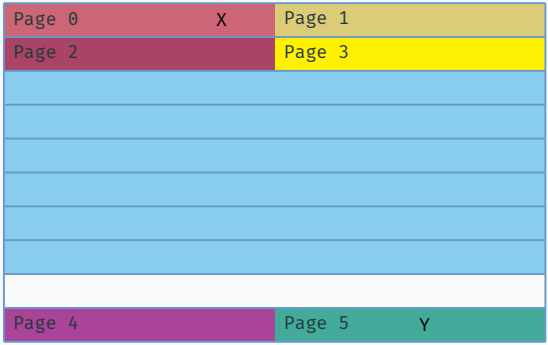
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

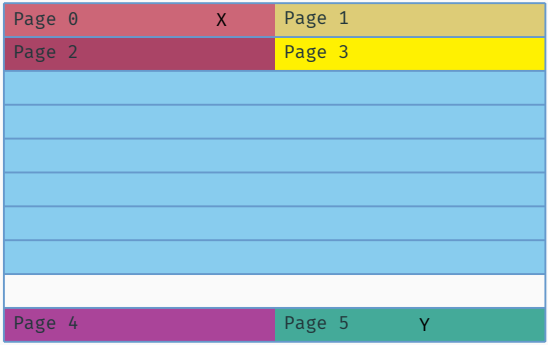
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

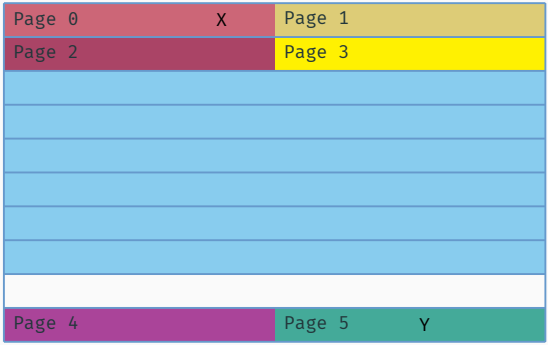
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

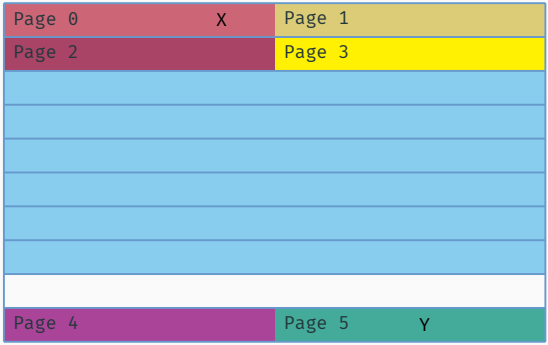
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

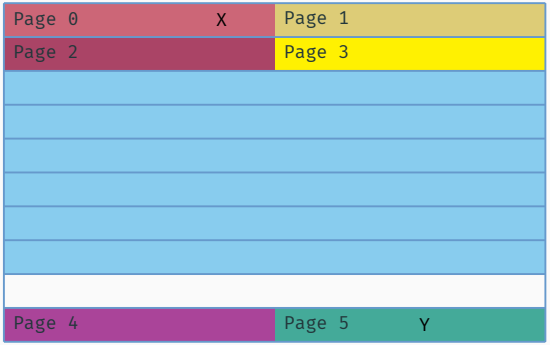
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

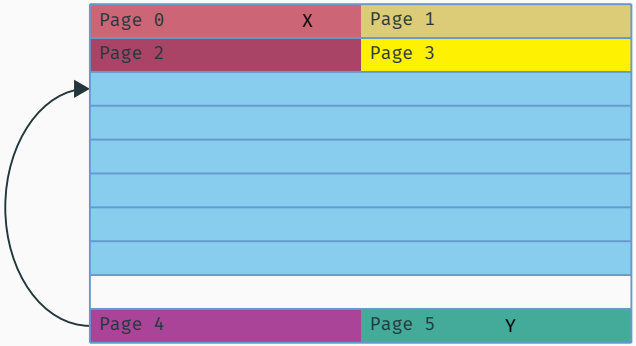
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

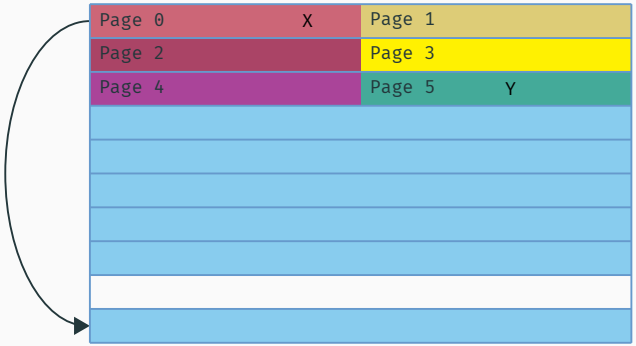
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```

Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```

Source code from Wikipedia

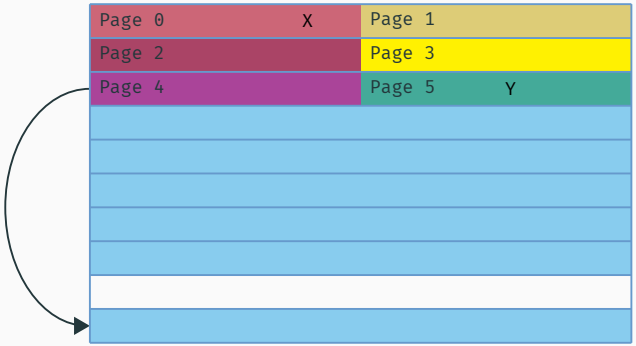
Rowhammer from the perspective of a DRAM bank

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```

Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

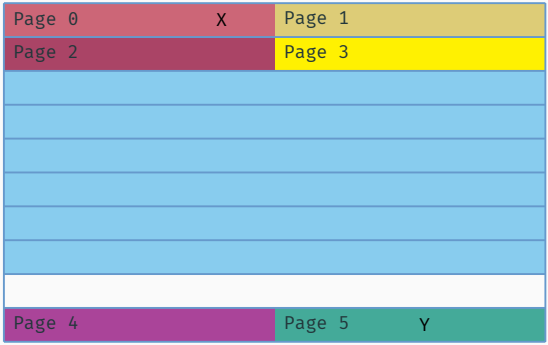
```
1   hammer:
2     mov eax, X
3     mov ebx, Y
4     clflush X
5     clflush Y
6     jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

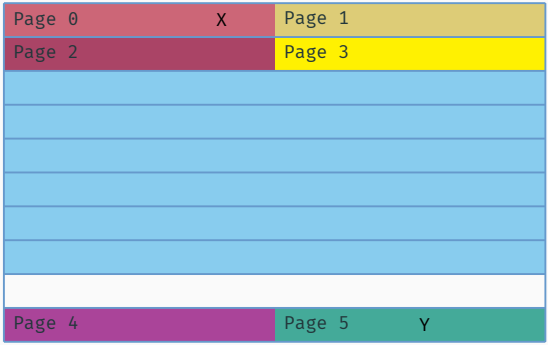
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

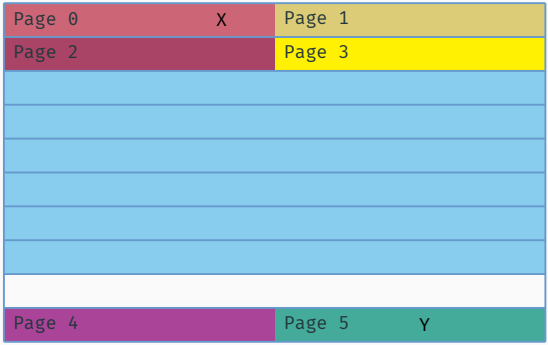
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

Rowhammer from the perspective of a DRAM bank

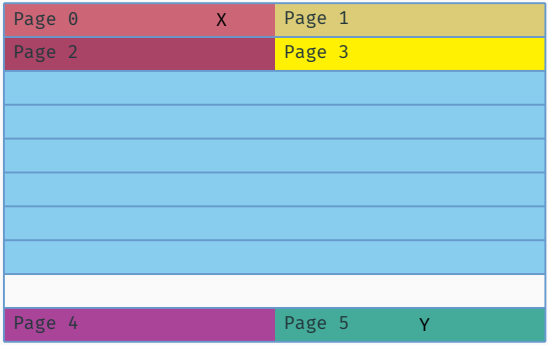
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

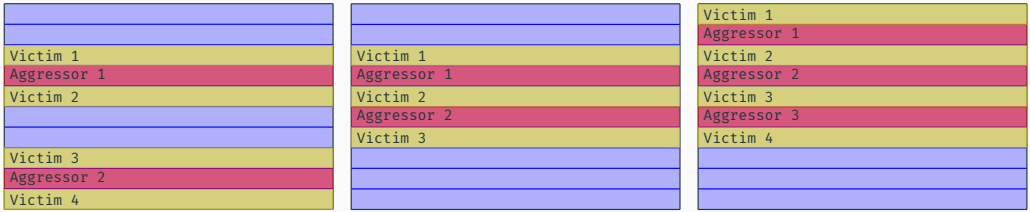
Rowhammer from the perspective of a DRAM bank

```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Source code from Wikipedia

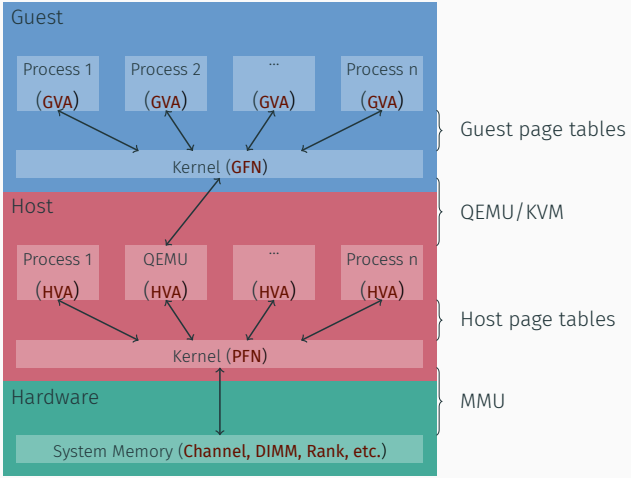
Rowhammer from the perspective of a DRAM bank (abstract)



(a) Single Side (b) Double Side (c) Three Side

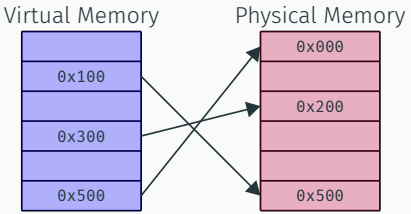
Figure 1: Examples of rowhammer patterns

Virtual Memory Management Part 3 — Virtual environments



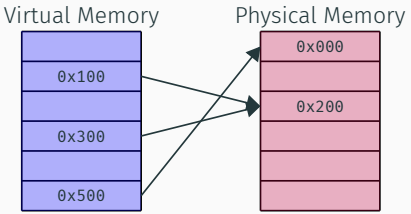
Kernel Samepage Merging (KSM)

- Mechanism in the Linux Kernel
- Deduplicates pages that are marked as merge candidates:
  - Multiple virtual pages with same content are stored once in physical memory
  - Page tables of all pages are adjusted so they reference to the same physical page
  - Copy On Write (COW) policy: Writing to a page copies the page, copy is accessed



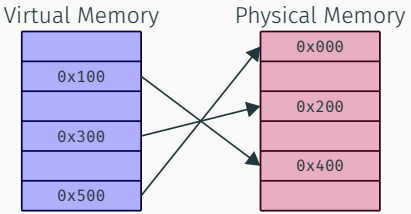
Kernel Samepage Merging (KSM)

- Mechanism in the Linux Kernel
- Deduplicates pages that are marked as merge candidates:
  - Multiple virtual pages with same content are stored once in physical memory
  - Page tables of all pages are adjusted so they reference to the same physical page
  - Copy On Write (COW) policy: Writing to a page copies the page, copy is accessed



Kernel Samepage Merging (KSM)

- Mechanism in the Linux Kernel
- Deduplicates pages that are marked as merge candidates:
  - Multiple virtual pages with same content are stored once in physical memory
  - Page tables of all pages are adjusted so they reference to the same physical page
  - Copy On Write (COW) policy: Writing to a page copies the page, copy is accessed



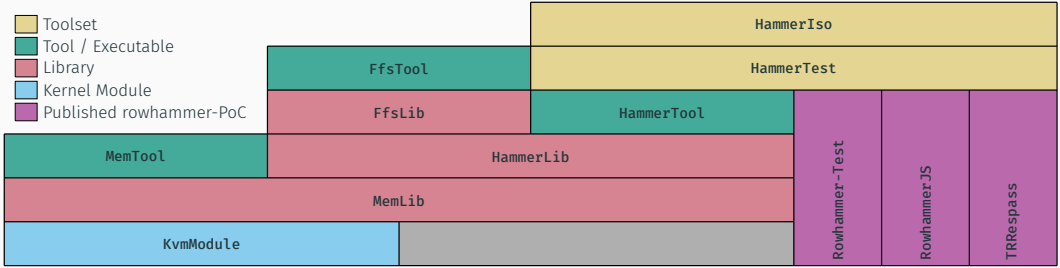
Flip Feng Shui (FFS)

- KSM merges pages across processes
- By default, QEMU/KVM marks all pages as KSM merge candidates
- When KSM is running, pages across VMs are deduplicated
- Idea from Razavi et al. [4]:
  - Select a *target page*
  - Chose a bit that should flip inside the target page
  - Find a location in memory that is vulnerable to rowhammer where the specified bit flips
  - Get this target page merged at a location found before
  - Execute the rowhammer attack at that location
  - The bit should be flipped at all pages that were merged to that location

Toolset `Hammertinger`

What is Hammertinger?

- All-in-One rowhammer and FFS exploitation framework



Practical rowhammer exploitation

- At first, no bit flips were found on the tested systems
- Reason: The systems had a BIOS version with a mitigation for rowhammer
- Idea: Downgrade BIOS to a version before the mitigation
- Problem: Downgrade locks (unable to perform the downgrade)

- At fi
- Rea
- Idea
- Prol

...mmer

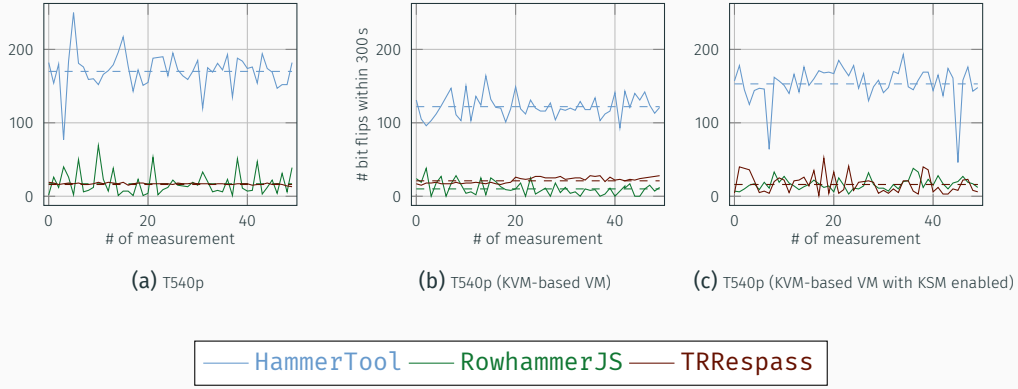
Solution

Manually downgrade the BIOS

Demo 1

Exploiting rowhammer with `HammerTool`

Evaluation of HammerTool



(a) T540p (b) T540p (KVM-based VM) (c) T540p (KVM-based VM with KSM enabled)

Figure 2: Number of bit flips found by different PoCs in 300 s

Amplification of rowhammer

Increase the amount of bit flips with KSM

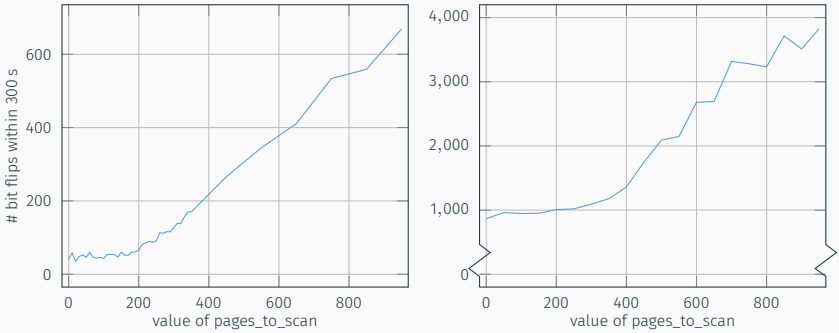


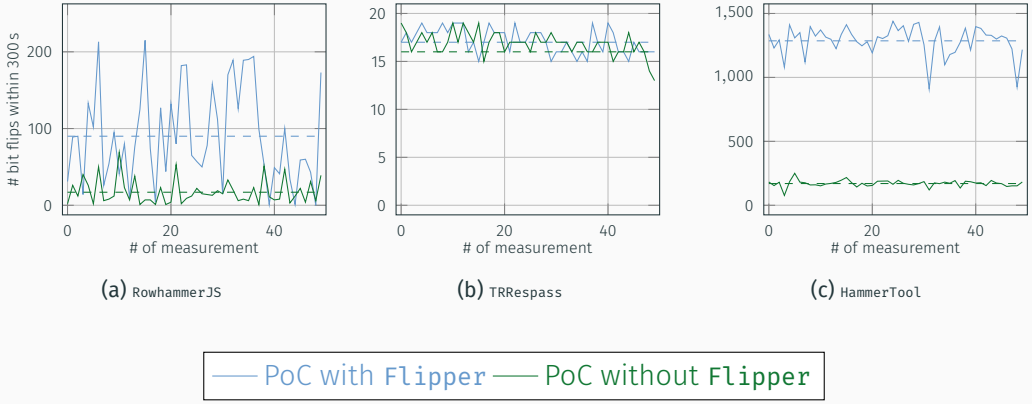
Figure 3: Number of bit flips found by `HammerTool` in dependence of `pages_to_scan`

Increase the amount of bit flips with KSM

```
1  int memcmp(const void *s1, const void *s2, size_t len)
2  {
3     bool diff;
4     asm("repe; cmpsb" CC_SET(nz)
5        : CC_OUT(nz) (diff), "+D" (s1), "+S" (s2), "+c" (len));
6     return diff;
7  }
```

From the Linux Kernel at arch/x86/boot/string.c

Increase the amount of bit flips with `Flipper`



(a) `RowhammerJS` (b) `TRRespass` (c) `HammerTool`

Figure 4: Number of bit flips found by different PoC in 300 s with and without `Flipper`

Demo 2

Increase the amount of bit flips with `Flipper`

Increase of bit flips with `Flipper` on a mitigated system

Reference	DIMM	without `Flipper`	with `Flipper`	ratio
M0	Samsung M471B5273DH0-CH9	$3.01 \cdot 10^{1}$	$1.8724 \cdot 10^{3}$	62.2
M1	Samsung M471B5273DH0-CH9	$1.384 \cdot 10^{1}$	$8.157 \cdot 10^{2}$	58.93
M2	Samsung M471B5273DH0-CH9	$4.18 \cdot 10^{0}$	$2.5348 \cdot 10^{2}$	60.64
M3	Samsung M471B5273DH0-CH9	$9.2 \cdot 10^{-1}$	$3.61 \cdot 10^{1}$	39.23
M4	Samsung M471B5273DH0-CH9	$6.54 \cdot 10^{0}$	$3.7384 \cdot 10^{2}$	57.16
M5	Kingston KVR16S11/4 99U5428-049.A00LF	$0 \cdot 10^{0}$	$0 \cdot 10^{0}$	
M6	Samsung M471B1G73BH0-YK0	$2 \cdot 10^{-2}$	$1.52 \cdot 10^{0}$	76

Table 1: Number of bit flips found by `HammerTool` with and without `Flipper`

Conclusion

Conclusion

- An attacker can flip bits in DRAM by accessing other memory locations
- There are sophisticated exploits using rowhammer
- Mitigation strategies do often not prevent rowhammer but make it unlikely
- `HammerTool` and `Flipper` increase the amount of bit flips found on a system siginificantly which can help bypassing the mitigation on DDR3 systems (maybe, DDR4 as well)

Conclusion

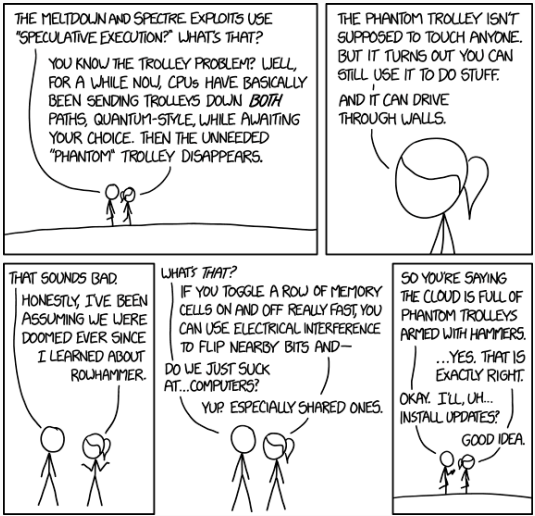


Image from XKCD

Questions?

References i

[1] Pietro Frigo et al. “TRRespass: Exploiting the Many Sides of Target Row Refresh”. In: *S&P*. Best Paper Award. May 2020. URL: https://download.vusec.net/papers/trrespass_sp20.pdf.

[2] Daniel Gruss, Clémentine Maurice, and Stefan Mangard. “Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript”. In: *CoRR* abs/1507.06955 (2015). arXiv: 1507.06955. URL: http://arxiv.org/abs/1507.06955.

References ii

[3] Yoongu Kim et al. “Flipping Bits in Memory without Accessing Them: An Experimental Study of DRAM Disturbance Errors”. In: *SIGARCH Comput. Archit. News* 42.3 (June 2014), pp. 361–372. ISSN: 0163-5964. DOI: 10.1145/2678373.2665726. URL: https://doi.org/10.1145/2678373.2665726.

[4] Kaveh Razavi et al. “Flip Feng Shui: Hammering a Needle in the Software Stack”. In: *USENIX Security*. June 2016. URL: https://download.vusec.net/papers/flip-feng-shui_sec16.pdf.

References iii

[5] Mark Seaborn and Thomas Dullien. *Exploiting the DRAM rowhammer bug to gain kernel privileges*. 2015. URL: https://www.cs.umd.edu/class/fall2019/cmsc818O/papers/rowhammer-kernel.pdf (visited on 11/16/2020).

[6] Andrei Tatar et al. "Throwhammer: Rowhammer Attacks over the Network and Defenses". In: *USENIX ATC*. Pwnie Award Nomination for Most Innovative Research. July 2018. URL: https://download.vusec.net/papers/throwhammer_atc18.pdf.

References iv

[7] Victor van der Veen et al. “Drammer: Deterministic Rowhammer Attacks on Mobile Platforms”. In: *CCS*. Pwnie Award for Best Privilege Escalation Bug, Android Security Reward, CSAW Best Paper Award, DCSR Paper Award. Oct. 2016. URL: https://vvdveen.com/publications/drammer.pdf.