Nethammer

Inducing Rowhammer Faults through Network Requests

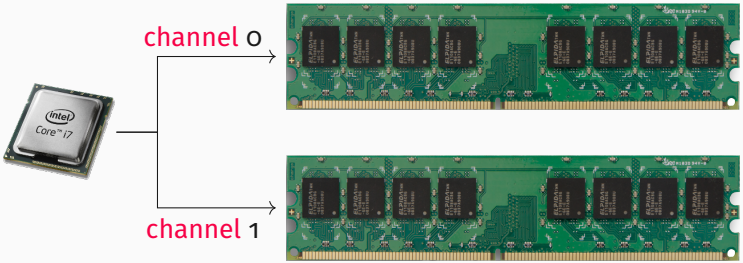
Moritz Lipp[1], Michael Schwarz[1], Lukas Raab[1], Lukas Lamster[1], Misiker Tadesse Aga[2], Clémentine Maurice[3], Daniel Gruss[1]

SILM Workshop 2020

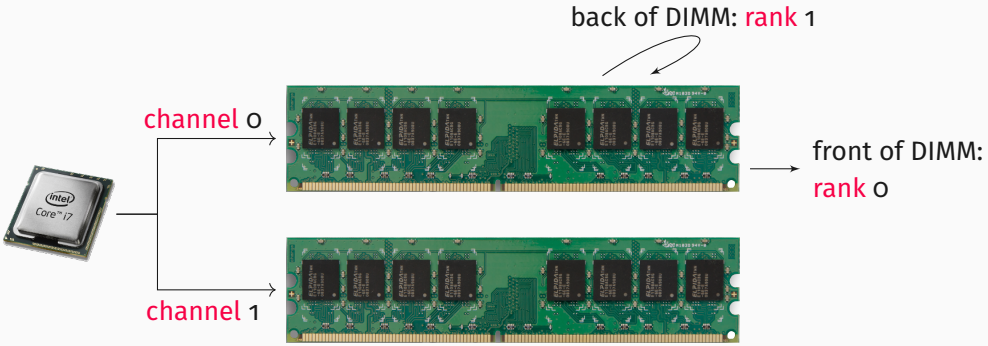
[1]Graz University of Technology, [2] University of Michigan, [3]Univ Rennes, CNRS, IRISA

DRAM organization

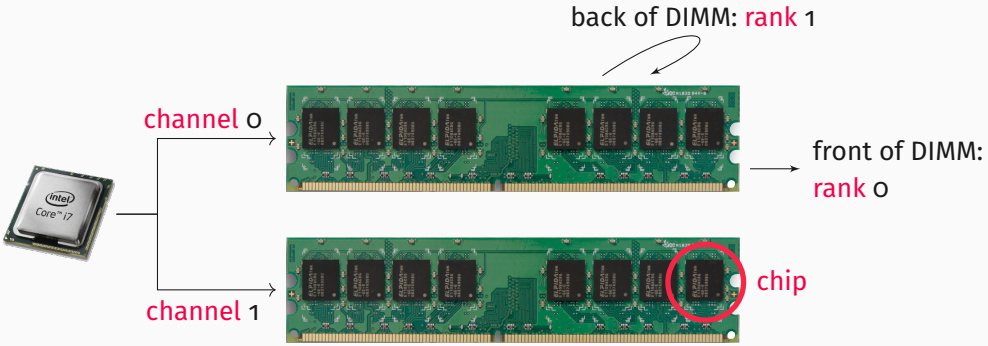
DRAM organization



DRAM organization

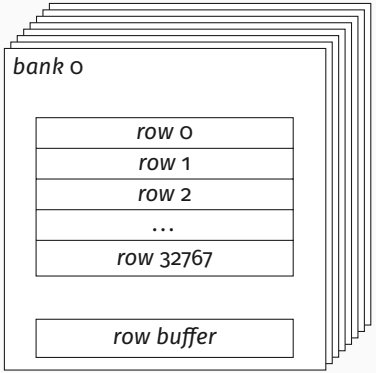


DRAM organization



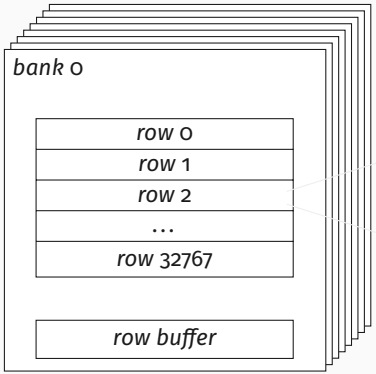
DRAM organization

chip



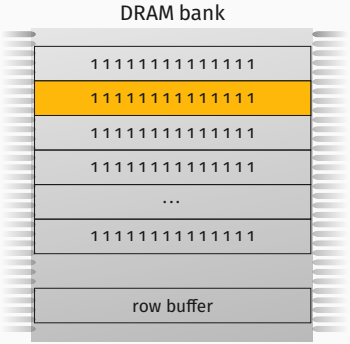
DRAM organization

chip



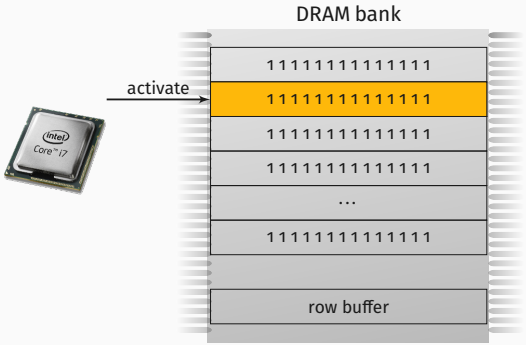
64k cells
1 capacitor,
1 transitor each

How reading from DRAM works



CPU wants to access row 1

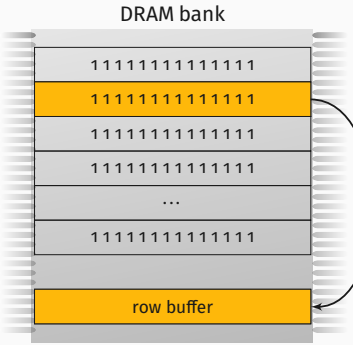
How reading from DRAM works



CPU wants to access row 1

$\rightarrow$ row 1 activated

How reading from DRAM works

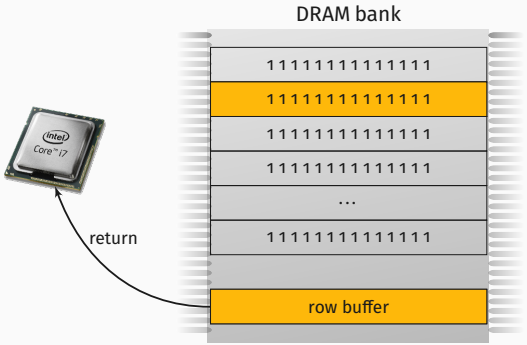


CPU wants to access row 1

$\rightarrow$ row 1 activated

$\rightarrow$ row 1 copied to row buffer

How reading from DRAM works

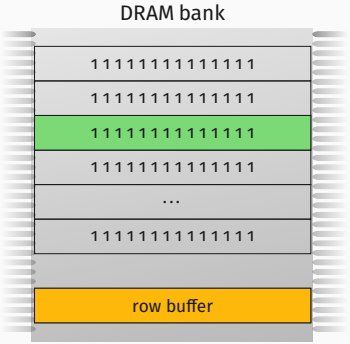


CPU wants to access row 1

$\rightarrow$ row 1 activated

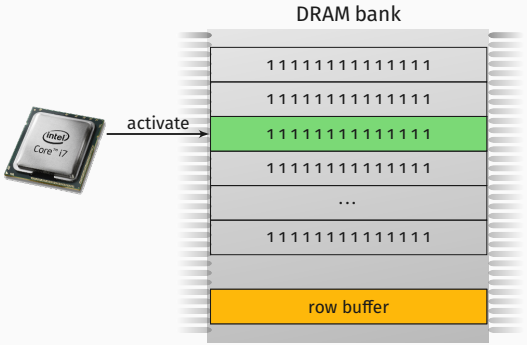
$\rightarrow$ row 1 copied to row buffer

How reading from DRAM works



CPU wants to access row 2

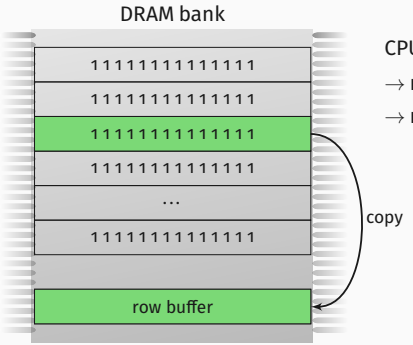
How reading from DRAM works



CPU wants to access row 2

$\rightarrow$ row 2 activated

How reading from DRAM works

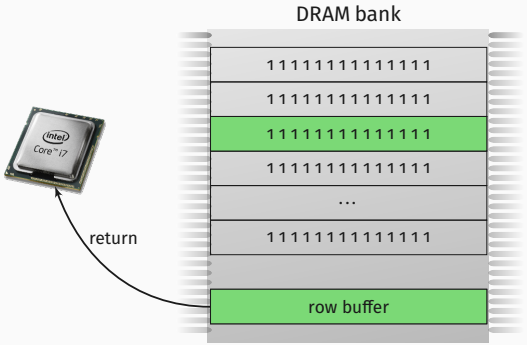


CPU wants to access row 2

$\rightarrow$ row 2 activated

$\rightarrow$ row 2 copied to row buffer

How reading from DRAM works

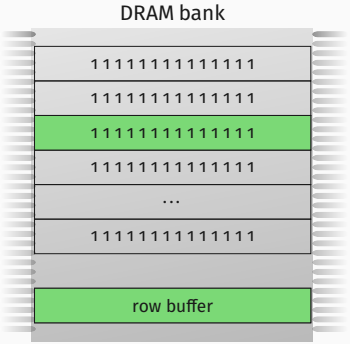


CPU wants to access row 2

$\rightarrow$ row 2 activated

$\rightarrow$ row 2 copied to row buffer

How reading from DRAM works



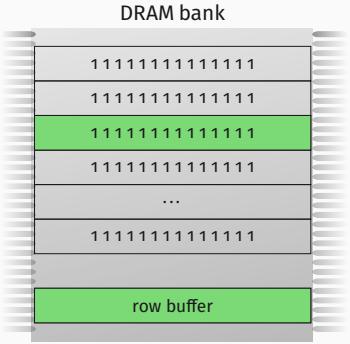
CPU wants to access row 2

→ row 2 activated

→ row 2 copied to row buffer

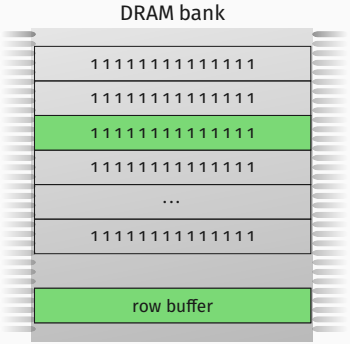
→ slow (row conflict)

How reading from DRAM works



CPU wants to access row 2—again

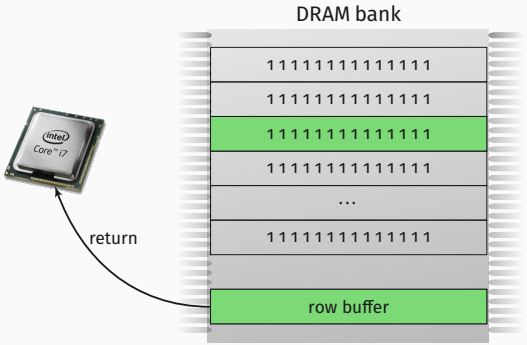
How reading from DRAM works



CPU wants to access row 2—again

$\rightarrow$ row 2 already in row buffer

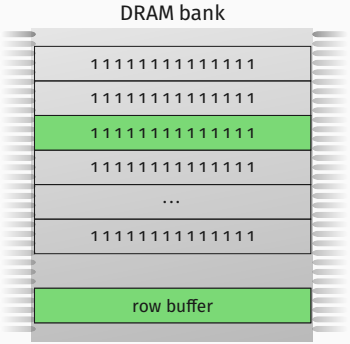
How reading from DRAM works



CPU wants to access row 2—again

$\rightarrow$ row 2 already in row buffer

How reading from DRAM works

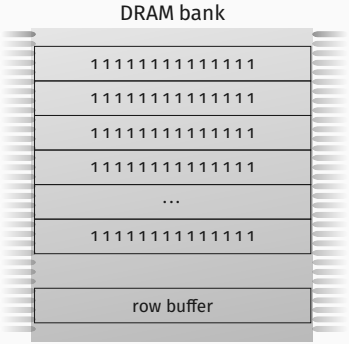


CPU wants to access row 2—again

$\rightarrow$ row 2 already in row buffer

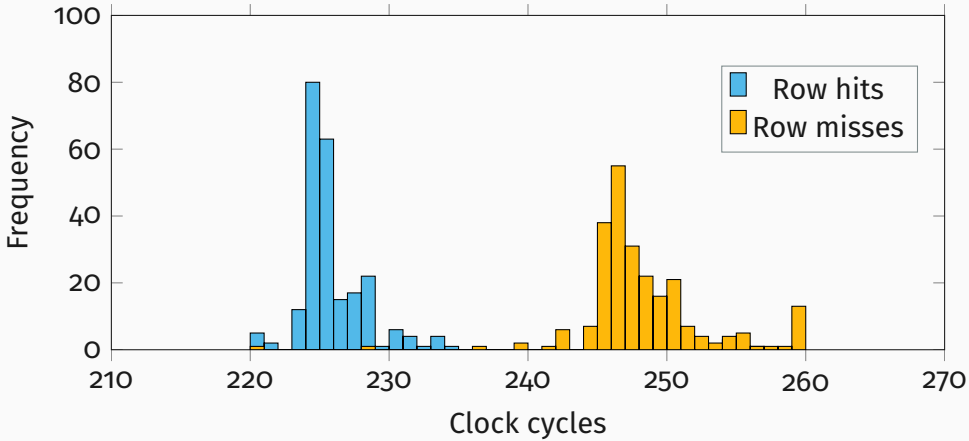
$\rightarrow$ fast (row hit)

How reading from DRAM works

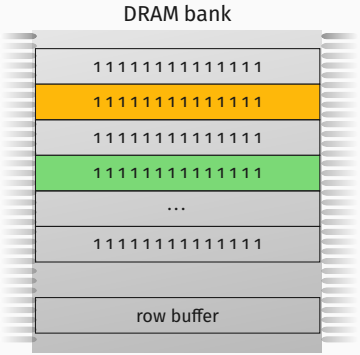


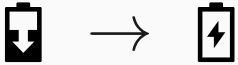
row buffer = cache

Timing difference

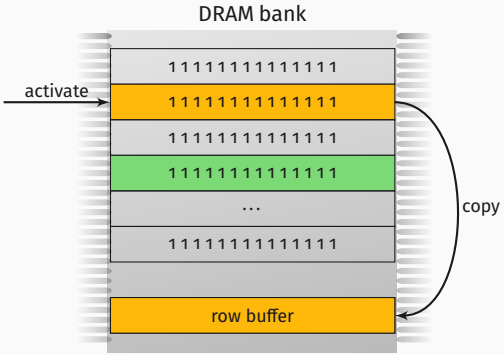


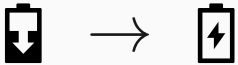
Rowhammer


DRAM bank
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
…
1 1 1 1 1 1 1 1 1 1 1 1 1 1
row buffer

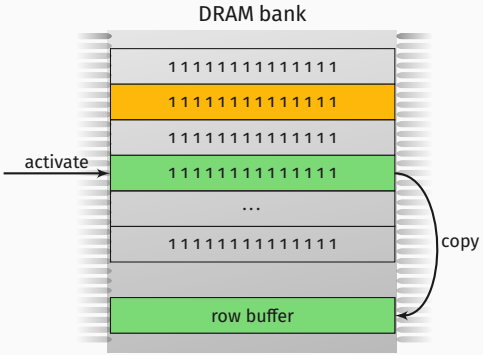


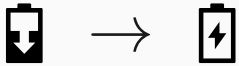
Rowhammer


DRAM bank
activate
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1
…
1 1 1 1 1 1 1 1 1 1 1 1 1 1
copy
row buffer



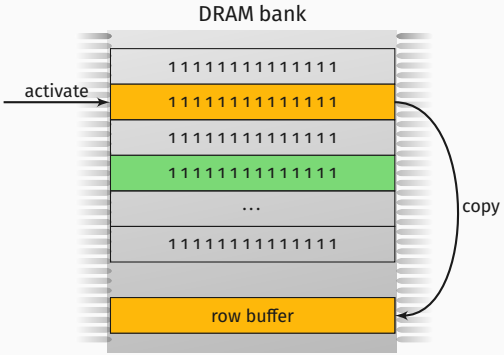
Rowhammer

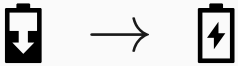




Cells leak faster upon proximate accesses $\rightarrow$ Rowhammer

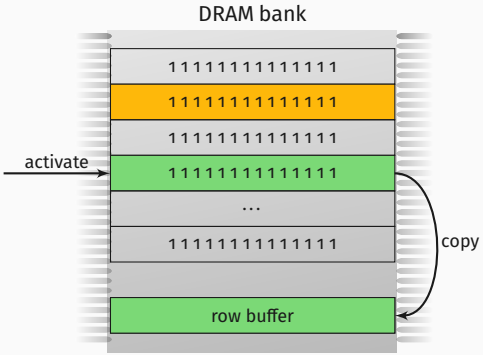
Rowhammer

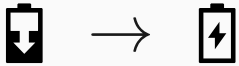




Cells leak faster upon proximate accesses $\rightarrow$ Rowhammer

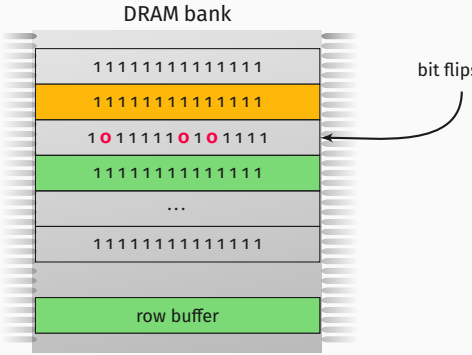
Rowhammer

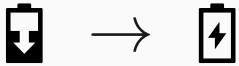




Cells leak faster upon proximate accesses $\rightarrow$ Rowhammer

Rowhammer



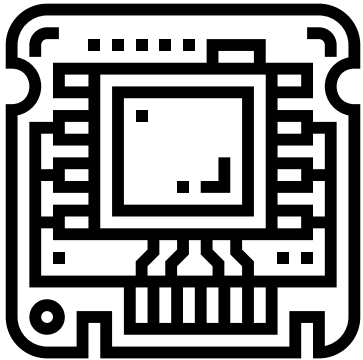


Cells leak faster upon proximate accesses $\rightarrow$ Rowhammer

Requirements

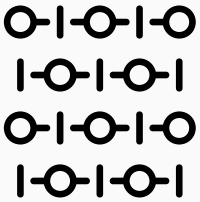
Memory accesses must be

- uncached: reach DRAM
- fast: race against the next row refresh
- targeted: reach specific row



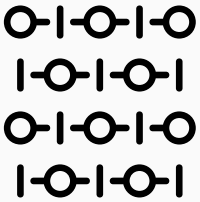
Memory-Controller Policies

Memory-Controller Policies



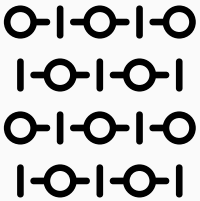
- **Open-page policy**: Keep row opened and buffered

Memory-Controller Policies



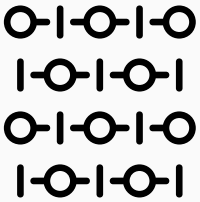
- **Open-page policy**: Keep row opened and buffered
  - Low latency for subsequent accesses to same row
  - High latency for accesses to any other row

Memory-Controller Policies



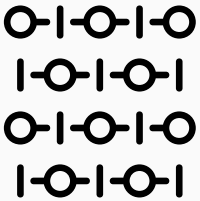
- **Open-page policy**: Keep row opened and buffered
  - Low latency for subsequent accesses to same row
  - High latency for accesses to any other row
- **Close-page policy**: Immediately close row, ready to open a new row

Memory-Controller Policies



- **Open-page policy**: Keep row opened and buffered
  - Low latency for subsequent accesses to same row
  - High latency for accesses to any other row
- **Close-page policy**: Immediately close row, ready to open a new row
  - Medium latency for accesses to any row

Memory-Controller Policies



- **Open-page policy**: Keep row opened and buffered
  - Low latency for subsequent accesses to same row
  - High latency for accesses to any other row
- **Close-page policy**: Immediately close row, ready to open a new row
  - Medium latency for accesses to any row
  - Perform better on multi-core systems [Dav+11]
- **Adaptive-page policy**: Change policy used

Memory-Controller Policies

- Policies that preemptively close rows, would allow one-location hammering

Memory-Controller Policies

- Policies that preemptively close rows, would allow one-location hammering
- We observed close-page policies on desktop computers

Memory-Controller Policies

- Policies that preemptively close rows, would allow one-location hammering
- We observed close-page policies on desktop computers
- Mobile devices (e.g., laptops) seem to use mostly open-page policies

Policies

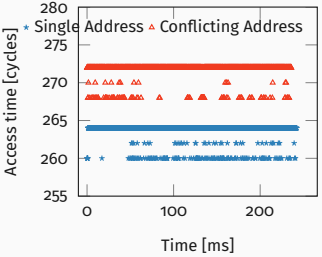


Figure 1: Open

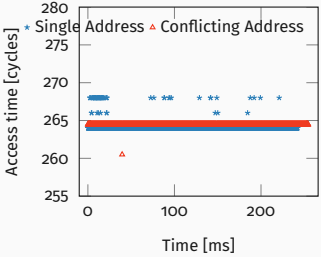


Figure 2: Closed

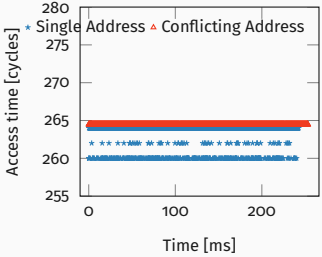
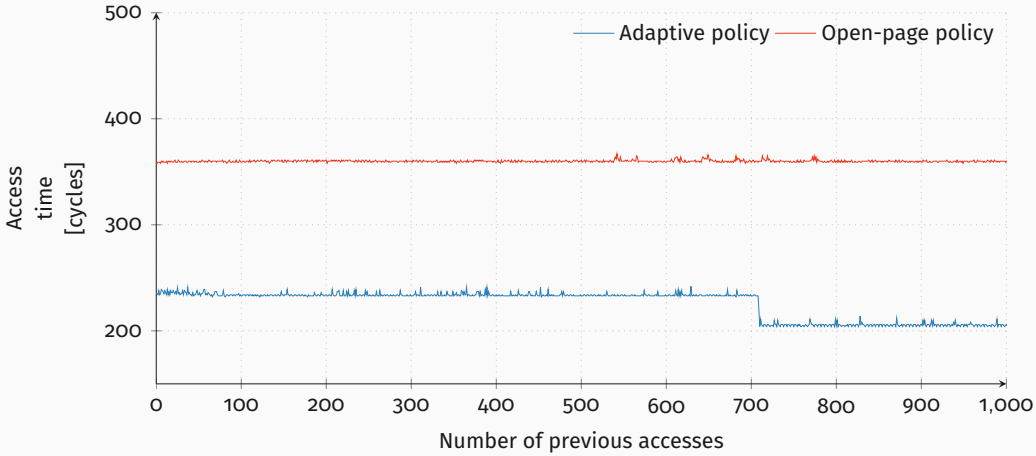
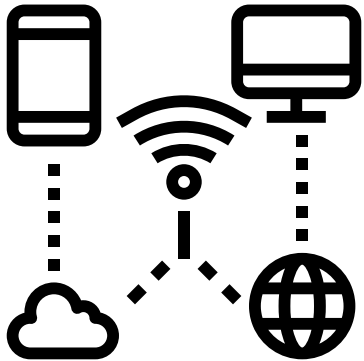


Figure 3: Adaptive

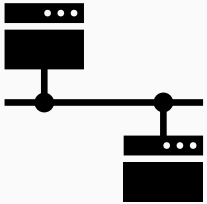
Automated Classification of Memory-Controller Page Policies





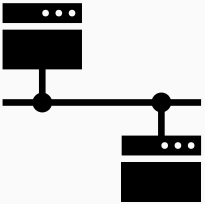
Rowhammer over the Network

Nethammer



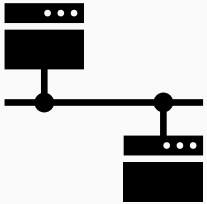
- Rowhammer always required local code execution

Nethammer



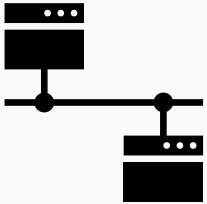
- Rowhammer always required local code execution
- Rowhammer.js required the victim to visit a website

Nethammer



- Rowhammer always required local code execution
- Rowhammer.js required the victim to visit a website
- *Is Rowhammer possible without any attracker-controlled code?*

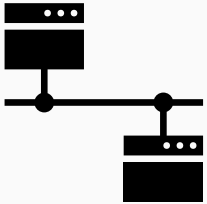
Nethammer



Inducing bit flips:

- Network stacks on ARM often use uncached memory

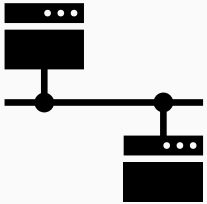
Nethammer



Inducing bit flips:

- Network stacks on ARM often use uncached memory
- Intel recommends Intel CAT for QoS

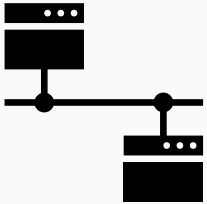
Nethammer



Inducing bit flips:

- Network stacks on ARM often use uncached memory
- Intel recommends Intel CAT for QoS
- Network reachable code might use `clflush` or non-temporal stores

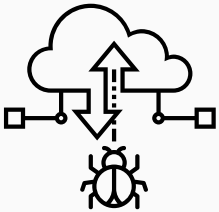
Nethammer



Inducing bit flips:

- Network stacks on ARM often use uncached memory
- Intel recommends Intel CAT for QoS
- Network reachable code might use `clflush` or non-temporal stores
- RDMA-enabled Networks (Throwhammer [Tat+18])

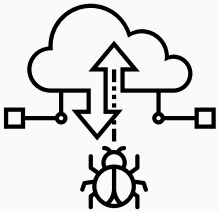
Exploiting Nethammer Bit Flips



Nethammer on …

- **SGX**: Powerful DoS

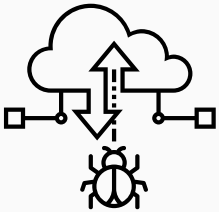
Exploiting Nethammer Bit Flips



Nethammer on ...

- **SGX**: Powerful DoS
- **File system**: Persistent DoS

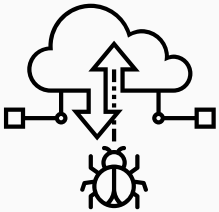
Exploiting Nethammer Bit Flips



Nethammer on ...

- **SGX**: Powerful DoS
- **File system**: Persistent DoS
- **DNS entries on a DNS server**: Bit-squatting attack

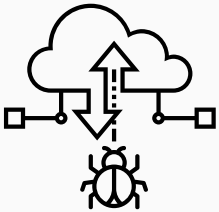
Exploiting Nethammer Bit Flips



Nethammer on ...

- **SGX**: Powerful DoS
- **File system**: Persistent DoS
- **DNS entries on a DNS server**: Bit-squatting attack
- **OCSP servers**: Make invalid certificates great again

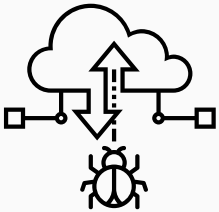
Exploiting Nethammer Bit Flips



Nethammer on ...

- **SGX**: Powerful DoS
- **File system**: Persistent DoS
- **DNS entries on a DNS server**: Bit-squatting attack
- **OCSP servers**: Make invalid certificates great again
- **Crypto**: Generate private keys for broken public keys

Exploiting Nethammer Bit Flips



Nethammer on ...

- **SGX**: Powerful DoS
- **File system**: Persistent DoS
- **DNS entries on a DNS server**: Bit-squatting attack
- **OCSP servers**: Make invalid certificates great again
- **Crypto**: Generate private keys for broken public keys
- **Crypto + GitLab**: Manipulate repositories in the name of someone else

Setup

- Different CPUs (Desktop, Server, Mobile) with different configurations

Setup

- Different CPUs (Desktop, Server, Mobile) with different configurations
  - Different Memory-Controller Page Policy

Setup

- Different CPUs (Desktop, Server, Mobile) with different configurations
  - Different Memory-Controller Page Policy
  - Intel CAT, `clflush` in a driver or userspace application

Setup

- Different CPUs (Desktop, Server, Mobile) with different configurations
  - Different Memory-Controller Page Policy
  - Intel CAT, `clflush` in a driver or userspace application
- Connected in a LAN

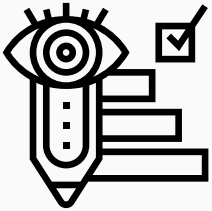
Setup

- Different CPUs (Desktop, Server, Mobile) with different configurations
  - Different Memory-Controller Page Policy
  - Intel CAT, `clflush` in a driver or userspace application
- Connected in a LAN
- Flooded with UDP packets

Setup

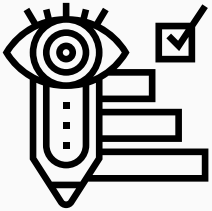
- Different CPUs (Desktop, Server, Mobile) with different configurations
  - Different Memory-Controller Page Policy
  - Intel CAT, `clflush` in a driver or userspace application
- Connected in a LAN
- Flooded with UDP packets
- Monitoring tool that checks for bitflips periodically

Evaluation



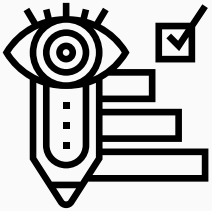
- Observed bit flips with different policies used …

Evaluation



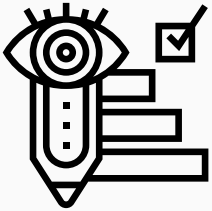
- Observed bit flips with different policies used …
  - *Closed*: 11 after 4 hours

Evaluation



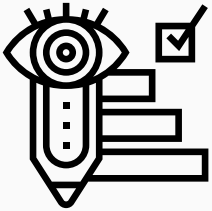
- Observed bit flips with different policies used …
  - *Closed*: 11 after 4 hours
  - *Open*: 0 after 4 hours (46 after 10 hours)

Evaluation



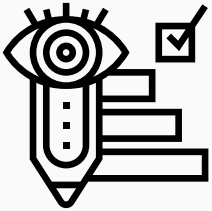
- Observed bit flips with different policies used …
  - *Closed*: 11 after 4 hours
  - *Open*: 0 after 4 hours (46 after 10 hours)
  - *Adaptive*: 10 after 4 hours

Evaluation



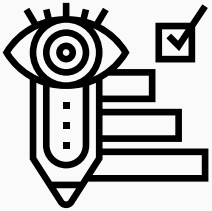
- Observed bit flips with different policies used …
  - *Closed*: 11 after 4 hours
  - *Open*: 0 after 4 hours (46 after 10 hours)
  - *Adaptive*: 10 after 4 hours
- Did not observe bit flips on the ARM device remotely

Evaluation



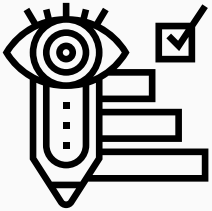
- Observed bit flips with different policies used …
  - *Closed*: 11 after 4 hours
  - *Open*: 0 after 4 hours (46 after 10 hours)
  - *Adaptive*: 10 after 4 hours
- Did not observe bit flips on the ARM device remotely
  - Demonstrated One-Location Hammering on ARM locally

Evaluation



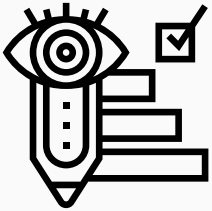
- We observed bit flips …
  - Causing the system not to boot anymore

Evaluation



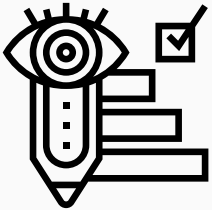
- We observed bit flips …
  - Causing the system not to boot anymore
  - Halting the system (flips in drivers)

Evaluation

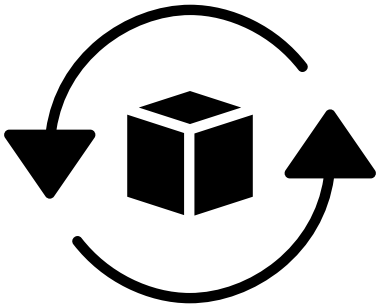


- We observed bit flips …
  - Causing the system not to boot anymore
  - Halting the system (flips in drivers)
  - Bit flips in SGX EPC region (memory controller lock down)

Evaluation

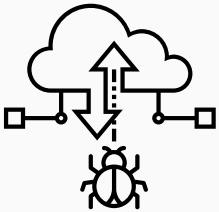


- We observed bit flips …
  - Causing the system not to boot anymore
  - Halting the system (flips in drivers)
  - Bit flips in SGX EPC region (memory controller lock down)
  - Crashing applications



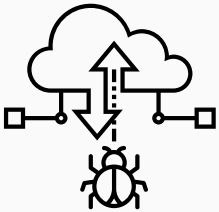
Target Row Refresh (TRR)

TRR



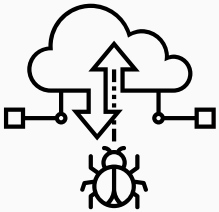
- **Idea:** Refresh adjacent rows if targeted row is accessed with a high frequency

TRR



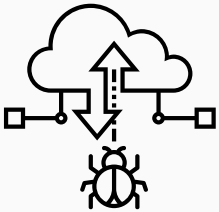
- **Idea:** Refresh adjacent rows if targeted row is accessed with a high frequency
- We were able to induce bit flips …

TRR



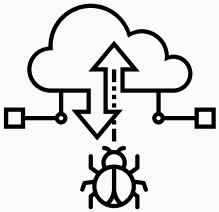
- **Idea:** Refresh adjacent rows if targeted row is accessed with a high frequency
- We were able to induce bit flips …
  - i7-6700K: Adjacent rows and further away (TRR not active)

TRR



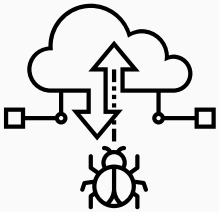
- **Idea:** Refresh adjacent rows if targeted row is accessed with a high frequency
- We were able to induce bit flips …
  - i7-6700K: Adjacent rows and further away (TRR not active)
  - E5-1630 v4: Only further away (TRR active)

TRR



- **Idea:** Refresh adjacent rows if targeted row is accessed with a high frequency
- We were able to induce bit flips …
  - i7-6700K: Adjacent rows and further away (TRR not active)
  - E5-1630 v4: Only further away (TRR active)
- TRR insufficient in mitigating Rowhammer attacks

TRR



- **Idea:** Refresh adjacent rows if targeted row is accessed with a high frequency
- We were able to induce bit flips …
  - i7-6700K: Adjacent rows and further away (TRR not active)
  - E5-1630 v4: Only further away (TRR active)
- TRR insufficient in mitigating Rowhammer attacks
- Frigo et al. [Fri+20] analyzed TRR in more depth

Takeaways

- Remote Rowhammer attack

Takeaways

- Remote Rowhammer attack
- Automatically identify page policies used by the memory controller

Takeaways

- Remote Rowhammer attack
- Automatically identify page policies used by the memory controller
- One-location hammering on ARM

Takeaways

- Remote Rowhammer attack
- Automatically identify page policies used by the memory controller
- One-location hammering on ARM
- TRR is insufficient in mitigating Rowhammer attacks

Nethammer

Inducing Rowhammer Faults through Network Requests

Moritz Lipp[1], Michael Schwarz[1], Lukas Raab[1], Lukas Lamster[1], Misiker Tadesse Aga[2], Clémentine Maurice[3], Daniel Gruss[1]

SILM Workshop 2020

[1]Graz University of Technology, [2] University of Michigan, [3]Univ Rennes, CNRS, IRISA

References i

Howard David et al. Memory power management via dynamic voltage/frequency scaling. In: ACM International Conference on Autonomic Computing. 2011.

Pietro Frigo et al. TRRespass: Exploiting the Many Sides of Target Row Refresh. In: S&P. 2020.

Andrei Tatar et al. Throwhammer: Rowhammer Attacks over the Network and Defenses. In: USENIX ATC. 2018.