= Row hammer =

Row hammer (also written as rowhammer) is an unintended side effect in dynamic random @-@ access memory (DRAM) that causes memory cells to leak their charges and interact electrically between themselves , possibly altering the contents of nearby memory rows that were not addressed in the original memory access . This circumvention of the isolation between DRAM memory cells results from the high cell density in modern DRAM , and can be triggered by specially crafted memory access patterns that rapidly activate the same memory rows numerous times .

The row hammer effect has been used in some privilege escalation computer security exploits . Different hardware @-@ based techniques exist to prevent the row hammer effect from occurring , including required support in some processors and types of DRAM memory modules .

= = Background = =

In dynamic RAM (DRAM) , each bit of stored data occupies a separate memory cell that is electrically implemented with one capacitor and one transistor . The charge state of a capacitor (charged or discharged) is what determines whether a DRAM cell stores " 1 " or " 0 " as a binary value . Huge numbers of DRAM memory cells are packed into integrated circuits , together with some additional logic that organizes the cells for the purposes of reading , writing and refreshing the data .

Memory cells (blue squares in the illustration provided in this section) are further organized into matrices and addressed through rows and columns . A memory address applied to a matrix is broken into the row address and column address , which are processed by the row and column address decoders (in the illustration , vertical and horizontal green rectangles , respectively) . After a row address selects the row for a read operation (the selection is also known as row activation) , bits from all cells in the row are transferred into the sense amplifiers that form the row buffer (red squares in the illustration) , from which the exact bit is selected using the column address . Consequently , read operations are of a destructive nature because the design of DRAM requires memory cells to be rewritten after their values have been read by transferring the cell charges into the row buffer . Write operations decode the addresses in a similar way , but as a result of the design entire rows must be rewritten for the value of a single bit to be changed .

As a result of storing data bits using capacitors that have a natural discharge rate , DRAM memory cells lose their state over time and require periodic rewriting of all memory cells , which is a process known as refreshing . As another result of the design , DRAM memory is susceptible to random changes in stored data , which are known as soft memory errors and attributed to cosmic rays and other causes . There are different techniques that counteract soft memory errors and improve the reliability of DRAM , of which error @-@ correcting code (ECC) memory and its advanced variants (such as lockstep memory) are most commonly used .

= = Overview = =

Increased densities of DRAM integrated circuits (ICs) have led to physically smaller memory cells capable of storing smaller charges , resulting in lower operational noise margins , increased rates of electromagnetic interactions between memory cells , and greater possibility of data loss . As a result , disturbance errors have been observed , being caused by cells interfering with each other 's operation and manifesting as random changes in the values of bits stored in affected memory cells . The awareness of disturbance errors dates back to the early 1970s and Intel 1103 as the first commercially available DRAM IC ; since then , DRAM manufacturers have employed various mitigation techniques to counteract disturbance errors , such as improving the isolation between cells and performing production testing . However , researchers proved in a 2014 analysis that commercially available DDR3 DRAM chips manufactured in 2012 and 2013 are susceptible to disturbance errors , while using the term row hammer to name the associated side effect that led to observed bit flips .

The opportunity for the row hammer effect to occur in DDR3 memory is primarily attributed to DDR3 's high density of memory cells and the results of associated interactions between the cells , while rapid DRAM row activations have been determined as the primary cause . Frequent row activations cause voltage fluctuations on the associated row selection lines , which have been observed to induce higher @-@ than @-@ natural discharge rates in capacitors belonging to nearby (adjacent , in most cases) memory rows , which are called victim rows ; if the affected memory cells are not refreshed before they lose too much charge , disturbance errors occur . Tests show that a disturbance error may be observed after performing around 139 @,@ 000 subsequent memory row accesses (with cache flushes) , and that up to one memory cell in every 1 @,@ 700 cells may be susceptible . Those tests also show that the rate of disturbance errors is not substantially affected by increased environment temperature , while it depends on the actual contents of DRAM because certain bit patterns result in significantly higher disturbance error rates .

A variant called double @-@ sided hammering involves targeted activations of two DRAM rows surrounding a victim row : in the illustration provided in this section , this variant would be activating both yellow rows with the aim of inducing bit flips in the purple row , which in this case would be the victim row . Tests show that this approach may result in a significantly higher rate of disturbance errors , compared to the variant that activates only one of the victim row 's neighbouring DRAM rows .

= = = Mitigation = = =

Different methods exist for more or less successful detection , prevention , correction or mitigation of the row hammer effect . Tests show that simple ECC solutions , providing single @-@ error correction and double @-@ error detection (SECDED) capabilities , are not able to correct or detect all observed disturbance errors because some of them include more than two flipped bits per memory word . A less effective solution is to introduce more frequent memory refreshing , with the refresh intervals shorter than the usual 64 ms , but this technique results in higher power consumption and increased processing overhead ; some vendors provide firmware updates that implement this type of mitigation . One of the more complex prevention measures performs counter @-@ based identification of frequently accessed memory rows and proactively refreshes their neighboring rows ; another method issues additional infrequent random refreshes of memory rows neighboring the accessed rows regardless of their access frequency . Research shows that these two prevention measures cause negligible performance impacts .

Since the release of Ivy Bridge microarchitecture , Intel Xeon processors support the so @-@ called pseudo target row refresh (pTRR) that can be used in combination with pTRR @-@ compliant DDR3 dual in @-@ line memory modules (DIMMs) to mitigate the row hammer effect by automatically refreshing possible victim rows , with no negative impacts on performance or power consumption . When used with DIMMs that are not pTRR @-@ compliant , these Xeon processors by default fall back on performing DRAM refreshes at twice the usual frequency , which results in slightly higher memory access latency and may reduce the memory bandwidth by up to 2 ? 4 % .

The LPDDR4 memory standard published by JEDEC includes optional hardware support for the so @-@ called target row refresh (TRR) that prevents the row hammer effect without negatively impacting performance or power consumption . Additionally , some manufacturers implement TRR in their DDR4 products , although it is not part of the DDR4 memory standard published by JEDEC . Internally , TRR identifies possible victim rows , by counting the number of row activations and comparing it against predefined chip @-@ specific maximum activate count (MAC) and maximum activate window (tMAW) values , and refreshes these rows to prevent bit flips . The MAC value is the maximum total number of row activations that may be encountered on a particular DRAM row within a time interval that is equal or shorter than the tMAW amount of time before its neighbouring rows are identified as victim rows ; TRR may also flag a row as a victim row if the sum of row activations for its two neighboring rows reaches the MAC limit within the tMAW time window .

Due to their necessity of huge numbers of rapidly performed DRAM row activations , row hammer exploits issue large numbers of uncached memory accesses that cause cache misses , which can be detected by monitoring the rate of cache misses for unusual peaks using hardware performance counters . Version 6 @.@ 0 @.@ 0 of the memtest86 memory diagnostic software , released on February 13 , 2015 , includes a so @-@ called hammer test that checks whether computer hardware is susceptible to disturbance errors .

= = Implications = =

Memory protection , as a way of preventing processes from accessing memory that has not been assigned to each of them , is one of the concepts behind most modern operating systems . By using memory protection in combination with other security @-@ related mechanisms such as protection rings , it is possible to achieve privilege separation between processes , in which programs and computer systems in general are divided into parts limited to the specific privileges they require to perform a particular task . Using privilege separation can also reduce the extent of potential damage caused by computer security attacks by restricting their effects to specific parts of the system .

Disturbance errors (explained in the section above) effectively defeat various layers of memory protection by " short circuiting " them at a very low hardware level , practically creating a unique attack vector type that allows processes to alter the contents of arbitrary parts of the main memory by directly manipulating the underlying memory hardware . In comparison , " conventional " attack vectors such as buffer overflows aim at circumventing the protection mechanisms at the software level , by exploiting various programming mistakes to achieve alterations of otherwise inaccessible main memory contents .

= = = Exploits = = =

The initial research into the row hammer effect , publicized by a group of authors in June 2014 , described the nature of disturbance errors and indicated the potential for constructing an attack , but did not provide any examples of a working security exploit . Another research paper , created by a group of authors and published in October 2014 , did not imply the existence of any security @-@ related issues arising from the row hammer effect .

On March 9 , 2015 , Google 's Project Zero revealed two working privilege escalation exploits based on the row hammer effect , establishing its exploitable nature on the x86 @-@ 64 architecture . One of the revealed exploits targets the Google Native Client (NaCl) mechanism for running a limited subset of x86 @-@ 64 machine instructions within a sandbox , exploiting the row hammer effect to escape from the sandbox and gain the ability to issue system calls directly . This NaCl vulnerability , tracked as CVE @-@ 2015 @-@ 0565 , has been mitigated by modifying the NaCl so it does not allow execution of the clflush (cache line flush) machine instruction , which has been found to be required for constructing an effective row hammer attack .

The second exploit revealed by Project Zero runs as an unprivileged Linux process on the x86 @-@ 64 architecture , exploiting the row hammer effect to gain unrestricted access to all physical memory installed in a computer . By combining the disturbance errors with memory spraying , this exploit is capable of altering page table entries (PTEs) used by the virtual memory system for mapping virtual addresses to physical addresses , which results in the exploit gaining unrestricted memory access . Due to its nature and the inability of the x86 @-@ 64 architecture to make clflush a privileged machine instruction , this exploit can hardly be mitigated on computers that do not use hardware with built @-@ in row hammer prevention mechanisms . While testing the viability of exploits , Project Zero found that about half of the 29 tested laptops experienced disturbance errors , with some of them occurring on vulnerable laptops in less than five minutes of running row @-@ hammer @-@ inducing code ; the tested laptops were manufactured between 2010 and 2014 and used non @-@ ECC DDR3 memory .

In July 2015 , a group of security researchers published a paper that describes an architecture- and instruction @-@ set @-@ independent way for exploiting the row hammer effect . Instead of relying on the clflush instruction to perform cache flushes , this approach achieves uncached memory accesses by causing a very high rate of cache eviction using carefully selected memory access patterns . Although the cache replacement policies differ between processors , this approach overcomes the architectural differences by employing an adaptive cache eviction strategy algorithm . The proof of concept for this approach is provided both as a native code implementation , and as a pure JavaScript implementation that runs on Firefox 39 . The JavaScript implementation , called Rowhammer.js , uses large typed arrays and relies on their internal allocation using large pages ; as a result , it demonstrates a very high @-@ level exploit of a very low @-@ level vulnerability .