Rowhammer

When thinking about potential computer vulnerabilities, I think it's fair to say that software code and operating system flaws are the predominant targets that are exploited. We have seen buffer overflows, where an application tries to put more data in a buffer than the buffer can handle, which can be exploited to crash an application, run malicious code, or cause corruption of data. There are SQL injection flaws where attackers change SQL queries which, once executed, could allow the attacker to not require authentication gaining access to the database information. These are two examples of well-known software security flaws and by just looking at the number of software security patches and updates available weekly and monthly one can conclude that skilled attackers have been focused on software vulnerabilities for many years. Another type of computer exploitation, called the rowhammer bug, is unique because this is a hardware flaw. This rowhammer bug is a hardware fault found in many DRAM memory modules manufactured from 2010 onward. Basically, continued refreshing rows of memory cells can cause bits to flip in adjacent rows. With an x86 running Linux, if you can induce corruption into DRAM, then potentially you could also discover methods to take over the kernel. Looking at how memory is structured will give us an understanding of the rowhammer bug, how it's creatively exploited, and what steps memory designers can take to mitigate or remove the potential threat.

Computer memory starts at the cell level. A cell represents a bit of 1 or 0. Memory cells are set up in rows and columns so they can be addressed and accessed. This matrix of cell memory, once written to, maintains the charged single bit state by using capacitors. A capacitor is a device that can temporarily store an electrical charge. Once the voltage sourced to the capacitor goes to zero volts the capacitor, by nature, begins to leak or discharge. As computer

software requires higher capacity memory modules, these memory integrated circuits and memory cells within them become much denser. The fact that memory cells are much closer together allows the varying electromagnetic field associated with the flow of current to cause bits flips in adjacent cells. As mentioned earlier, a memory cell is charged during a write and the charge is maintained by capacitors. The capacitors, by nature, leak and discharge so they need to be refreshed. The refresh frequency for DRAM cells is within 64ms. The memory cell contents need to be refreshed, or basically re-written to, without any change in order to maintained the data. This refresh process when a row of cells is refreshed, or energized too often can cause disturbance errors. We know that there is a magnetic field present with the flow of electrons. Electromagnetic radiation can be generated as electronic and associated magnetic fields fluctuate at the same time. Continued refreshing or hammering a row of cells allows this electromagnetic radiation to flip bits in adjacent rows. Once a hacker can flip bits, this opens the opportunity to exploit the inherent weakness of highly dense memory integrated circuits.

Some very skilled hardware engineers targeted this DRAM vulnerability to exploit bit flips of the rowhammer bug. Bit flips are a known occurrence in DRAM, and for computer servers Error-Correcting Code (ECC) memory was developed to identify and correct single bit memory errors. These engineers found that they could take steps to identify cells that could be flipped, then induce the hammered rows bypassing cache, and writing directly to the memory cells. The goal is to repeatedly write to a memory address and normally repeated writes would go to cache. One way to bypass the normal write process to cache on an x86 machine is to use the CLFLUSH instruction that flushes the contents of cache. This is an unprivileged instruction that can be used by any process and cannot be disabled. To reduce latency, memory is designed into equal sized banks. Writing to two rows back and forth allows changes to the current row. A

software program was used to first identify addresses where bits are flipping then an algorithm was used to locate two physical addresses in the same memory bank. They used a technique referred to as double-sided hammering. This technique hammers the rows above and below the row they wanted to flip bits to increase the electromagnetic field around the cells, increasing the frequency of bit flips. They noticed that after finding bits that would flip these same bits could be induced to flip regularly.

Now that rows were identified where bits would flip, the rowhammer bug was exploited in multiple forms. One way was to flip bits in paging tables, gaining the ability to write to the paging tables, which in turn could be used to obtain write access to physical memory. After getting write access to physical memory through the paging tables code could be located, changed, and overwritten. Examples of exploitations observed by research teams were that higher system privileges could be gained by an untrusted application, and designed security could be breached that stopped malicious code from influencing a Linux operating system. Also, many websites use JavaScript. JavaScript could hold the code to be used within a web browser to find the bits that can be flipped through rowhammer. Then the rows of memory cells are continuously refreshed or hammered which can cause cells in adjacent rows to flip. By flipping adjacent rows of cells containing those bits you can get access once again to physical memory. Google reports there are over one billion active Android devices worldwide. Researchers have now designed an attack on Android devices using rowhammer to get unrestricted root access reportedly in a matter of seconds.

Surprisingly, my research discovered that the largest manufacturers of DRAM were aware of this potential security flaw of bit flipping for many years. The assumption appears to be that the cost to redesign DRAM, versus the chance of the memory flaw exploitation being

discovered and implemented, was a calculated business decision. The Google Project Zero team and Carnegie Mellon University were two of the first to document the flaw and exploitation techniques. Google's Project Zero team tested 29 laptops, manufactured from 2010 to 2014, using DDR3 DRAM including 8 models using 5 memory manufacturers. Bit flips occurred in the memory of 15 laptops. Since this is a hardware bug there is no patch or update to fix it. The recommended measures to mitigate or stop the rowhammer bug include: Running ECC memory which identifies and corrects single bit errors, Target Row Refresh (TRR) which counts the number of times a row is refreshed and refreshes the adjacent rows once the determined threshold is reached, and lastly doubling the refresh rate for DRAM which would require more power impacting overall system performance. At first, it appeared that this flaw only impacted DDR3 memory, but further research determined that some DDR4 memory is also susceptible to the flaw. This may make the best solution just to replace memory that software diagnostics identifies as being susceptible to DRAM memory cell bit flips.

Organizations, like the IEEE Computer Society Center for Secure Design, made up of software security professionals, discover and identify design flaws that contribute to cyber security breaches. Since the beginning of software development there have always been flaws and bugs within the code that become exploited by hackers. By identifying design flaws that became security breaches these organizations develop strategies for software developers to follow during the design phase of development. In 2014, the National Vulnerability Database listed the average number of 19 operating and application security vulnerabilities per day. Looking at 2014 data Apple Mac OS X, Linux Kernel, and Microsoft versions all experienced over 100 vulnerabilities. Microsoft IE, Google Chrome, and Mozilla Firefox web browsers all experienced over 100 vulnerabilities. Adobe and Java both once again had over 100

vulnerabilities during 2014. All these vulnerabilities have one thing in common. They all are software bugs or flaws. Once these software weaknesses are discovered updates and patches are created to remove the threat. I think it's safe to say that when thinking about computer security vulnerabilities one automatically thinks of software code. Therefore, researching rowhammer was extremely interesting. It was my first examples of a computer vulnerability that surfaced because of the design of DRAM. The hardware bug was due to requirements of software to have inexpensive DRAM memory modules of increasingly higher capacity. A flaw known by the largest DRAM manufacturers but apparently not taken seriously. The type of DRAM that can suffer from the rowhammer bug would be memory modules for laptops, desktops, smart phones, and other devices. Most server memory used ECC memory that can correct single bit memory errors. This can still potentially be a serious problem as a laptop could contain classified data or sensitive intellectual property. A hacked smart phone could be a serious security break. Hopefully the awareness of the rowhammer bug will drive manufacturers of computer hardware to incorporate security in the design phase of hardware architecture much like software developers have been doing for years.

REFERENCES

Walker James, "Rowhammer memory flaw puts millions of smartphones at risk", Digital Journal,
October 24, 2016 *
http://www.digitaljournal.com/tech-and-science/technology/rowhammer-memory-flaw-puts-millions-of-smartphones-at-risk/article/477987

Mutlu Onur, "The Row Hammer Problem and Others Issues We May Face as Memory Becomes Denser", Invited Talk and Paper in Proceedings of the 53rd Design Automation Conference, Austin TX, June 2016 *
https://users.ece.cmu.edu/~omutlu/projects.htm

Seaborn Mark, Dullien Thomas, "Exploiting the DRAM rowhammer bug to gain kernel privileges", Project Zero, March 2015
https://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html

Jean-Pharuns Alix, "Rowhammer.js Is the Most Ingenious Hack I've Ever Seen", Motherboard, July 2015
https://motherboard.vice.com/en_us/article/rowhammerjs-is-the-most-ingenious-hack-ive-ever-seen

Row hammer. (2017, March 11). In *Wikipedia, The Free Encyclopedia*. Retrieved 17:11, April 9, 2017, from https://en.wikipedia.org/w/index.php?title=Row_hammer&oldid=769721113