RAMBleed

Reading Bits in Memory Without Accessing Them

RAMBleed is a side-channel attack that enables an attacker to read out physical memory belonging to other processes. The implications of violating arbitrary privilege boundaries are numerous, and vary in severity based on the other software running on the target machine. As an example, in our paper we demonstrate an attack against OpenSSH in which we use RAMBleed to leak a 2048 bit RSA key. However, RAMBleed can be used for reading other data as well.

RAMBleed is based on a previous side channel called Rowhammer, which enables an attacker to flip bits in the memory space of other processes. We show in our paper that an attacker, by observing Rowhammer-induced bit flips in her own memory, can deduce the values in nearby DRAM rows. Thus, RAMBleed shifts Rowhammer from being a threat not only to integrity, but confidentiality as well. Furthermore, unlike Rowhammer, RAMBleed does not require persistent bit flips, and is thus effective against ECC memory commonly used by server computers.

We will present our paper titled "RAMBleed: Reading Bits in Memory Without Accessing Them" at the 41st IEEE Symposium on Security and Privacy (https://www.ieee-security.org/TC/SP2020/) in May, 2020.

Read the Paper (docs/20190603-rambleed-web.pdf)

Cite

People

RAMBleed was discovered by the following joint team of academic researchers:

- Andrew Kwong (https://andrewkwong.org) at University of Michigan (https://www.cse.umich.edu)
- Daniel Genkin (https://web.eecs.umich.edu/~genkin/) at University of Michigan (https://www.cse.umich.edu)
- Daniel Gruss (https://gruss.cc/) at Graz University of Technology (https://www.tugraz.at/home/)
- Yuval Yarom (https://www.adelaide.edu.au/directory/yuval.yarom) at University of Adelaide (https://www.adelaide.edu.au/) and Data 61 (https://www.data61.csiro.au/).

(https://www.cse.umich.edu)

(http://iaik.tugraz.at/)

(https://www.adelaide.edu.au)

(https://www.data61.csiro.au/)

Q&A

What is the Rowhammer bug?

The trend towards increasing DRAM cell density and decreasing capacitor size over the past decades has given rise to a reliability issue known as Rowhammer. Specifically, repeated accesses to rows in DRAM can lead to bit flips in neighboring rows (not only the direct neighbors), even if these neighboring rows are not accessed.

Attackers can exploit these cross process bit flips for a myriad of security breaches. Researchers have demonstrated how to abuse Rowhammer for privilege escalation (https://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html), RSA modulus factorization (https://www.vusec.net/projects/flip-feng-shui/), and more.

What is RAMBleed?

Previous attacks exploited the Rowhammer effect to write (or flip) bits in the victim's memory. RAMBleed is different in that it uses Rowhammer for reading data stored inside the computer's physical memory. As the physical memory is shared among all process in the system, this puts all processes at risk.

What data can be read by RAMBleed?

While the end-to-end attack we demonstrated read out OpenSSH 7.9's RSA key, RAMBleed can potentially read any data stored in memory. In practice, what can be read depends on the victim program's memory access patterns.

■ You extracted an OpenSSH key!? Does that mean that I should stop using SSH?

There is nothing particularly vulnerable about OpenSSH, it was simply a convenient target to demonstrate RAMBleed's security implications. We don't recommend that you stop using SSH any more than we recommend that you stop using the internet.

■ What technologies are affected by RAMBleed?

RAMBleed relies on Rowhammer-induced bit flips to read privileged memory. As such, any system that uses Rowhammer-susceptible DIMMs is vulnerable. Previous research has demonstrated bit flips on both DDR3 (https://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html) and DDR4 (http://www.thirdio.com/rowhammer.pdf) with TRR (targeted row refresh) enabled. While we demonstrated our attack on a desktop machine and an ECC enabled server machine, Rowhammer attacks have been demonstrated against both mobile (https://www.vusec.net/projects/drammer/) devices and laptops (https://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html). As such, we suspect that many classes of computers are susceptible to RAMBleed.

■ Does ECC (Error Correcting Code) memory prevent RAMBleed?

No! RAMBleed uses bit flips as a read side channel, and as such does not require bit flips to be persistent. Instead, the attacker merely needs to know that a bit flip occurred; the secret information leaks regardless of whether or not ECC corrects the flip.

If ECC corrects the flip, how can the attacker determine whether or not a bit has flipped in her memory? The attacker can read her memory and use the ECC timing side channel to determine if the bit flipped. As described by Cocojar et al. (https://www.vusec.net/projects/eccploit/), when the hardware corrects the bit flip, a large delay is induced on that particular memory access. On our setup, we found an even stronger signal than previously reported, with a 1,000,000 X slowdown over the common case.

■ How can I mitigate this issue?

Users can mitigate their risk by upgrading their memory to DDR4 with targeted row refresh (TRR) enabled. While Rowhammer-induced bit flips have been demonstrated on TRR, it is harder to accomplish in practice.

Memory manufacturers can help mitigate this issue by more rigorously testing for faulty DIMMs. Furthermore, publicly documenting vendor specific TRR implementations will facilitate a stronger development process as security researchers probe such implementations for weaknesses.

■ Can RAMBleed be detected by antivirus?

We believe that it is very unlikely that any antivirus software on the market currently detects RAMBleed.

■ Was RAMBleed ever exploited in the wild?

It is not possible for us to say definitively, but we believe it to be unlikely.

■ How does RAMBleed work?

Rowhammer induced bit flips are data dependent, i.e. a bit is more likely to flip when the bits above and below it have the opposite charge. This creates a data-dependent side channel, wherein an attacker can deduce the values of bits in nearby rows by observing bit flips in her own memory rows. Finally, as the data in nearby rows might belong to a different process, this leakage breaks the isolation boundaries enforced by the operating system.

To exploit this effect, we developed novel memory massaging techniques to carefully place the victim's secret data in the rows above and below the attacker's memory row. This causes the bit flips in the attacker's rows to depend on the values of the victim's secret data. The attacker can then use Rowhammer to induce bit flips in her own memory, thereby leaking the victim's secret data.

■ Is there a CVE number?

Yes, see CVE-2019-0174 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-0174).

■ Why is it called RAMBleed?

Due to deficiencies in the memory modules, the RAM bleeds its contents, which we then recover through a side-channel.

■ Can I use the logo?

All rights to the logo have been waived through CC0 (https://creativecommons.org/publicdomain/zero/1.0/). Marina Minkin (http://www-personal.umich.edu/~minkin/index.html) designed the logo.

Acknowledgments

This research was partially supported by Intel.