Meltdown "basically melts security boundaries which are normally enforced by the hardware." Spectre, meanwhile, "breaks the isolation between different applications" allowing "an attacker to trick error-free programs, which follow best practices, into leaking their secrets."

Essentially, either of these vulnerabilities could be theoretically exploited to steal sensitive data, like passwords, off our computer. Spectre is also a threat to our Smartphone, so no escape there.

Meltdown got its name because it "melts" security boundaries normally enforced by hardware. By exploiting Meltdown, an attacker can use a program running on a machine to gain access to data from all over that machine that the program shouldn't normally be able to see, including data belonging to other programs and data that only administrators should have access to. Meltdown doesn't require too much knowledge of how the program the attacker hijacks works, but it only works with specific kinds of Intel chips. This is a pretty severe problem but fixes are being rolled out.

By exploiting the Spectre variants, an attacker can make a program reveal some of its own data that should have been kept secret. It requires more intimate knowledge of the victim program's inner workings, and doesn't allow access to other programs' data, but will also work on just about any computer chip out there. Spectre's name comes from speculative execution but also derives from the fact that it will be much trickier to stop — while patches are starting to become available, other attacks in the same family will no doubt be discovered. That's the other reason for the name. Spectre will be haunting us for some time.

Spectre and Meltdown both open up possibilities for dangerous attacks. For instance, JavaScript code on a website could use Spectre to trick a web browser into revealing user and password information. Attackers could exploit Meltdown to view data owned by other users and even other virtual servers hosted on the same hardware, which is potentially disastrous for cloud computing hosts. But beyond the potential specific attacks themselves lies the fact that the flaws are fundamental to the hardware platforms running beneath the software we use every day. Even code that is formally secure as written turns out to be vulnerable, because the assumptions underlying the security processes built into the code — indeed, built into all of computer programming — have turned out to be false.