Jane Dee Horwitz Parson
November 2025
Computer Security

The 2024 Crowdstrike Break

On July 19th. 2024, 8.5 million computers running Microsoft Windows crashed, entering inescapable boot-loops that prevented any manner of recovery. They would open, display a BSOD (blue screen of death), attempt to restart, and repeat. Hundreds of thousands of websites went down, but those weren't the worst of the consequences. Bank servers stopped being able to process transactions- their payment processing servers having been afflicted. Five thousand flights were cancelled, as no passengers were able to check in at airports with compromised systems. 911 services for some cities in America were rendered completely nonfunctional- the calls being routed through a now broken computer. It would have been the largest cybersecurity attack to ever occur– if it were a cybersecurity attack, that is.

It wasn't. It was a bug, one caused by a company named Crowdstrike

Crowdstrike is a cybersecurity corporation that specializes in large-scale industry solutions– they manage firewalls, serve as on-call technicians, and generally ensure that companies technology remains untampered with by neer-do-wells of all sorts. A faulty update sent to their Kernel-level EDR software, 'Falcon,' would crash the computer on startup through making requests to non-accessible memory. There was no way to undo the change, and until recovery software was produced, the only repair method was to revert the computer to a backup; something impossible for some computers. How did something like this happen? How could it spread so easily, and how could it do so much damage?

Before we can answer that, it's important to understand a few of the nonsense words I threw around. Namely, what is a piece of 'Kernel-Level" software?

Operating systems such as Windows run on a series of rings of trust. The innermost, 0, has the most control, and as the number increases, the amount of power a piece of software has decreases. The Kernel sits at level 0, and is the ultimate piece of software that controls every function of a computer, including communication between parts of the machine. Traditional software runs at Level 3, or 2 with administrator permissions applied.

In this case, Kernel-Level meant that Falcon had complete control over every aspect of the computer, and could read every piece of information involved. This was because Falcon is an *Endpoint Detection and Response* software (EDR, for short), a style of antivirus software that middlemans every single interaction a computer holds. In order to do that, it needed to see all those interactions- and necessarily to see those interactions, it needed permission to operate with ultimate control of the machine. This would have been fine, should it have continued to function properly, but with the faulty update this meant that the computer's functionality was now tethered at the most basic level to a broken piece of software.

Another question might come, now. How did the update break *every* computer? Shouldn't have people found out and chosen not to install the new update?

Because Crowdstrike wouldn't let them. The system responsible for updating Crowdstrike's software does so automatically, with no method of intervention on the part of the user. Once the faulty update was distributed, every system that was powered on automatically received and installed it, and the failure in the system prevented any replacement patches from being installed.

This was, on Crowdstrike's part, for two reasons. One is that malware develops consistently enough that a machine *needs* those constant updates to defend itself properly. Another is that these machines are designed to run without direct interaction (outside of remote sysadmin intervention) for long periods of time. If someone had to get to every single screen in LAX to install updates, it would hobble their infrastructure. On the other hand, this was a remote uncontrollable backdoor to the kernel level installed in seven million computers.

Now that we've established how this failure spread and caused so much harm, we can ask another question. What *was* this failure?

Crowdstrike published an incident report after the fact that's pretty telling. Crowdstrike Falcon functions on the reading and monitoring of Interprocess Communication methods, or IPCs for short. These are the ways that different processes in the computer talk to one another, through verifiable 'Pipes' and other channels. Crowdstrike has a file to monitor these named pipes, a template file that marked twenty-one communication inputs of which to monitor. However, the integration within the computer only legitimately linked twenty of them.

This raised no issues in tests, due to the majority of stress-testing running on virtual machines that assumed all twenty-one pipes were matched up. The computer was made to match these specifications, and in tests on physical machines the pipe was only checked with *wildcard* matching (assuming some placeholder exists that will be substituted in later, ergo, won't throw any error if there's nothing there.)

This caused no problems, at first. The update was made in February 2024. Five months later was the first time a test made use of the 21st pipe without the use of a wildcard modifier. The error was discovered within an hour of distribution, and they immediately pulled the patch to issue a fixed replacement. However, by that point, it was already too late. Any computers that were powered on for the duration of the update were infected and couldn't receive the fix.

To compound this, there *is* a way to manually fix the problem– users could enter Windows Safe Mode, a version of the system that doesn't run any programs on startup, *including* kernel linked ones, and delete the individual faulty driver. From there, the computer would restart as normal, and it would work as intended. It's unfortunate then that no industry computers could use this technique, as 1) almost all interaction required remote access, from which you could not enter safe mode, and 2) BitLocker instances would immediately clamp down on the server, requiring a key to unlock that was often stored on *another broken server*. Even when windows released specific recovery tools, namely a boot media that could be loaded on USB, admins would have

to manually plug the device into every computer, and *still* required the BitLocker key if their original settings were strict enough.

Crowdstrike doesn't elaborate on how errors failed to be discovered in testing, aside from the aforementioned wildcard oversight, but consider: In Texas, where Crowdstrike's central offices are, the update was shipped at 10:00 PM on a Friday.

If you're reading this, you were almost certainly around when the Crowdstrike attacks actually happened, but it bares repeating just what kind of computers used this software. Check in systems for airports, banking services, Security systems, Ground Shipping, Election Systems, first response call systems, ***Hospitals***. A failure on the part of a privately owned company hobbled the entire internet, and almost certainly killed people in the process. So what consequences did the company face, for such a grievous oversight?

None.

No part of their failure violated the law, you see- and in the contract that all parties signed to use the software, they waived their right to anything further than a cash restitution for failures on the part of Crowdstrike. Negligence law would be *incredibly* difficult to find charges with, because almost all of it was written before the Internet's widespread adoption. All the law is, really, except for new bits like the DMCA that only account for the profits of major corporations. There is no legal structure for governing the internet, and that gap prevented Crowdstrike from facing any repercussions, at least legally.

You might wonder the same for financially, though- and it would be fair. After such a visible failure on their part to defend the computers they were meant to, their stock did plummet $150, as you can see below:

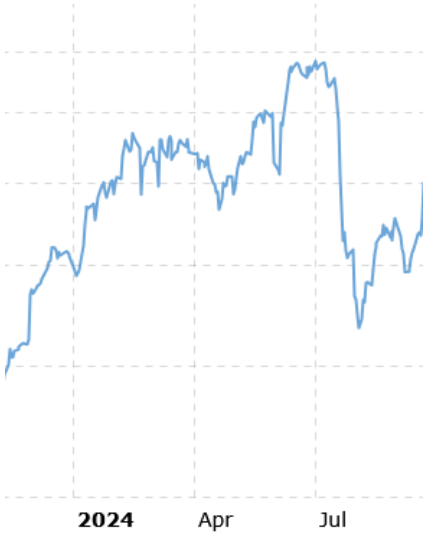
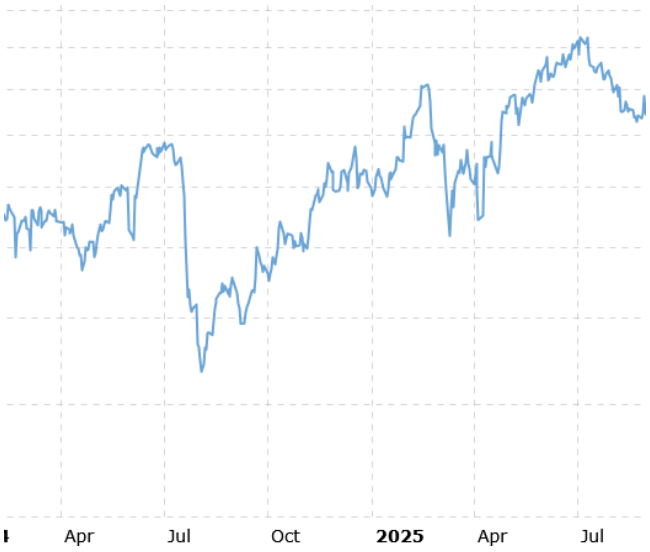


Image credit MacroTrends.com - Stock ticker of Crowdstrike

If you've heard the saying 'All press is good press,' it applies here. Crowdstrike might have taken a hit, but it was now *visible*, and people who might not have known it existed now had one cybersecurity firm on the top of their mind when they went to purchase. They recovered by the end of the year.



They didn't change their structure, either. Their methods of testing builds have changed internally (not that they'll give much in terms of specifics), but the matter of fact is that Crowdstrike has not truly changed, and will not change unless they explicitly *need* to. Even Microsoft has dragged feet on potentially limiting what pieces of software can have kernel-level authority, but any definition that they could use would include antivirus software such as this.

With tech so omnipresent in our society, and more and more able to cause severe harm, companies like Crowdstrike need legal oversight. There needs to be safeguards and consequences, ways to prevent the move-fast-and-break-things mindset of Silicon Valley from breaking things that matter to everyone.

This is what happens on *accident.* Can you imagine what would happen if it wasn't?