

Hacker Lexicon: What Is a Supply Chain Attack?

 [wired.com/story/hacker-lexicon-what-is-a-supply-chain-attack/](https://www.wired.com/story/hacker-lexicon-what-is-a-supply-chain-attack/)

Andy Greenberg

May 31, 2021



Cybersecurity truisms have long been described in simple terms of trust: Beware email attachments from unfamiliar sources, and don't hand over credentials to a fraudulent website. But increasingly, sophisticated hackers are undermining that basic sense of trust and raising a paranoia-inducing question: What if the legitimate hardware and software that makes up your network has been compromised at the source?

That insidious and increasingly common form of hacking is known as a "supply chain attack," a technique in which an adversary slips malicious code or even a malicious component into a trusted piece of software or hardware. By compromising a single supplier, spies or saboteurs can hijack its distribution systems to turn any application they sell, any software update they push out, even the physical equipment they ship to customers, into Trojan horses. With one well-placed intrusion, they can create a springboard to the networks of a supplier's customers—sometimes numbering hundreds or even thousands of victims.

"Supply chain attacks are scary because they're really hard to deal with, and because they make it clear you're trusting a whole ecology," says Nick Weaver, a security researcher at UC Berkeley's International Computer Science Institute. "You're trusting every vendor whose code is on your machine, *and* you're trusting every vendor's vendor."

The severity of the supply chain threat was demonstrated on a massive scale last December, when it was revealed that Russian hackers—later identified as working for the country's foreign intelligence service, known as the SVR—had hacked the software firm SolarWinds and planted malicious code in its IT management tool Orion, allowing access to as many as 18,000 networks that used that application around the world. The SVR used that foothold to burrow deep into the networks of at least nine US federal agencies, including NASA, the State Department, the Department of Defense, and the Department of Justice.

But as shocking as that spy operation was, SolarWinds wasn't unique. Serious supply chain attacks have hit companies around the world for years, both before and since Russia's audacious campaign. Just last month, it was revealed that hackers had compromised a software development tool sold by a firm called CodeCov that gave the hackers access to hundreds of victims' networks. A Chinese hacking group known as Barium carried out at least six supply chain attacks over the past five years, hiding malicious code in the software of computer maker Asus and in the hard-drive cleanup application CCleaner. In 2017 the Russian hackers known as Sandworm, part of the country's GRU military intelligence service, hijacked the software updates of the Ukrainian accounting software MEDoc and used it to push out self-spreading, destructive code known as NotPetya, which ultimately inflicted \$10 billion in damage worldwide—the costliest cyberattack in history.

In fact, supply chain attacks were first demonstrated around four decades ago, when Ken Thompson, one of the creators of the Unix operating system, wanted to see if he could hide a backdoor in Unix's login function. Thompson didn't merely plant a piece of malicious code that granted him the ability to log into any system. He built a compiler—a tool for turning readable source code into a machine-readable, executable program—that secretly placed the backdoor in the function when it was compiled. Then he went a step further and corrupted the compiler that *compiled* the compiler, so that even the source code of the user's compiler wouldn't have any obvious signs of tampering. "The moral is obvious," Thompson wrote in a lecture explaining his demonstration in 1984. "You can't trust code that you did not totally create yourself. (Especially code from companies that employ people like me.)"

That theoretical trick—a kind of double supply chain attack that corrupts not only a widely used piece of software but the tools used to create it—has since become a reality too. In 2015, hackers distributed a fake version of XCode, a tool used to build iOS applications, that secretly planted malicious code in dozens of Chinese iPhone apps. And the technique appeared again in 2019, when China's Barium hackers corrupted a version of the Microsoft Visual Studio compiler so that it let them hide malware in several video games.

The rise in supply chain attacks, Berkeley's Weaver argues, may be due in part to improved defenses against more rudimentary assaults. Hackers have had to look for less easily protected points of ingress. And supply chain attacks also offer economies of scale; hack one software supplier and you can get access to hundreds of networks. "It's partially that you

want bang for your buck, and partially it's just that supply chain attacks are indirect. Your actual targets are not who you're attacking," Weaver says. "If your actual targets are hard, this might be the weakest point to let you get into them."



Preventing future supply chain attacks won't be easy; there's no simple way for companies to ensure that the software and hardware they buy hasn't been corrupted. Hardware supply chain attacks, in which an adversary physically plants malicious code or components inside a piece of equipment, can be particularly hard to detect. While a [bombshell report from Bloomberg in 2018 claimed](#) that tiny spy chips had been hidden inside the SuperMicro motherboards used in servers inside Amazon and Apple data centers, all the companies involved vehemently denied the story—as did the NSA. But the classified leaks of Edward Snowden revealed that the [NSA itself has hijacked shipments of Cisco routers and backdoored them for its own spying purposes](#).

The solution to supply chain attacks—on both software and hardware—is perhaps not so much technological as organizational, argues Beau Woods, a senior adviser to the Cybersecurity and Infrastructure Security Agency. Companies and government agencies need to know who their software and hardware suppliers are, vet them, hold them to certain standards. He compares that shift to how companies like Toyota seek to control and limit their supply chains to ensure reliability. The same now has to be done for cybersecurity. "They look to streamline the supply chain: fewer suppliers and higher-quality parts from those suppliers," Woods says. "Software development and IT operations have in some ways been relearning those supply chain principles."

The Biden White House's [cybersecurity executive order](#) issued earlier this month may help. It sets new minimum security standards for any company that wants to sell software to federal agencies. But the same vetting is just as necessary across the private sector. And private companies—just as much as federal agencies—shouldn't expect the epidemic of supply chain compromises to end any time soon, Woods says.

Ken Thompson may have been right in 1984 when he wrote that you can't fully trust any code that you didn't write yourself. But trusting code from suppliers you trust—and have vetted—may be the next best thing.

More Great WIRED Stories

-  The latest on tech, science, and more: [Get our newsletters!](#)
- The Arecibo Observatory was like family. [I couldn't save it](#)
- The hostile takeover of a [Microsoft Flight Simulator server](#)
- Goodbye Internet Explorer—and good riddance
- How to take a slick, professional [headshot with your phone](#)
- Online dating apps are actually [kind of a disaster](#)
-  Explore AI like never before with [our new database](#)

- 🎮 WIRED Games: Get the latest tips, reviews, and more
- ✨ Optimize your home life with our Gear team's best picks, from robot vacuums to affordable mattresses to smart speakers