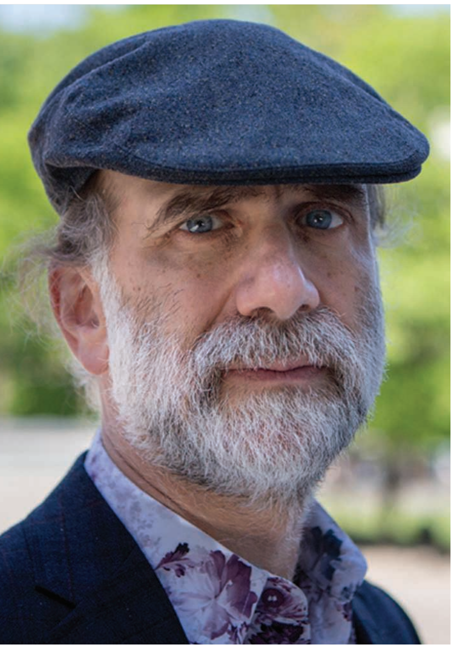

Bruce Schneier
Harvard University

Hacking the Tax Code

The tax code isn't software. It doesn't run on a computer. But it's still code. It's a series of algorithms that takes an input—financial information for the year—and produces an output: the amount of tax owed. It's incredibly complex code; there are a bazillion details and exceptions and special cases. It consists of government laws, rulings from the tax authorities, judicial decisions, and legal opinions.

Like computer code, the tax code has bugs. They might be mistakes in how the tax laws were written. They might be mistakes in how the tax code is interpreted, oversights in how parts of the law were conceived, or unintended omissions of some sort or another. They might arise from the exponentially huge number of ways different parts of the tax code interact.

A recent example comes from the 2017 Tax Cuts and Jobs Act. That law was drafted in both haste and secret, and quickly passed without any time for review—or even proofreading. One of the things in it was a typo that accidentally categorized military death benefits as earned income. The practical effect of that mistake is that surviving family members were hit with surprise tax bills of US$10,000 or more.

That's a bug, but not a vulnerability. An example of a vulnerability is the "Double Irish with a Dutch Sandwich." It arises from the interactions of tax laws in multiple countries, and it's how companies like Google and Apple have avoided paying U.S. taxes despite being U.S. companies. Estimates are that U.S. companies avoided paying nearly US$200 billion in taxes in 2017 alone.

In the tax world, vulnerabilities are called *loopholes*. Exploits are called *tax avoidance strategies*. And there are thousands of black-hat researchers who examine every line of the tax code looking for exploitable vulnerabilities—tax attorneys and tax accountants.

Some vulnerabilities are deliberately created. Lobbyists are constantly trying to insert this or that provision into the tax code that benefits their clients financially. That same 2017 U.S. tax law included a special tax break for oil and gas investment partnerships, a special exemption that ensures that fewer than 1 in 1,000 estates will have to pay estate tax, and language specifically expanding a pass-through loophole that industry uses to incorporate companies offshore and avoid U.S. taxes. That's not hacking the tax code. It's hacking the processes that create them: the legislative process that creates tax law.

We know the processes to use to fix vulnerabilities in computer code. Before the code is finished, we can employ some sort of secure development processes, with automatic bug-finding tools and maybe source code audits. After the code is deployed, we might rely on vulnerability finding by the security community, perhaps bug bounties—and most of all, quick patching when vulnerabilities are discovered.

What does it mean to "patch" the tax code? Passing any tax legislation is a big deal, especially in the United States where the issue is so partisan and contentious. (That 2017 earned income tax bug for military families hasn't yet been fixed. And that's an easy one; everyone acknowledges it was a mistake.) We don't have the ability to patch tax code with anywhere near the same agility that we have to patch software.

We can patch some vulnerabilities, though. The other way tax code is modified is by IRS and judicial rulings. The 2017 tax law capped income tax deductions for property taxes. This provision didn't come into force in 2018, so someone came up with the clever hack to prepay 2018 property taxes in 2017. Just before the end of the year, the IRS ruled about when that was legal and when it wasn't. Short answer: most of the time, it wasn't.

There's another option: that the vulnerability isn't patched and isn't explicitly approved, and slowly becomes part of the normal way of doing things. Lots of tax loopholes end up

Digital Object Identifier 10.1109/MSEC.2020.3007049
Date of current version: 3 September 2020

continued on p. 79

Last Word

continued from p. 80

like this. Sometimes they're even given retroactive legality by the IRS or Congress after a constituency and lobbying effort gets behind them. This process is how systems evolve. A hack subverts the intent of a system. Whatever governing system has jurisdiction either blocks the hack or allows it—or does nothing and the hack becomes the new normal.

Here's my question: what happens when artificial intelligence and machine learning (ML) gets hold of this problem? We already have ML systems that find software vulnerabilities. What happens when you feed a ML system the entire U.S. tax code and tell it to figure out all of the ways to minimize the amount of tax owed? Or, in the case of a multinational corporation, to feed it the entire planet's tax codes? What sort of vulnerabilities would it find? And how many? Dozens or millions?

In 2015, Volkswagen was caught cheating on emissions control tests. It didn't forge test results; it got the cars' computers to cheat for them. Engineers programmed the software in the car's onboard computer to detect when the car was undergoing an emissions test. The computer then activated the car's emissions-curbing systems, but only for the duration of the test. The result was that the cars had much better performance on the road at the cost of producing more pollution.

ML will result in lots of hacks like this. They'll be more subtle. They'll be even harder to discover. It's because of the way ML systems optimize themselves, and because their specific optimizations can be impossible for us humans to understand. Their human programmers won't even know what's going on.

Any good ML system will naturally find and exploit hacks. This is because their only constraints are the rules of the system. If there are problems, inconsistencies, or loopholes in the rules, and if those properties lead to a "better" solution as defined by the program, then those systems will find them. The challenge is that you have to define the system's goals completely and precisely, and that that's impossible.

The tax code can be hacked. Financial markets regulations can be hacked. The market economy, democracy itself, and our cognitive systems can all be hacked. Tasking a ML system to find new hacks against any of these is still science fiction, but it's not stupid science fiction. And ML will drastically change how we need to think about policy, law, and government. Now's the time to figure out how. ■

Bruce Schneier is a security technologist, fellow, and lecturer at the Harvard Kennedy School and the chief of security architecture of Inrupt, Inc. He blogs at www.schneier.com and is currently writing a book on this essay's topic (expected to be published in 2021). Contact him at schneier@schneier.com.