The Last Byte

Hacking in the Dark

Scott Davidson

I Haven't Been to DAC for a while, so I was happy to learn of the Hack@DAC initiative. This exercise is especially useful for security, the subject of the Hack@DAC highlighted in this issue of Design&Test. Security threats are always going to be distributed in the sense that they can come from anywhere, can attack a product no matter where it is and will be initiated by attackers with a wide variety of backgrounds and skills—wider than any security team can ever hope to have. A distributed defense from giving a wide range of hackers in the good sense the charter of trying to break a system can find flaws before the bad guys do.

One of the articles in this issue mentions that security workers inside a design team have the advantage of being able to ask about the design from those who created it. I can understand one thinking this when struggling to figure out what a chunk of RTL code does. But there is a downside.

Back when I started, designers were responsible for writing tests to detect manufacturing faults and defects. When we were able to fault simulate these tests, we found the coverage was much worse than the test writers' estimates. Designers think of a design in terms of what it is supposed to do, which includes how it reacts to out of bounds inputs. Defects cause designs to operate in ways that designers can hardly imagine. Even if they have a good imagination, there are an immense number of ways that something as simple as an arithmetic logic unit can fail. Covering them all functionally would tax the patience of the most dedicated designer, not to mention the time and money budget of the project.

Automatic test generators do not have these constraints. I looked at the automatically generated test for a counter. It exercised the counter in ways I'd never think of and did it more efficiently than I could.

The situation for security is similar. Security holes might be things that the designer missed or some function of the design that was unintended. A designer might be sure that a situation putting a design at risk will never come up—a Hack@DAC participant, not knowing this, might find that it does.

Perhaps AI would be a good avenue for security analysis, since learning systems approach problems in ways humans don't. I didn't find any such work in a quick search, but if there is some perhaps that would be a good special issue of D&T.

Not that it would help. Those companies likely to deploy such a system, and those companies who'd hire Hack@DAC veterans, are not likely to be the problem. Think of Internet of Things (IoT) devices out there with obsolete and insecure operating systems, basic mistakes like encoding passwords in firmware and sloppy design practices. How are we going to get those companies to pay attention to hardware security? Given the number of people today who are irresponsible despite putting other's lives at risk, and aren't even making a profit by being irresponsible, not likely.

Maybe the solution will be for an AI system to win Hack@DAC someday—and then go on to publish security reviews of IoT devices on social media. Perhaps shame will work where responsibility doesn't. ■

Digital Object Identifier 10.1109/MDAT.2020.3045853
Date of current version: 10 March 2021.

Direct questions and comments about this article to Scott Davidson; davidson.scott687@gmail.com; Twitter: @scottd687.

2168-2356/21©2021 IEEE
Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC