

Title: We need a verifiable electronic voting system

Summary: Now more than ever, says IBM computer scientist Rich Cardone, voters need an electronic voting system whose results are easy to confirm.

Duration: 5 minutes, 40 seconds

Introduction

If you've ever seen Preston Sturges' [The Great McGinty](#), you know exactly what can go wrong in an electoral process that cannot be held to account.

IBM computer scientist Rich Cardone is an ardent advocate of an electronic voting system whose processes and results are verifiable. In this episode of [Podium](#), we get his take on the subject.

Presentation

I'm Rich Cardone and I'm a researcher at Watson Research Center. I'm based in Austin, Texas.

In the United States, the current voting systems are really in a state of flux. After the 2000 presidential election, there was a great move to change the voting systems and to move from paper to electronic voting systems. The electronic voting systems have exhibited problems in accuracy and reliability, and they're pretty expensive. But

probably more of concern to most people is the fact that there are security problems and people don't necessarily trust the system and they don't have a way of verifying elections with lots of the systems.

The idea of "verifiable" really springs from the idea of the basic requirement that the public has to trust their voting system. They have to believe the voting system that they're using is going to return fair results and accurate results. Without that social buy-in, the system breaks down.

So the verifiability goes to the question of, is there a way to cross-check or double-check the results?

The approach that we're looking at is to stick with the touch-screen style voting systems. They're called Direct Recording Electronic, DRE systems, but improve them.

So a voter's experience would be, they would walk into a booth, or at least a private area, and they would have some sort of a console with some sort of an input device and they would be presented a ballot on the display, and they would have a way of navigating and making their choices.

We really need to be very sure about the software we're

running on those machines, and that has two components. You have to know exactly what software is on the machine and then you have to believe that that software is correct.

The way we're approaching the problem is to use the trusted computing platform, which is an industry standard, as the way to verify that the code that gets loaded is the code that you installed on the machine.

And the second component is you make all the code on the machine open source. And along with that is the idea to use commodity hardware so that keeps down the costs and also leads to the familiarization that people have with the hardware.

As far as gaps in the technology: To actually put a system together that is manageable and configurable and easy enough to manage for the intended users, which would be the election authorities in various parts of the country and various parts of the world, you have to spend a lot of time thinking about how do these systems get their code installed, how do they get their code updated, how do you deploy the systems in a secure way?

I think probably the biggest gaps is in that area. The actual basic technology of trusted computing platform and

the hardware it depends on, the chips that are in the computers, those have been out for quite a while and they're pretty well understood and pretty well behaved.

There will probably also be gaps somewhere in the software stack, but each layer of the software stack checks the layer above it before it invokes it.

So, for instance, the bios checks the boot loader and the boot loader does the same thing with the operating system. So this happens all the way up the chain. You form a chain of trust. We know it goes all the way up to the operating system. There are versions of Linux, for instance, Open Source Linux, that have the whole software stack up to them covered. We need to go further. We need to go right to the application. There will still be some sort of a technical challenge to make sure it works with the software we would use.

The other part of the question is whether it's going to require some technicians or poll workers to occasionally tweak the machine in some manner or to adjust things. That should not happen. These machines, when you turn them on, they should run for the 12 or 16 hours that day or, if there's prolonged voting over period of time, for that amount of time. And that's doable.

The ultimate social value is that the public has confidence in the electoral process.

The idea is you publish all the results from every machine in the system that you have. It would not be outside the resources of many people to actually run the tally themselves and say, you know, given all the data that they had, it seems like every ballot that was cast is accounted for and here they are and the tallies add up.

This is not a problem of technology. We have all the technology we need to solve this problem. Computers can add much quicker than we need for voting machines to add up votes.

Of course it requires some technology that isn't there and combinations of technology that exist but used in ways that haven't been used before. But if we're willing to, you know, fly jetliners by wire – these are electronic devices that people's lives depend on – why can't we tally votes up with electronic machines? I mean that's what electronic machines can do.

This is a need that needs to go away. We need to just get over this, fix it and move on to the next problem.

IBM. Podium.

Series producer: Barbara Finkelstein

Music: [Comparsa](#) by Kevin MacLeod