Impossible —
Secure Computer Votecounting

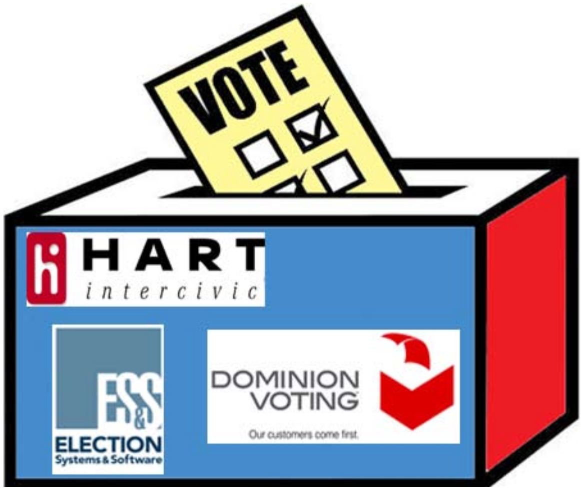
by Dr. Ethan Scarl

"People make mistakes, but computers do not" is the classic justification for America's great investment in counting its votes electronically.

But a computer's code can be corrupted, in many and subtle ways, and fraudulent counting can have extremely high payoffs from public policies enacted by those elected, with very low risk of detection.

So here are the many steps would it actually take to know that electronic elections are accurate, and honest.

1. **Check the design of the computer's program.** This requires the often impossible task of acquiring and evaluating the system's Requirements and Specifications documents. Without these, one has only their intuition of what the machine is intended to do.
2. **Obtain and Validate Source Code** for all machines. This is the code that is written and readable by humans. Without it, there is no way to test the system rigorously; testing becomes a haphazard business that might find some, but not all faults, but can never prove correctness. Unfortunately, American voting machine corporations are allowed to declare their code "proprietary," meaning that it cannot be examined by citizens or even by government officials who contract for it.
3. **Verify the Source Code.** This ensures that the program correctly implements its requirements. This is done by human expertise, often with software assistance, but the huge number of possible paths through the code typically makes any guarantee impossible.
4. **Verify verification software.** Although verification software is not an easy vehicle for intentional fraud, it could hide corruption in the voting software.
5. **Compile only verified source code.** The Compiler is the software that translates the human-written source code into the Object Code that actually runs on the vote-counting machines. It is critical that this object code actually came from the verified source. Protecting both source and object Codes during transport to and from the compiler is one of several "chain of custody" challenges to trusting a votecount.
6. **Certify code on all machines** that count votes. Even with careful transport, we need to ensure that this object code actually came from verified source code. This requires checking every voting machine and tabulator, a difficult task in both principle and practice.
7. **Be sure that the compiler itself is clean.** A corrupted compiler can insert erroneous or malicious code into the object code while compiling legitimate source code. If the compiler is compromised, we can verify and validate source code till doomsday and never spot the most blatant fraud. This means that we have to do the same verification on the compiler's own code, a daunting prospect rarely attempted.
8. **Guarantee all machine hardware.** Can we be certain there is nothing hiding in the memory cards or mother boards? Or even chips hiding in the cabling? Maybe something that accepts remote inputs? This is difficult to rule out definitively.
9. **Guarantee the overall chain of custody.** Finally, we have to guarantee our loaded machines' physical and electronic isolation, allowing no access before Election Day. There are histories of vendor agents installing uncertified code "patches" in "selected" precincts just before elections. Any patched system (not just the patch) needs to repeat this entire validation process.



graphic: Inspired by Ethan Scarl

If each of these steps is not completed, the election results may be erroneous or fraudulent in ways virtually impossible even for expert observers to detect. Keep in mind that if any machine can be remotely accessed, then machines and code can be corrupted from step 5 on down, and even open source becomes irrelevant.

The inescapable conclusion is that guaranteeing a secure and uncorrupted computerized votecounting system has never been done, nor even attempted. To do so would be fantastically difficult and expensive.

The inescapable conclusion is that guaranteeing a secure and uncorrupted computerized votecounting system has never been done, nor even attempted.

Therefore, the only plausible way to have any confidence in computerized votecounting is to test its output with statistically valid hand-counted auditing of paper ballots.

Alternatively, we could save money and dramatically reduce risk by hand-counting all ballots in the first place, under proper supervision, as we once did and as is now done in many other countries without problem.

Dr. Ethan Scarl is a computer scientist on the AfD Council and has pursued election integrity issues since 2008.