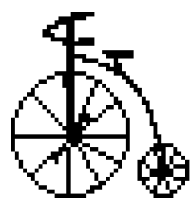
e-Voting Machines: Measuring *What*?

Matt Bishop
Dept. of Computer Science
University of California, Davis
Davis, CA 95616-8562

phone: +1 (530) 752-8060

UC Davis
Computer Security Laboratory



Opening Thought

There's no sense in being precise when you don't even know what you're talking about.

— John von Neumann

Outline

- What is assurance?
- Attacking an e-voting system
- Assurance and measurement

What is "Assurance"?

Trustworthy entity: one for which there is sufficient credible evidence leading one to believe that the entity will meet a set of given requirements

Assurance: confidence that an entity meets its requirements based on specific evidence provided by the application of assurance techniques

Measures of Assurance

- What is target audience?
  - Computer scientists, election officials, politicians, *average person*
- How convincing is assurance evidence to target audience?

High Assurance

- Requirements assurance
  - Defines "security policy" (among other things)
- Design assurance
  - Show design satisfies requirements
- Implementation assurance
  - Show implementation matches design

High Assurance

- Deployment, operation assurance
  - Show deployment, operation meets requirements
- Maintenance assurance
  - Show maintenance (upgrades, *etc.*) meets requirements

Development Process

- Gives assurance that holes not introduced during development
  - Necessary but not sufficient (think "Trojan Horse")
- Must also check assumptions underlying development correctly reflect environment in which systems are developed

Key Ideas

- Requirements tracing
- Layering: build on …
  - Trusted computing base
    - What your system trusts
  - Reference monitor
    - Complete, verifiable, tamperproof
  - Security kernel

Clichéd but True

- You can never prove an actual system is secure
- You can only prove (or argue) one layer is secure *with respect to the layer beneath it*
- Hardware can have bugs, too …

Top 10 New Intel Slogans for the Pentium

9.99999732. It's a **Flaw**, Not a Bug !
8.99991633. It's Close Enough, We Say So
7.99994146. Nearly 300 Correct Opcodes
6.99998315. You Don't Need to Know What's Inside
5.99998351. Redefining the PC—and Mathematics As Well
4.99999990. We Fixed It, Really!
3.99982459. Division Considered Harmful
2.99915236. Why Do You Think They Call It *Floating* Point?
1.99991017. We're Looking for a Few Good Flaws
0.99999998. The Errata Inside

Testing

- Part of assurance
- Can *never prove security*; can only *disprove*
- Penetration testing: captures environment, policy and procedures, *etc.*
  - Can be very structured using Flaw Hypothesis Methodology
  - Other testing methods typically don't capture these

Vulnerabilities Models

- Research Into Secure Operating Systems (Abbott, 1976)
- Program Analysis project (Bisbey, 1978)
- NRL Model (Landwehr, 1989)
- Aslam's Model (Aslam, 1995)
- UC Davis Model (Bishop *et al.*, 1999+)

Attack Models

- Requires/Provides models of attack:
  - Attack *requires* certain conditions to hold; process can have capabilities that satisfy these
  - A successful attack *provides* the attacking process certain capabilities

Attacking an e-Voting System

- Benaloh's paper, first phase

"Each voter prepares and casts an encrypted ballot that represents the voter's intended selection. Once encrypted, these ballots–and even the identity of the voter that cast each one–can be made public."

Availability

- Systems are to be at polling stations and are to be running
  - Can I knock out power?
  - Can I put my own lock on the door?
  - Will poll workers all get sick (how many poll workers signed up)?
  - Can I do something to the machines *before* the election, while they are in storage?

Voting

- Some mechanism interacts with the user to prepare a ballot
  - Can handicapped user be able to use system (visual cues for blind, auditory cues for deaf)?
  - Is the ballot being voted on the same as the sample ballot mailed to voters?

User Choice

- Users can express choices on system
  - What happens when users given wrong ballot?
  - Can non-technical users use the system?
  - Can the machine misrecord choices?
    - Miscalibrated touch screens
    - Software bugs (or worse)

Identity

- Identity of voter not tied to ballot
  - Are serial numbers tied to voters?
  - Can we embed a marker in ballot to reveal order of voting?
  - How does the system handle provisional ballots?

More Identity

- What happens if someone votes for an odd write-in candidate in addition to the votes for which you are being paid?
  - Assumes I can see the raw ballots
- Will more than 1 voter vote at the precinct?
  - Common assumption for non-electronic voting, too

Privacy

- Voter cannot see how others voted
  - Does machine *wipe* vote from all parts of the system that interact with voters?
  - Does a security kernel control access to previous votes?

Accuracy

- Voters' choices are accurately recorded
  - How does the voter verify their vote?
    - Screens do not show actual ballot, but only a representation
  - How do you *know* the voter verifies vote?
  - When is out-of-bands media used?
    - Paper normally used only for recounts (1% or challenge)

Recounts

- Use trusted media for the recount
  - What is the media?
    - California (formerly): generate paper ballots from machine, count them, compare
  - Which is authoritative, paper or bits?
    - Bits get scrambled
    - Paper gets mangled, smudged, marred unreadable

Attacking an e-Voting System

- Benaloh's paper, second phase

"Once all participating voters have cast their encrypted ballots, the set of encrypted ballots is cryptographically processed to produce a tally and a proof that the tally matches the set of ballots cast. In some cases, the original decrypted ballots are revealed, but the individual associations between the revealed ballots and the identity of voters are removed."

Software

- Works correctly
  - Standard tricks: give bad input, type weird key combinations, pull out wires or plug and see what happens, try boundary cases (drag finger over 3 boxes to edge, hit two boxes at same time), and so forth
  - Be perverse and look for assumptions

Proof Is Correct

- Proofs of software, system, protocols, etc.
- How do you know?
  - 1879: Four-Color Conjecture proved!
    - 1900: error found
  - 1890s: Fermat's Last Theorem proved!
    - 1900s: proof disproved
  - Lots of other examples

Proof is Convincing

- *Proofs, assurance evidence are also social and how convincing they are depends on audience*
  - Good paper:

    De Millo, Lipton, and Perlis, "Social Processes and Proofs of Theorems and Programs", *Communications of the ACM* **22**(5) pp. 271–280 (May 1979).
  - In English: what would convince you would not necessarily convince my father
  - Attack: challenge proof, sow doubt

Sanitization

- Revealed ballots are *completely* sanitized
  - Is revealing ballots legal?
  - Do you really take into account *all* the external knowledge an attacker can have?
  - Can you define and counter all inferencing attacks?
  - Sanitization (de-identification) is a *very* deep and tricky problem!

Key Management

- System may use key management scheme
  - Default keys hardcoded in memory
    - Diebold: 1997, 2003, **forcefully** pointed out in 2004, still there in 2005
  - PKI: how do you know keys are properly bound to identities?
    - Identity theft
    - Prime the pump with a fake key

Reporting Results

- Usually preliminary results are reported and *publicly* accepted as final
  - So can I make the final ones differ from these to undermine public confidence?

Measure X Parable

Hypothetical Example

- SSL set to provide *confidentiality*, not integrity, for phoning in initial tallies
  - I intercept the call, send in bogus results, Measure X wins overwhelmingly
- Official counts are on flash cards
  - They arrive at county seat, get counted
  - Oops! Official tally: Measure X lost
- And no-one in Davis trusts the election process again

Disclaimer: Yolo County does not use DREs!!!

Measures

What is the effect of changing over to an e-voting system?

Leads to question of how secure an *election process* is

Process vs. Machines

- Machine is component of process
  - Policies, procedures can be designed to mitigate/eliminate threats from machines
- Do we measure qualities, properties of machine or process?
  - Most work focuses on machine
  - Some work focuses on process

Consistency

- Differing jurisdictions require different measures
  - Maryland can revote *precincts* if problems arise (court order only?)
  - California cannot; State Supreme Court can order *entire statewide election* rerun
- How does this affect the measurement of California's and Maryland's processes?

Certification

- Need to trust evaluators
  - ITAs don't seem to be doing as good a job as they should
- Need to certify to meaningful standards
  - Standards lack threat, system models; mix functional, testing requirements
  - Standards certify machines, not processes; processes can weaken secure systems

Usability

- *Critical to security*
  - Especially important here as *many operators will be computer-illiterate* or *non-technical* and employed only for one day (poll workers)
- Secure systems operated non-securely are non-secure (to put it mildly)

Transparency

- Must be as clear to voters as current system
- Anyone can observe *every* step of election except:
  - With DREs, cannot observe tallying of votes at per machine level
    - May be at per precinct level
  - With paperless DREs, cannot verify those tallies either

What's the Question?

- Not "how secure is this system"
- Right question will have several parts:
  - What properties do I care about?
  - What is the ideal for those properties (taken as a whole)?
  - How close to that ideal can we come?
  - How do we convince others that our measurements are good?

Final Random Thoughts

- Biba
- Noninterference and nondeducibility
  - For preventing information leakage, such as deduction of previous votes
- Data sanitization, de-identification; database inferencing
  - If you want to post ballots (which may not be legal!), you need to sanitize them

Conclusion

- We need to measure with respect to system development, and *not* just measure the end result
  - Measures must be *against proper set of requirements*
- We need to *design and build e-voting systems in such a way that we can measure security properties*

Conclusion

- We need to think in terms of *elections that use e-voting machines* and not about e-voting machines
  - Measures must take target environment into consideration
  - View the election process holistically

Closing Thought

To those accustomed to the precise, structured methods of conventional system development, exploratory development techniques may seem messy, inelegant, and unsatisfying. But it's a question of congruence: precision and flexibility may be just as dysfunctional in novel, uncertain situations as sloppiness and vacillation are in familiar, well-defined ones. Those who admire the massive, rigid bone structures of dinosaurs should remember that jellyfish still enjoy their very secure ecological niche.

— Beau Sheil, "Power Tools for Programmers"