March 12, 2014

Requirements for an Election

- Voter validation (authenticated, registered, has not yet voted)
- Ballot validation (voter uses right ballot, results of marking capture intent of voter)
- Voter privacy (no association between voter, ballot; includes voter showing others how he/she voted)
- Integrity of election (ballots not changed, vote tallied accurately)

Requirements for an Election

- Voting availability (voter must be able to vote, materials must be available)
- Voting reliability (voting mechanisms must work)
- Election transparency (audit election process, verify everything done right)
- Election manageability (process must be usable by those involved, including poll workers)

Add In E-Voting

- System must meet state certification requirements
  - Usually these incorporate the FEC standards
- Systems used must be certified
- Systems must be available on Election Day
  - No re-runs allowed (in California, at least—some states apparently do allow them)
- Systems must be "secure"
  - Properties must hold in face of (limited) conspiracy to undermine them

Assurance

- Provide sufficient evidence of assurance to target audience that using e-voting systems makes elections at least as secure, accurate, etc. as current elections
- Who is "target audience"?
  - Computer scientists, election officials, politicians, *average person*

Standards

Each state sets its own; most based on Federal standards

- Performance and Test Standards for Punchcard, Marksense, and Direct Recording Electronic Voting Systems (1990)
- Voting Systems Performance and Test Standards (2002)
- Voluntary Voting Systems Guidelines (2005)
  - Took effect Dec. 2007
- New ones under development (time frame uncertain)

Why Standards?

If systems are certified to meet standards, then people can have confidence they work!

- How good are the standards?
- How good is the testing?

Current Standards

- Goal: "address what a voting system should reliably do, not how system components should be configured to meet these requirements"
- Security concerns that have been raised, including:
  - System integrity during build and deployment
  - Voter anonymity
  - Access control policies
  - Availability
  - Poor design and implementation
  - Data transmission
  - Language
  - Unclear bases

System Integrity

- No procedural mechanisms required to ensure the software submitted for qualification is the exact software used in production units
- Integrity of ROMs must be validated before each election
- No requirement that integrity be maintained throughout election

Consequences

- In 2006, several California counties used uncertified software
  - Diebold downloaded last-minute fixes just before an election
- Also happened in other states such as Indiana and Colorado

Availability

- Required: $\frac{MTBF}{MTBF+MTTR} \geq 0.99$ "during normal operation for the functions indicated above"
  - Reliability: measure MTBF over at least 163 hours
  - Mathematical model to predict availability (vendor); validate model (testing authority)

Problems

- Testing done under laboratory conditions
  - Actual conditions of use may be different
  - Physical attacks like yanking wires or jamming cards typically not tested
- Availability models are problematic
  - Method of validating model not specified; up to tester

Unclear Bases

- Some numbers given but not explained
- Example: "achieve a target error rate of no more than one in 10,000,000 ballot positions"
  - Why this? Why not 1,000,000 or 100,000,000?
- Determine MTBF over 163 hours of testing
  - Again, why 163? Why not 14, or 48, or 168?

Lack of Threat Model

Against what threats should the systems be protected?

- Standards silent on this model
- Without it, basis for many requirements unclear and requirements themselves vague

Lack of System Model

Key question: in what environment, and under what processes, will the system be used?

- Standards also silent on this model
- Without it, vague requirements about processes, procedures, assumptions

Testing for Conformance

- Testing performed by independent testing authorities (ITAs)
  - Vendors pay for testing
  - Vendors can choose any ITA certified as such
  - Testing methodology up to ITA

Diebold AccuBasic

- Intent: add a scripting language to a report writing facility on the AccuVote-OS optical scan and AccuVote-TSx DREs
- CA required that it be "not possible to compromise an election in any way through the (mis)use of AccuBasic, including an unintentional error or malicious AccuBasic script" (request for ITA review)

ITA Findings

- Three violations allow manipulation, reading data in global space but can only be exploited by modified AccuBasic object file
- Bounds checking on stack, heap segments not detected, but bounds checking performed inside the code
- Interpreters lack proper degree of error checking to identify, recover from key failures in damaged environment

ITA Findings

- "Three security vulnerabilities and a small number of requirements violations that were not capable of being exploited by malicious code or operators"
- TSx ready for election; AV-OS needs to have these problems corrected
- If memory cards not tampered with between AV-OS and GEMS, existing units ready for election

VSTAAB Independent Review

Led by David Wagner of UC Berkeley

- Asked questions:
  - What kind of damage can malicious person do to undermine election if he can arbitrarily change contents of memory card?
  - How can such attacks be neutralized?
- Found code problems:
  - Buffer overflows (12 in AV-OS, 8 in TSx)
  - Other problems (4 in AV-OS, 2 in TSx)

VSTAAB Findings

- 16 security problems in AV-OS, 10 in TSx
  - All code problems, easily fixed
- If you can tamper with memory cards, you can undetectably rig election
- TSx has memory cards digitally signed . . . using keys for which defaults are hard-coded
- Interpreters disallowed by FEC standards!

Summary

- ITA clearly missed many problems
- ITA report not very detailed ($\sim$ 5 pages); VSTAAB report very detailed ($\sim$ 33 pages)

CA Top-to-Bottom Review

Undertaken to "restore the public's confidence in the integrity of the electoral process and to ensure that California voters are being asked to cast their ballots on machines that are secure, accurate, reliable, and accessible."

Structure

- UC teams provided technical data for CA Secretary of State
  - UC Berkeley (Wagner): source code review, document review
  - UC Davis (Bishop): red team testing, accessibility testing
  - Both groups used people from around the country
- Secretary used this data and other data to make decision
  - Policies, procedures, and their implementation
  - Each county has its own

Goals of the Study

"to identify and document vulnerabilities, if any, to tampering or error that could cause incorrect recording, tabulation, tallying or reporting of votes or that could alter critical election data such as election definition or system audit data."
Assume attackers could be anyone (voters, poll workers, election officials, vendors, etc.)

Constraints

- Time
  - Exercise lasted 5 weeks for 3 vendors (ended July 20)
- Lack of information and vendor software
  - Some documents delivered on July 13
  - Some software delivered on July 18
- Secretary, staff *exceptionally* supportive throughout

Example Threats

Attacker modifies "firmware" to misrecord votes

Case 1: Paper trail modified to reflect misrecorded votes unless voter corrects it, so no discrepancies between paper and stored ballots

Case 2: Paper trail records correct vote, disagreeing with stored ballots, creating discrepancy

Results

"security mechanisms provided for all systems analyzed were inadequate to ensure accuracy and integrity of the election results and of the systems that provide those results"

Example: Diebold

- Election management server
  - Delivered unpatched
  - Not all security-related actions logged
  - Remotely accessible account that by default does not require password
  - GEMS users can conceal actions from GEMS logging
- Precinct count AccuVote-OS
  - Low-tech attacks to stop it from reading ballots

Example: Diebold

- AccuVote TSx
  - Physical security: bypass locks; disable printer
  - Firmware: overwritten; virus attack possible
  - Escalate privileges from voter to election official, and erase votes, close polls, etc.
  - Security keys: well-known key used as default
  - Malicious voter input: made machine act erratically (no time to craft working exploits)
  - Paper trail: can easily be put out of service; could destroy records before and after attack, in a way voters wont notice

What Secretary Bowen Did

- ES&S
  - Certification and approval for use withdrawn
  - ES&S could undergo testing
- Diebold, Sequoia
  - Certification and approval for use withdrawn
  - 1 system per polling place (to comply with HAVA)
  - Vendors could fix problems and request recertification
- Hart
  - Jurisdictions must reinstall all software and firmware on all systems before each election
  - Vendor must present procedures to prevent virus propagation and to harden system

Later Version: Diebold

- Diebold added cryptography in the version after the one California reviewed
  - Not examined in TTBR because it wasn't certified in California
- Florida did examine it as part of certification process
  - Led by Prof. Alec Yasinsac of Florida State University

The Crypto

Signature is a SHA-1 160-bit digest signed using RSA:

sign: write M, S_{2048}

where $S_{2048} = RSA(privkey, 0_{1888}|SHA1(M)_{160})$

The Crypto

Signature is a SHA-1 160-bit digest signed using RSA:

sign: write M, S_{2048}

where $S_{2048} = RSA(privkey, 0_{1888}|SHA1(M)_{160})$

verify: read M, S_{2048}

if $RSA(pubkey, S_{2048})_{160} = SHA1(M)_{160}$, accept M

The Crypto

Signature is a SHA-1 160-bit digest signed using RSA:

sign: write M, S_{2048}

where $S_{2048} = RSA(privkey, 0_{1888}|SHA1(M)_{160})$

verify: read M, S_{2048}

if $RSA(pubkey, S_{2048})_{160} = SHA1(M)_{160}$, accept M

But . . .

- *privkey* is 3
- Verify step above just checks the low-order 160 bits!

Summary

- Standards, testing are not enough
- You need to know what the systems are to do
- You need to know under what constraints they will need to function
  - Environment
  - Policies and procedures
- You need to know with what assurance you can trust the systems