CSci 5271 Introduction to Computer Security Day 12: Electronic voting

Stephen McCamant University of Minnesota, Computer Science & Engineering

Outline

Elections and their security

- System security of electronic voting
- Announcements intermission
- End-to-end verification
- More Unix access control

Elections as a challenge problem

- Elections require a tricky balance of openness and secrecy
- Important to society as a whole But not a big market
- Computer security experts react to proposals that seem insecure

History of (US) election mechanisms

- For first century or so, no secrecy
 Secret ballot adopted in late 1800s
- Punch card ballots allowed machine counting
 - Common by 1960s, as with computers
 Still common in 2000, decline thereafter
- How to add more technology and still have high security?

Election integrity Tabulation should reflect actual votes No valid votes removed No fake votes inserted Best: attacker can't change votes Easier: attacker can't change votes without getting caught

Secrecy, vote buying and coercion

- Alice's vote can't be matched with her name (unlinkable anonymity)
- Alice can't prove to Bob who she voted for (receipt-free)
- 🖲 Best we can do to discourage:
 - Bob pays Alice \$50 for voting for Charlie
 - Bob fires Alice if she doesn't vote for Charlie

Election verifiability

- We can check later that the votes were tabulated correctly
- Alice, that her vote was correctly cast
- Anyone, that the counting was accurate
- In paper systems, "manual recount" is a privileged operation

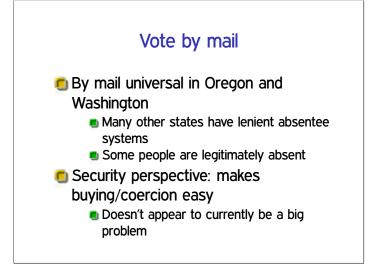
Politics and elections

- In a stable democracy, most candidates will be "pro-election"
- But, details differ based on political realities
- "Voting should be easy and convenient"
 Especially for people likely to vote for me
- "No one should vote who isn't eligible"
 Especially if they'd vote for my opponent

Errors and Florida Detectable mistakes: Overvote: multiple votes in one race Undervote: no vote in a race, also often intentional Undetectable mistakes: vote for wrong candidate 2000 presidential election in Florida illustrated all these, "wake-up call"

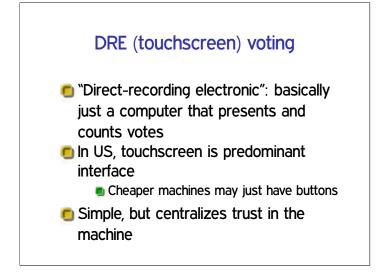
Precinct-count optical scan

- Good current paper system, used here in MN
- Voter fills in bubbles with pen
- Ballot scanned in voter's presence
 Can reject on overvote
- Paper ballot retained for auditing



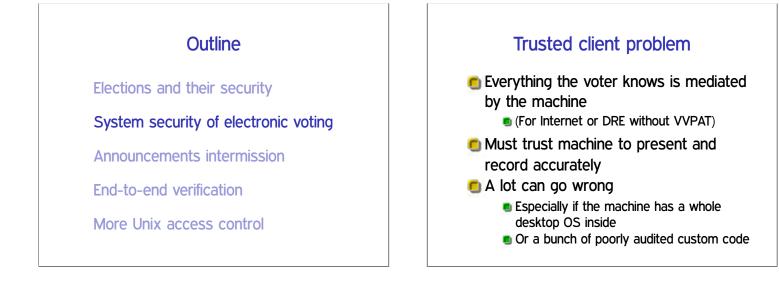
Vote by web?

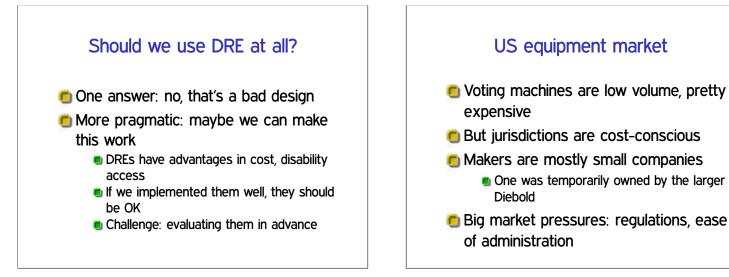
- 🖲 An obvious next step
- But, further multiplies the threats
- 🖲 No widespread use in US yet
- Unusual adversarial test in D.C. thoroughly compromised by U. Michigan team

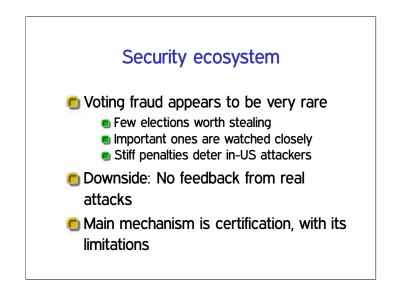


Adding an audit trail

- VVPAT: voter-verified paper audit trail
- DRE machine prints a paper receipt that the voter looks at
- Goal is to get the independence and verifiability of a paper marking system







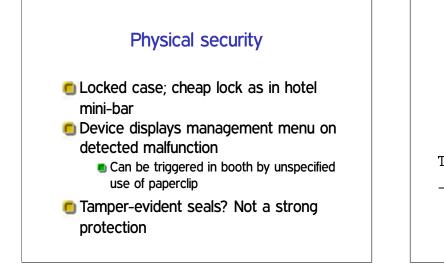
Diebold case study

Major manufacturer in early 2000s

- During a post-2000 purchasing boom
 Since sold and renamed
- Thoroughly targeted by independent researchers
 - Impolitic statement, blood in the water

Later state-authorized audits found comprehensive problems

Your reading: from California



Buffer overflows, etc.

Format string vulnerability
 "Page %d of %d"
 Was this audited?

TCHAR name; _stprintf(&name, _T("\\Storage Card\\%s"),

findData.cFileName);

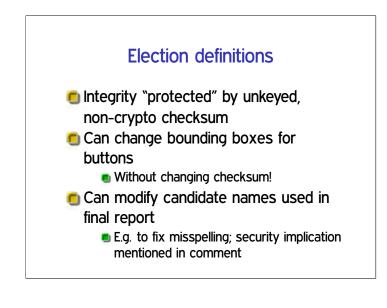


In management workstation software:

- SQL injection
- Authentication logic encoded only in enabled/disabled UI elements
 - E.g., buttons grayed out if not administrator
 - Not quite as obviously wrong as in web context
 - But still exploitable with existing tools

OpenSSL mistakes

Good news: they used OpenSSL
 Bad news: old, buggy version
 Insufficient entropy in seeding PRNG
 Good interface from desktop Windows missing in WinCE
 Every device ships with same certificate and password



Secrecy problems

- Limited, since the DRE doesn't see registration information
- But, records timestamp and order of voting
- Could be correlated with hidden camera or corrupted poll worker

Voting machine viruses

- Two-way data flow between voting and office machines
- Hijacking vuln's in software on both sides
- $\bigcirc \rightarrow$ can write virus to propagate between machines
- Leverage small amount of physical access

Subtle ways to steal votes

- Change a few votes your way, revert if the voter notices
 - Compare: flip coin to split lunch
- Control the chute for where VVPAT receipts go
- Exchange votes between provisional and regular voters

Outline

Elections and their security

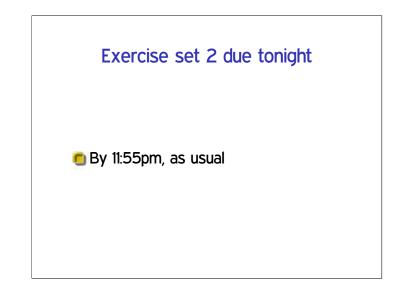
System security of electronic voting

Announcements intermission

End-to-end verification

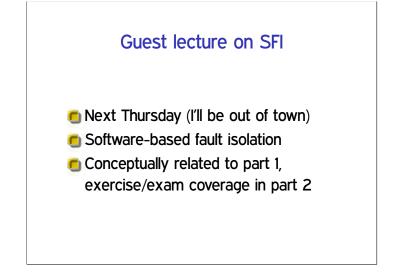
More Unix access control

BCLPR design changes Avoid unnecessary changes to benign functionality Restricting length or character sets of arguments Though, what is the benign functionality? Not a great candidate for privilege separation 500 LOC is not large! (LPRNG 45k, CUPS 371k)



Midterm exam Tuesday

- Usual location, starting promptly at 2:30
- Open books/notes/printouts, no computers or other electronics
- Sample exam w/solutions (2013's) posted



Outline

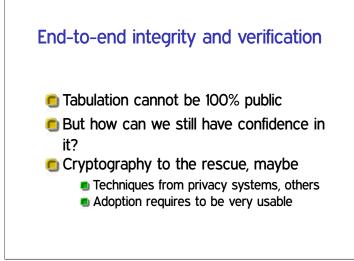
Elections and their security

System security of electronic voting

Announcements intermission

End-to-end verification

More Unix access control



Commitment to values

- Two phases: commit, later open
 Similar to one use of envelopes
- Binding property: can only commit to a single value
- Hiding property: value not revealed until opened

Randomized auditing

- How can I prove what's in the envelope without opening it?
- n envelopes, you pick one and open the rest
 - \blacksquare Chance 1/n of successful cheating
- Better protection with repetition

Election mix-nets

- Independent election authorities similar to Tor nodes
- Multi-encrypt ballot, each authority shuffles and decrypts
- Extra twist: prove no ballots added or removed, without revealing permutation
 Instance of "zero-knowledge proof"
- Privacy preserved as long as at least one authority is honest

Pattern voting attack

 Widely applicable against techniques that reveal whole (anonymized) ballots)
 Even a single race, if choices have

enough entropy

3-choice IRV with 35 candidates: 15 bits

- Buyer says: vote first for Bob, then 2nd and 3rd for Kenny and Xavier
 - Chosen so ballot is unique

Fun tricks with paper: visual crypto

- Want to avoid trusted client, but voters can't do computations by hand
- Analogues to crypto primitives using physical objects
- One-time pad using transparencies:



Scantegrity II

- Designed as end-to-end add-on to optical scan system
- Fun with paper 2: invisible ink
- Single trusted shuffle
 - Checked by random audits of commitments

Outline

Elections and their security

- System security of electronic voting
- Announcements intermission

End-to-end verification

More Unix access control

Special case: /tmp

- We'd like to allow anyone to make files in /tmp
- So, everyone should have write permission
- But don't want Alice deleting Bob's files
- Solution: "sticky bit" 01000

Special case: group inheritance

- When using group to manage permissions, want a whole tree to have a single group
- When 02000 bit set, newly created entries with have the parent's group
 (Historic BSD behavior)
- Also, directories will themselves inherit 02000

"POSIX" ACLS Based on a withdrawn standardization More flexible permissions, still fairly Unix-like Multiple user and group entries Decision still based on one entry Default ACLs: generalize group inheritance

Command line: getfacl, setfacl

ACL legacy interactions

- Hard problem: don't break security of legacy code
 - Suggests: "fail closed" Contrary pressure: don't want
- Contrary pressure: don't want to break functionality
 - Suggests: "fail open"
- POSIX ACL design: old group permission bits are a mask on all novel permissions

"POSIX" "capabilities"

- Divide root privilege into smaller (~35) pieces
- Note: not real capabilities
- First runtime only, then added to FS similar to setuid
- 🖲 Motivating example: ping
- 🖲 Also allows permanent disabling

Privilege escalation dangers

- Many pieces of the root privilege are enough to regain the whole thing
 - Access to files as UID 0
 - CAP_DAC_OVERRIDE
 - CAP_FOWNER
 - CAP_SYS_MODULE
 - CAP_MKNOD
 - CAP_PTRACE
 - CAP_SYS_ADMIN (mount)

