Electronic Voting & Security

Dr. Avi Rubin
Information Security Institute
Johns Hopkins University

Things everybody agrees on

- Punch card ballots result in mistakes by voters
- Computers can be useful in improving voting

Our democracy hinges on the quality of our voting systems and the confidence people have in them.

E-voting controversy

- We all want fair and secure elections
- Some disagreement on how to achieve
- My position:
  - There must be a voter-verifiable audit trail
  - Insider threat is real
  - Software is dangerous
  - Logic & Accuracy tests do not test security
    - e.g. can’t find Easter eggs

Election Procedures

- Good procedures are no excuse for deploying machines that are grossly insecure
- Procedures might detect tampering, but then what?
  - better to avoid tampering in the first place, if possible
- In the event that a procedure is not followed or does not work, the election should still be secure
- Not reasonable to place the burden of securing our elections on the poll workers

- Kim Zetter (Wired magazine) trained as a poll worker in California and found many lapses in security procedures

Last Election

- Washington Post 11/6
  - Software glitch in November's election in Virginia
  - Advanced Voting Solutions touchscreen machines
  - "Voters in three precincts reported that when they attempted to vote for [Thompson], the machines initially displayed an 'x' next to her name but then, after a few seconds, the 'x' disappeared. In response to Thompson's complaints, county officials tested one of the machines in question yesterday and discovered that it seemed to subtract a vote for Thompson in about 'one out of a hundred tries,' said Margaret K. Luca, secretary of the county Board of Elections. "

http://www.washingtonpost.com/wp-dyn/articles/A6291-2003Nov5.html

Last Election (Cont.)

- Indianapolis Star 11/9
  - Software glitch in November's election
    - 19,000 registered voters
    - 144,000 votes tallied
    - actual number of votes cast was 5,352
  - MicroVote touchscreen machines

http://www.indystar.com/articles/6/091021-1006-009.html

Voter verifiable audit

- enables recounts
- voter confidence
- harder to tamper with the election
- probably involves paper
- surprise recounts

The very piece of paper that is verified by the voter is used in the recount

Insider threat

- Easy to hide code in large software packages
- Virtually impossible to detect back doors
- Skill level needed to hide malicious code is much lower than needed to find it
- Anyone with access to development environment is capable
- Requires
  - background checks
  - strict development rules
  - physical security

Example

- Recent hidden trap door in Linux
- Allows attacker to take over a computer
- Practically undetectable change
- Discovered by rigorous software engineering process - not code inspection

```
                schedule();
                goto repeat;
        }
        if ((options == (__WCLONE|__WALL)) && (current->uid = 0))
                retval = -EINVAL;
        retval = -ECHILD;
end_wait4:
        current->state = TASK_RUNNING;
```

Example #2

- Rob Harris case - slot machines
  - an insider: worked for Gaming Control Board
- Malicious code in testing unit
  - when testers checked slot machines
    - downloaded malicious code to slot machine
  - was never detected
  - special sequence of coins activated “winning mode”
- Caught when greed sparked investigation
  - $100,000 jackpot

Software dangers

- Software is complex
  - top metric for measuring number of flaws is lines of code
- Windows Operating System
  - tens of millions of lines of code
  - new “critical” security bug announced every week
- Unintended security flaws *unavoidable*
- Intentional security flaws *undetectable*

Example #3

- Breeder's cup race
  - Upgrade of software to phone betting system
  - Insider, Christopher Harn, rigged software
  - Allowed him and accomplices to call in
    - change the bets that were placed
    - undetectable
  - Caught when got greedy
    - won $3 million

Case Study:

Diebold voting machines

Code analysis

- 56-bit DES in CBC mode with static IVs used to encrypt votes and audit logs (not compression, as Diebold claims in their “technical” analysis)

```
#define DESKEY ((des_key*)"F2654hD4")
```

- Unkeyed public function (CRC) used for integrity protection
- No authentication of smartcard to voting terminal
- Insufficient code review

// LCG - Linear Conguential Generator
// used to generate ballot serial numbers
// A psuedo-random-sequence generator
// (per Applied Cryptography,
// by Bruce Schneier, Wiley, 1996)

- BallotResults.cpp
Diebold Election Systems

// LCG - Linear Conguential Generator
// used to generate ballot serial numbers
// A psuedo-random-sequence generator
// (per Applied Cryptography,
// by Bruce Schneier, Wiley, 1996)

- BallotResults.cpp
Diebold Election Systems

“Unfortunately, linear congruential generators cannot be used for cryptography” - Page 369,
Applied Cryptography
by Bruce Schneier

“this is a bit of a hack for now.”

AudioPlayer.cpp

“the BOOL beeped flag is a hack so we don't beep twice. This is really a result of the key handling being gorped.”

WriteIn.cpp

“the way we deal with audio here is a gross hack.”

BallotSelDlg.cpp

“need to work on exception *caused by audio*. I think they will currently result in double-fault.”

BallotDlg.cpp

Code Fragment

```
void CBallotRelSet::Open(const CDistrict* district, const CBaseunit* baseunit,
const CVGroup* vgroup1, const CVGroup* vgroup2)
{
 ASSERT(m_pDB != NULL);
 ASSERT(m_pDB->IsOpen());
 ASSERT(GetSize() == 0);
 ASSERT(district != NULL);
 ASSERT(baseunit != NULL);
 if (district->KeyId() == -1) {
  Open(baseunit, vgroup1);
 } else {
  const CDistrictItem* pDistrictItem = m_pDB->Find
  if (pDistrictItem != NULL) {
   const CBaseunitKeyTa
   int count = baseunitTable.G
   for (int i= 0; i < count; i++
    const CBaseunit& curBa
    if (baseunit->KeyId() ==
     const CBallotRelationshipItem* pBalRelItem = NULL
     while ((pBalRelItem = m_pDB->FindNextBal
      if (!vgroup1 || vgroup1->KeyId() == -1
        (*vgroup1 == pBalRelIte
        (vgro
        *vg
        Add
      }
     }
    }
    m_CurIndex = 0;
    m_Open = TRUE;
   }
  }
 }
}
```

Zero Comments

Other problems

- Ballot definition file on removable media unprotected
- Smartcards use no cryptography
- Votes kept in sequential order
- Several glaring errors in cryptography
- Inadequate security engineering practices
- Default Security PINs of 1111 on administrator cards

SAIC Study

- 2/3 of the report redacted
  - due to “security” reasons
  - goes against a basic tenet of computer security
- Diebold claims everything will be fixed
  - if so, then why hide details of the report from the public?
- It is very important that the entire report be made public
- Long term plan, suggestion:
  - Maryland require SAIC to sign off on improved Diebold machines before using them

Recommendation #1

- Separate vote casting from tabulating
  - Touch screen machine produces paper ballot
    - need not be as trusted as today's DREs
  - voter can use or destroy
  - scanning and tabulating machine
    - small code base
    - open source
    - extensive testing and certification
    - different manufacturer from touch screen

Recommendation #2

- Transparency
  - Require designs of machines to be public
  - Require security audit of machines by qualified experts
    - Require public report of this audit
  - Require open source for vote tabulation code
    - necessary but not sufficient

Recommendation #3

- Quality control
  - Establish criteria for testing the expertise of manufacturers
    - NIST could play this role
  - Require source code analysis for certification
  - Establish standards for policies and procedures
    - Aim for simplicity:
      - The more complicated and burdensome, the less likely to be followed

Conclusions & Advice

- Security of voting should be a non-partisan issue
  - Only democrats have approached me:
    - Holt, Kucinich, Moseley-Braun, Kaptur, DNC
  - Too much is at stake for party politics
- Keys to future work on voting systems:
  - transparency
  - openness
  - accountability & audit
  - public review
- Computer Scientists and Politicians should work together

Additional slides

(if needed for Q & A)

Diebold's response

- The code we looked at was old and not the one that runs in their machines
  - We do not believe that
  - Several people have matched the version numbers
  - The code compiled and ran - no accident
  - SAIC looked at the "current" code and found the same flaws

Diebold's response

- These machines have been used in many elections with no problems
  - This says nothing about the security of the machines
  - Attacks are more likely to happen when more is at stake
  - You don't always know when someone has hacked the system

Diebold's response

- We ran the code on a different platform from the one used in the voting machines
  - Nothing in our analysis has to do with the fact that we ran the code
  - We only ran the code to see if it was real code
  - Since it compiled and ran on our machine, the platform had to be similar, but this is an unimportant point
  - This response by Diebold is an intentional diversion from the security problems in their machines

Diebold's response

- My role as an advisor to Votehere Inc. introduces bias into the study
  - I was on the technical advisory board of Votehere and 7 other security companies
  - Votehere is not a competitor of Diebold's
    - Johns Hopkins concluded in a review of the matter
  - My 3 collaborators had no affiliation with Votehere
  - Our results have been confirmed by the security community and the SAIC study
  - I resigned my advisory position and never had any financial gain from that relationship