Access Control and Privacy Policies (2)

Email: christian.urban at kcl.ac.uk
Office: S1.27 (1st floor Strand Building)
Slides: KEATS (also homework is there)

This Course is about "Satan's Computer"

Ross Anderson and Roger Needham wrote:

> "In effect, our task is to program a computer which gives answers which are subtly and maliciously wrong at the most inconvenient possible moment... we hope that the lessons learned from programming Satan's computer may be helpful in tackling the more common problem of programming Murphy's."

This Course is about "Satan's Computer"

Ross Anderson and Roger Needham wrote:

> "In effect, our task is to program a computer which gives answers which are subtly and maliciously wrong at the most inconvenient possible moment... we hope that the lessons learned from programming Satan's computer may be helpful in tackling the more common problem of programming Murphy's."

Murphy's computer

Satan's computers

User-Tracking Without Cookies

Can you track a user **without**:

- Cookies
- Javascript
- LocalStorage/SessionStorage/GlobalStorage
- Flash, Java or other plugins
- Your IP address or user agent string
- Any methods employed by Panopticlick

→ https://panopticlick.eff.org/

Even when you disabled cookies entirely, have Javascript turned off and use a VPN service.

User-Tracking Without Cookies

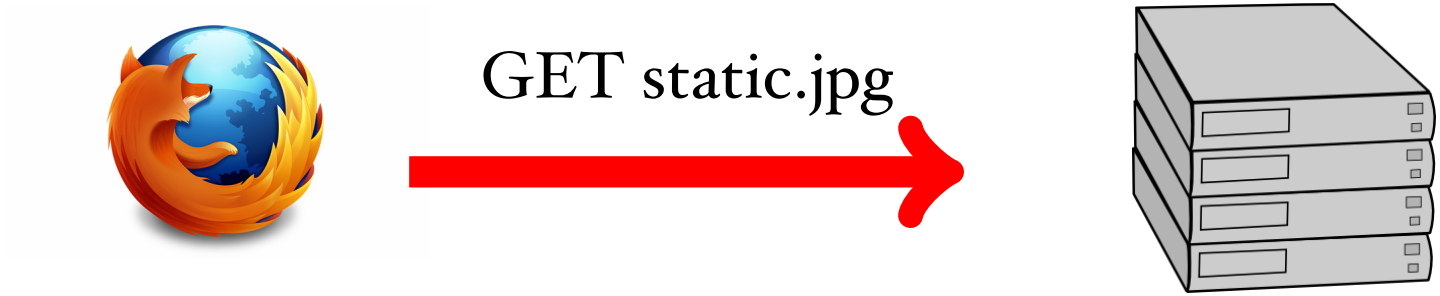
Can you track a user **without**:

- Cookies
- Javascript
- LocalStorage/SessionStorage/GlobalStorage
- Flash, Java or other plugins
- Your IP address or user agent string
- Any methods employed by Panopticlick

→ https://panopticlick.eff.org/

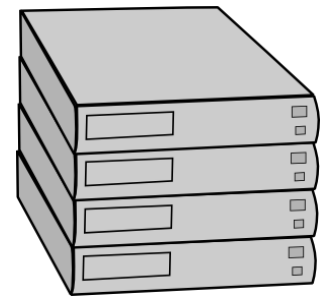
Even when you disabled cookies entirely, have Javascript turned off and use a VPN service. And numerous sites already use it (Google).

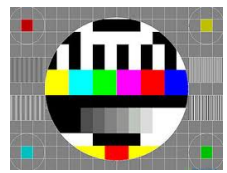
Web-Protocol

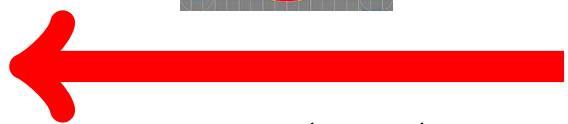


Web-Protocol

GET static.jpg



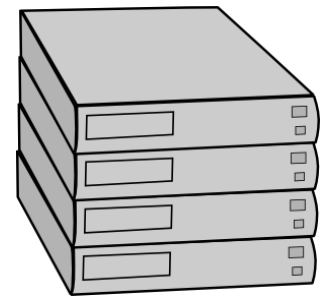


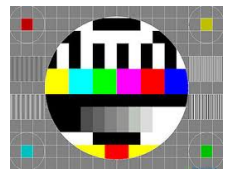


ETag: 7b33de1

Web-Protocol

GET static.jpg



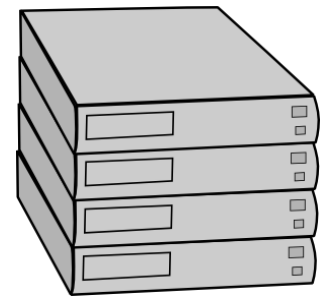


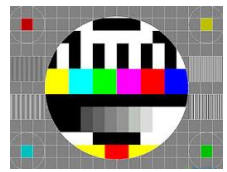
ETag: 7b33de1

GET static.jpg ETag: 7b33de1

Web-Protocol

GET static.jpg





ETag: 7b33de1

GET static.jpg ETag: 7b33de1

HTTP/1.1 304 (Not Modified)

Today's Lecture

online banking
solved

e-voting
unsolved

Voting as Security Problem

What are the security requirements of a voting system?

Voting as Security Problem

What are the security requirements of a voting system?

- Integrity

- The outcome matches with the voters' intend.
- There might be gigantic sums at stake and need to be defended against.

Voting as Security Problem

What are the security requirements of a voting system?

- Integrity
- Ballot Secrecy

Voting as Security Problem

What are the security requirements of a voting system?

- Integrity
- Ballot Secrecy

- Nobody can find out how you voted.
- (Stronger) Even if you try, you cannot prove how you voted.

Voting as Security Problem

What are the security requirements of a voting system?

- Integrity
- Ballot Secrecy
- Voter Authentication

- Only authorised voters can vote up to the permitted number of votes.

Voting as Security Problem

What are the security requirements of a voting system?

- Integrity
- Ballot Secrecy
- Voter Authentication
- Enfranchisement

- Authorised voters should have the opportunity to vote.

Voting as Security Problem

What are the security requirements of a voting system?

- Integrity
- Ballot Secrecy
- Voter Authentication
- Enfranchisement
- Availability

- The voting system should accept all authorised votes and produce results in a timely manner.

Problems with Voting

Integrity vs. Ballot Secrecy

Authentication vs. Enfranchisement

Problems with Voting

Integrity vs. Ballot Secrecy

Authentication vs. Enfranchisement

Further constraints:

- costs
- accessibility
- convenience
- intelligibility

Traditional Ballot Boxes

Traditional Ballot Boxes

they need a “protocol”

E-Voting

- The Netherlands between 1997 - 2006 had electronic voting machines
  (hacktivists had found: they can be hacked and also emitted radio signals revealing how you voted)
- Germany had used them in pilot studies
  (in 2007 a law suit has reached the highest court and it rejected electronic voting on the grounds of not being understandable by the general public)
- UK used optical scan voting systems in a few polls

E-Voting

- US used mechanical machines since the 30s, later punch cards, now DREs and optical scan voting machines
- Estonia used in 2007 the Internet for national elections (there were earlier pilot studies in other countries)
- India uses e-voting devices since at least 2003 ("keep-it-simple" machines produced by a government owned company)
- South Africa used software for its tallying in the 1993 elections (when Nelson Mandela was elected) (they found the tallying software was rigged, but they were able to tally manually)

A Brief History of Voting

- Athenians
  - show of hands
  - ballots on pieces of pottery
  - different colours of stones
  - “facebook”-like authorisation

  problems with vote buying / no ballot privacy

- French Revolution and the US Constitution got things “started” with paper ballots (you first had to bring your own; later they were pre-printed by parties)

Ballot Boxes

Security policies involved with paper ballots:

1. you need to check that the ballot box is empty at the start of the poll / no false bottom (to prevent ballot stuffing)
2. you need to guard the ballot box during the poll until counting
3. tallied by a team at the end of the poll (independent observers)

Which security requirements do paper ballots satisfy better than voice voting?

- Integrity
- Enfranchisement
- Ballot secrecy
- Voter authentication
- Availability

Paper Ballots

What can go wrong with paper ballots?

Paper Ballots

What can go wrong with paper ballots?

William M. Tweed, US Politician in 1860's
"As long as I count the votes, what are you going to do about it?"

Paper Ballots

What can go wrong with paper ballots?

Chain Voting Attack

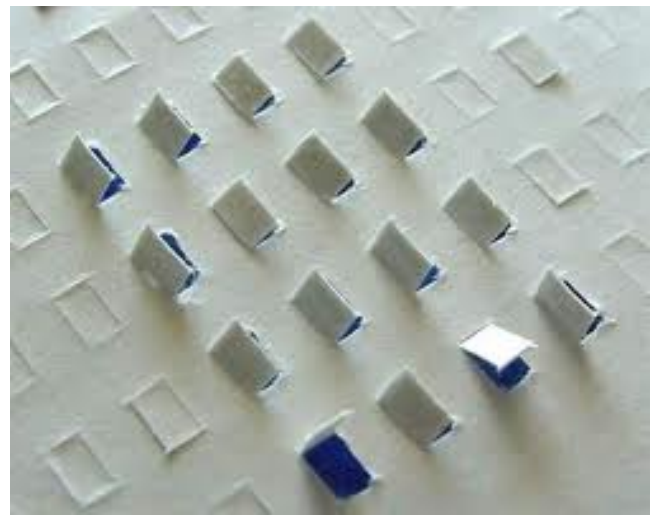
1. you obtain a blank ballot and fill it out as you want
2. you give it to a voter outside the polling station
3. voter receives a new blank ballot
4. voter submits prefilled ballot
5. voter gives blank ballot to you, you give money
6. goto 1

Mechanical Voting Machines

- Lever Voting Machines (ca. 1930 - 1990)

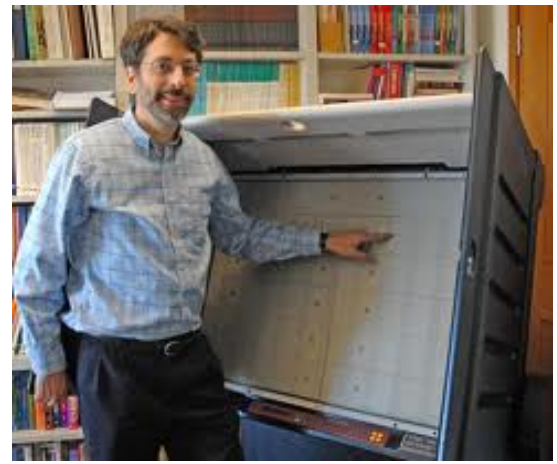
Mechanical Voting Machines

- Lever Voting Machines (ca. 1930 - 1990)
- Punch Cards (ca. 1950 - 2000)



Electronic Voting Machines

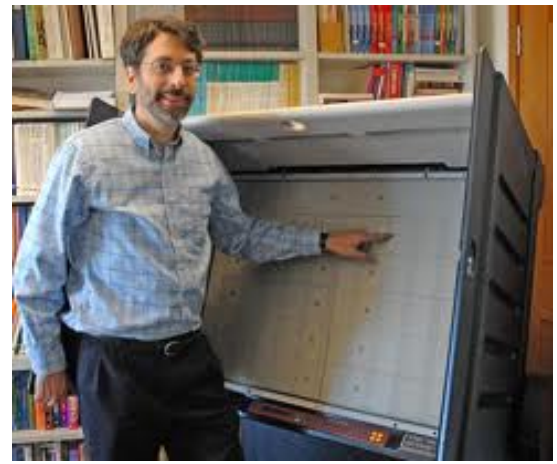
DREs



Optical Scan

Electronic Voting Machines

DREs



Optical Scan

all are computers

DREs

Direct-recording electronic voting machines
(votes are recorded for example on memory cards)
typically touchscreen machines
usually no papertrail

Diebold Machines

The work by J. Alex Halderman:

- acquired a machine from an anonymous source
- they try to keep secret the source code running the machine

Diebold Machines

The work by J. Alex Halderman:

- acquired a machine from an anonymous source
- they try to keep secret the source code running the machine
- first reversed-engineered the machine (extremely tedious)
- could completely reboot the machine and even install a virus that infects other Diebold machines
- obtained also the source code for other machines

Diebold Machines

What could go wrong?

Diebold Machines

What could go wrong? Failure-in-depth.

Diebold Machines

What could go wrong? Failure-in-depth.

A non-obvious problem:

- you can nowadays get old machines, which still store old polls
- the paper ballot box needed to be secured during the voting until counting; e-voting machines need to be secured during the entire life-time

Paper Trail

Conclusion:
Any electronic solution should have a paper trail.

Paper Trail

Conclusion:
Any electronic solution should have a paper trail.

You still have to solve problems about voter registration, voter authentication, guarding against tampering

E-Voting in India

Their underlying engineering principle is "keep-it-simple":

E-Voting in India

Their underlying engineering principle is "keep-it-simple":

Official claims: "perfect", "tamperproof", "no need for technical improvements" ,"infallible"

Lessons Learned

- keep a paper trail and design your system to keep this secure
- make the software open source (avoid security-by-obscurity)

  source code for Estonian e-vote included

  http://goo.gl/oRMHAI
- have a simple design in order to minimise the attack surface

Lessons Learned

- keep a paper trail and design your system to keep this secure
- make th
  securit
- have a
  attack

```
def analyze(ik, vote, votebox):

# TODO: implement security checks
# such as verifying the correct size
# of the encrypted vote

return []
```

Lessons Learned

- keep a paper trail and design your system to keep this secure
- make the software open source (avoid security-by-obscurity)

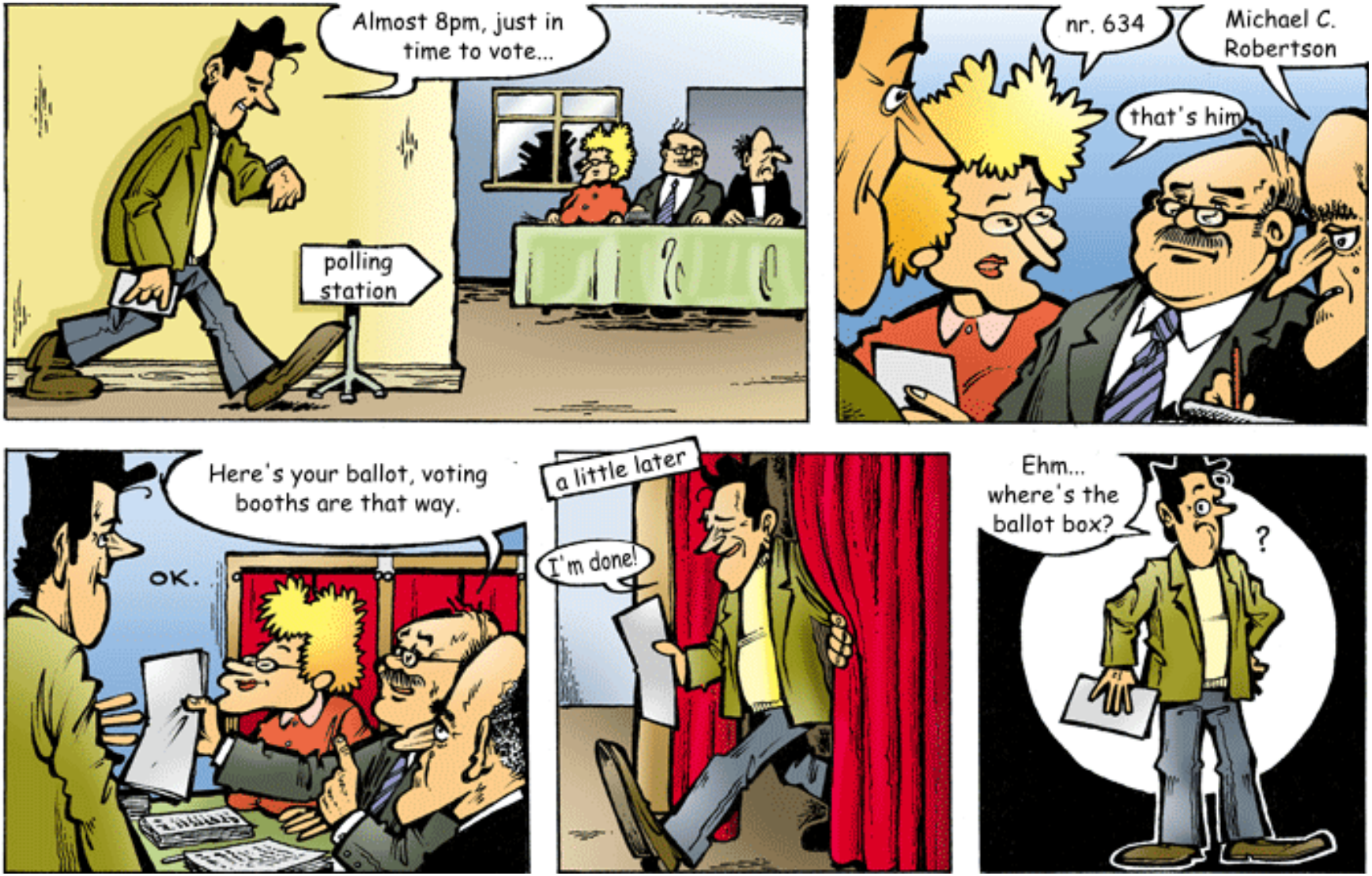
  source code for Estonian e-vote included

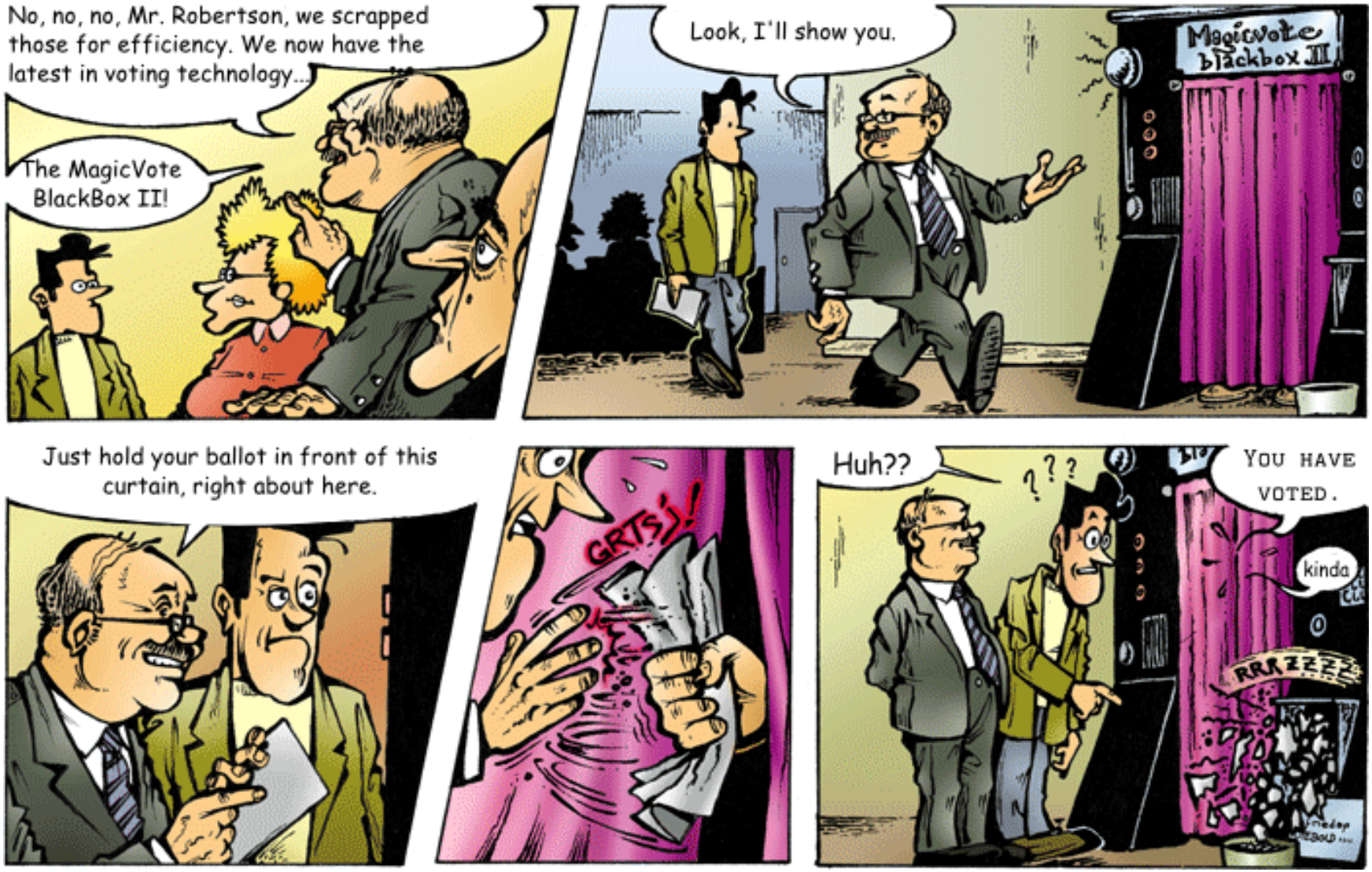
  http://goo.gl/oRMHAI
- have a simple design in order to minimise the attack surface

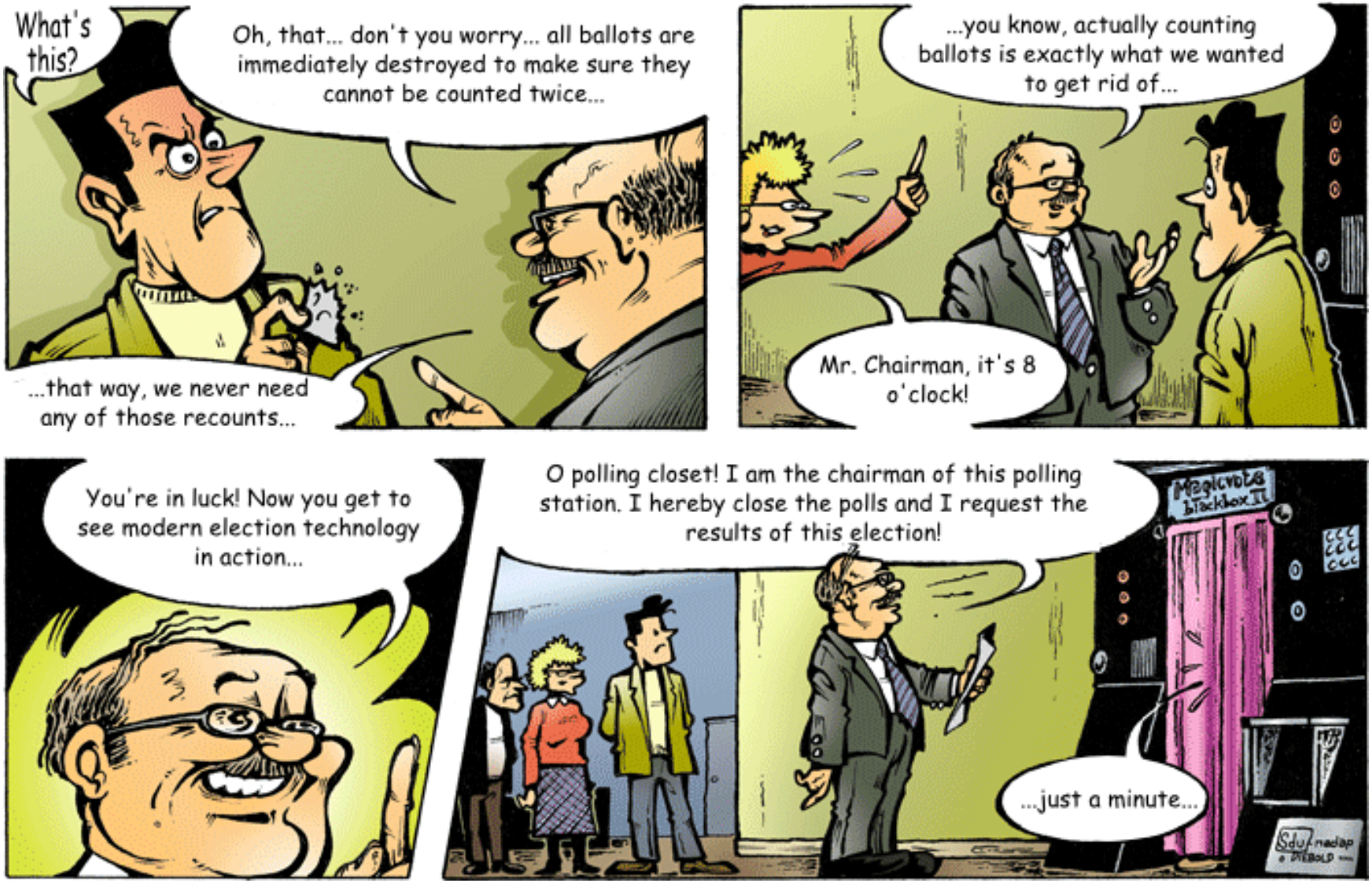
Online Banking vs. E-Voting

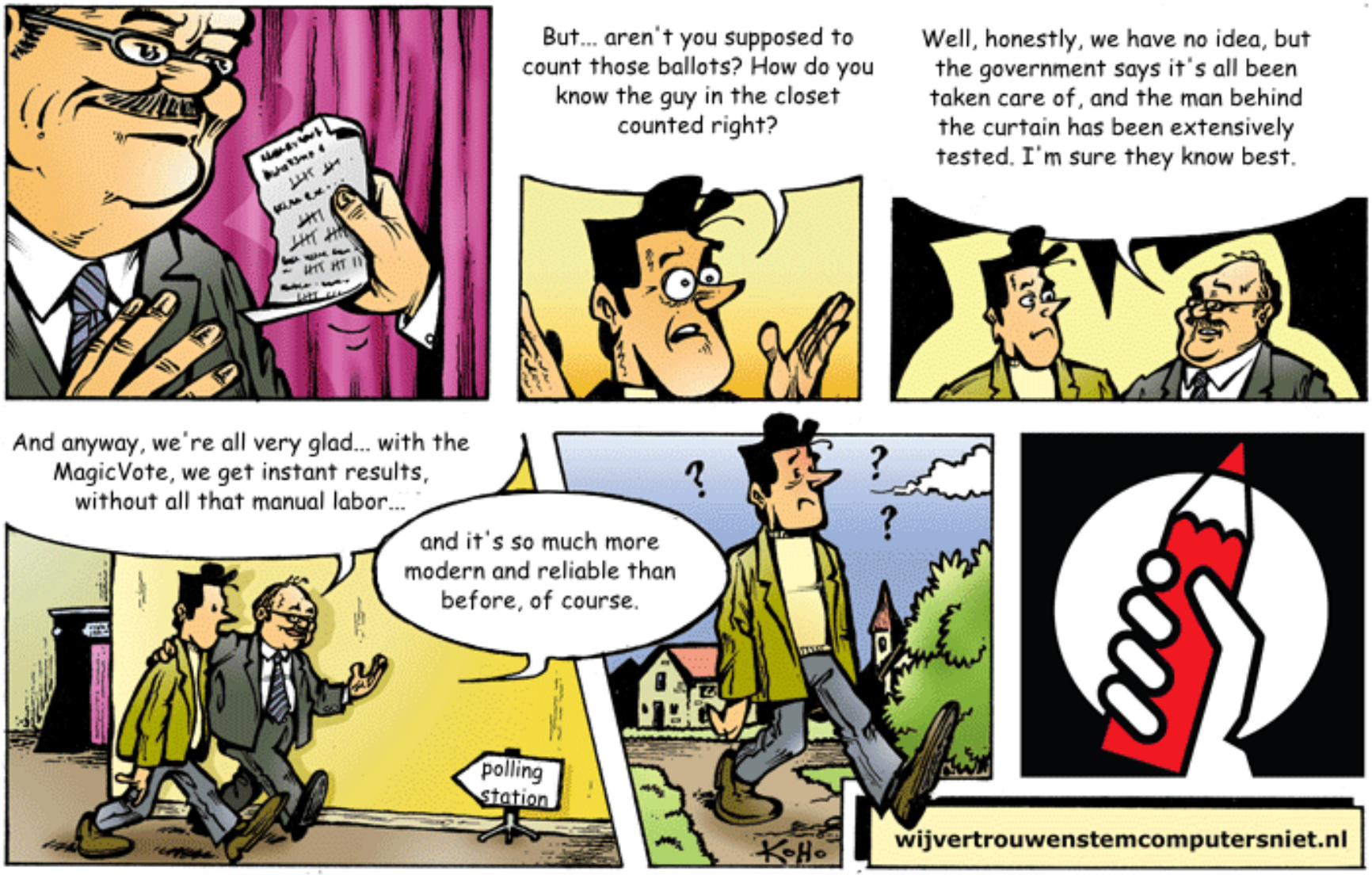
- online banking: if fraud occurred you try to identify who did what (somebody's account got zero)
- e-voting: some parts can be done electronically, but not the actual voting (final year project: online voting)

The adventures of citizen Michael C. Robertson




No, no, no, Mr. Robertson, we scrapped those for efficiency. We now have the latest in voting technology...
The MagicVote BlackBox II!
Look, I'll show you.
Magicvote blackbox III
Just hold your ballot in front of this curtain, right about here.
GRTS'Y!
Huh??
YOU HAVE VOTED.
kinda
RRRZZZZ


What's this?
Oh, that... don't you worry... all ballots are immediately destroyed to make sure they cannot be counted twice...
...that way, we never need any of those recounts...
...you know, actually counting ballots is exactly what we wanted to get rid of...
Mr. Chairman, it's 8 o'clock!
You're in luck! Now you get to see modern election technology in action...
O polling closet! I am the chairman of this polling station. I hereby close the polls and I request the results of this election!
...just a minute...
Magicvote blackbox II



Drawings: Koen Hottentot — Story: Rop Gonggrijp / Barry Wels — Color: Adam Swiecky — Translation: Jaap Weel

Unix-Style Access Control

How to do access control? In Unix you have

- you have users and you have groups/roles:
- some special roles: root

Unix-Style Access Control

- Q: "I am using Windows. Why should I care?"
  A: In Windows you have similar AC:

  administrators group
  (has complete control over the machine)
  authenticated users
  server operators
  power users
  network configuration operators

- Modern versions of Windows have more fine-grained AC than Unix; they do not have a setuid bit, but have runas (asks for a password).

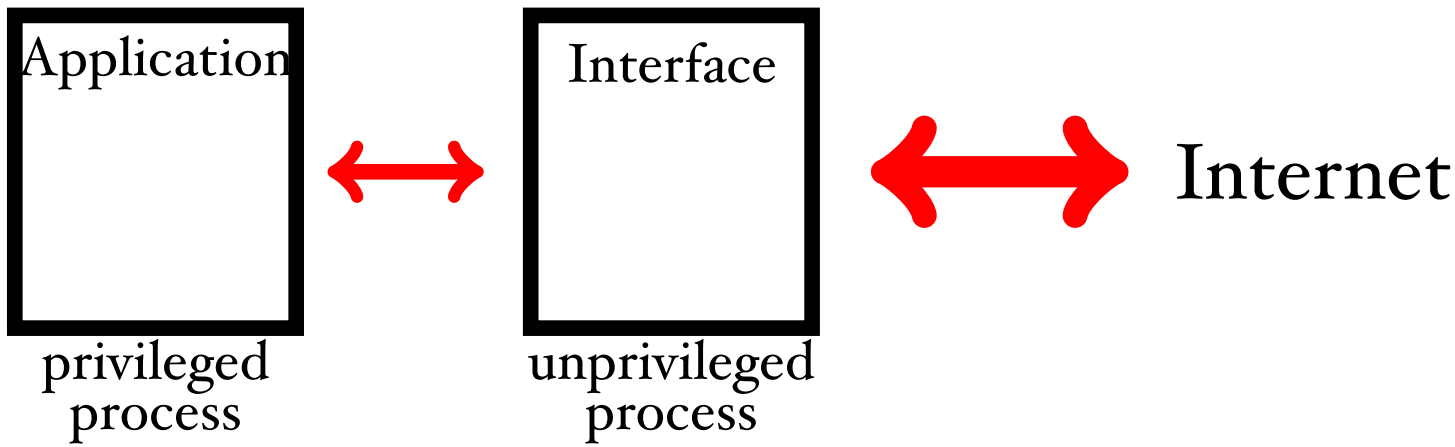
Unix-Style Access Control

- Q: "I am using Windows. Why should I care?"
  A: In Windows you have similar AC:

  administrators group
  (has complete control over the machine)
  authenticated users
  server operators
  power users
  network configuration operators

- Modern versions of Windows have more fine-grained AC than Unix; they do not have a setuid bit, but have `runas` (asks for a password).
- OS-provided access control can **add** to your security.

Network Applications: Privilege Separation



- the idea is make the attack surface smaller and mitigate the consequences of an attack

Lessons from Access Control

Not just restricted to Unix:

- if you have too many roles (i.e. too finegrained AC), then hierarchy is too complex
  you invite situations like...let's be root

- you can still abuse the system...

A "Cron"-Attack

The idea is to trick a privileged person to do something on your behalf:

- root:
  rm /tmp/*/*

A "Cron"-Attack

The idea is to trick a privileged person to do something on your behalf:

- root:
  rm /tmp/*/*

the shell behind the scenes:
rm /tmp/dir_1/$file_1$ /tmp/dir_1/$file_2$ /tmp/dir_2/$file_1$ ...

this takes time

A "Cron"-Attack

1. **attacker** (creates a fake passwd file)
   mkdir /tmp/a; cat > /tmp/a/passwd
2. **root** (does the daily cleaning)
   rm /tmp/*/*

   records that /tmp/a/passwd

   should be deleted, but does not do it yet
3. **attacker** (meanwhile deletes the fake passwd file, and establishes a link to the real passwd file)
   rm /tmp/a/passwd; rmdir /tmp/a;
   ln -s /etc /tmp/a
4. **root now deletes the real passwd file**

A "Cron"-Attack

1. attacker (creates a fake passwd file)
   mkdir /tmp/a; cat > /tmp/a/passwd
2. ro
   rm

To prevent this kind of attack, you need additional policies (don't do such operations as root).

should be deleted, but does not do it yet

3. attacker (meanwhile deletes the fake passwd file, and establishes a link to the real passwd file)
   rm /tmp/a/passwd; rmdir /tmp/a;
   ln -s /etc /tmp/a
4. root now deletes the real passwd file