Security Engineering (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

Defence in Depth

```
urbanc:$6$3WWbKfr1$4vblknvGr6FcDeF92R5xFn3mskfdnEn...:...
```

- hashes help when password databases are leaked
- salts help with protecting against dictionary attacks and help people who have the same password on different sites

- but they do not protect against a focused attack against a single password and also do not make poorly chosen passwords any better

Subtle Points

- in our web-application the salt needed to remain secret; in password files the salt is public
- the NYT has the "resource" unlocked at first and locks it depending on the cookie data
- our "web-application" has the resource locked at first, and unlocks it depending on the cookie data

How to Salt?

1salt	⇒	8189effef4d4f7411f4153b13ff72546dd682c69
2salt	⇒	1528375d5ceb7d71597053e6877cc570067a738f
3salt	⇒	d646e213d4f87e3971d9dd6d9f435840eb6a1c06
4salt	⇒	5b9e85269e4461de0238a6bf463ed3f25778cbba

- in Unix systems: `hash(salt + password)`, or even $\texttt{hash}^{1500}$`(salt + password)`

How to Salt?

```
1salt  ⇒  8189effef4d4f7411f4153b13ff72546dd682c69
2salt  ⇒  1528375d5ceb7d71597053e6877cc570067a738f
3salt  ⇒  d646e213d4f87e3971d9dd6d9f435840eb6a1c06
4salt  ⇒  5b9e85269e4461de0238a6bf463ed3f25778cbba
```

- in Unix systems: `hash(salt + password)`, or even $\texttt{hash}^{1500}$`(salt + password)`
- Bruce Schneier in cases messages are long:
  instead of `m` ↦ `hash(m)`,
  use `m` ↦ `hash(hash(m) + m)`

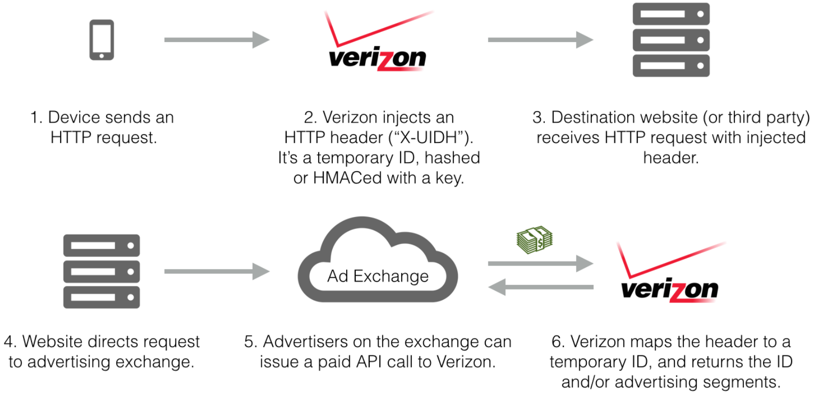
User-Tracking Without Cookies

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 also ...

Verizon



1. Device sends an HTTP request.

2. Verizon injects an HTTP header (“X-UIDH”). It’s a temporary ID, hashed or HMACed with a key.

3. Destination website (or third party) receives HTTP request with injected header.

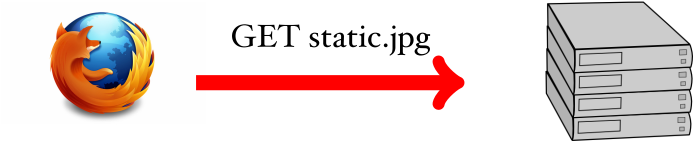
4. Website directs request to advertising exchange.

5. Advertisers on the exchange can issue a paid API call to Verizon.

6. Verizon maps the header to a temporary ID, and returns the ID and/or advertising segments.

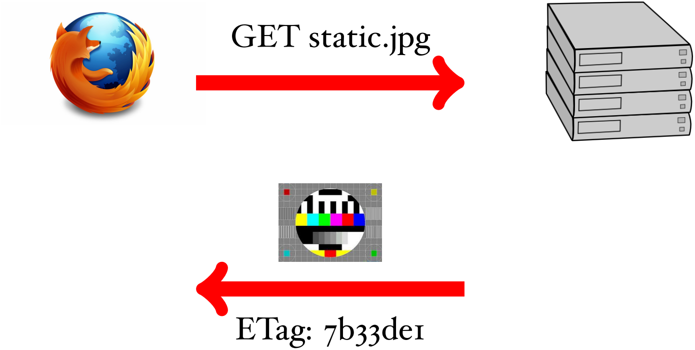
http://webpolicy.org/2014/10/24/how-verizons-advertising-header-works

Web-Protocol



GET static.jpg

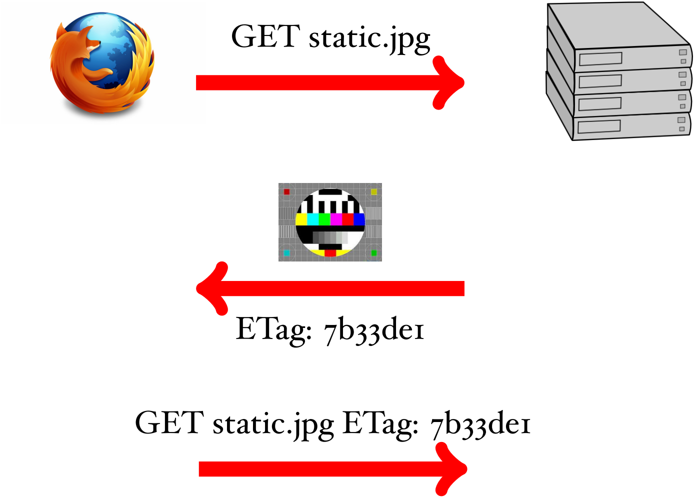
Web-Protocol



GET static.jpg

ETag: 7b33de1

Web-Protocol

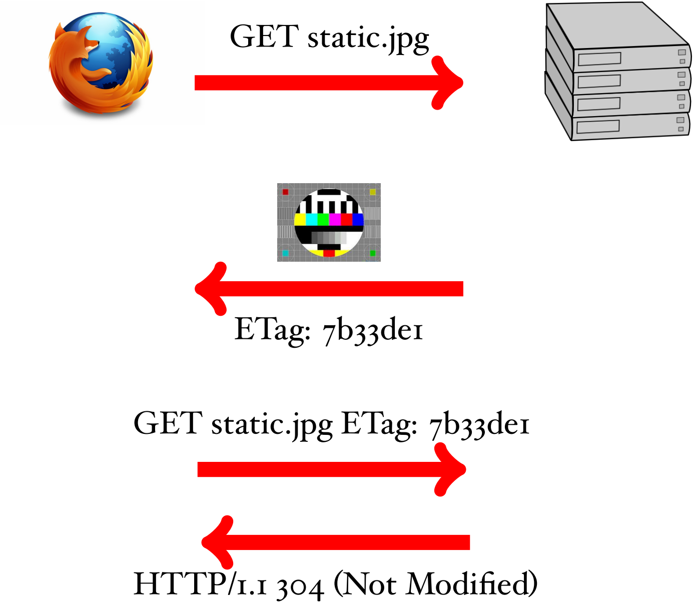


GET static.jpg

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	vs	e-voting
solved		unsolved

E-Voting

"Any electronic voting system should provide at least the same security, privacy and transparency as the system it replaces."

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

mechanical, but they need a “protocol”

Motives for E-Voting

- 76% of pensioners in the UK vote, but only 44% of the under-25s
- convenience
- speed

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 test polls, but abandoned any wide deployment

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)

E-Voting in Estonia

- worlds first general election that used internet voting (2007)
- builds on the Estonian ID card (a smartcard like CC)
- Internet voting can be used before the election (votes can be changed an unlimited amount of times, last vote is tabulated, you can even change your vote on the polling day in person)
- in the 2011 parliamentary election 24% voted via Internet

E-Voting in Estonia

- worlds first general election that used internet voting (2007)
- builds on the Estonian ID card (a smartcard like CC)
- Internet voting can be used before the election (votes can be changed an unlimited amount of times, last vote is tabulated, you can even change your vote on the polling day in person)
- in the 2011 parliamentary election 24% voted via Internet

- needs to trust the integrity of voters' computers, central server components and the election staff from https://github.com/vvk-ehk/evalimine/

```
#!/usr/bin/python2.7
# -*- coding: UTF8 -*-

”””
Copyright: Eesti Vabariigi Valimiskomisjon
(Estonian National Electoral Committee), www.vvk.ee
Written in 2004-2013 by Cybernetica AS, www.cyber.ee

This work is licensed under the Creative Commons
Attribution-NonCommercial-NoDerivs 3.0 Unported License.
To view a copy of this license, visit
http://creativecommons.org/licenses/by-nc-nd/3.0/.
”””

def analyze(ik, vote, votebox):

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

    return []
```

E-Voting in **Theory**

- Alice prepares and audits a ballot, then casts an encrypted ballot, which requires her to authenticate to a server.
- A bulletin board posts Alice's name and encrypted ballot. Anyone, including Alice, can check the bulletin board and find her encrypted vote posted.
- When the election closes, all votes are shuffled and the system produces a non-interactive proof of a correct shuffling. (zero-knowledge-proofs)
- After a reasonable complaint period to let auditors check the shuffling, all shuffled ballots are decrypted, and the system provides a decryption proof for each decrypted ballot. (zero-knowledge-proofs)
- Perform a tally of the decrypted votes.
- An auditor can download the entire election data and verify the shuffle, decryptions and tally.

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 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?

THE BALLOT
IN COUNTING THERE IS STRENGTH

"THAT'S WHAT'S THE MATTER."

Boss Tweed. "As long as I count the Votes, what are you going to do about it? say?"

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

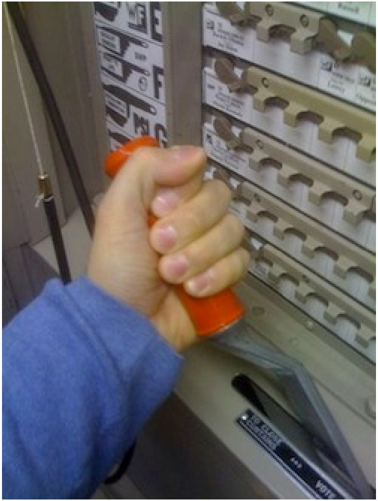
What can go wrong with paper ballots?

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

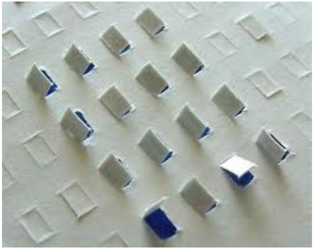
Mechanical Voting Machines

- 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

Alex Halderman:

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

Diebold Machines

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)
- 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 the software open source (avoid security-by-obscurity)
- have a simple design in order to minimise the attack surface

But overall, in times of NSA/state sponsored cyber-crime, e-voting is too hard with current technology.

Online Banking vs. E-Voting

- online banking: if fraud occurs 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

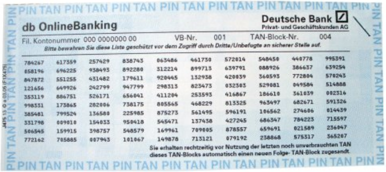
Student In-Lecture Polling

- can guarantee anonymity
- integrity by electronic means

- how to achieve the same in “software”?

Anonymity

- anonymity through one-time pads



db OnlineBanking

Deutsche Bank
Privat- und Geschäftskunden AG

Fil. Kontonummer 000 0000000 00 VB-Nr. 001 TAN-Block-Nr. 004

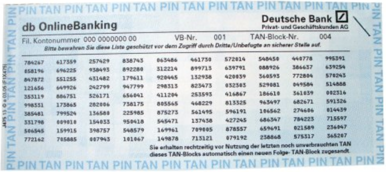
Bitte bewahren Sie diese Liste geschützt vor dem Zugriff durch Dritte/Unbefugte an sicherer Stelle auf.

784267	617359	257429	838743	063486	461730	572014	548450	440778	995391
058196	696225	930493	892280	312214	899713	639791	088926	386637	639254
847872	551255	431482	179611	920445	132938	420039	360593	772804	570243
121656	649926	242799	947799	298313	823473	032303	529081	049584	514888
353119	886751	526171	656041	411204	253593	416867	186610	361039	002316
998331	173865	282006	738175	805565	468229	813325	963497	682671	591326
385481	799524	136580	225985	875273	561495	596191	106562	274606	014439
331798	009010	154033	950418	545471	137438	427245	686347	784225	715597
506545	159915	398757	548579	169961	709005	878557	659691	021589	236047
772162	705885	007943	101067	149878	713121	079192	238868	575317	365207

Sie erhalten rechtzeitig vor Nutzung der letzten noch unverbrauchten TAN dieses TAN-Blocks automatisch einen neuen Folge- TAN-Block zugesandt.

Anonymity

- anonymity through one-time pads



db OnlineBanking

Deutsche Bank
Privat- und Geschäftskunden AG

Fil. Kontonummer 000 0000000 00 VB-Nr. 001 TAN-Block-Nr. 004

Bitte bewahren Sie diese Liste geschützt vor dem Zugriff durch Dritte/Unbefugte an sicherer Stelle auf.

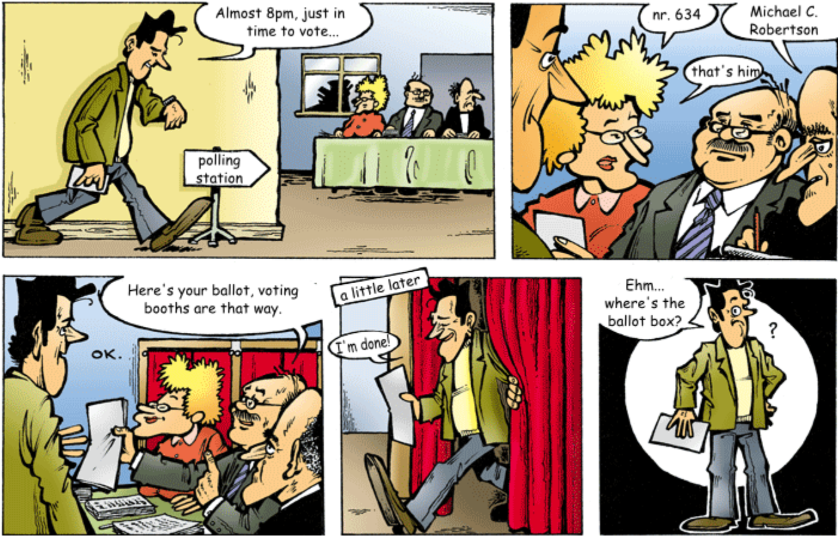
784267	617359	257429	838743	063486	461730	572014	548450	440778	995391
658196	696225	930493	892280	312214	899713	639791	088926	386637	639254
847872	551255	431482	179611	920445	132938	420039	360593	772804	570243
121656	649926	242799	947799	298313	823473	032303	529081	049584	514888
353119	886751	526171	656041	411204	253593	416867	186610	361039	002316
998331	173865	282006	738175	805565	468229	813325	963497	682671	591326
385481	799524	136580	225985	875273	561495	596191	106562	274606	014439
331798	009010	154033	950418	545471	137438	427245	686347	784223	715597
506545	159915	398757	548579	169961	709005	878557	659691	021589	236047
772162	785885	007943	101067	149878	713121	079192	238868	575317	365207

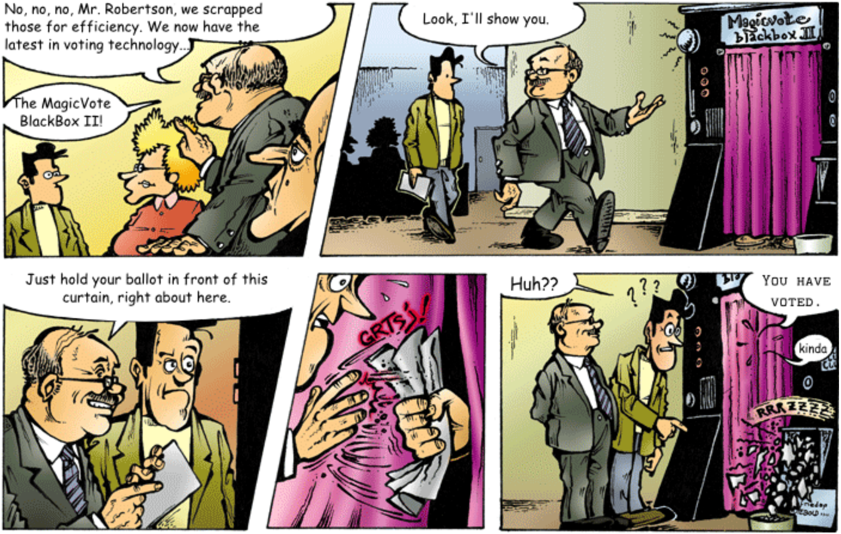
Sie erhalten rechtzeitig vor Nutzung der letzten noch unverbrauchten TAN dieses TAN-Blocks automatisch einen neuen Folge- TAN-Block zugesandt.

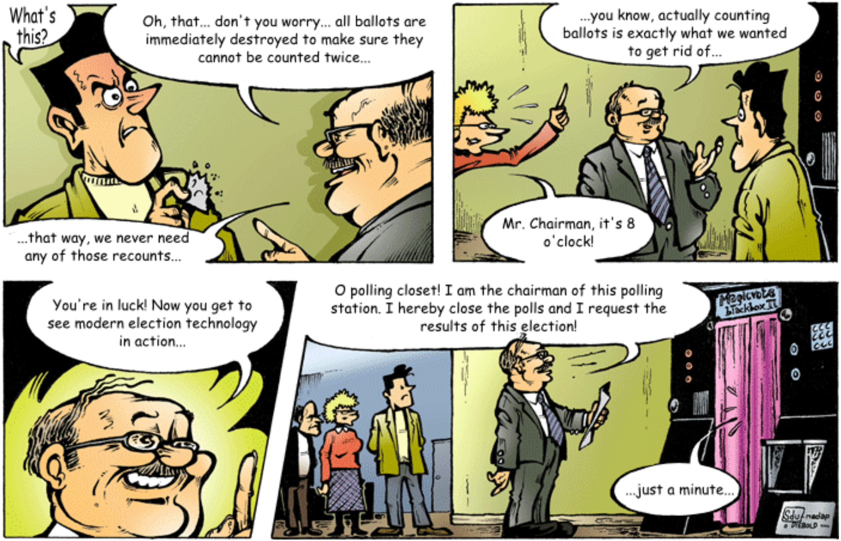
J475 13 03.06 [F1X475]

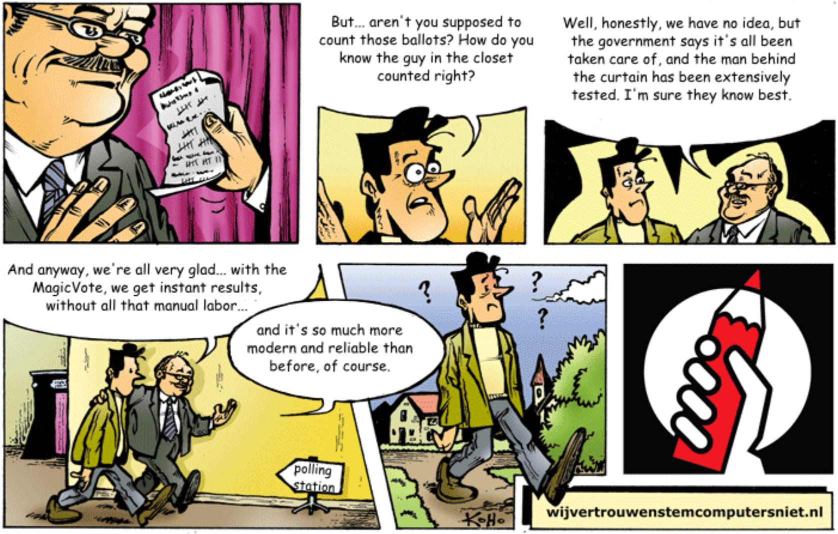
- solving the problem of distribution

The adventures of citizen Michael C. Robertson


Almost 8pm, just in time to vote...
polling station
nr. 634
Michael C. Robertson
that's him
Here's your ballot, voting booths are that way.
OK.
a little later
I'm done!
Ehm... where's the ballot box?
?


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 II
Just hold your ballot in front of this curtain, right about here.
GRTSJ!
Huh??
? ? ?
YOU HAVE VOTED.
kinda
RRRZZZ


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...


But... aren't you supposed to count those ballots? How do you know the guy in the closet counted right?
Well, honestly, we have no idea, but the government says it's all been taken care of, and the man behind the curtain has been extensively tested. I'm sure they know best.
And anyway, we're all very glad... with the MagicVote, we get instant results, without all that manual labor...
and it's so much more modern and reliable than before, of course.
polling station
?
?
?
KoHo
wijvertrouwenstemcomputersniet.nl

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

Unix-Style Access Control

How to do control access? In Unix 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).

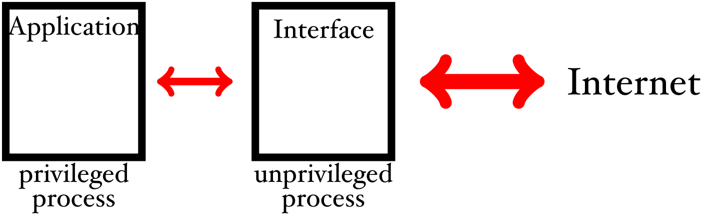
Unix-Style Access Control

- 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. (defence in depth)

Network Applications: Privilege Separation



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

Weaknesses of Unix AC

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

   should be deleted, but does not do it yet

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

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

Buffer Overflow Attacks

first lecture

Buffer Overflow Attacks

first lecture

next week