Security Principles & Sandboxes

CS 161: Computer Security

Prof. Raluca Ada Popa

January 25, 2018

Some slides credit Nick Weaver or David Wagner.

Announcements

- Homework 1 is out, due on Monday
- Midterm 1 date Feb 15, 7-8:30pm: conflict?

Security principles

- Set of principle that help to design more secure software
- Keep them in mind when you build your own systems

Principle: “Know your threat model”

What will a document safe protect against?

(1) The average unskilled burglar
(2) Skilled burglar
(3) None of the above

Principle: “Know your threat model”

- Most "safes" you buy at Office Depot are not actual safes...
  - They are not rated nor even well designed to keep out a burglar
- Rather, they are "fire safes": designed to prevent damage in case of a fire
- Two big categories
  - Documents/guns/etc: Keeps temperatures below 350F
    - Will keep that passport from burning
  - Data safes: Keeps temperatures below 125F and humidity below 80%
    - Computer media much more delicate
- Security lesson: Know what you are protecting and what your threat is
  - Don't expect a document-rated fire safe to keep a hard drive safe from damage in a fire
  - Don't expect either to meaningfully stop a teenager with a crowbar
- And do your threat modeling ***before*** you commit to a security procedure!

Lesson: checkbox security and real security are often different..

- Some safes are concerned with rather low-threat attackers
  - Toddlers and the like
- Classic example are CA state mandated "gun" locks
  - A long list of "approved" devices
  - That often can't even keep a toddler out!

- Not enough to be security compliant, work on making your system secure for real

Principle: “Security is economics”

- Make it expensive for an attacker to attack, ideally more expensive than the reward the attacker gets by attacking a resource
- Often, there is a tradeoff in cost and level of security of a security mechanism

TL-15= will take 15 min to break using common tools

TL-30 = will take at least 30 min to break using common tools

TRTL-30 = at least 30 min with common tools and cutting torch

Q: what other principle does the choice of this safe remind you of?

A: know your threat model

TXTL-60 = 60 minutes, working on all 6 sides, and the attacker even gets to use 8 ounces of explosives!

These are conservative ratings: they assume an attacker with the proper set of tools ***and knowledge of the safe's construction***

Principle: “Don’t rely on security through obscurity.”

- Shannon's Maxim: "The enemy knows the system"
- Kerckhoffs's Principle: "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge."

The attacker will eventually find out the algorithm or how the system works since that part does not change (unlike a key) and that algorithm/system will be doomed forever (not as easy as changing a key)

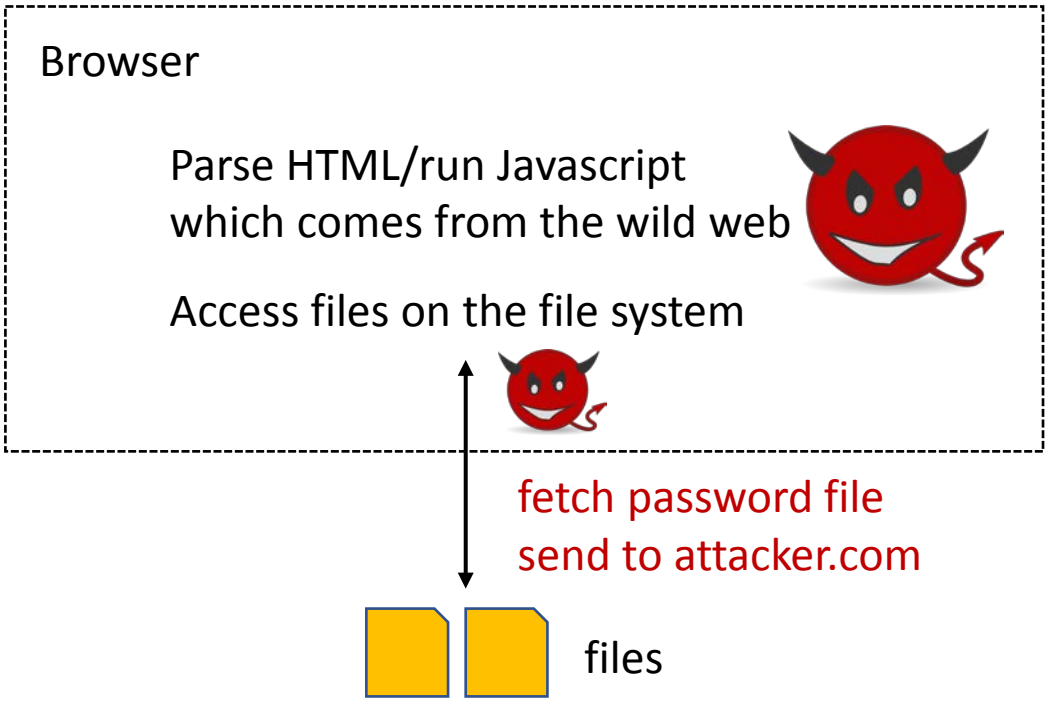
A software lock: sandbox

- A sandbox is a mechanism (often, a software program) that isolates a process from the rest of the environment
- If some code is from an untrusted source or takes inputs from potential attacking users, running in a sandbox aims to prevents the exploits from spreading outside of the sandbox
- Various sandbox designs exist, but typically the sandbox:
  - provides a controlled set of resources for guest programs to run in (e.g., memory, disk)
  - Do not allow network access, file system access, other I/O

Example of sandbox implementation: seccomp

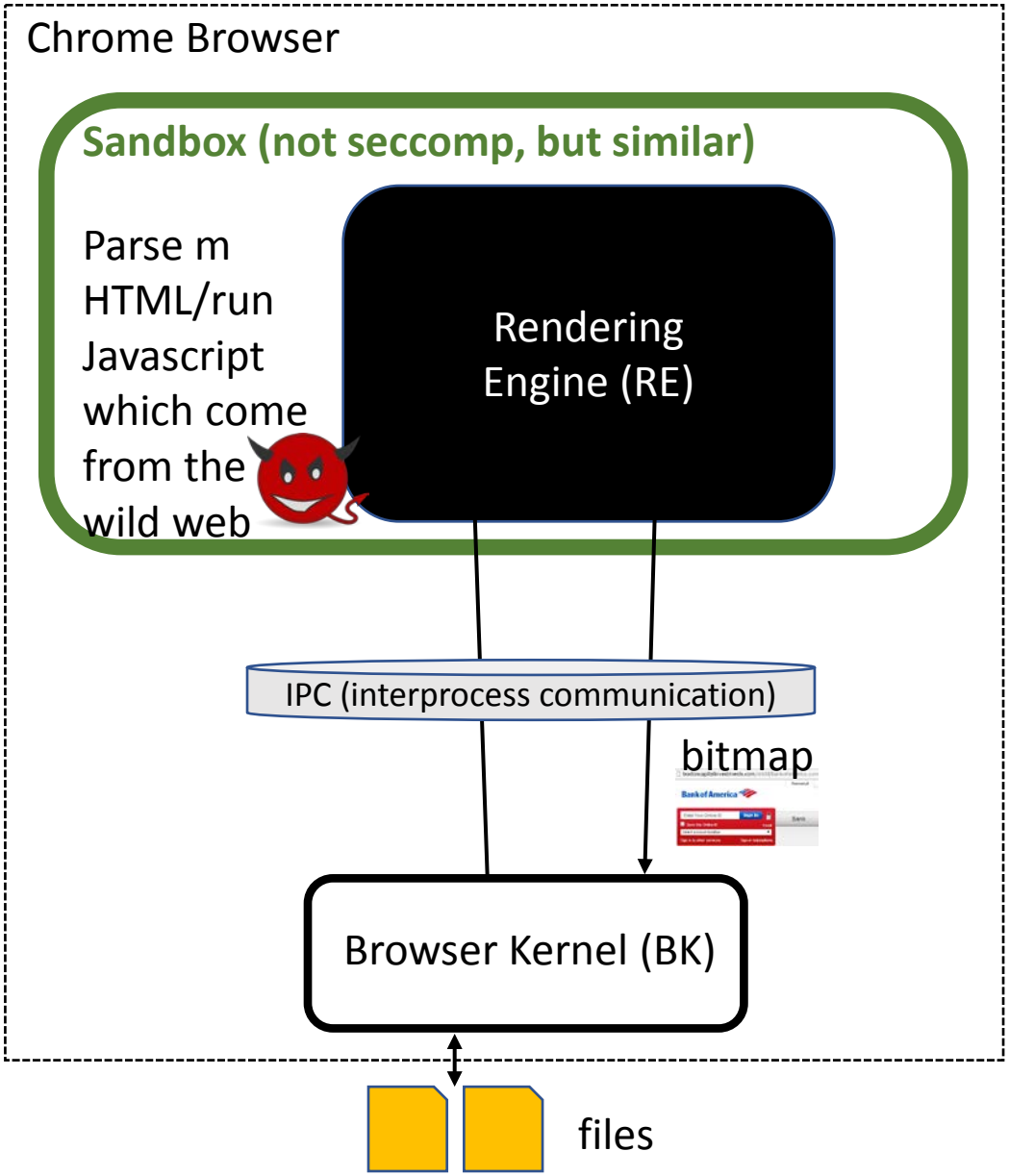
- Facility in the Linux Kernel
- seccomp = “secure computing”
- Allows a process to transition into secure mode in a “one-way mode” from which it can only return with an exit, or it can only read and write files that are already open, but cannot open new files or make other system calls
- So caller of sandbox opens the files it wants the dangerous process to work with and nothing more, and calls seccomp to create the sandbox and run that process

Recall monolithic browser design issue



An exploit in the parsing/rendering part of the code can result in an attacker having access to the file system in the absence of isolation

Recall: the Chrome browser

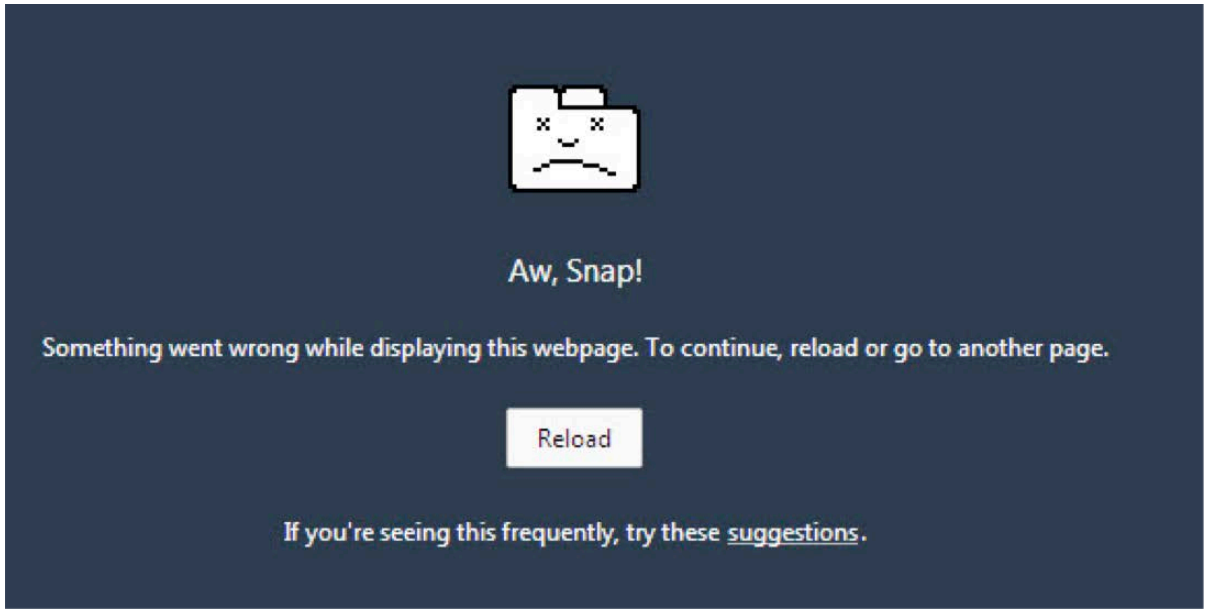


Browser kernel only provides a restricted API to RE:

- RE can send a bitmap image with webpage to display to the user, but not active code
- RE cannot access what files it wishes. BK might ask the user to upload a file the *user chooses* when RE requests so, but BK will not fetch if RE asks for it

"Least Privilege" in action: RE does not get more privilege than it needs

The sandboxed process can now fail gracefully and not take the whole system down



TCB

- Recall: the trusted computing base (TCB) is the subset of the system that has to be correct, for some security goal to be achieved

- Q: What is the TCB that ensures that if the RE fails, the browser does not fail?
- A: the sandbox
- Q: What is the TCB that ensures the RE cannot fetch whatever files it wants from the file system?
- A: the sandbox and the browser kernel

Principle: “don’t reinvent the wheel”

- Use tools that have already been worked on extensively
- They already dealt with many bugs and exploits. Writing your new tool will likely suffer from issues
- Example: for end-to-end encrypted communications, Open SSL: more than 185 vulnerasbilities reported on the CVE website
- For sandboxing, use existing tools such as:
    - For Windows:
        - Wrapper around Chrome's sandbox
    - For Linux:
        - Uses seccomp as the building block
    - A set of others, investigate what you need for your situation

Principle: “Defense in depth.”

If you don’t want to be caught with your pants around your ankles, wear both a belt and suspenders…

If you use multiple redundant protections, then all of them would need to
be breached before the system’s security will be endangered

Q: when you use two or more mechanisms for securing a system, what is important about them?

A: that they be as different as possible, so if one fails, its unlikely the second one fails

Q: What are some other examples of defense in depth?

Two-factor authentication

Authentication using two of:

- Something you know (account details or passwords)
- Something you have (tokens or mobile phones)
- Something you are (biometrics)

Example

Online banking:

- Hardware token or card (“smth you have”)
- Password (“smth you know”)

Mobile phone two-factor authentication:

- Password (“smth you know”)
- Code received via SMS (“smth you have”)

Another example

- Password

+

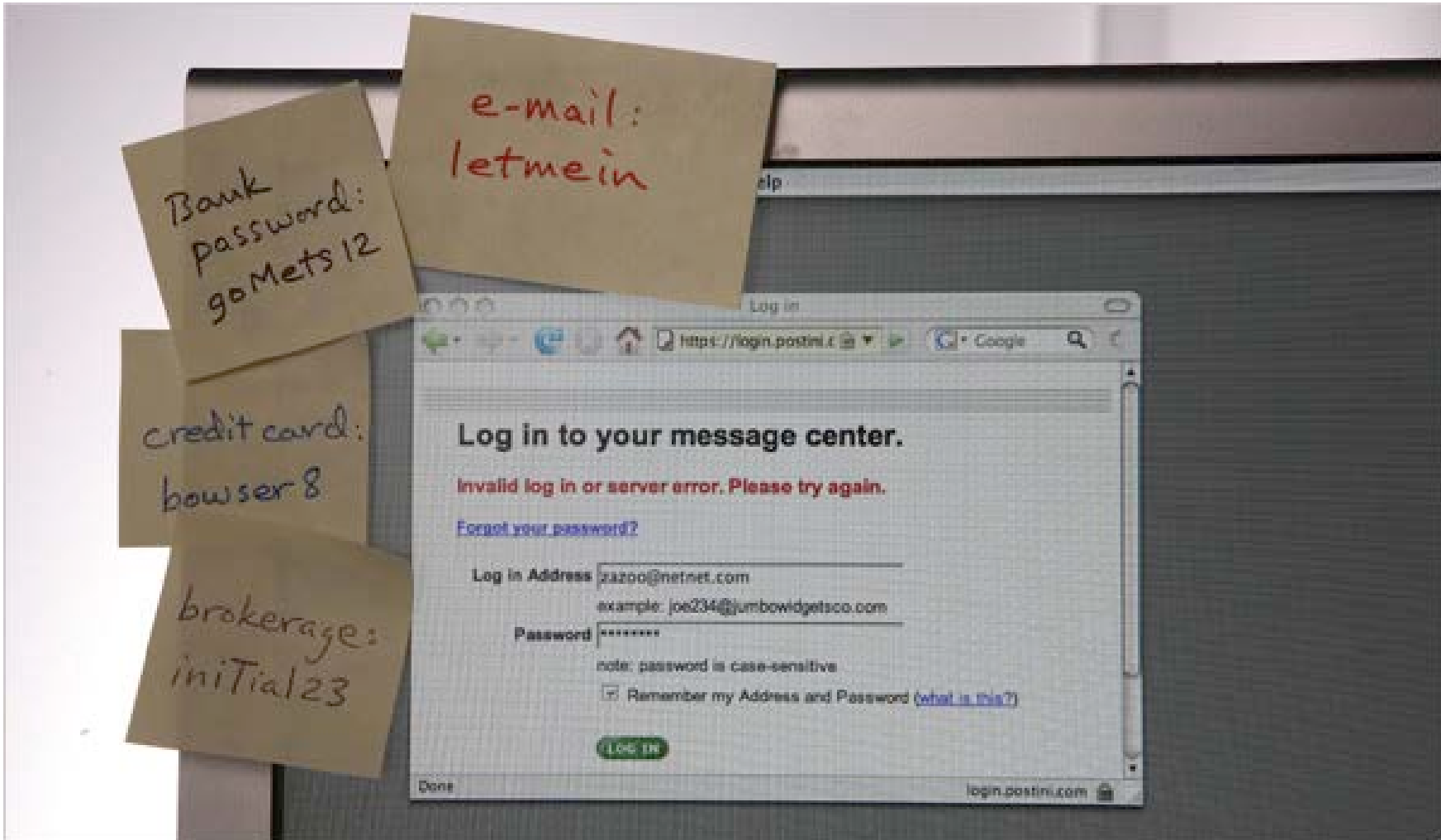
- Answer to security question

This is not two-factor authentication because both of the factors are something you know

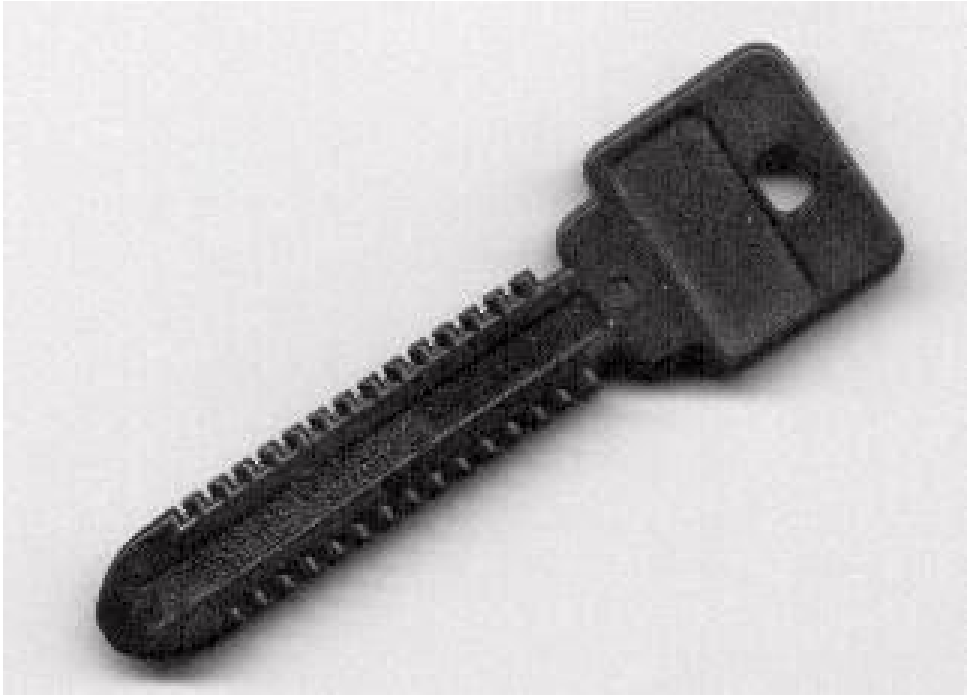
Principle: “Consider human factors.”

“Company policy: passwords must be at least 10 characters long, contain at least 2 digits, 1 uppercase character, 1 lowercase character, and 1 special character.”

company portal
Password: 1secret



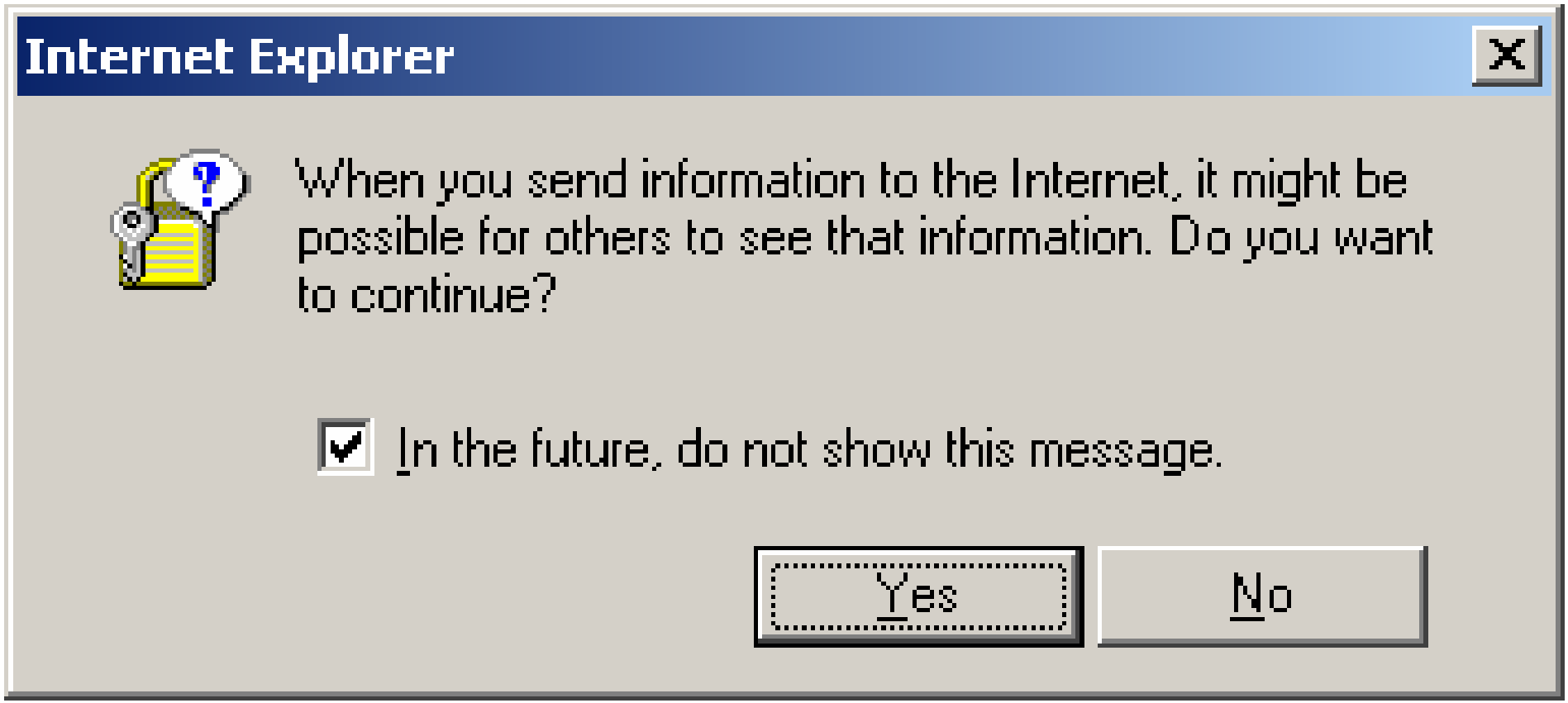
Example: storage device for cryptographic keys

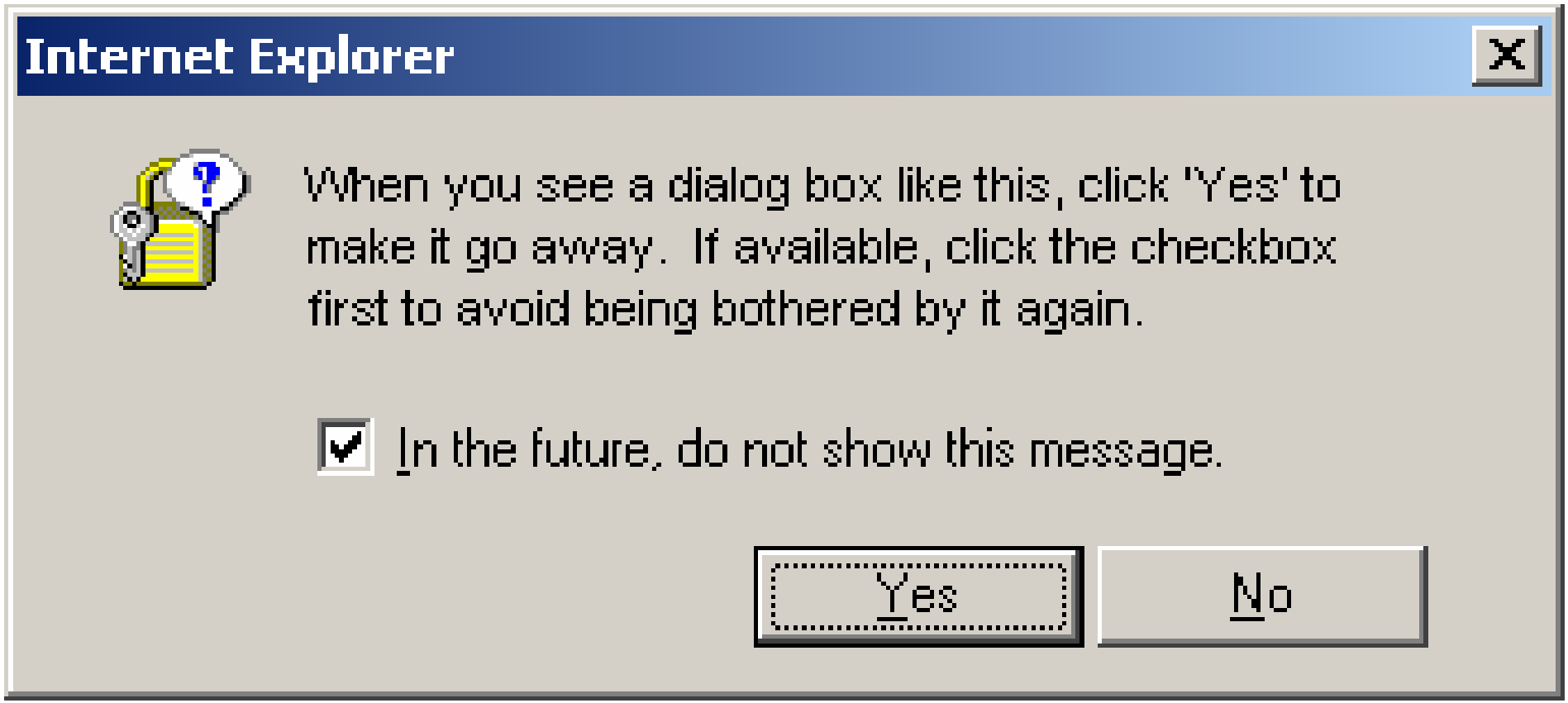


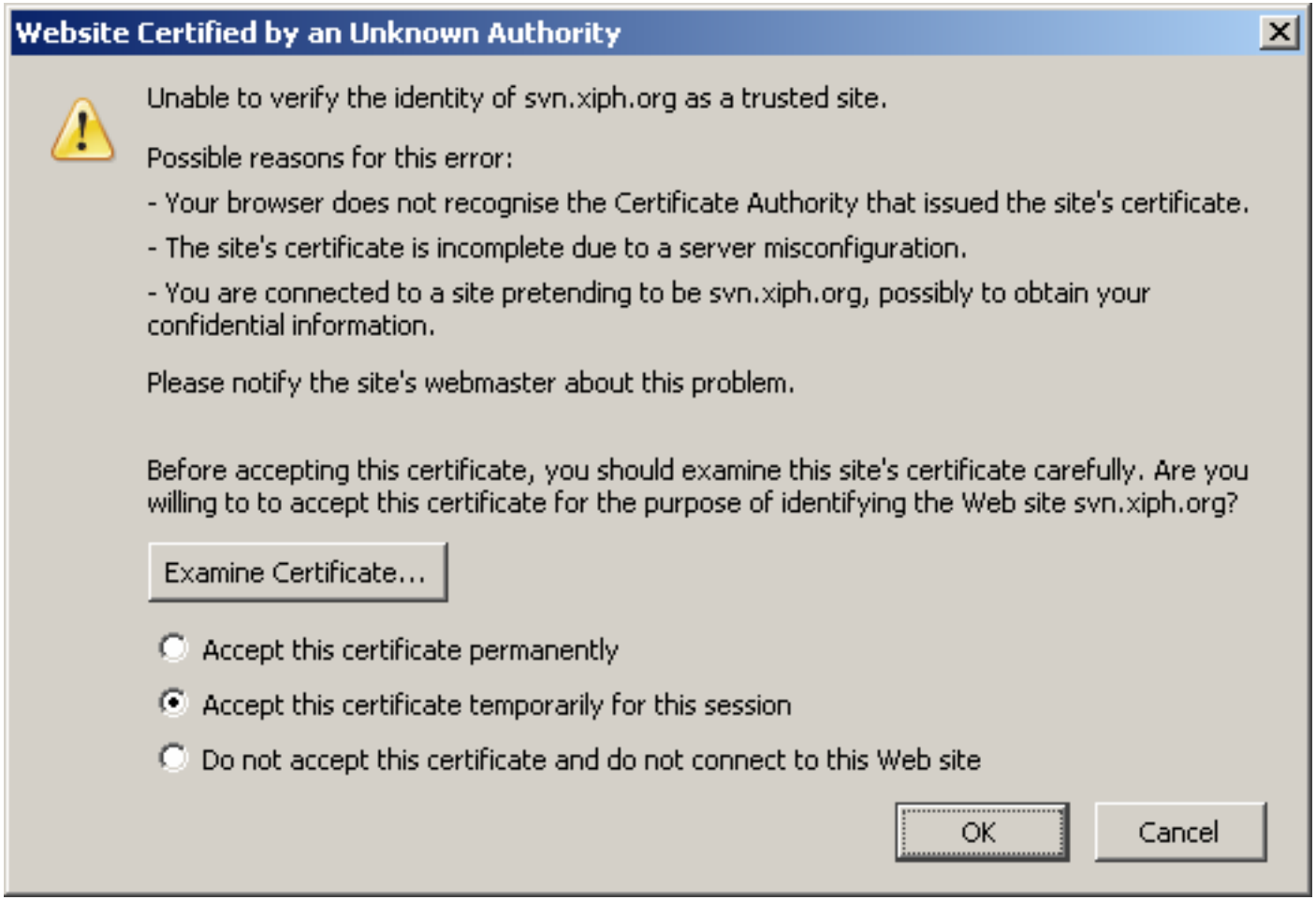
Q: what is good about this design?

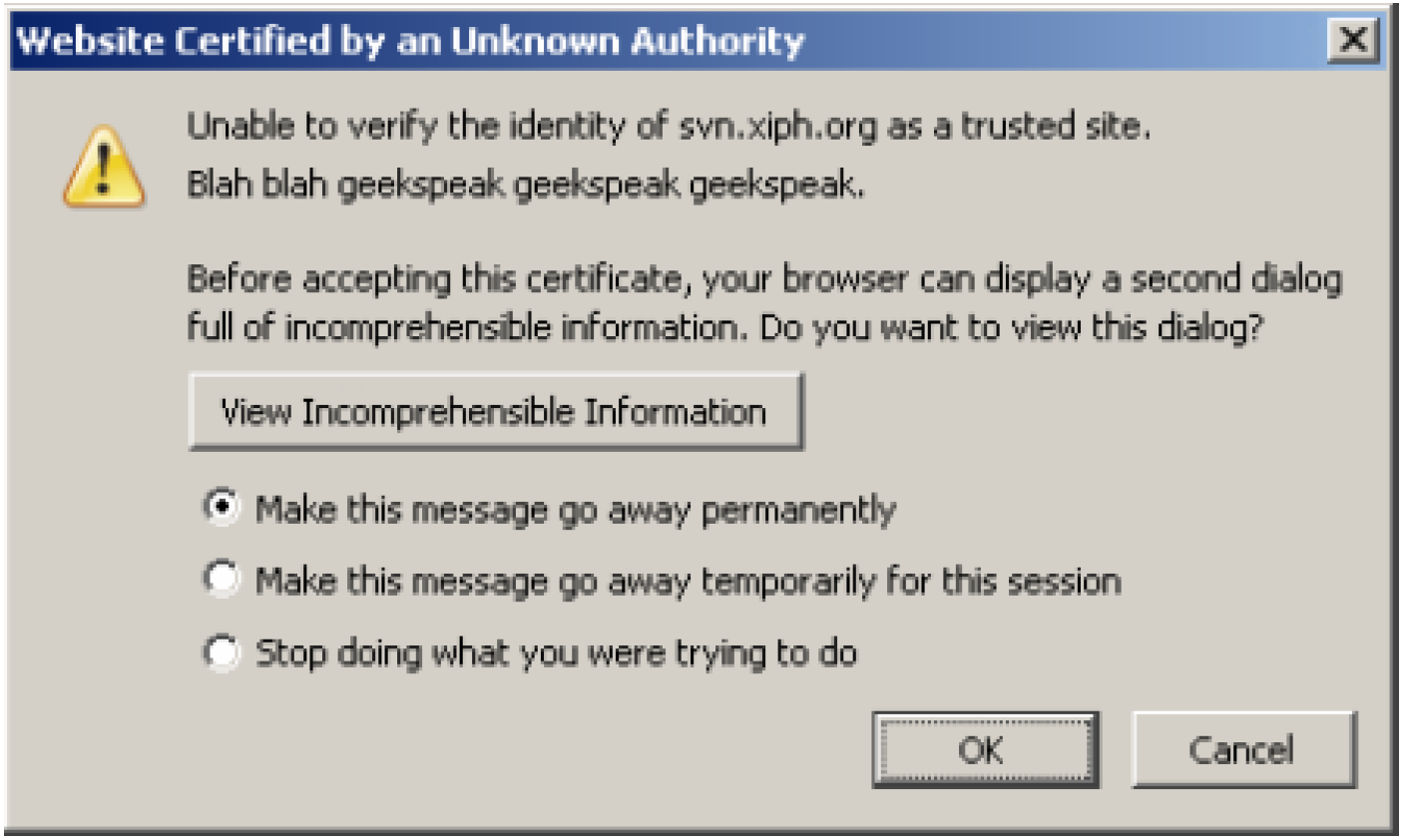
A: tells users they need to protect it

Principle: ”Make your system easy to use in a secure way”
- make sure users understand how to use the system securely
- related to consider human factors


Internet Explorer
When you send information to the Internet, it might be possible for others to see that information. Do you want to continue?
In the future, do not show this message.
Yes
No


Internet Explorer
When you see a dialog box like this, click 'Yes' to make it go away. If available, click the checkbox first to avoid being bothered by it again.
In the future, do not show this message.
Yes
No


Website Certified by an Unknown Authority
Unable to verify the identity of svn.xiph.org as a trusted site.
Possible reasons for this error:
- Your browser does not recognise the Certificate Authority that issued the site's certificate.
- The site's certificate is incomplete due to a server misconfiguration.
- You are connected to a site pretending to be svn.xiph.org, possibly to obtain your confidential information.
Please notify the site's webmaster about this problem.
Before accepting this certificate, you should examine this site's certificate carefully. Are you willing to to accept this certificate for the purpose of identifying the Web site svn.xiph.org?
Examine Certificate...
Accept this certificate permanently
Accept this certificate temporarily for this session
Do not accept this certificate and do not connect to this Web site
OK
Cancel


Website Certified by an Unknown Authority
Unable to verify the identity of svn.xiph.org as a trusted site.
Blah blah geekspeak geekspeak geekspeak.
Before accepting this certificate, your browser can display a second dialog full of incomprehensible information. Do you want to view this dialog?
View Incomprehensible Information
Make this message go away permanently
Make this message go away temporarily for this session
Stop doing what you were trying to do
OK
Cancel

Principle: “Only as secure as the weakest link.”

(recall Sarah Palin email example)


ВТБ
VISA
VISA
Windows Product Activation
Your Windows product must be activated within 7 days.
Do you want to activate Windows now?
Yes
No

Wells Fargo Web-enables 6,200 ATMs

The Windows-based infrastructure enables remote upgrades

By Lucas Mearian

March 3, 2005 12:00 PM ET Add a comment

+1 Like 0 + Briefcase More

Computerworld - Wells Fargo & Co. announced this week that it has completed a five-year project to Web-enable its 6,200 ATMs in 23 states. The Windows-based infrastructure is designed to allow Wells Fargo to update and add services such as new languages and envelope-free deposits to its entire network remotely.

Wells Fargo took all "the rational steps you'd take to harden any operating system," such as closing unused ports. But, "the reality is you can't buy a new ATM that runs OS/2. This is where the industry is," he said.

Principle:
“Design security in from the start.”

(Beware of *bolt-on security*.)


ACCIDENT
ON
MOTORWAY


ACCIDENT
ON
MOTORWAY


CAUTION!
ZOMBIES!
AHEAD!!!
RUN FOR
COLD
CLIMATES
THE END
IS NEAR!
!!!!!!!!!


RAPTORS
AHEAD
CAUTION

“Ensure complete mediation.”

Ensuring Complete Mediation

- To secure access to some capability/resource, construct a *reference monitor*
- Single point through which all access must occur
  - E.g.: a network firewall
- Desired properties:
  - Un-bypassable ("complete mediation")
  - Tamper-proof (is itself secure)
  - Verifiable (correct)
  - (Note, just restatements of what we want for TCBs)
- One subtle form of reference monitor flaw concerns *race conditions* …

TOCTTOU Vulnerability

TOCTTOU = Time of Check To Time of Use

```
procedure withdrawal(w)
    // contact central server to get balance
    1. let b := balance

    2. if b < w, abort
```

Balance could have decreased at this point due to another action

```
    // contact server to set balance
    3. set balance := b - w

    4. dispense $w to user
```

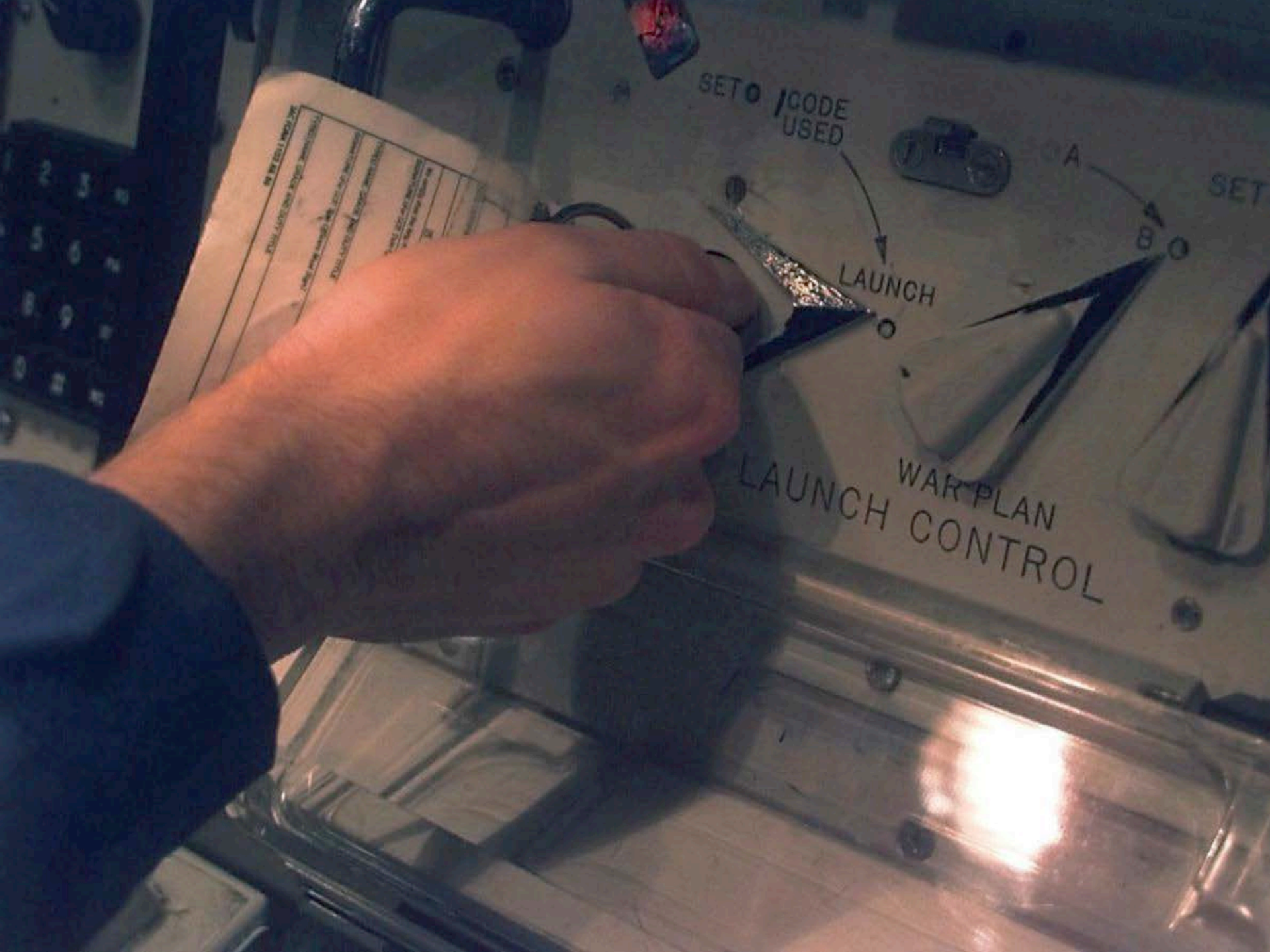
```
public void buyItem(Account buyer, Item item) {
   if (item.cost > buyer.balance)
     return;
   buyer.possessions.put(item);
   buyer.possessionsUpdated();
   buyer.balance -= item.cost;
   buyer.balanceUpdated();
 }
```

Q: Where can TOCTOU happen here?

A: Both threads of execution pass the if statement but are bfore the subtraction

“Division of trust.”
- reduce the trust in each party

Missile silo, it has
two launch
officers who mus
both agree


SET
CODE USED
LAUNCH
A
B
SET
WAR PLAN
LAUNCH CONTROL


NO LONE ZONE
SAC TWO MAN POLICY
MANDATORY
CAUTION
DO NOT KEY RTMX IN L/D
EXCEPT IN CASE OF AN
EMERGENCY-MUST BE AT
LEAST 5FT FROM MSL
E-F-G


California
California
The IDES OF MARCH
PARANORMAL ACTVITY 3
THE
RUM DIARY

Summary: security principles

- Know your threat model
- Security is economics
- Don’t rely on security through obscurity
- Least privilege
  - helpful tool here is sandboxing
- Defense in depth
- Consider human factors
- Make your system easy to use in a secure way
- Defense in depth
- Design security in from the start
- Ensure complete mediation
- Division of trust