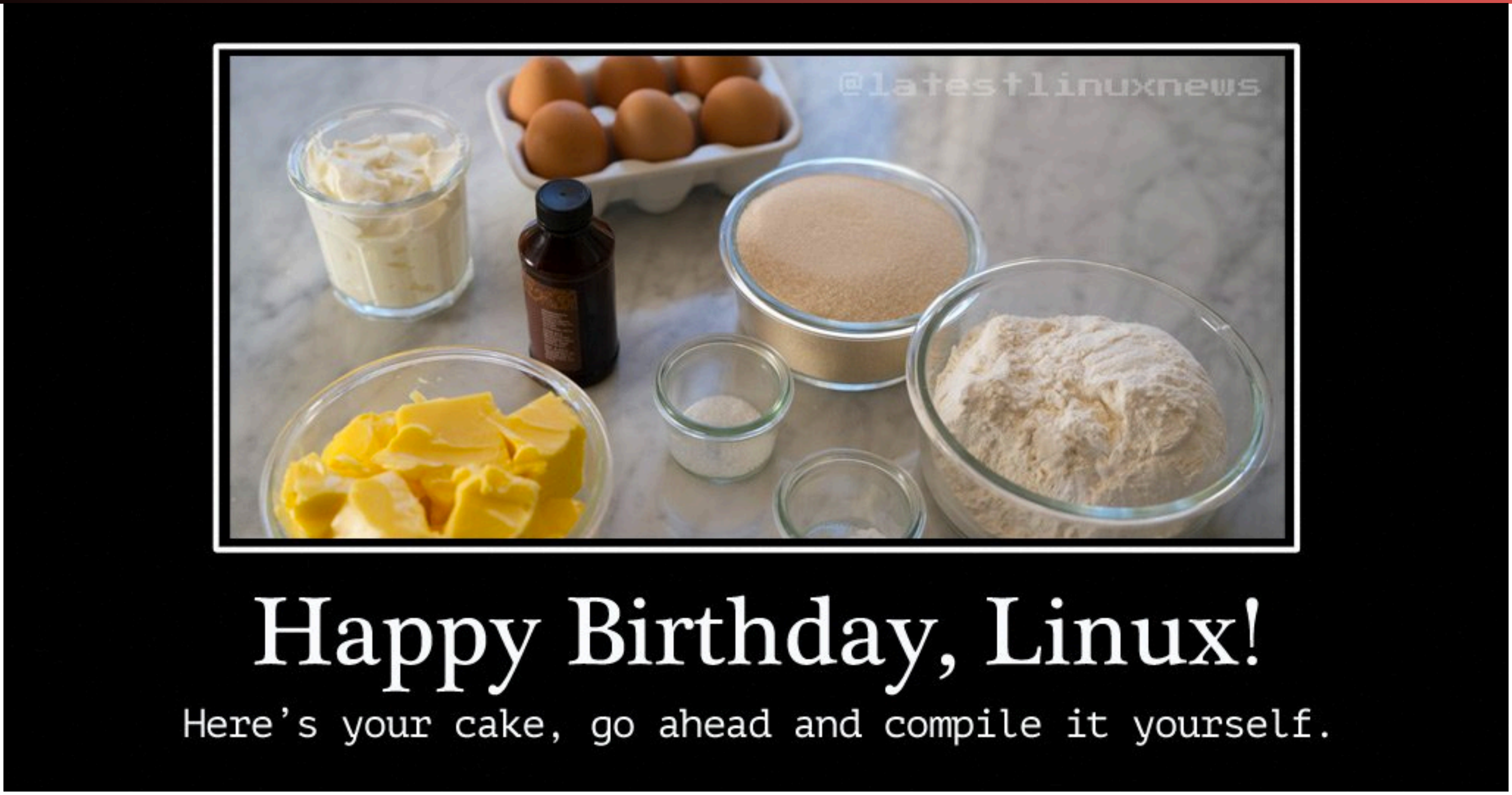

@latestlinuxnews
Happy Birthday, Linux!
Here's your cake, go ahead and compile it yourself.

Lecture 2: Security Principles Cont

https://cs161.org

Reminder on Chat...

- If you like it, great...
  - But please keep it a bit more professional
  - Especially feel free to ask questions...
- If you don't:
  - Open the chat window and move it off the screen: Effectively suppresses notifications
  - I try to repeat questions
- In particular, please flag when I forget to define new terms!
- No need for "F to take attendance":
  we will get that data from Zoom (somehow)

The Properties We Want in a Safe

- We want the inside to be inaccessible to an attacker
  - But what ***sort*** of attacker?
  - But ***how much time*** does the attacker have?
- We want to ***measure*** how much time & capabilities needed for an attacker
  - For a safe, ratings communicate how much based on experts performing the attack
    - Such security ratings are much harder in the computer security side

Security Rating: A Real Safe

- TL-15:
  - An expert with common tools will take >= 15 minutes to break in
- May even have "relockers"
  - EG, a pane of glass which, if shattered when trying to drill for the combo lock, causes the safe to freeze closed!

Security Rating: A Stronger Safe

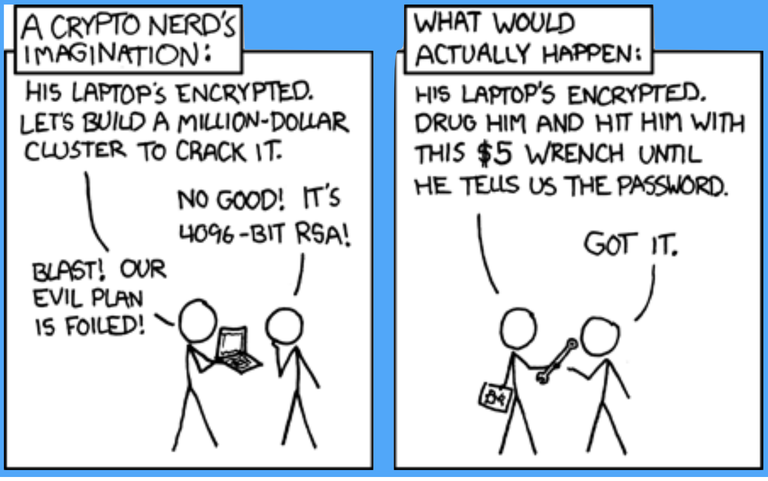
- TL-30:
  - The same expert and tools now takes 30 minutes

Security Rating:
Now We Are Talking

- TRTL-30
    - 30 minute to break with tools and/or a cutting torch

Security Rating: Maximum Overkill...

- TXTL-60:
  - 60 minutes with tools, torches, and up to 4 oz of ***explosives!***
  - Far easier to use "Rubber Hose Cryptanalysis" on someone who knows the combination

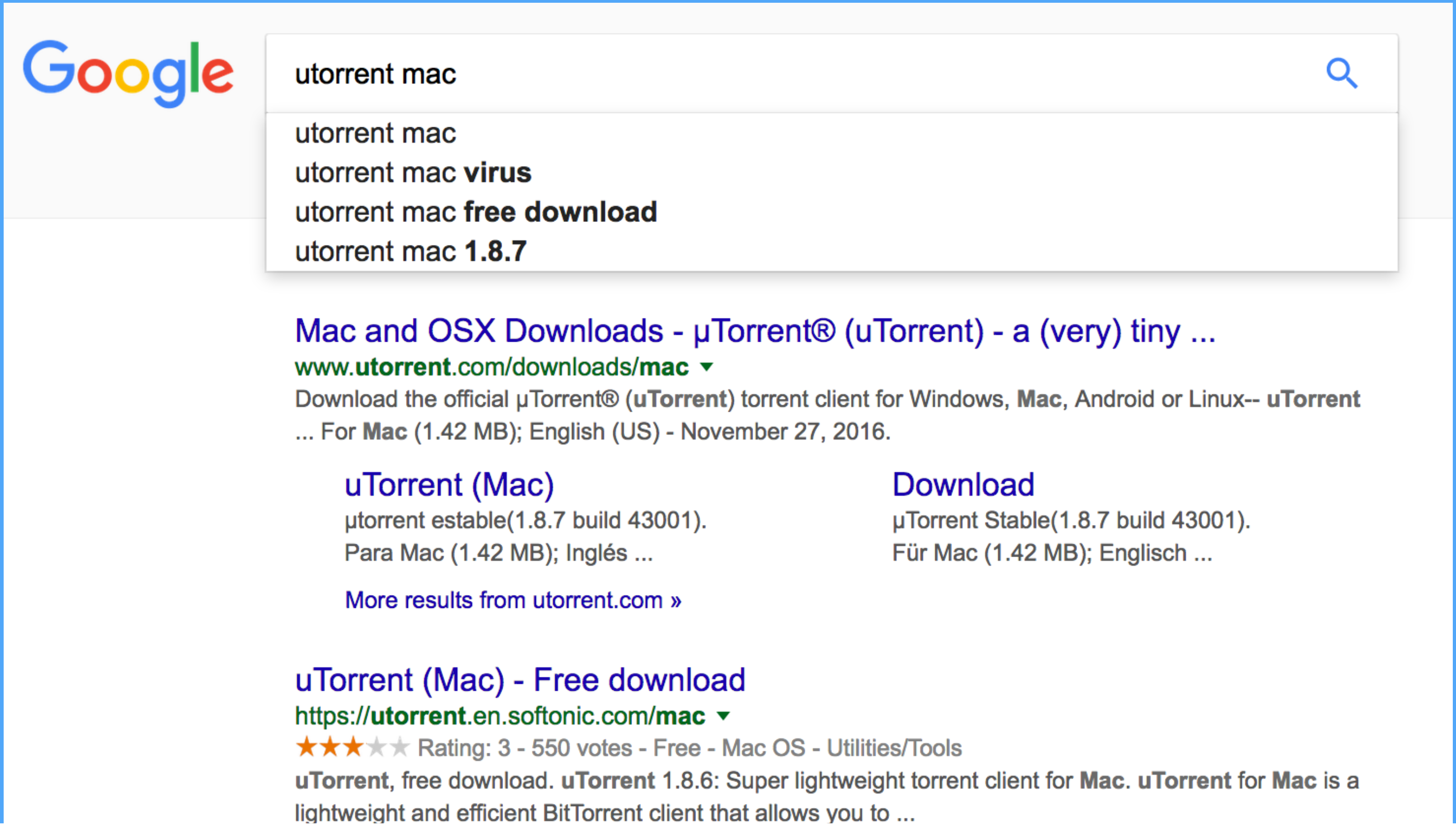

A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Security Rating: 😂😂😂😂😂😂😂😂😂...

- This is legally a "gun safe"
  - Meets the California requirements for "safe" storage of a handgun
- But it is practically ***snake oil***:
  - Cylindrical locks can often be opened with a Bic pen
  - Some safes like this open if you just ***drop them a foot!***
- So why do people buy this?
  - It creates an ***illusion*** of security
  - It meets the ***legal requirement*** for security

Lesson:
Security is economics

- More security (***generally***) costs more
  - If it costs the same or less and doesn't impose other costs,
    you'd always go with "more security"
- Standards often define security
  - The safe standards from Underwriters Laboratories
    - If you are selling a real safe to a customer who knows what they are buying, you have to meet theses standards
  - The "gun safe" standards from the California Department of Justice
    - Which are a joke
- The more purchasers makes security cheaper...
  - If you need a safe at home, buy a UL listed Residential Security Container ***gun safe!***
    - The gun owners are willing to pay for security, and so have created a market for security!


utorrent mac

utorrent mac
utorrent mac **virus**
utorrent mac **free download**
utorrent mac **1.8.7**

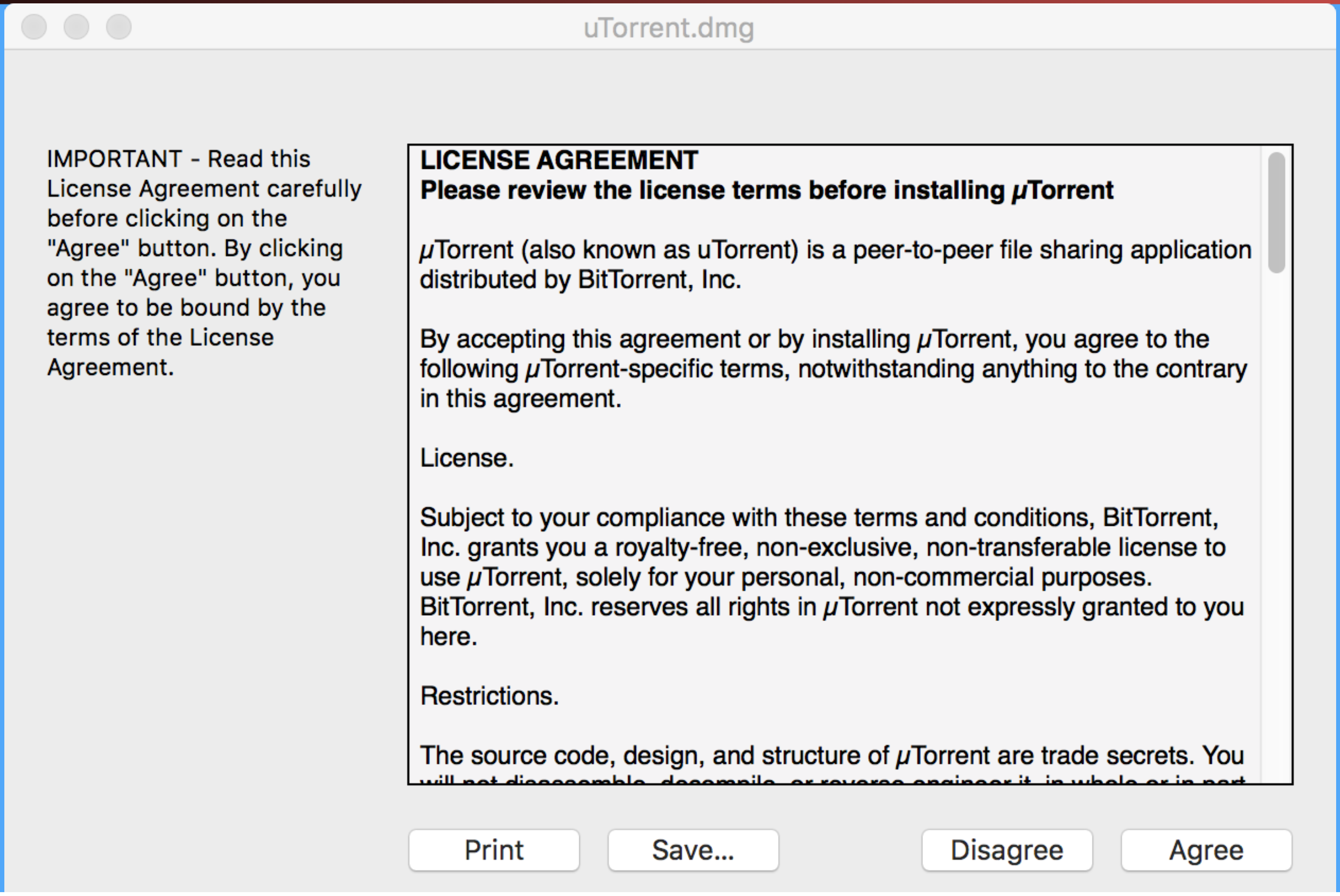
Mac and OSX Downloads - µTorrent® (uTorrent) - a (very) tiny ...
www.**utorrent**.com/downloads/**mac**
Download the official µTorrent® (**uTorrent**) torrent client for Windows, **Mac**, Android or Linux-- **uTorrent** ... For **Mac** (1.42 MB); English (US) - November 27, 2016.

uTorrent (Mac)
µtorrent estable(1.8.7 build 43001). Para Mac (1.42 MB); Inglés ...

Download
µTorrent Stable(1.8.7 build 43001). Für Mac (1.42 MB); Englisch ...

More results from utorrent.com »

uTorrent (Mac) - Free download
https://**utorrent**.en.softonic.com/**mac**
★★★★★ Rating: 3 - 550 votes - Free - Mac OS - Utilities/Tools
uTorrent, free download. **uTorrent** 1.8.6: Super lightweight torrent client for **Mac**. **uTorrent** for **Mac** is a lightweight and efficient BitTorrent client that allows you to ...



IMPORTANT - Read this License Agreement carefully before clicking on the "Agree" button. By clicking on the "Agree" button, you agree to be bound by the terms of the License Agreement.

LICENSE AGREEMENT
Please review the license terms before installing μTorrent

μTorrent (also known as uTorrent) is a peer-to-peer file sharing application distributed by BitTorrent, Inc.

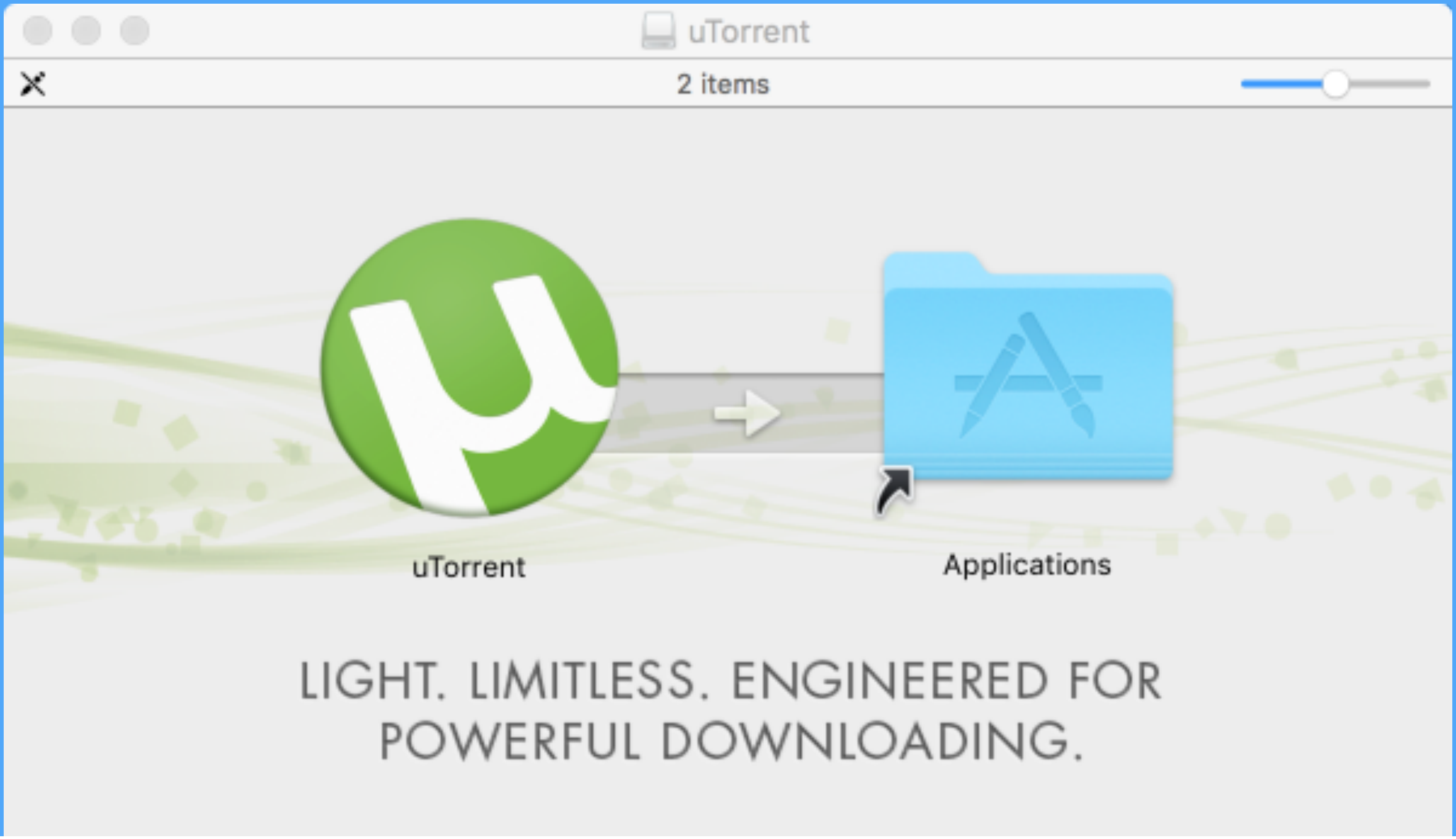
By accepting this agreement or by installing μTorrent, you agree to the following μTorrent-specific terms, notwithstanding anything to the contrary in this agreement.

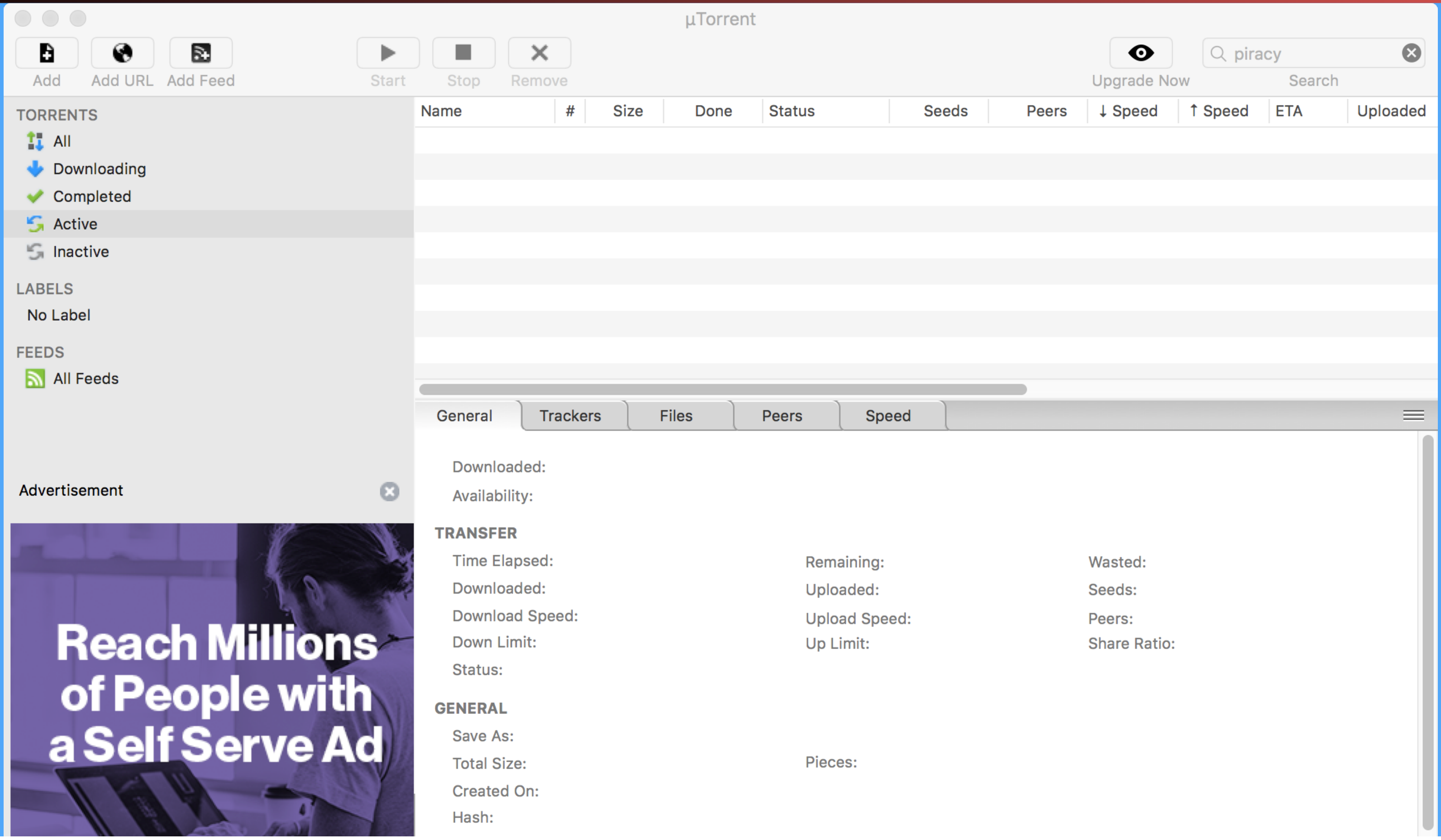
License.

Subject to your compliance with these terms and conditions, BitTorrent, Inc. grants you a royalty-free, non-exclusive, non-transferable license to use μTorrent, solely for your personal, non-commercial purposes. BitTorrent, Inc. reserves all rights in μTorrent not expressly granted to you here.

Restrictions.

The source code, design, and structure of μTorrent are trade secrets. You will not disassemble, decompile, or reverse engineer it, in whole or in part


uTorrent
2 items
uTorrent
Applications
LIGHT. LIMITLESS. ENGINEERED FOR POWERFUL DOWNLOADING.



Name	#	Size	Done	Status	Seeds	Peers	↓ Speed	↑ Speed	ETA	Uploaded

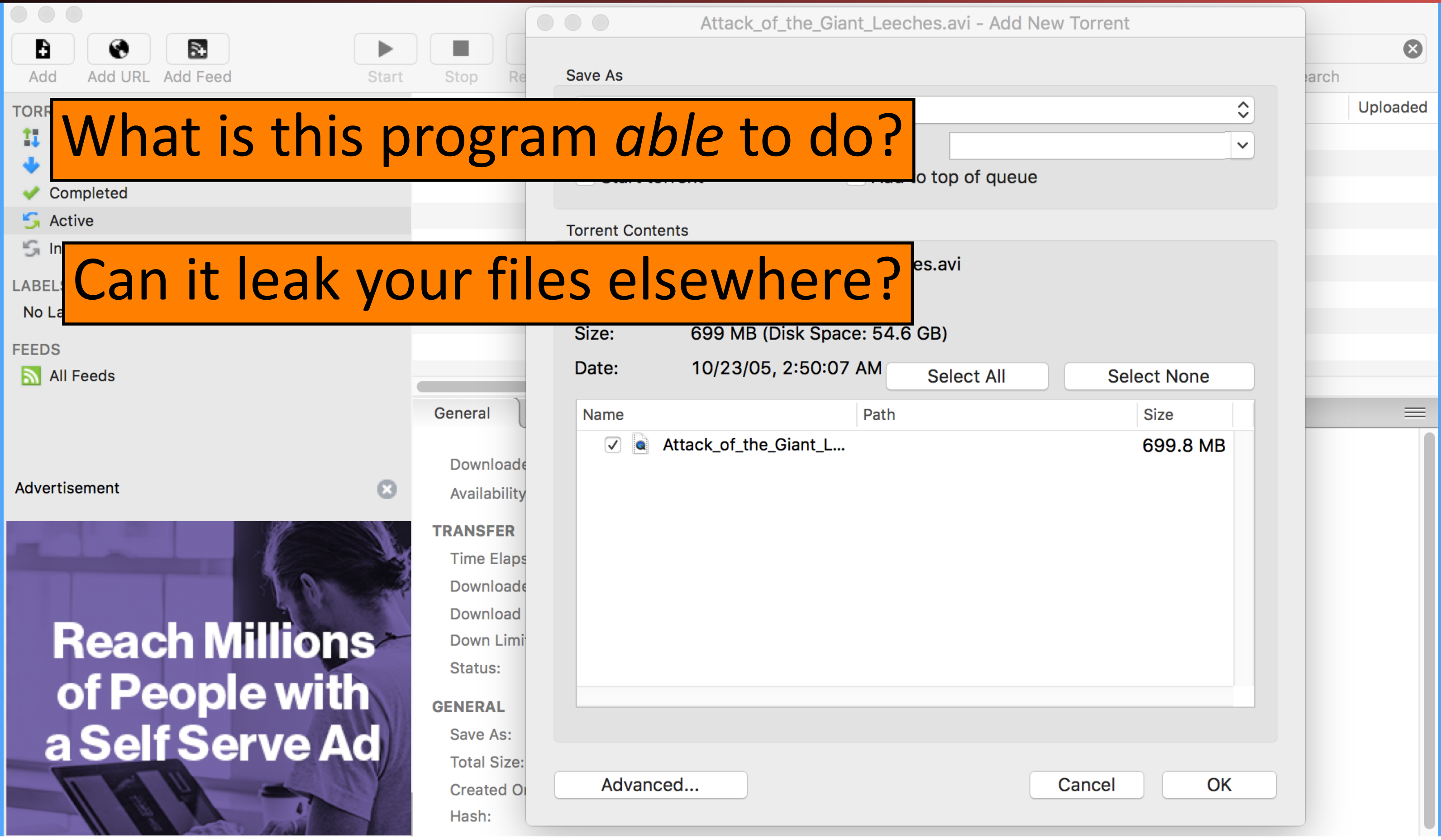
Downloaded:
Availability:

TRANSFER

Time Elapsed:	Remaining:	Wasted:
Downloaded:	Uploaded:	Seeds:
Download Speed:	Upload Speed:	Peers:
Down Limit:	Up Limit:	Share Ratio:
Status:		

GENERAL

Save As:	
Total Size:	Pieces:
Created On:	
Hash:	



Attack_of_the_Giant_Leeches.avi - Add New Torrent

Save As

Torrent Contents

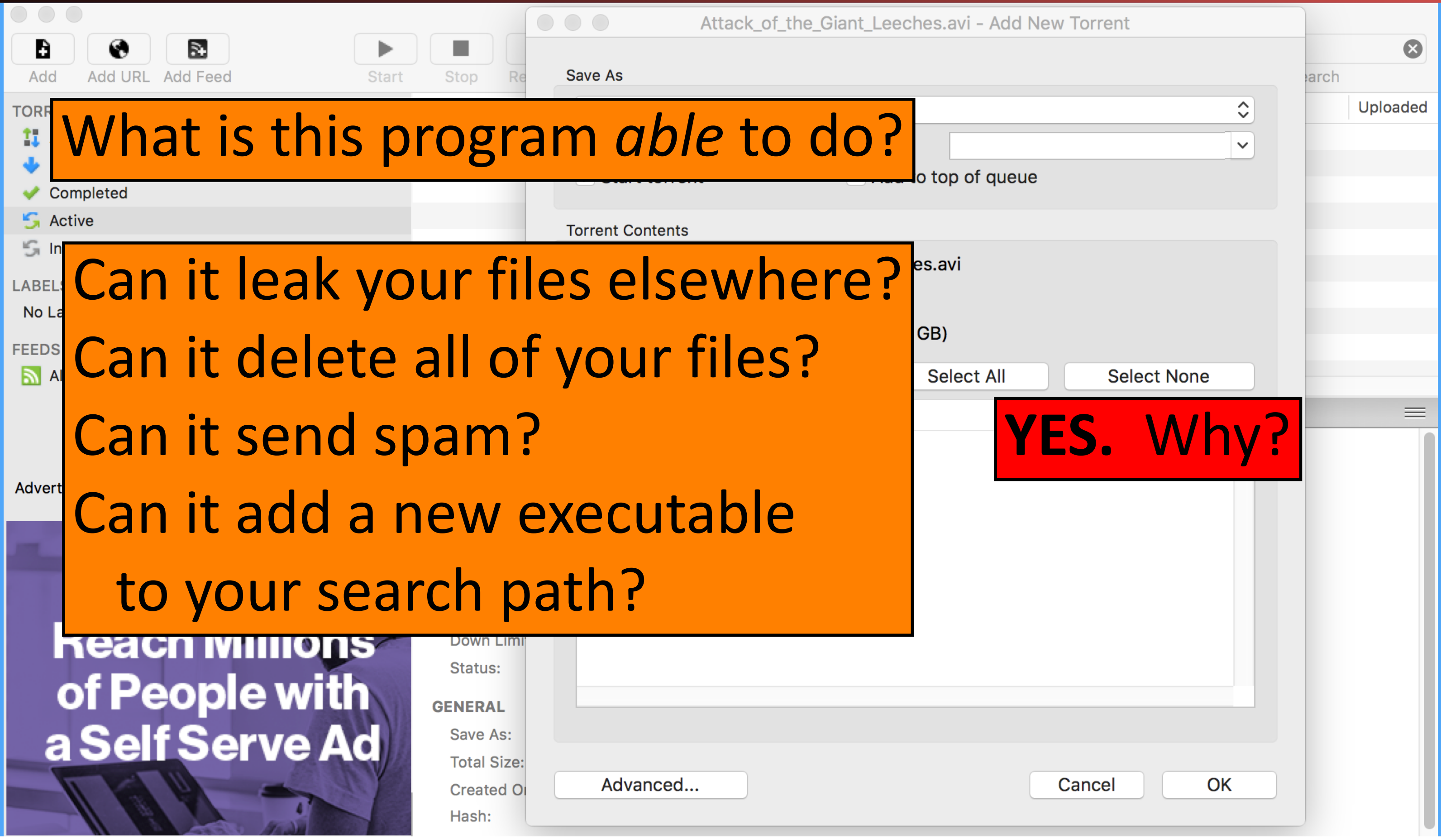
Size: 699 MB (Disk Space: 54.6 GB)

Date: 10/23/05, 2:50:07 AM

Name	Path	Size
✓ Attack_of_the_Giant_L...		699.8 MB

What is this program *able* to do?

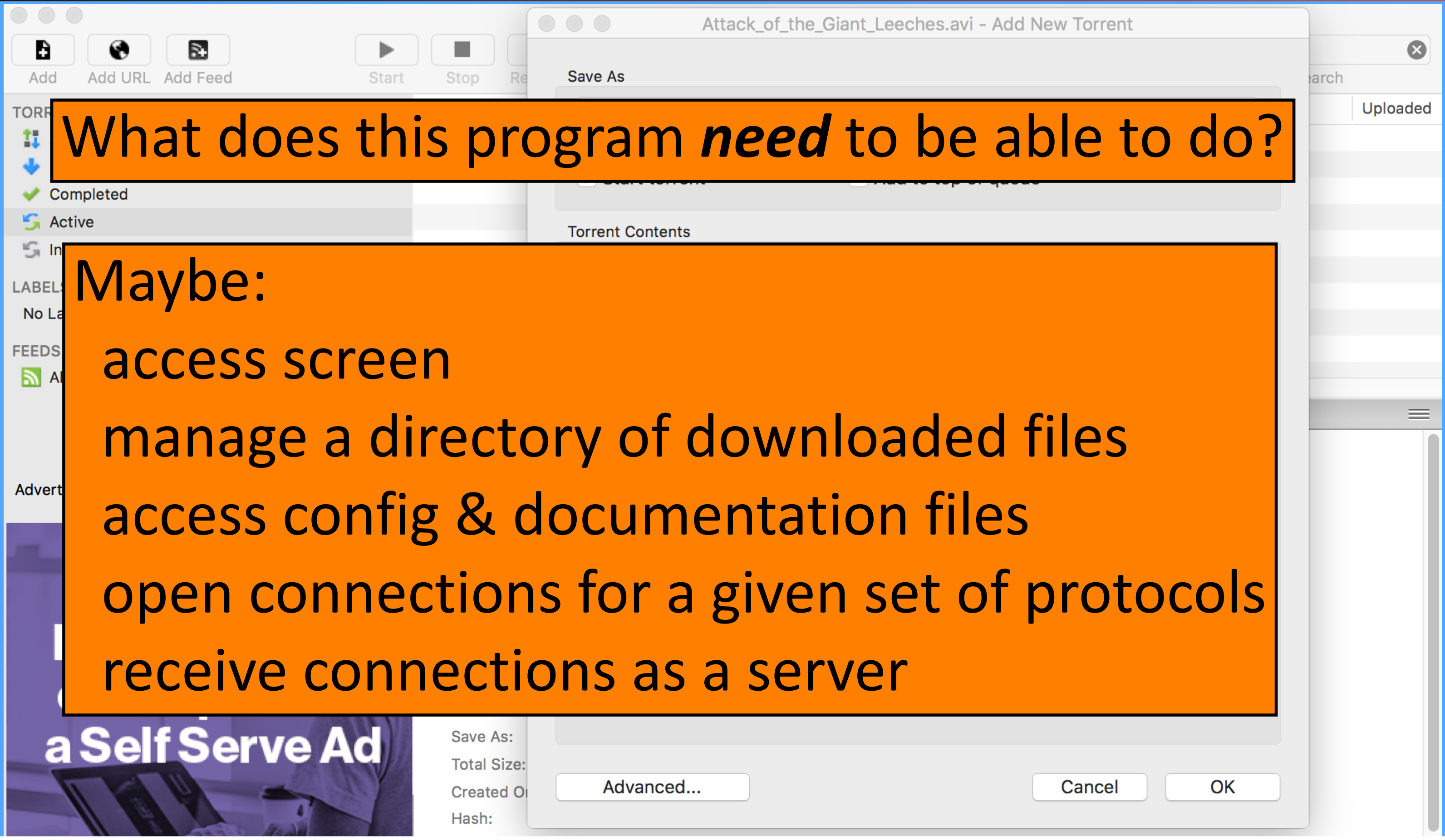
Can it leak your files elsewhere?



What is this program *able* to do?

Can it leak your files elsewhere?
Can it delete all of your files?
Can it send spam?
Can it add a new executable to your search path?

YES. Why?



What does this program *need* to be able to do?

Maybe:

access screen

manage a directory of downloaded files

access config & documentation files

open connections for a given set of protocols

receive connections as a server

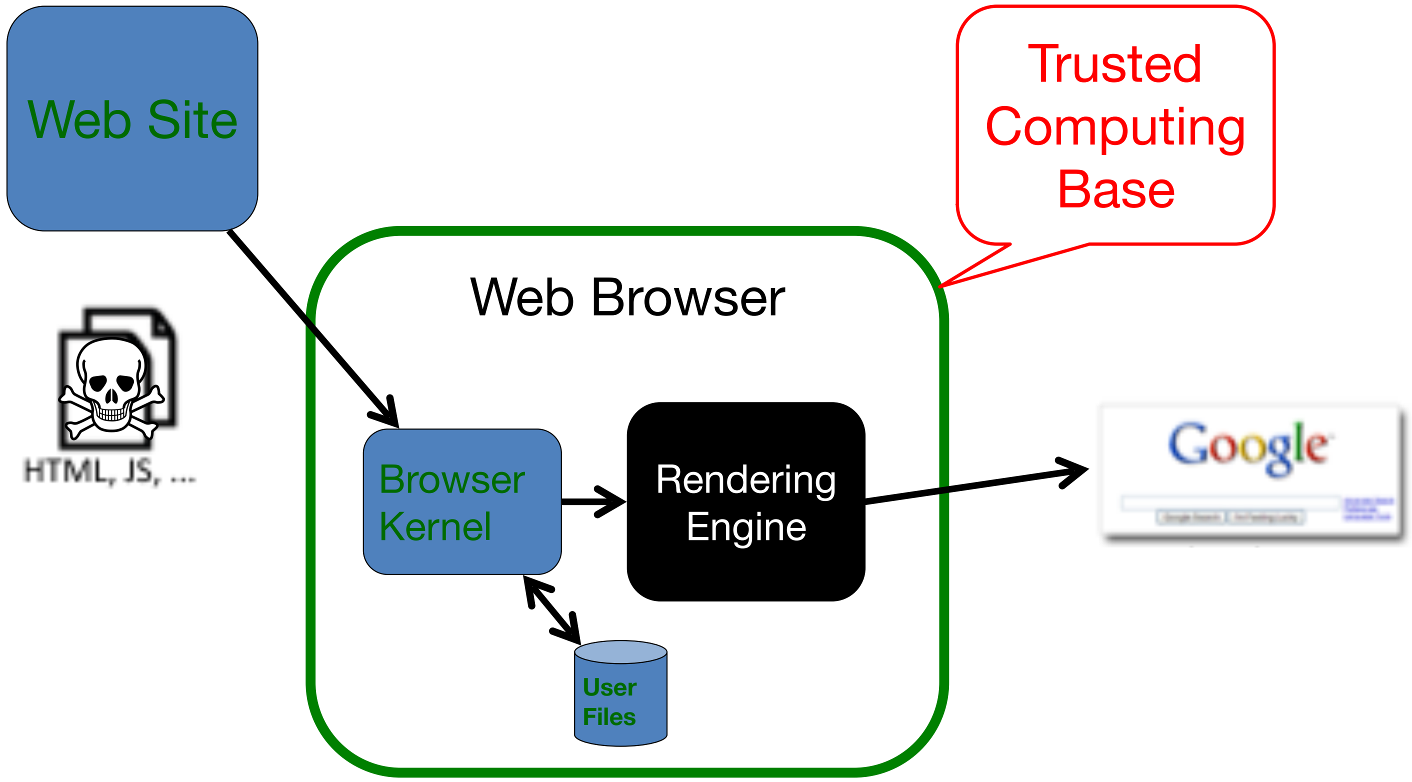
Thinking About Least Privilege

- When assessing the security of a system's design, identify the Trusted Computing Base (TCB).
  - What components does security ***rely upon***?
- Security requires that the TCB:
  - Is correct
  - Is complete (can't be bypassed)
  - Is itself secure (can't be tampered with)
- Best way to be assured of correctness and its security?
  - KISS = Keep It Simple, Stupid!
  - Generally, Simple = Small
- One powerful design approach: privilege separation
  - Isolate privileged operations to as small a component as possible

The Base for Isolation: The Operating System...

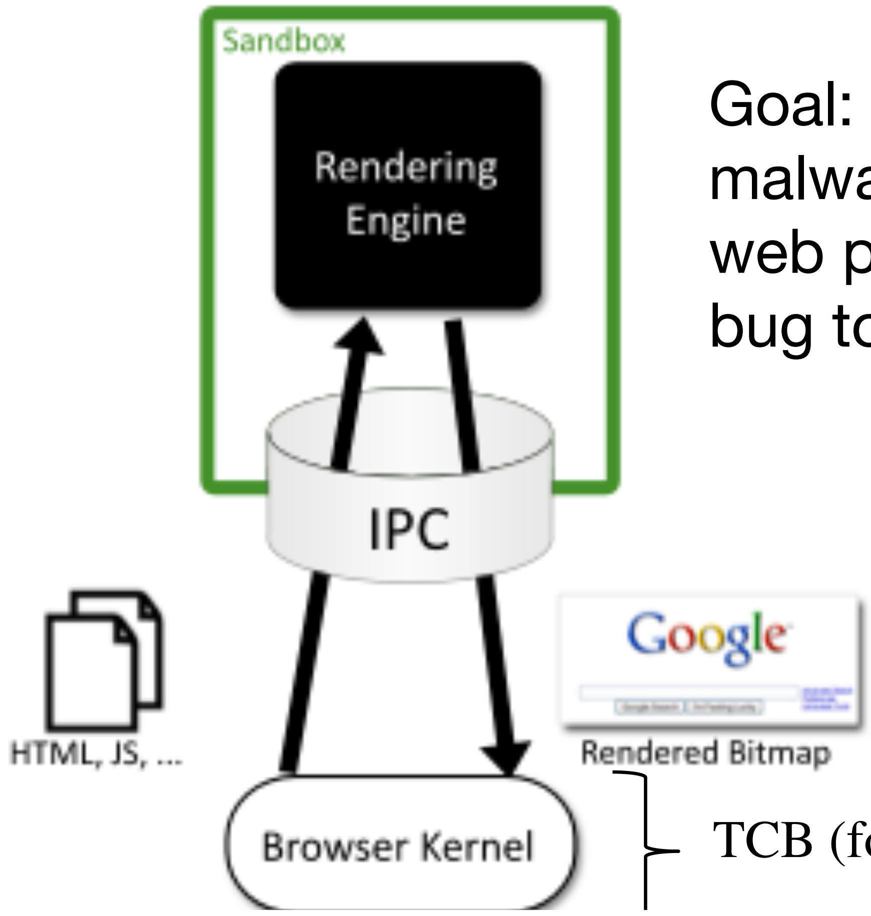
- The operating system ***process*** provide the following "guarantees" (you hope)
  - Isolation: A process can not access (read OR write) the memory of any other process unless both processes have set up a shared memory region
  - Permissions: A process can only change files etc if it has permission to
    - This ***usually*** means "Anything that the user can do" in something like Windows or MacOS
      - It can be considerably less in Android or iOS
- But even in Windows, MacOS, & Linux one can say "I don't want any permissions"
  - So if you have a process you can then have it "sandbox itself": peremptorily give up all rights

Web browser



“Drive-by malware”: malicious web page exploits browser bug to infect local files

The Chrome browser

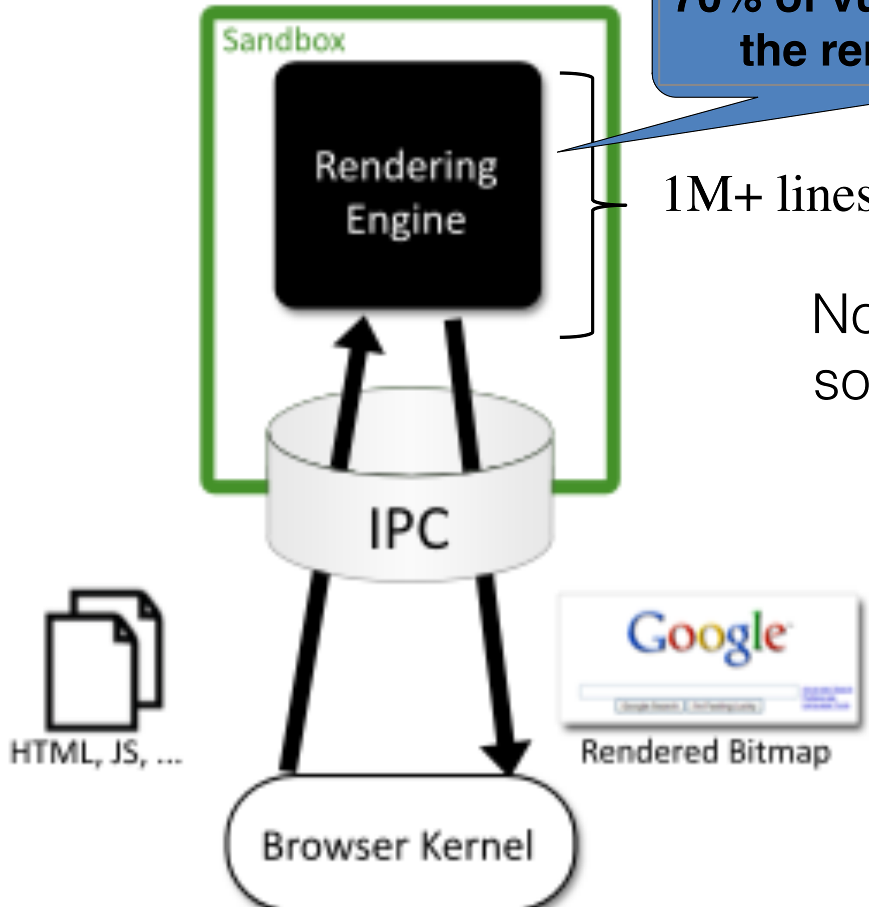


Goal: prevent “drive-by malware”, where a malicious web page exploits a browser bug to infect local files

$\}$ TCB (for this property)

The Chrome browser

70% of vulnerabilities are in the rendering engine.



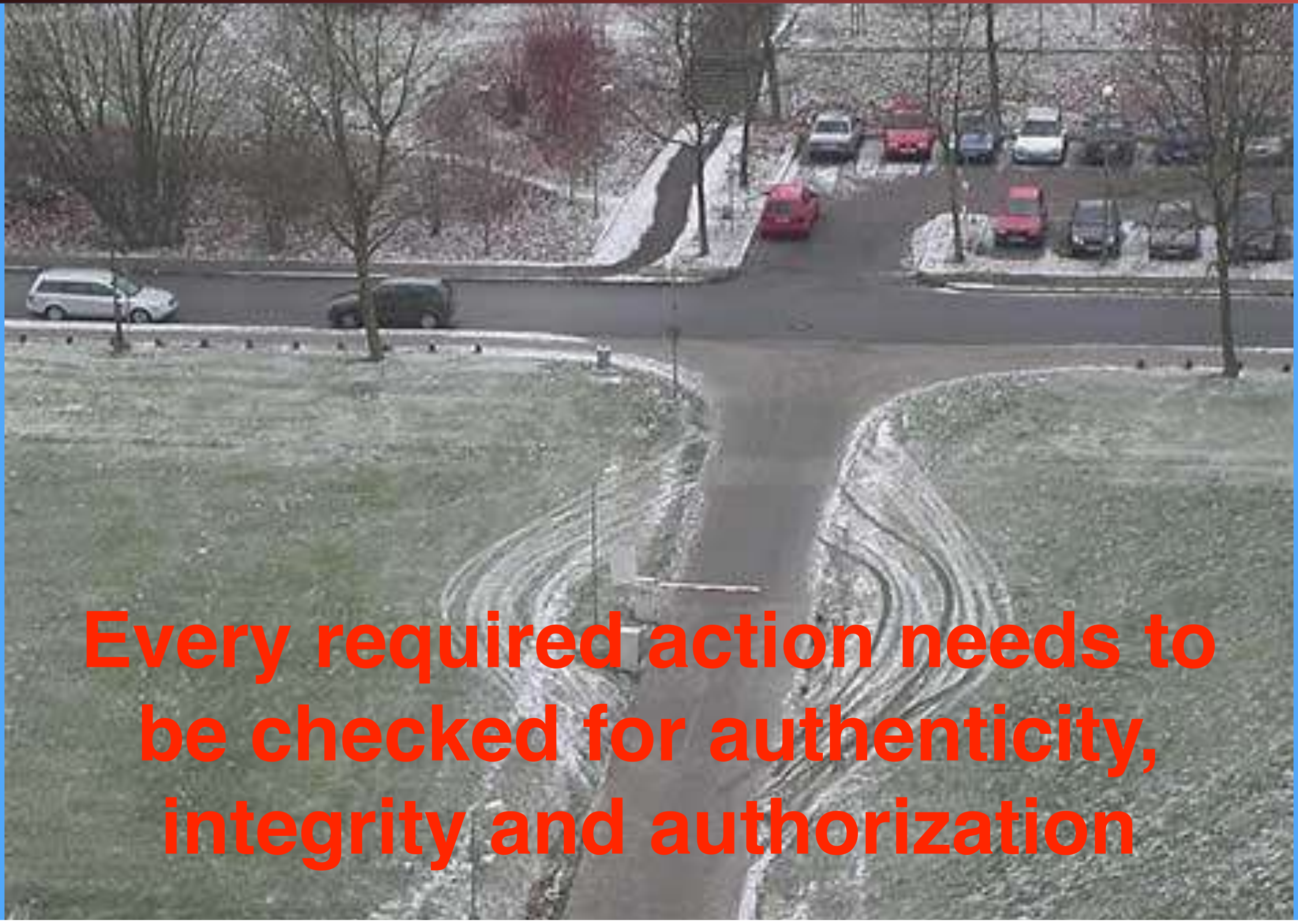
1M+ lines of code

Now it sandboxes ***each web context*** so you can't even read out other web page content (E.g. spectre)

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 …

A Failure of Complete Mediation



Every required action needs to be checked for authenticity, integrity and authorization

Time of Check to Time of Use Vulnerability: Race Condition

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

    2. if b < w, abort

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

    4. dispense $w to user
```

Suppose that *here* an attacker arranges to suspend first call, and calls `withdrawal` again **concurrently**

TOCTTOU = Time of Check To Time of Use

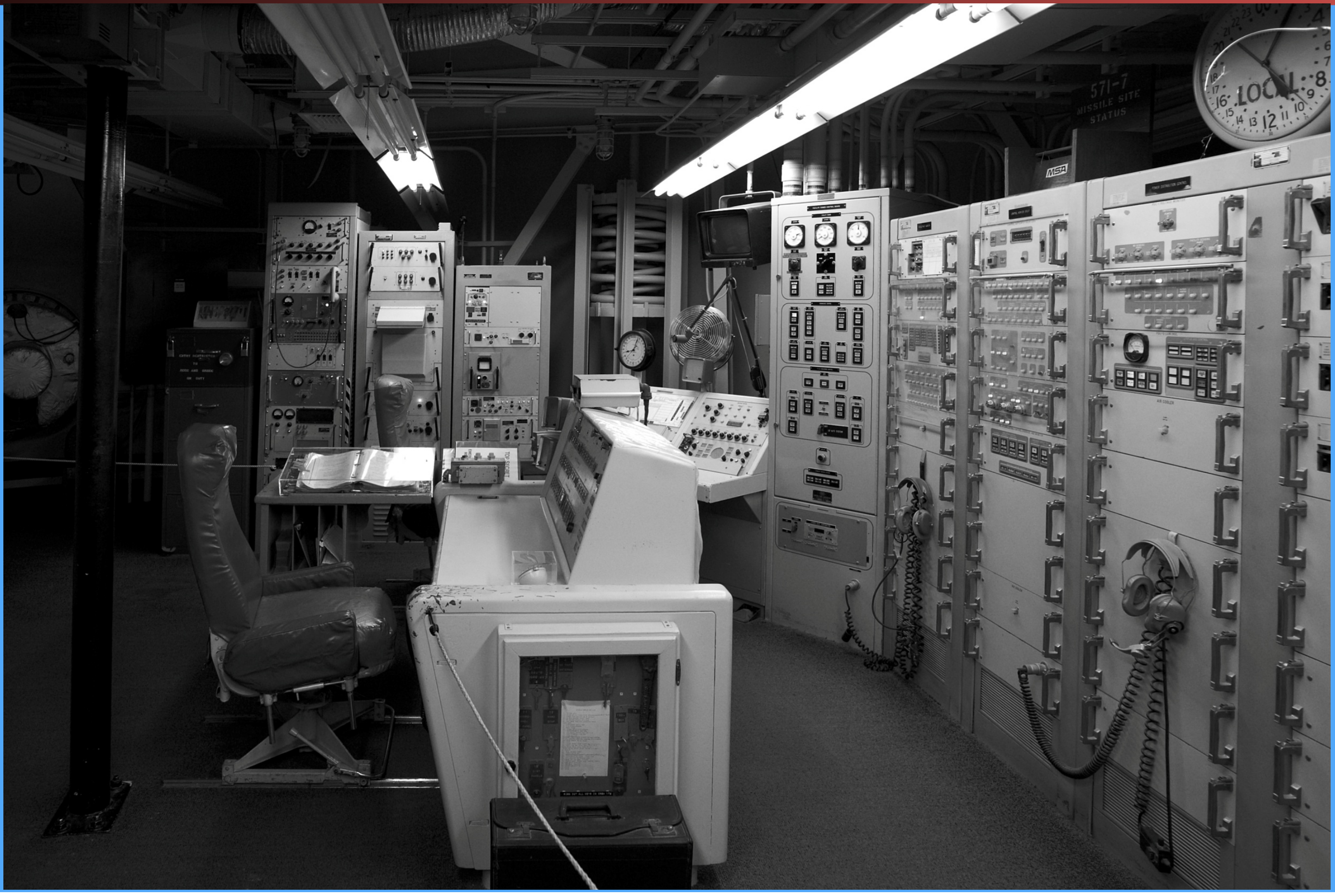
A Hundred Million Dollar TOCTTOU Bug...

- Ethereum is a cryptocurrency which offers "smart" contracts
  - Program you money in a language that makes JavaScript and PHP look beautiful and sane
- The DAO (Distributed Autonomous Organization) was an attempt to make a distributed mutual fund in Ethereum
  - Participants could vote on "investments" that should be made
    - Of course nobody actually had any idea what to do with the "investments" but hey, its the DAO! Gotta get in on the DAO!
- The DAO supported withdrawals as well
  - What is the point of a mutual fund that you couldn't take your money out of?

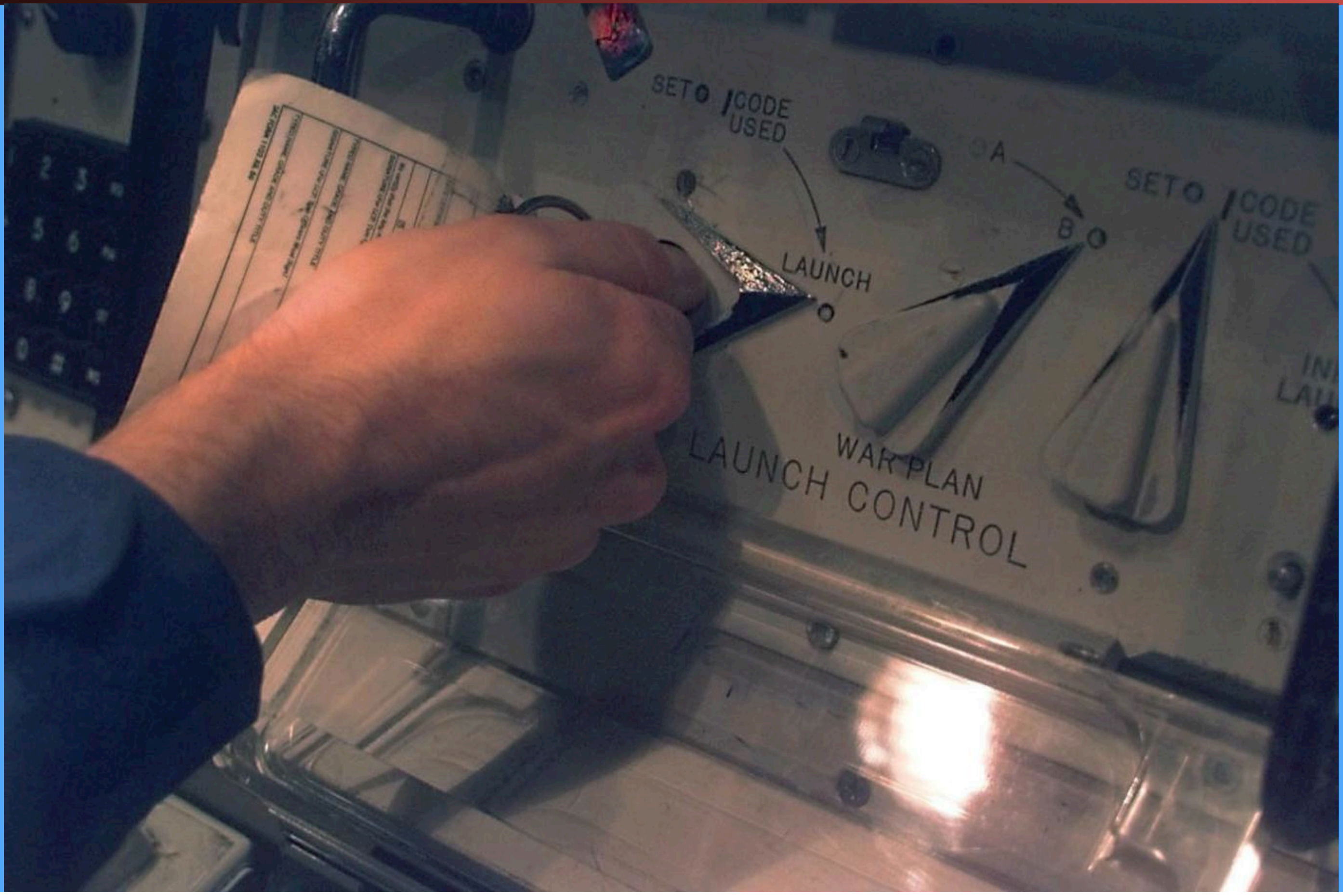
A "Feature" In The Smart Contract

- To withdraw, the code was:
  - Check the balance, then send the money, then decrement the balance
- But sending money in Ethereum can send to ***another program written by the recipient***
- So someone "invested", then did a withdraw to his program
  - Which would initiate another withdraw...

Welcome to a Nuclear Bunker



Two Man Control:
Each Needs To Turn the Key



Desired Security Property:
Only Want To Destroy The World ***On Purpose***

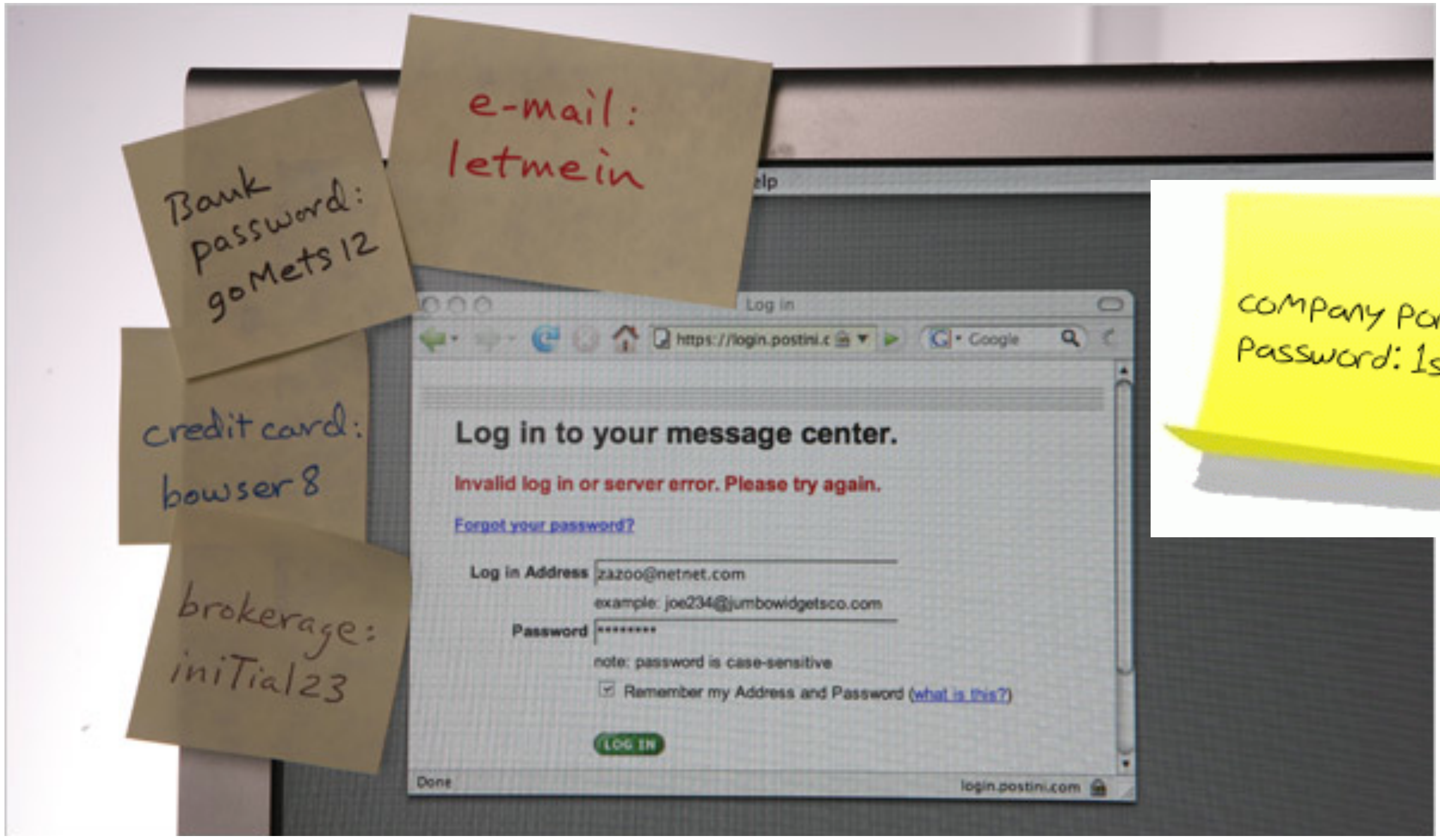
“Separation of responsibility.”

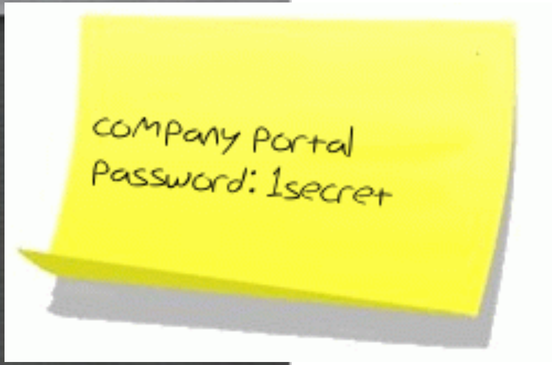
Independent audit

Summary:
Notions Regarding Managing Privilege

- Least privilege
  - The notion of avoiding having unnecessary privileges
- Privilege separation
  - A way to achieve least privilege by isolating access to privileges to a small Trusted Computing Base (TCB)
- Separation of responsibility
  - If you need to have a privilege, consider requiring multiple parties to work together (collude) to exercise it

Impact of a Password Policy

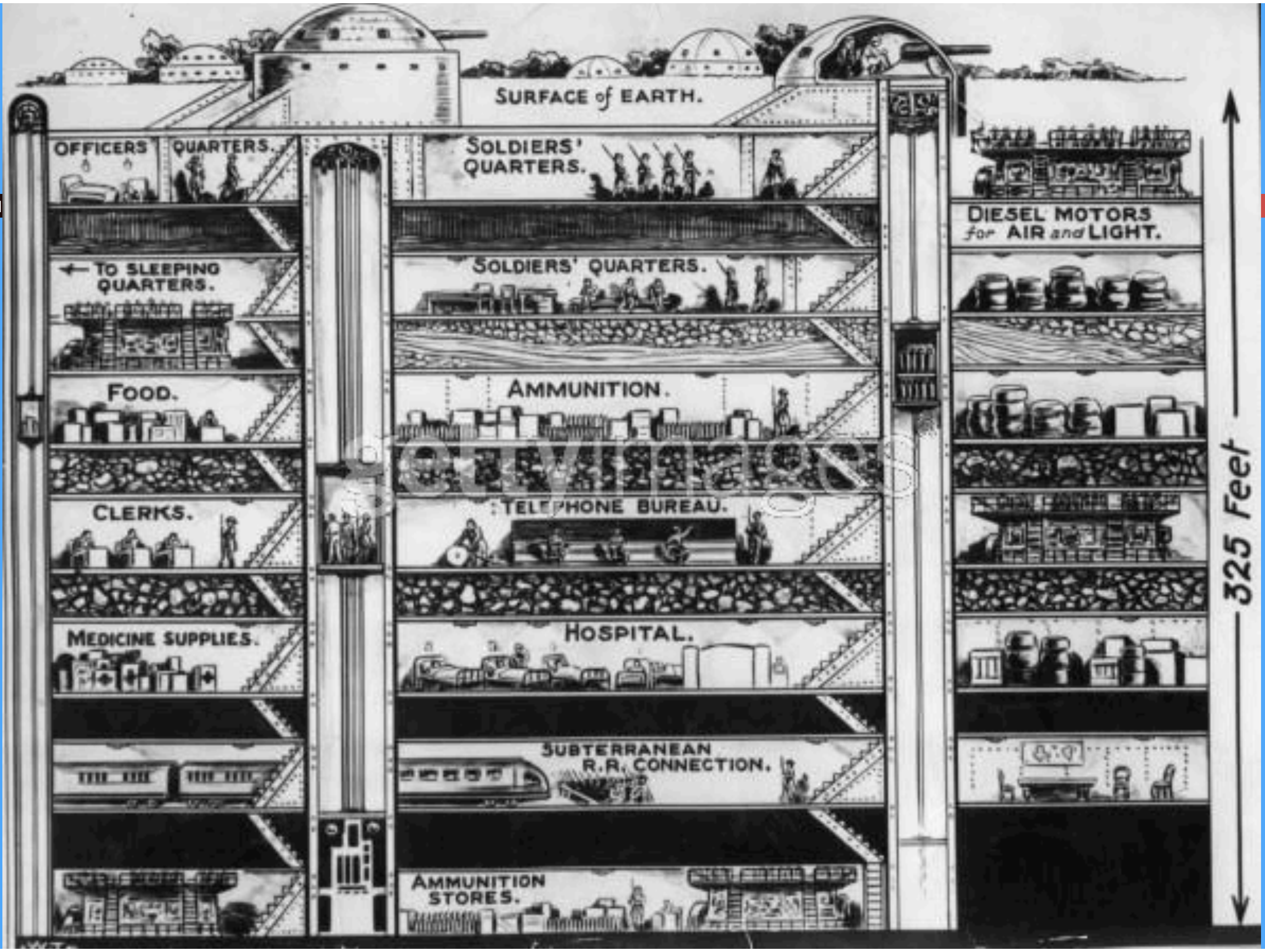

Bank password: goMets12
e-mail: letmein
credit card: bowser8
brokerage: iniTial23
Log In
https://login.postini.c
Google
Log in to your message center.
Invalid log in or server error. Please try again.
Forgot your password?
Log in Address zazoo@netnet.com
example: joe234@jumbowidgetsco.com
Password *********
note: password is case-sensitive
Remember my Address and Password (what is this?)
LOG IN
Done
login.postini.com

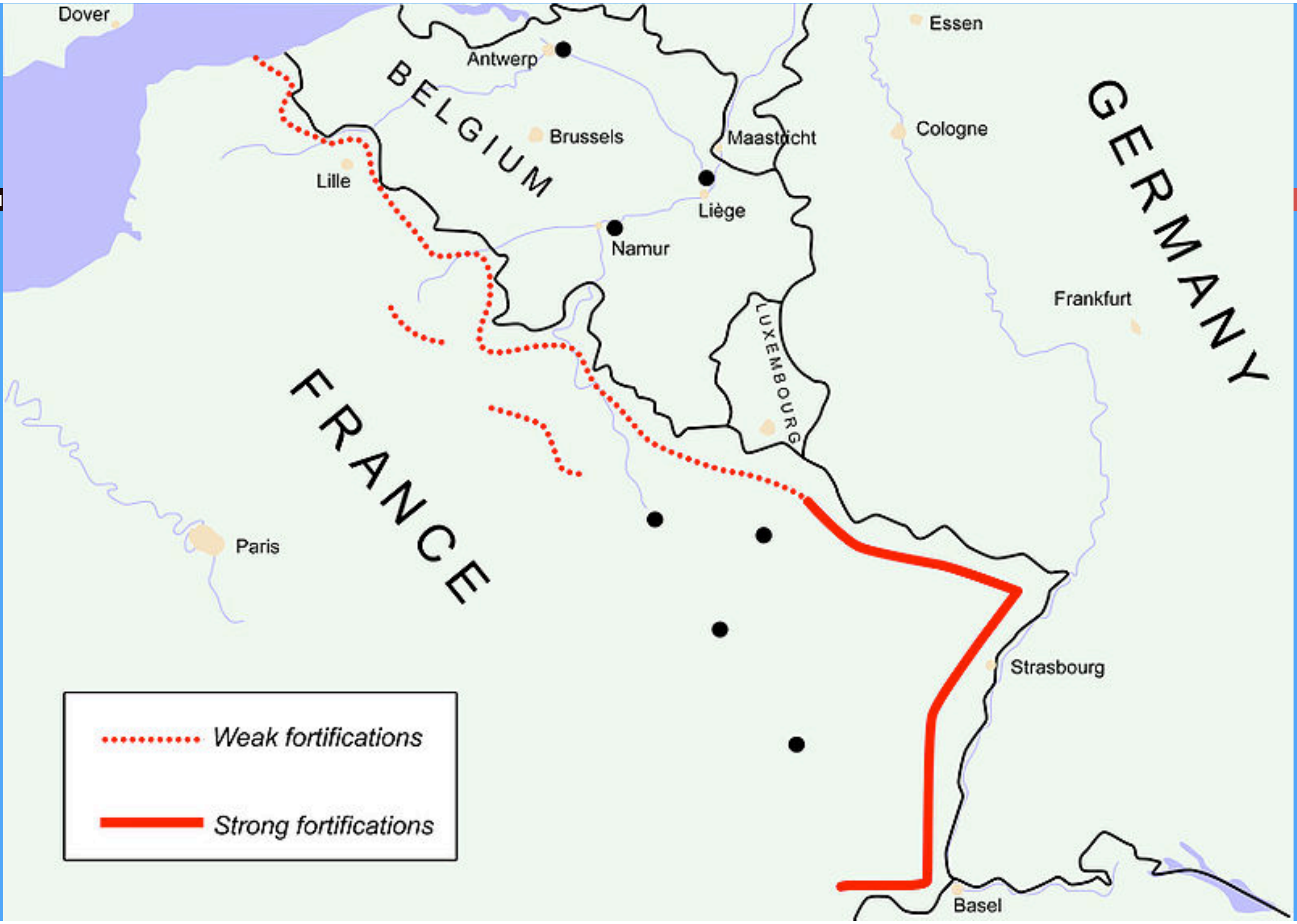


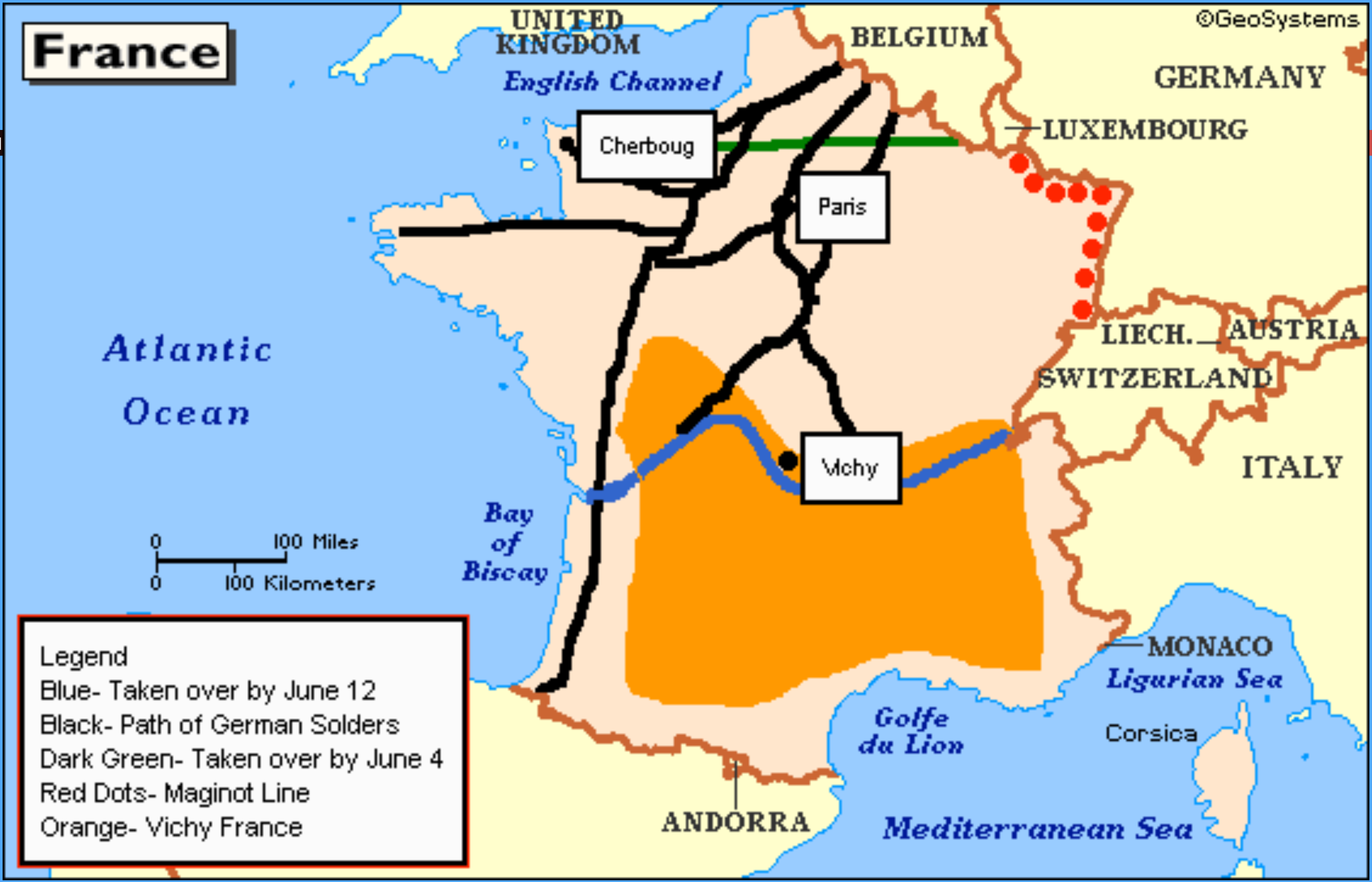
Summary:
Dealing with Users

- Psychological acceptability
  - Will users abide a security mechanism, or decide to subvert it?
    - Remember Rule 777...
- Consider human factors
  - Does a security mechanism assume something about human behavior when interacting with the system that might not hold, even in the absence of conscious decisions by the users to subvert
  - Have the computer do computer-y things, and humans do human-y things

2


SURFACE of EARTH.
OFFICERS QUARTERS.
SOLDIERS' QUARTERS.
DIESEL MOTORS for AIR and LIGHT.
← TO SLEEPING QUARTERS.
SOLDIERS' QUARTERS.
FOOD.
AMMUNITION.
CLERKS.
TELEPHONE BUREAU.
MEDICINE SUPPLIES.
HOSPITAL.
SUBTERRANEAN R.R. CONNECTION.
AMMUNITION STORES.
325 Feet

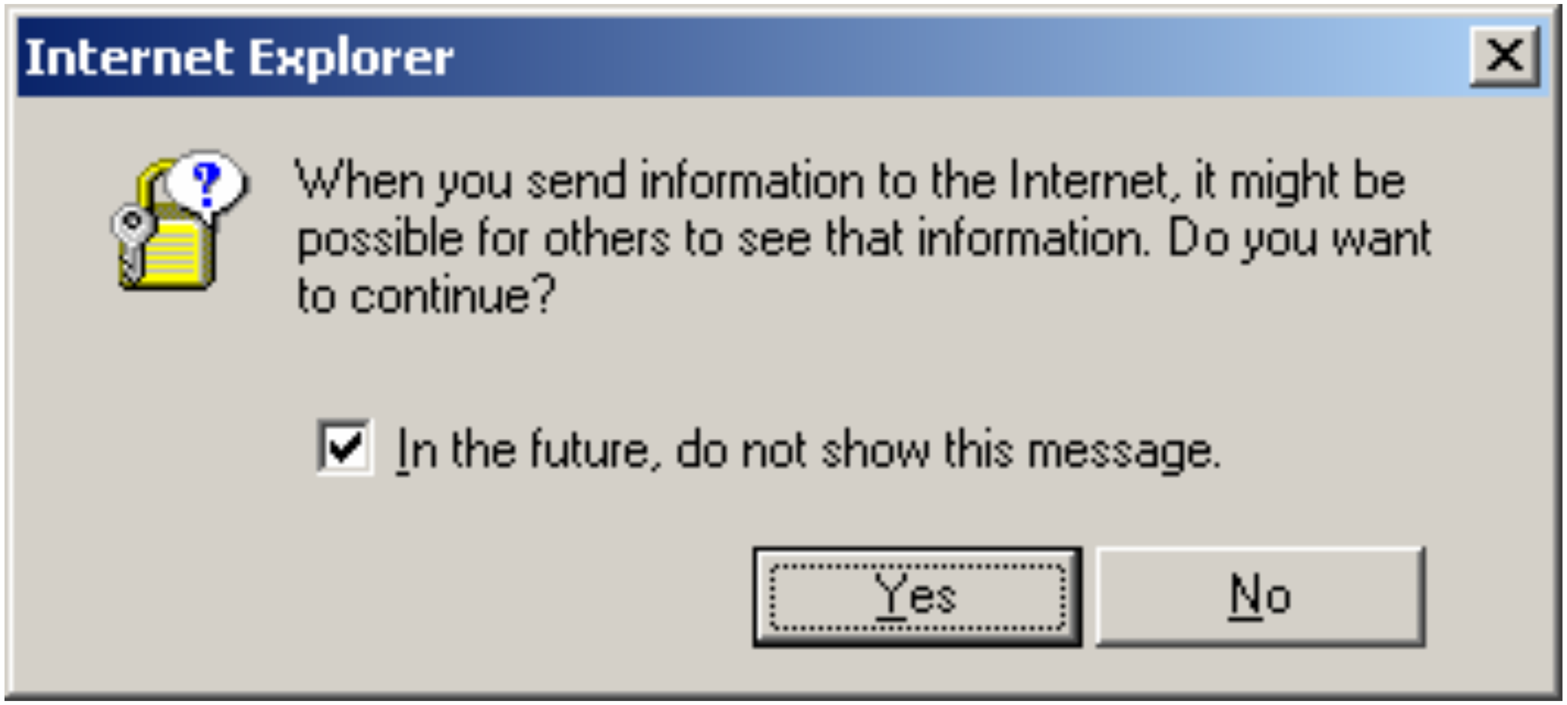

Dover
Antwerp
BELGIUM
Brussels
Maastricht
Liège
Namur
Lille
Essen
Cologne
GERMANY
Frankfurt
LUXEMBOURG
FRANCE
Paris
Strasbourg
Basel
Weak fortifications
Strong fortifications



Legend
Blue- Taken over by June 12
Black- Path of German Solders
Dark Green- Taken over by June 4
Red Dots- Maginot Line
Orange- Vichy France

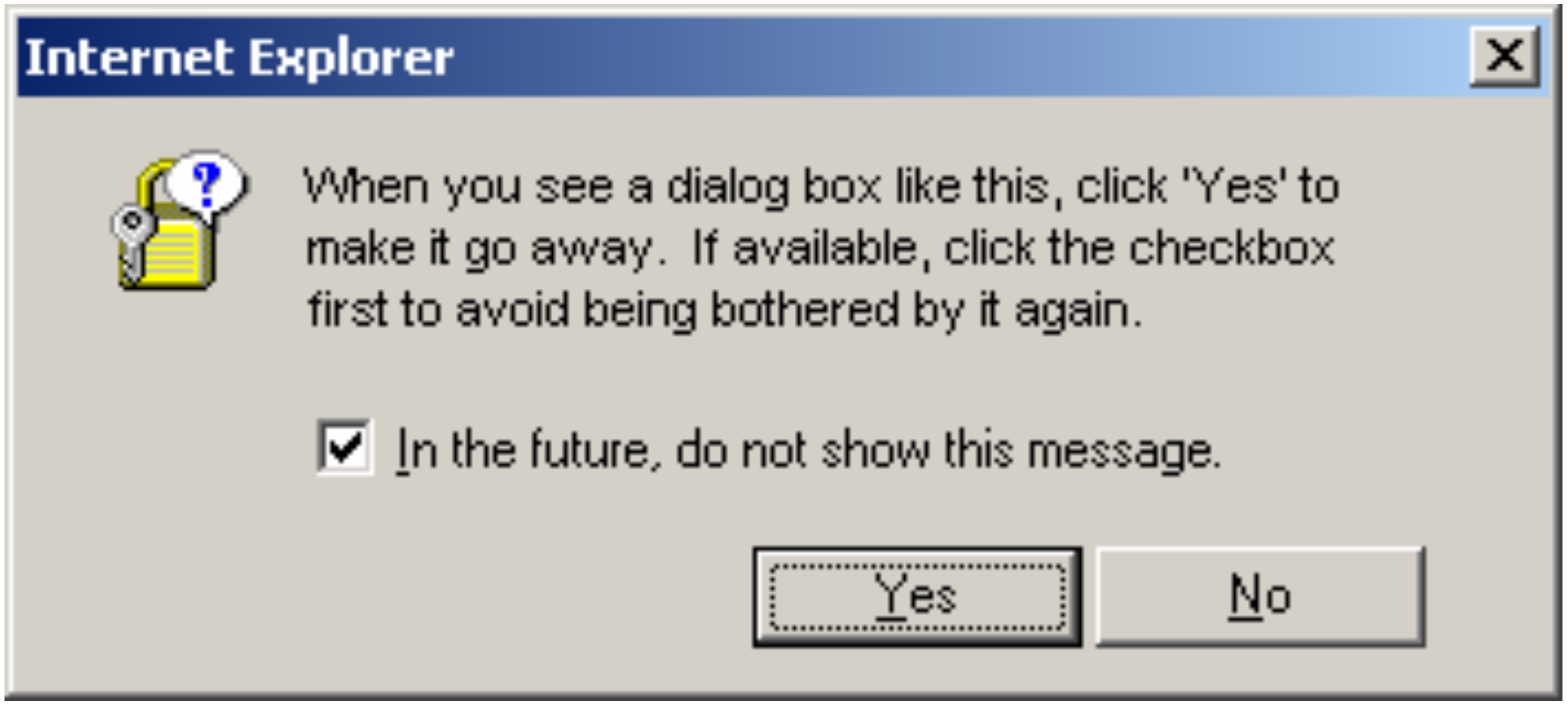
“Only as secure as the weakest link.”

- "A door lock is only as strong as the window"


Internet Explorer

When you send information to the Internet, it might be possible for others to see that information. Do you want to continue?

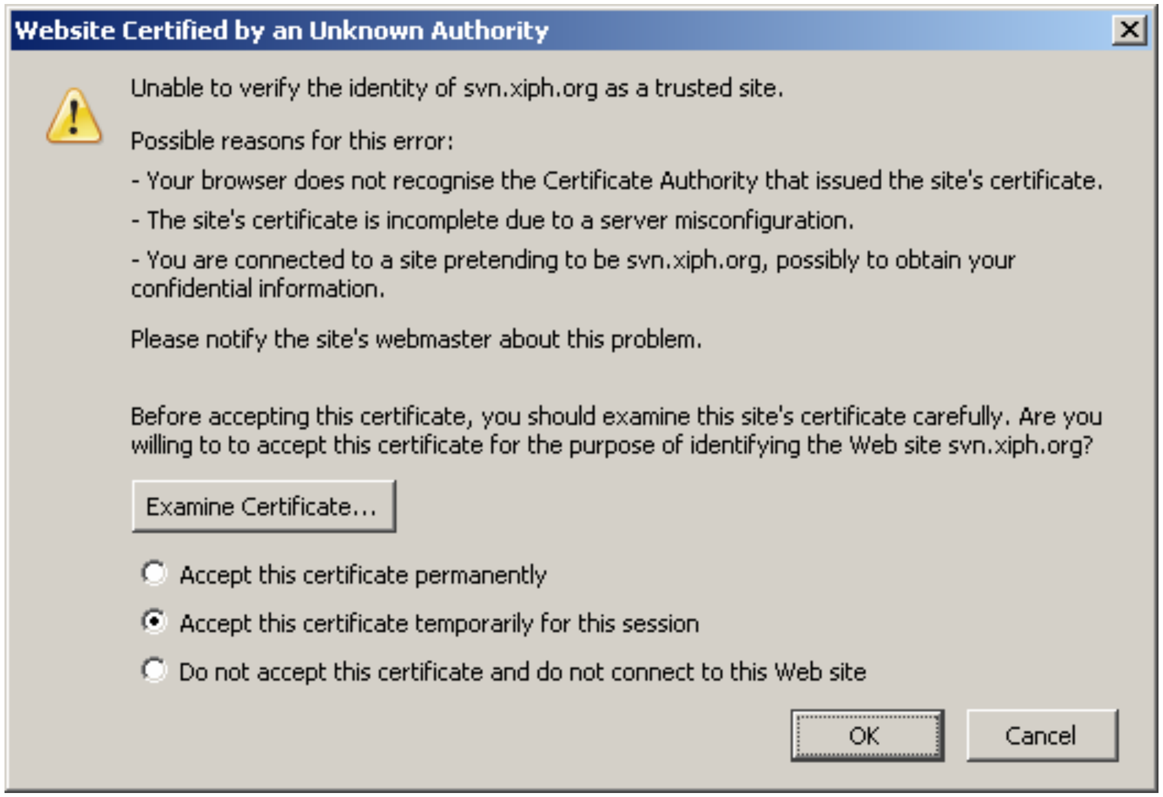
☑ In the future, do not show this message.



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.



Website Certified by an Unknown Authority

Unable to verify the identity of svn.xiph.org as a trusted site.

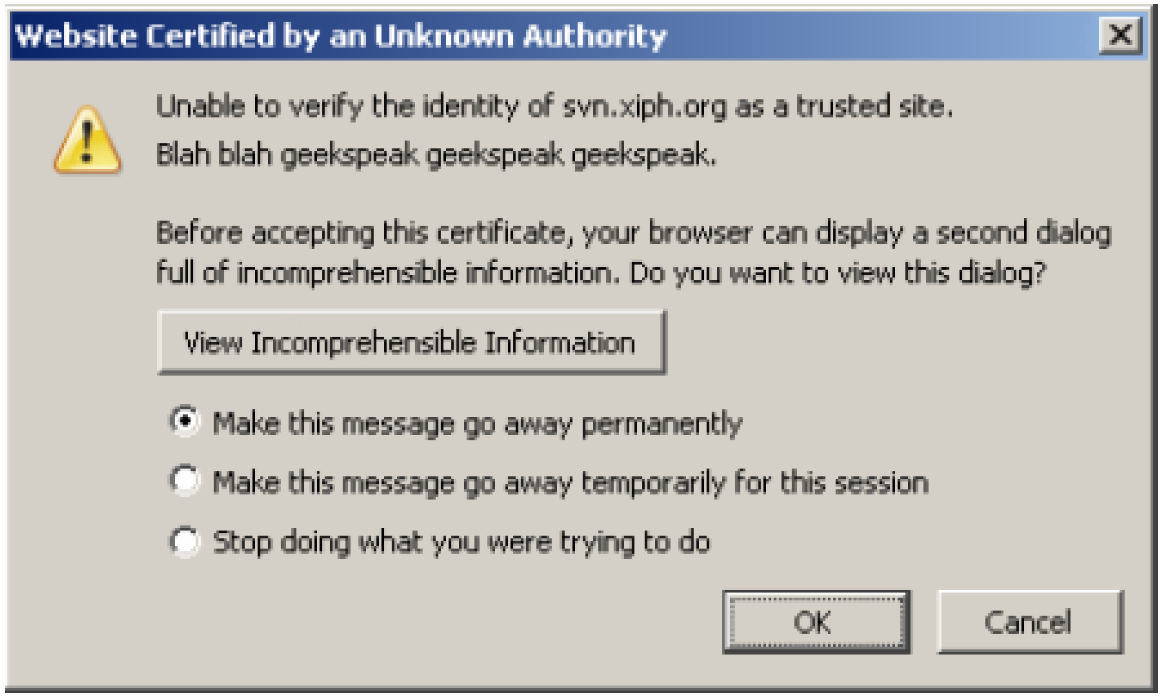
Possible reasons for this error:

- 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?

○ Accept this certificate permanently
[✓ Accept this certificate temporarily for this session]
○ Do not accept this certificate and do not connect to this Web site



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?

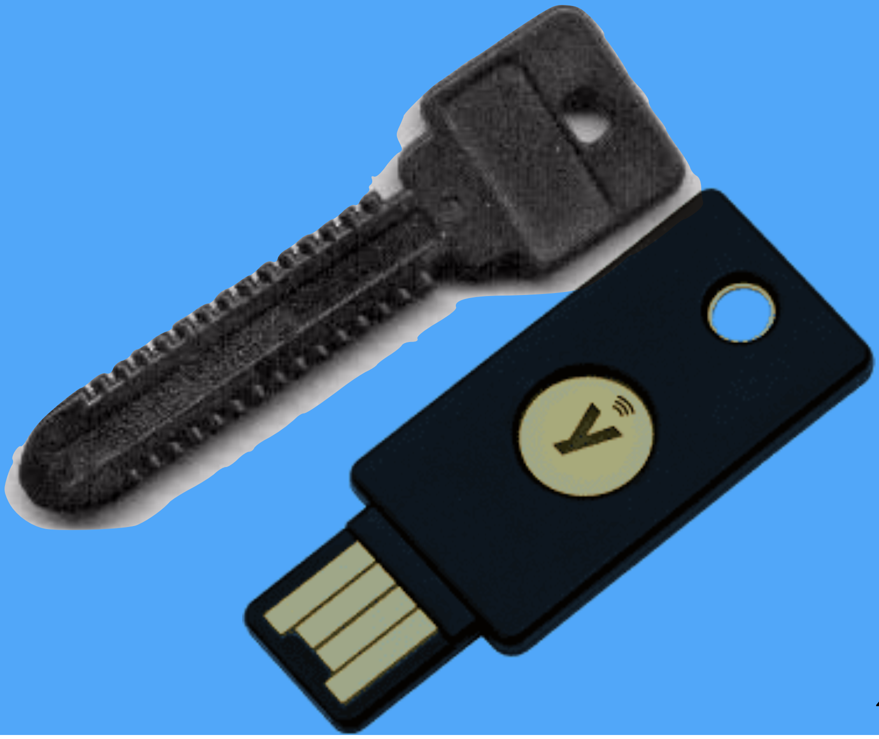
[✓ Make this message go away permanently]
Make this message go away temporarily for this session
Stop doing what you were trying to do

Whenever you confront the user like this it is probably "Blame the User" Security

- Given two choices, one catastrophically wrong...
  - Users are going to chose the "wrong" thing >50% of the time!

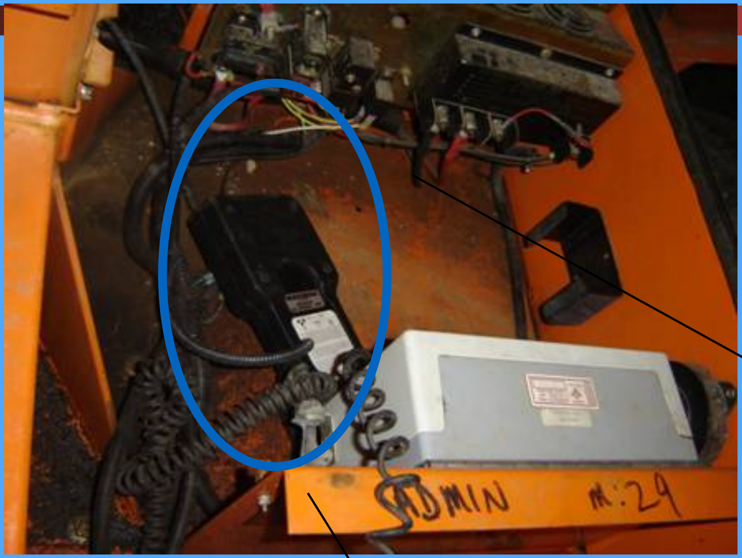
Security Keys and Human Factors

- This is a security key for storing key material for an encrypted military phone
  - Leverages a lifetime of knowledge in how to protect physical keys
- U2F security keys leverage the same knowledge!
- Product/design idea:
  A ***physical*** doorlock that uses a U2F key!



ACCIDENT
ON
MOTORWAY

ACCIDENT
ON
MOTORWAY


ADMIN

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

RAPTORS
AHEAD
CAUTION

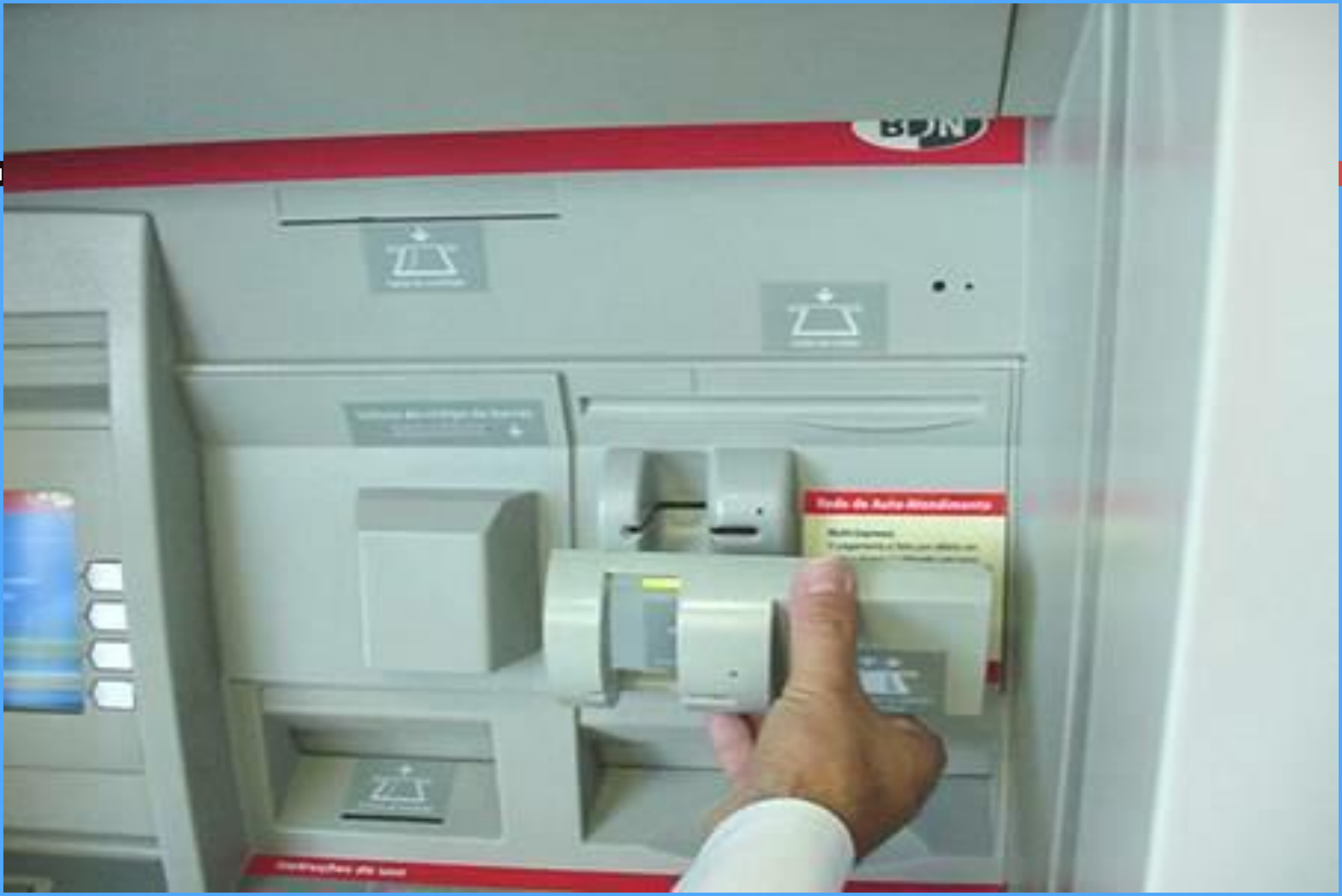
TRAPPED
IN SIGN
FACTORY

SEND
HELP!

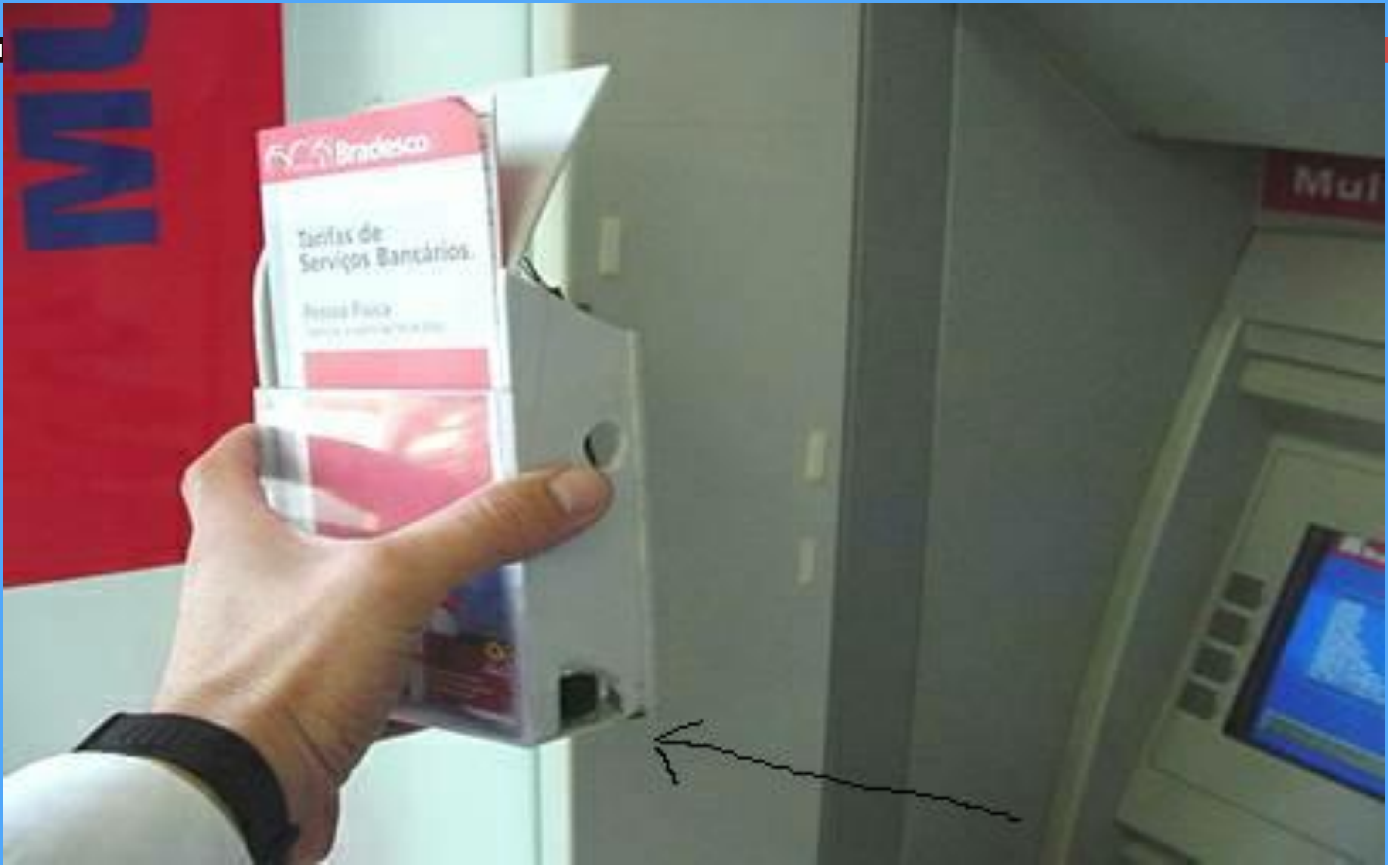
“Don’t rely on security through obscurity.”

- Because otherwise the raptors will get you...
- Obscurity does help but you need to design your system so that it fails...
- Kerckhoffs's Principle:
  - A cryptosystem should be secure even if everything about the system, ***except the key***, is public knowledge.
- Shannon's Maxim:
  - The enemy knows the system
- AND ***FOR FUCKS SAKE DON'T DO THIS YOURSELVES***!!!

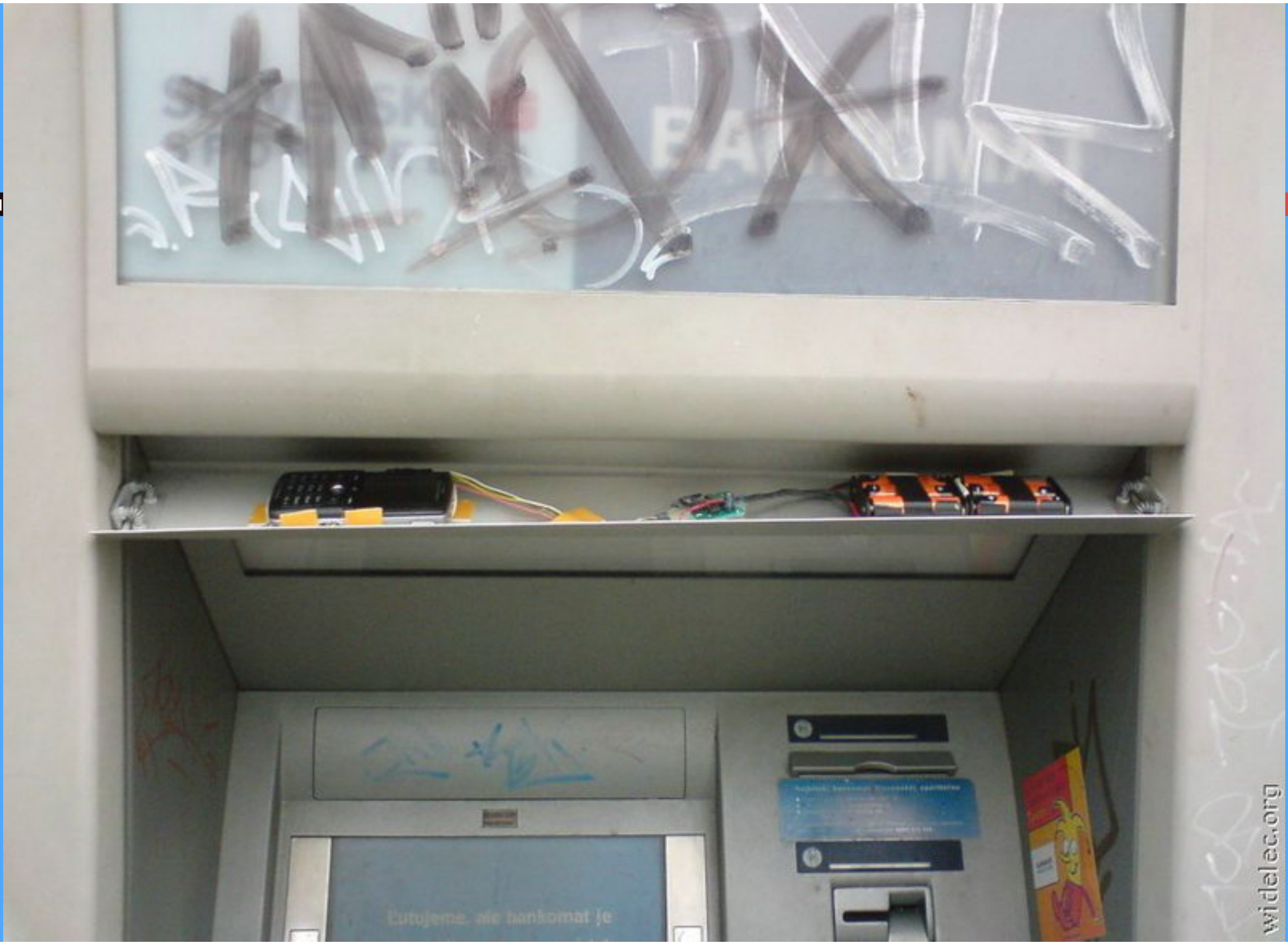
Leitura do código de barras
Instruções de uso



Bradesco
Tarifas de Serviços Bancários
Multi Expresso
Bradesco


Bradesco
Tarifas de
Serviços Bancários

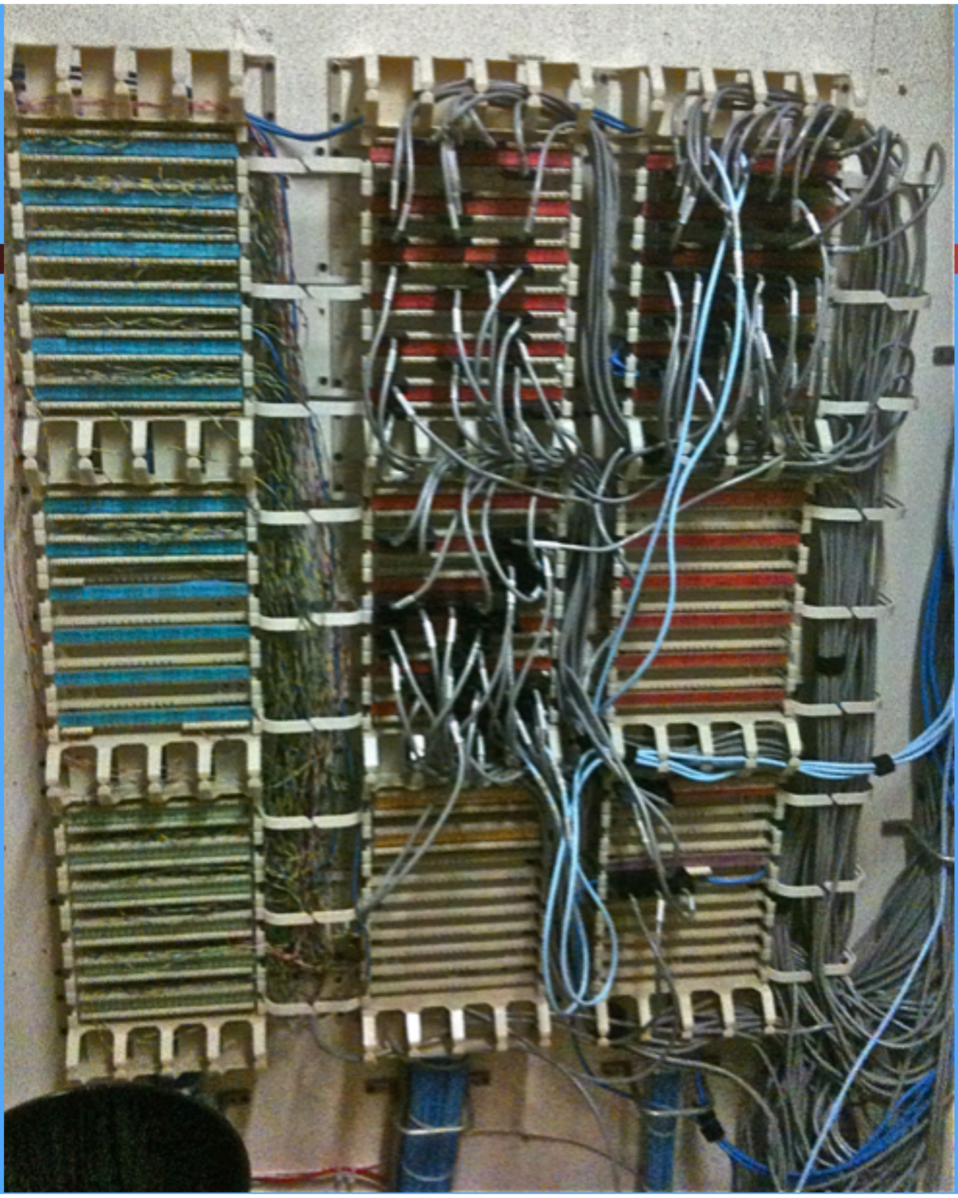
widelec.org


widelec.org
Lutujeme, ale bankomat je

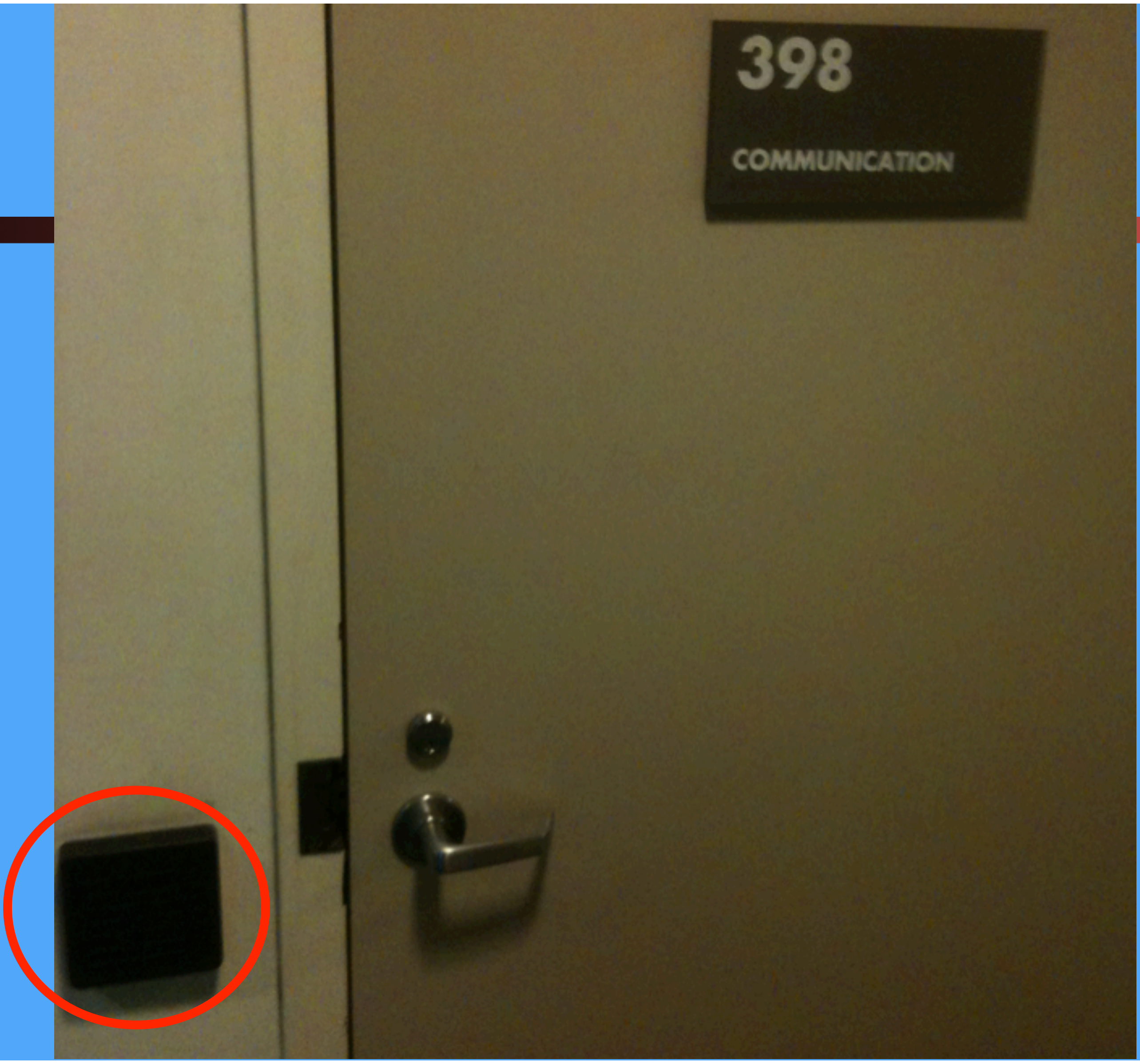
“Trusted path.”

- Users need to know they are talking with the legit system
- System needs to know its talking with the legit user
- These channels need to be unspoofable and private
    - ATM skimmers are a failure of the trusted path

Soda Hall wiring closet



Protection?


398
COMMUNICATION

“Use fail-safe defaults.”

- But it can often be hard to determine
- Default for access here is reasonable...
  - Deny all except for an allowed user list
- But when the power goes out...
  - Should the lock fail shut?
    Should the lock fail open?

Common Assumptions When Discussing Attacks

- (Note, these tend to be pessimistic … but prudent)
- Attackers can interact with our systems ***without particular notice***
  - Probing (poking at systems) may go unnoticed …
  - … even if highly repetitive, leading to crashes, and easy to detect
- It’s easy for attackers to know general information about their targets
  - OS types, software versions, usernames, server ports, IP addresses, usual patterns of activity, administrative procedures

Common Assumptions, con’t

- Attackers can obtain access to a copy of a given system to measure and/or determine how it works
  - Shannon's Maxim: "The Enemy Knows the System"
- Attackers can make energetic use of automation
  - They can often find clever ways to automate:
    If an attack has a 1 in 2^{30} chance of success, the attacker just tries a ***billion*** times!
- Attackers can pull off complicated coordination across a bunch of different elements/systems
- Attackers can bring large resources to bear if req’d
  - Computation, network capacity
  - But they are not super-powerful (e.g., control entire ISPs)

Common Assumptions, con’t

- If it helps the attacker in some way, ***assume they can obtain privileges***
  - But if the privilege gives everything away (attack becomes trivial), then we care about unprivileged attacks
- The ability to robustly detect that an attack has occurred ***does not replace desirability of preventing***
- Infrastructure machines/systems are well protected (hard to directly take over)
  - So a vulnerability that requires infrastructure compromise is less worrisome than same vulnerability that doesn’t

Common Assumptions, con’t

- Network routing is hard to alter … other than with physical access near clients (e.g., “wifi/coffeeshop”)
  - Such access helps fool clients to send to wrong place
  - Can enable Man-in-the-Middle (MITM) attacks
- We worry about attackers who are lucky
  - Since often automation/repetition can help “make luck”:
    If its 1 in a million, just try a million times!
- Just because a system does not have apparent value,
  it may still be a target
  - "Lets break into the Casino network... Through the fishtank"
- Attackers are mostly undaunted by fear of getting caught
  - There are exceptions

Patches & 0-days

- Systems have vulnerabilities all the time...
  - A ***patch*** is an update which is designed to remove such vulnerabilities.
- An "0-day" is an exploit where nobody but the attacker knows about
  - So there ***is*** no patch
- But 0-days are rare: Require independent discovery...
  - But it is straightforward to take a patch and find an exploit
- So patch religiously!
  - Similarly, the "patch" for influenza is the flu-shot. ***GET ONE***!
  - Just as the University requires that computers meet basic security standards, they are ***finally*** requiring that student immune systems meet basic security standards

And Most Exploits These Days Are Chains...

- EG, to pwn an iPhone...
  - Need an exploit for the browser to start running code within the browser's sandbox
  - And another exploit to break out of the sandbox and take over the OS kernel...
    - And that other exploit may actually be 2-3 exploits themselves chained together
- So e.g. on the massive Chinese campaign a year ago...
  - There was one known 0-day in the chains...
  - But taking over the browser MAY have only been 1-day:
    Take patch, derive exploit. (We just don't know...)