

Security Principles





Happy Birthday, Linux!

Here's your cake, go ahead and compile it yourself.

Announcement: Logistics

- Project 1 & Homework 1: Released on January 28th
- Get in your accommodation requests on midterms/final:
 - MT1: Feb 21, 7pm - 9pm.
 - MT2: Mar 13, 7pm - 9pm.
 - Final: May 16, 3pm - 6pm.

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...

Computer Science 161 Spring 2019

- 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



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 these standards
 - The "gun safe" standards from the California Department of Justice





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

Mac and OSX Downloads - μ Torrent® (uTorrent) - a (very) tiny ...

www.utorrent.com/downloads/mac ▾

Download the official μ Torrent® (uTorrent) torrent client for Windows, Mac, Android or Linux-- uTorrent ... For Mac (1.42 MB); English (US) - November 27, 2016.

uTorrent (Mac)

μ torrent estable(1.8.7 build 43001).

Para Mac (1.42 MB); Inglés ...

[More results from utorrent.com »](#)

Download

μ Torrent Stable(1.8.7 build 43001).

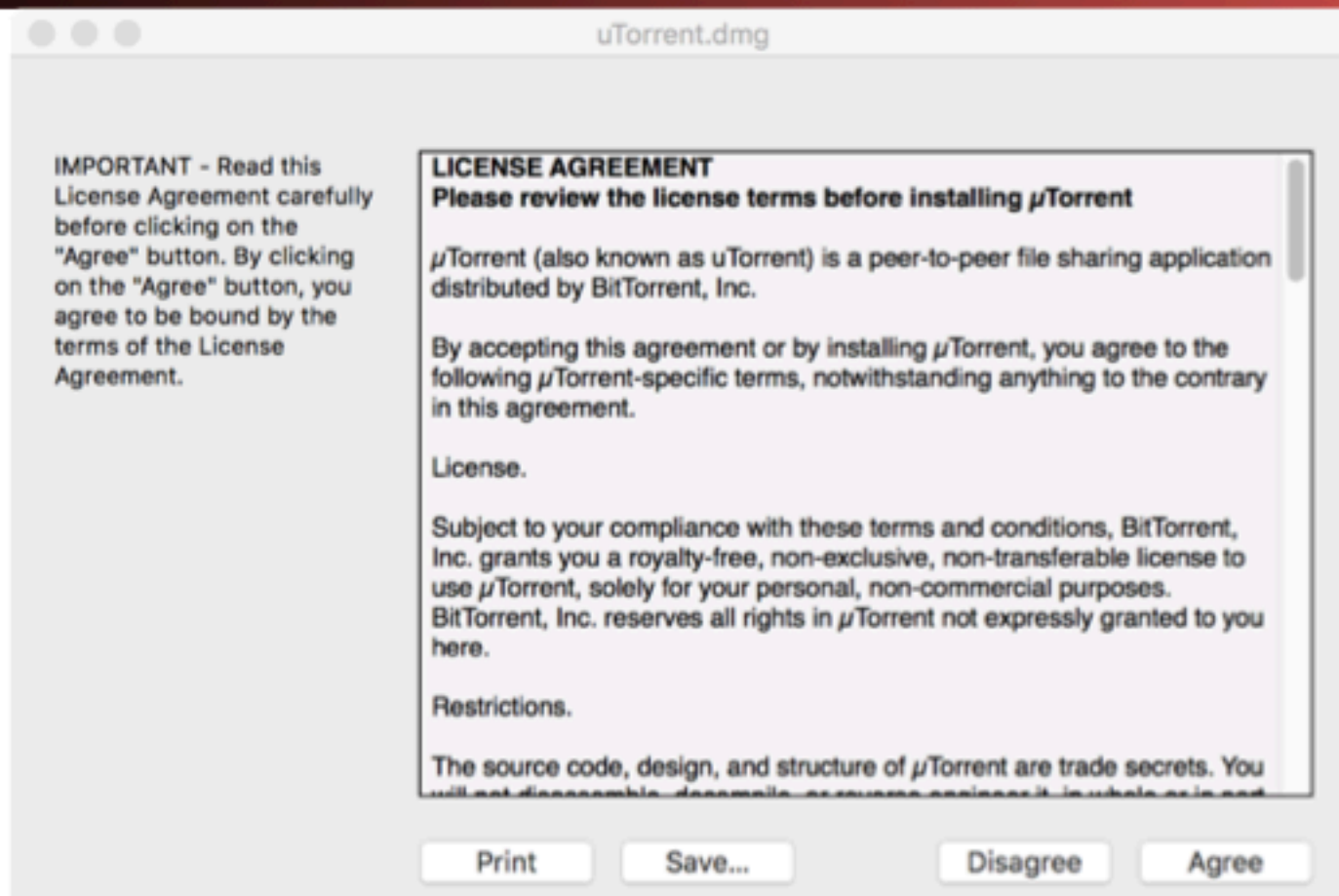
Für Mac (1.42 MB); Englisch ...

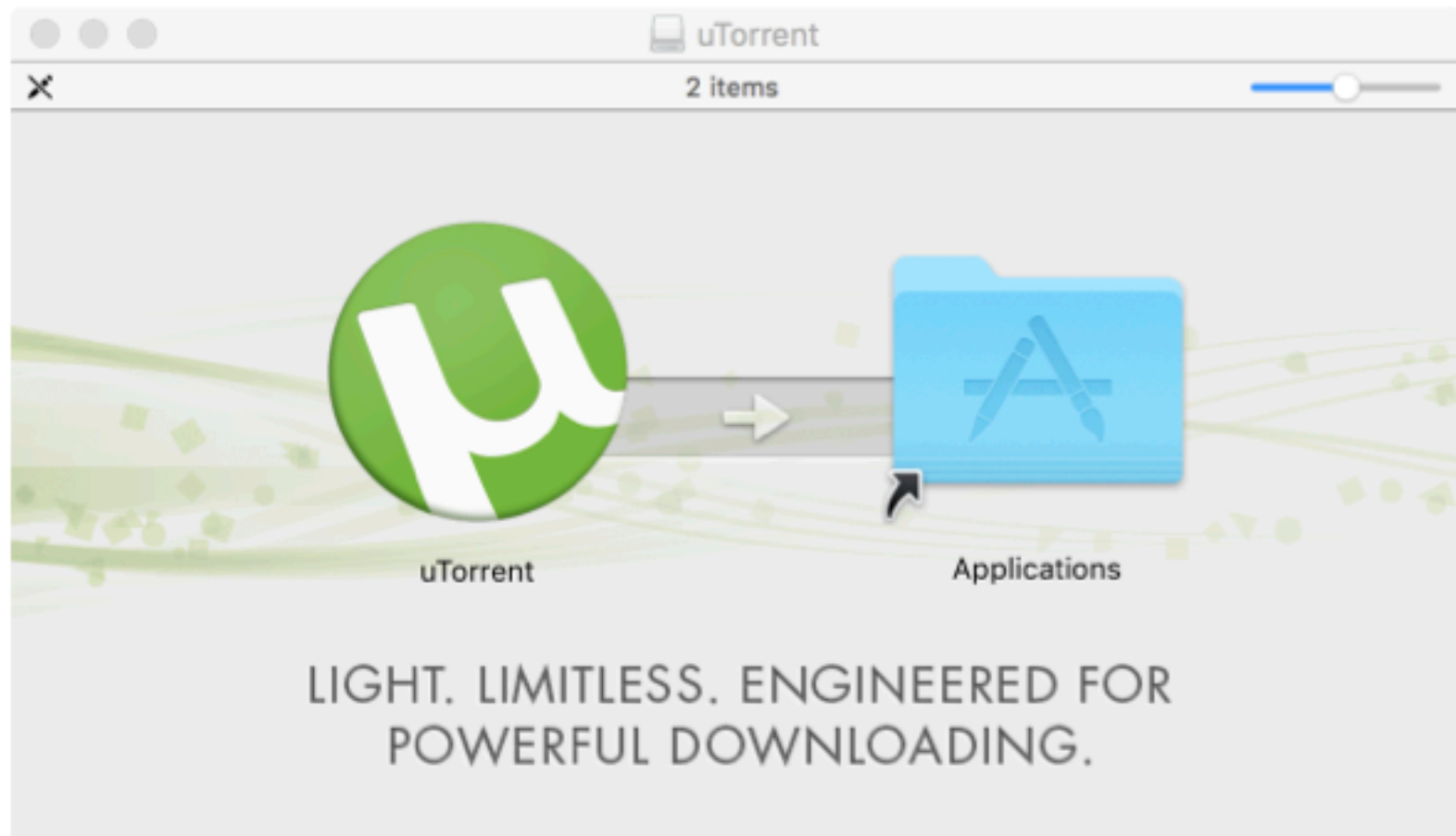
uTorrent (Mac) - Free download

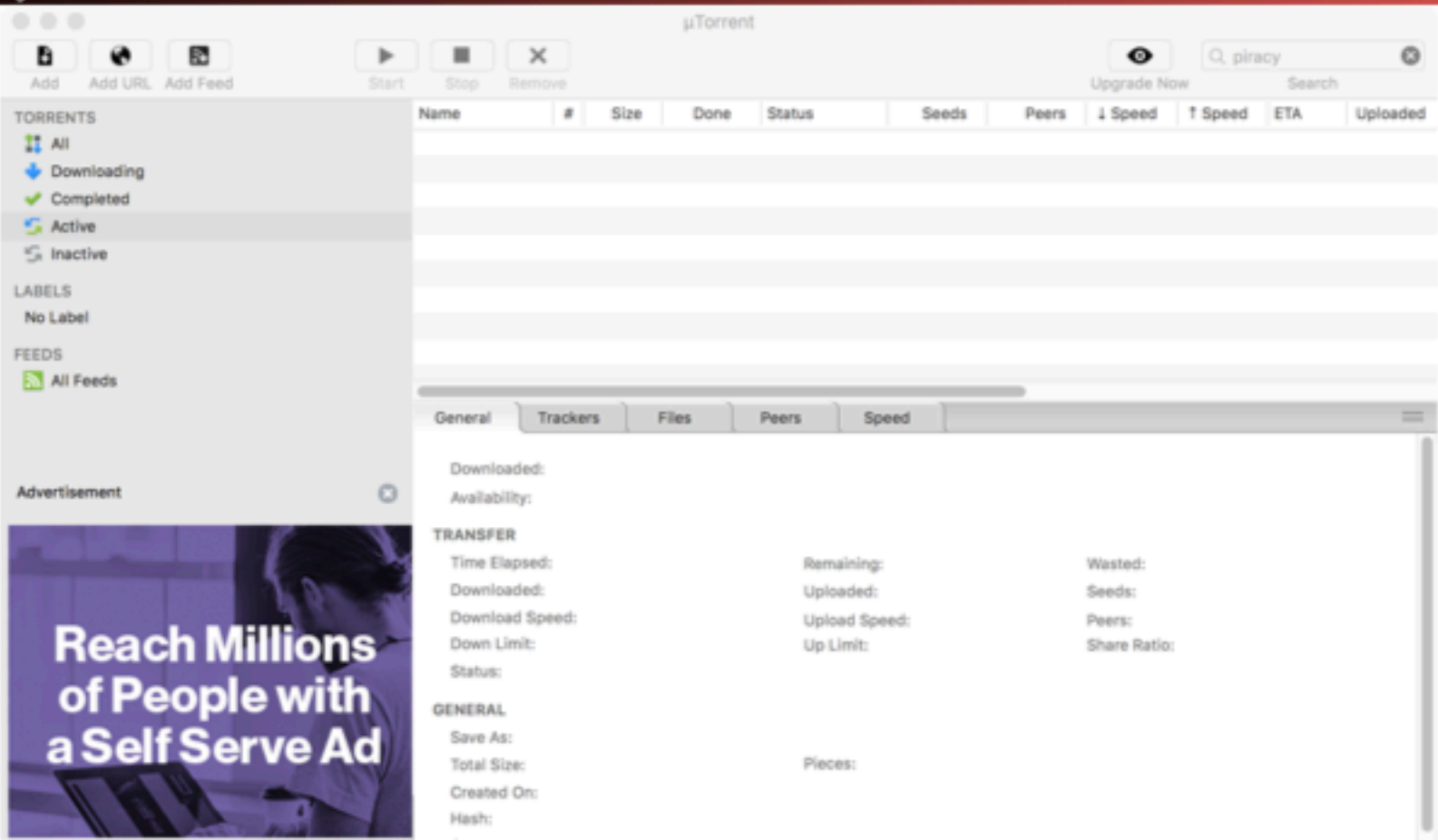
<https://utorrent.en.softonic.com/mac> ▾

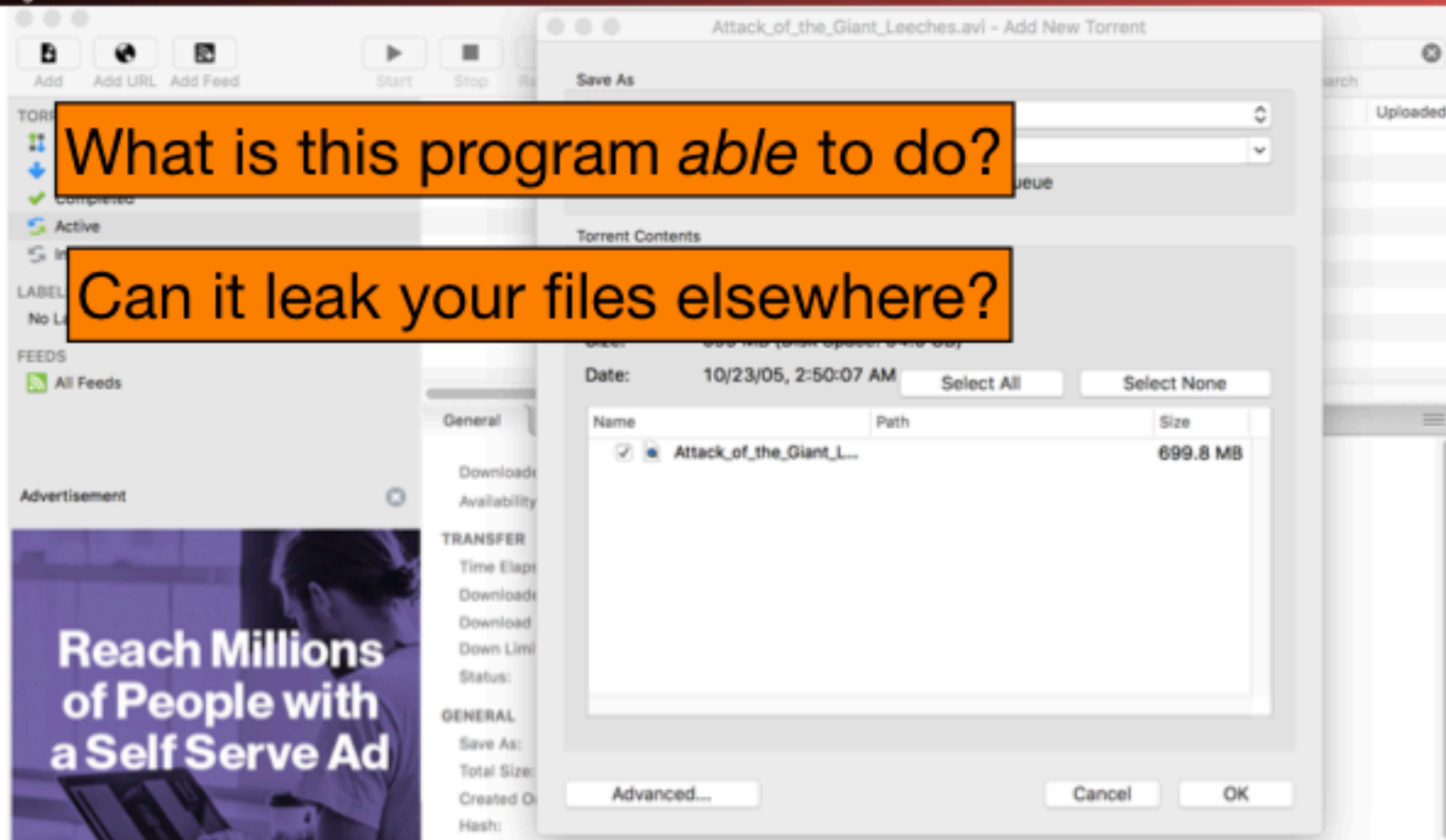
★ ★ ★ ☆ ☆ Rating: 3 - 550 votes - Free - Mac OS - Utilities/Tools

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 ...









What is this program *able* to do?

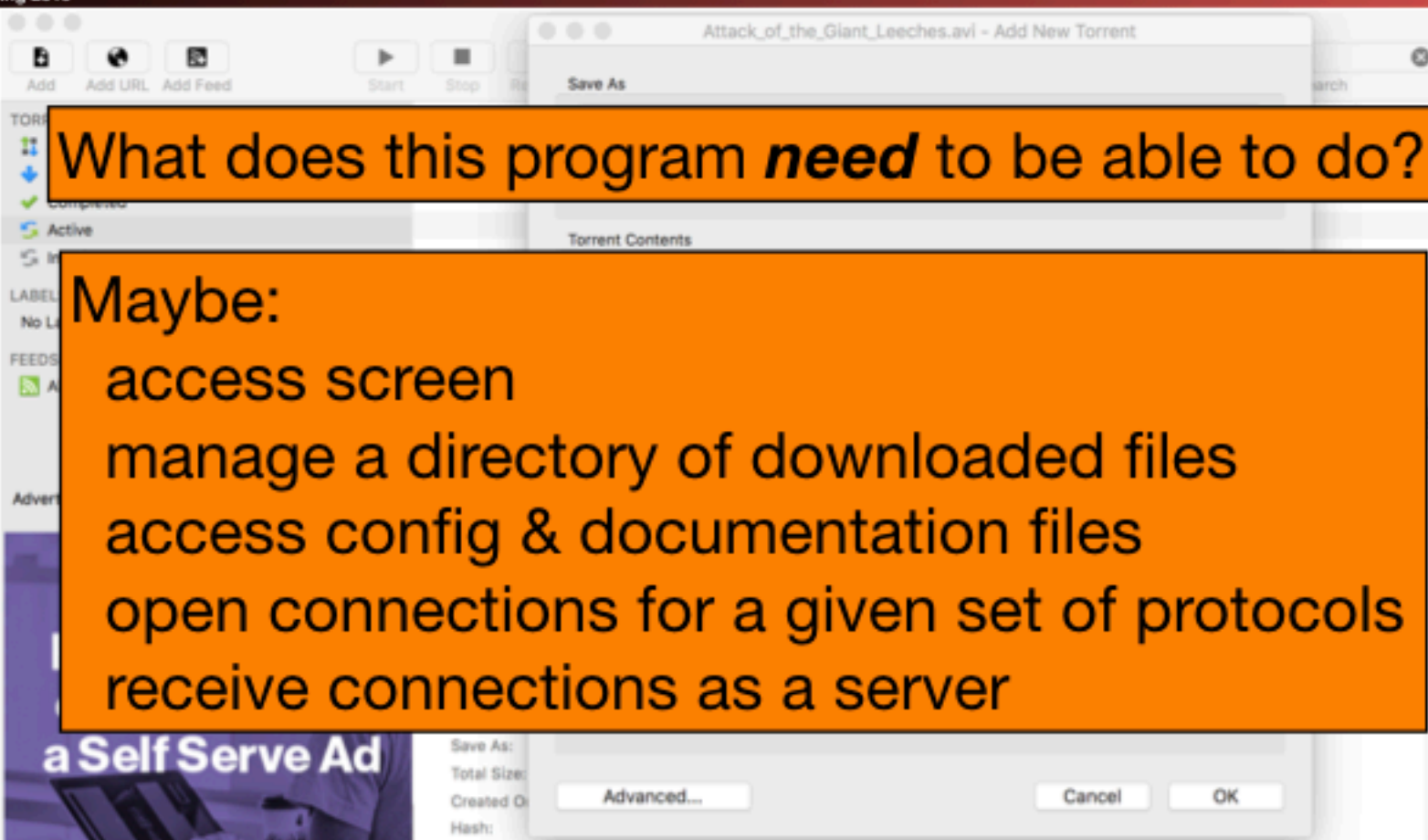
Can it leak your files elsewhere?

Can it delete all of your files?

Can it send spam?

Can it add a new executable to your search path?

YES. Why?

The image shows a screenshot of a torrent client interface. At the top, there are buttons for 'Add', 'Add URL', 'Add Feed', 'Start', and 'Stop'. Below these is a list of torrents, with one titled 'Attack_of_the_Giant_Leeches.avi'. A 'Save As' dialog box is open over the torrent, showing fields for 'Total Size', 'Created On', and 'Hash', along with 'Advanced...', 'Cancel', and 'OK' buttons. An orange text box is overlaid on the interface, containing the text: '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'.

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

Check for Understanding

- We've seen that laptop/desktop platforms grant applications a lot of privileges
- Quiz: Name a platform that does a better job of least privilege

So What Do You Think Here?

**Allow “Adult Cat Finder” to
access your location while
you use the app?**

We use your location to find nearby
adorable cats.

Don't Allow

Allow

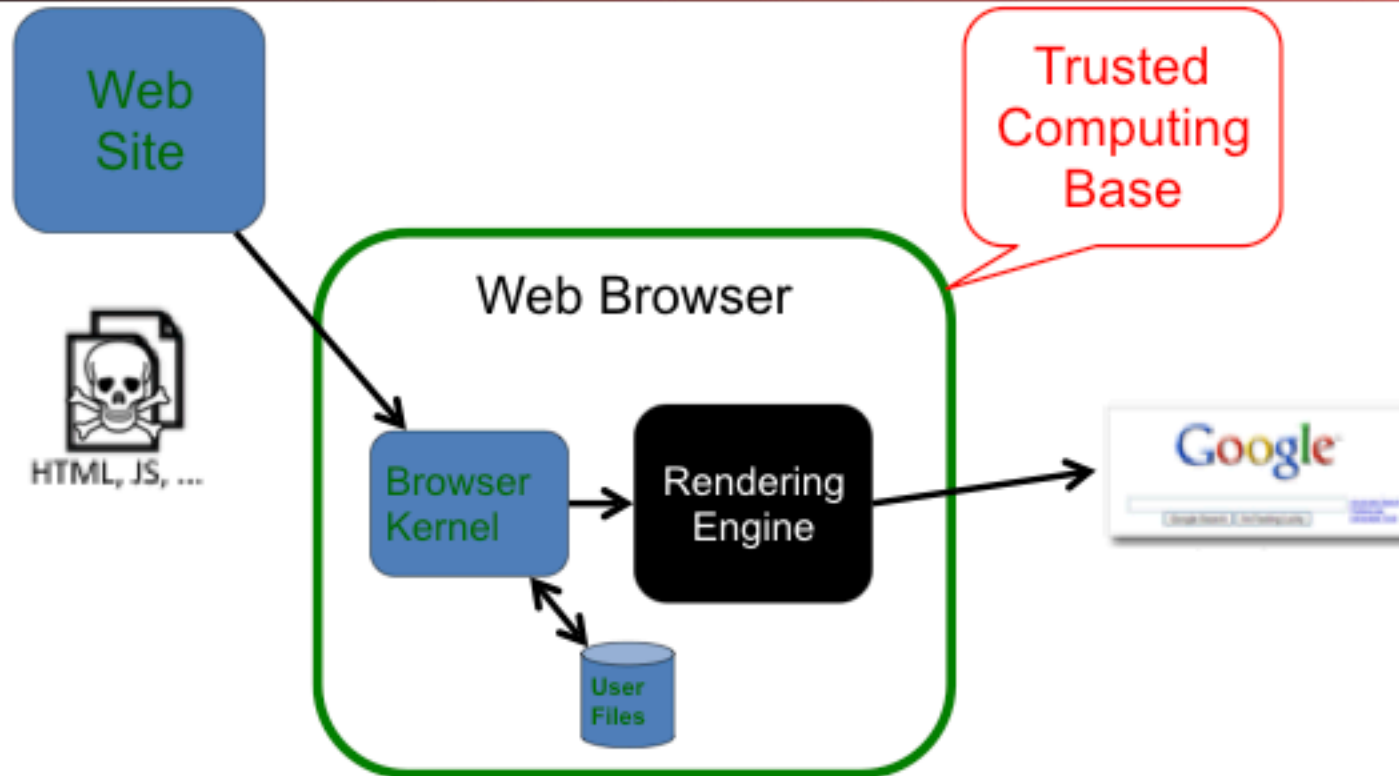
Thinking About Least Privilege

- 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...

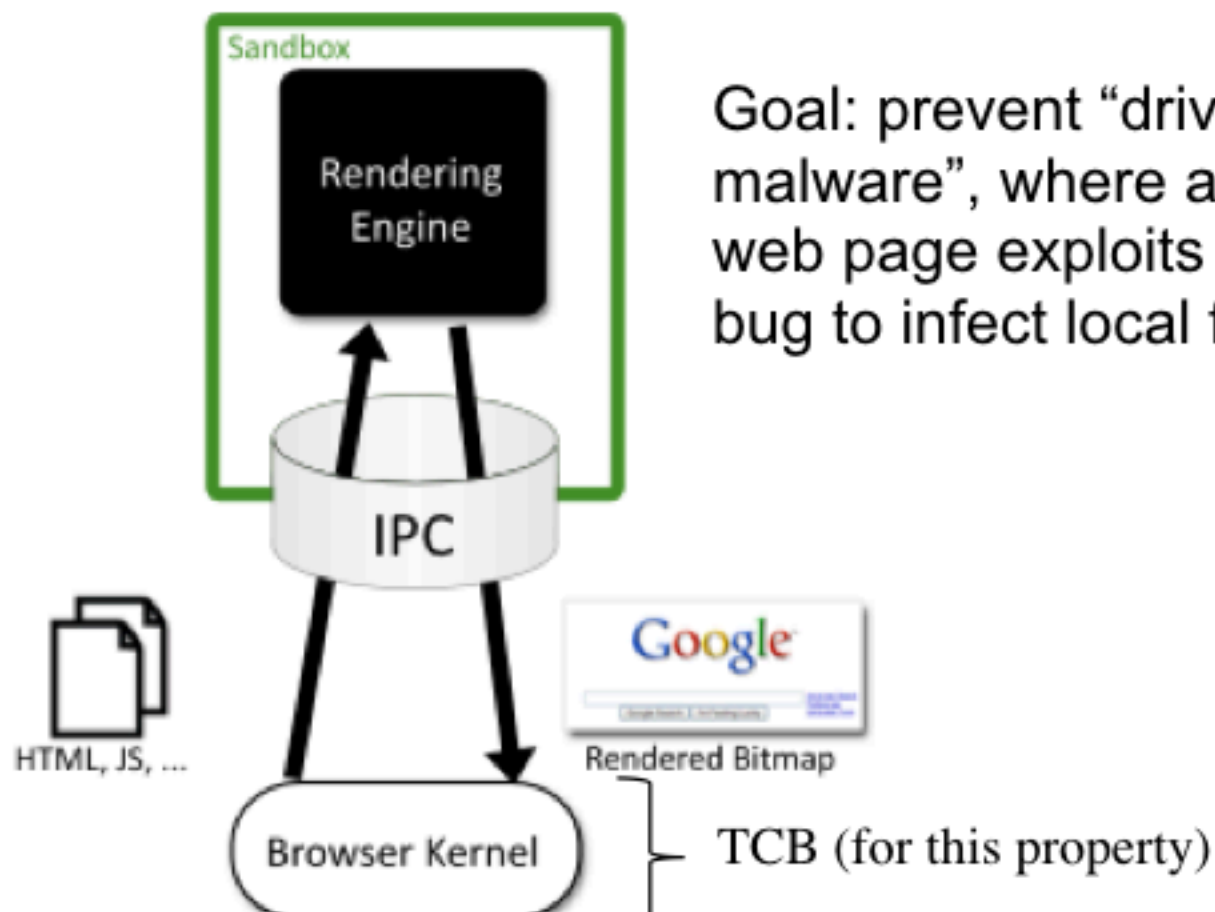
- The operating system **process** provide the following "guarentees" (you hope)
 - Isolation: A process can not access (read or write) the memory of any other process
 - Permissions: A process can only change files etc if it has permission to
 - This **usually** means "Anything that the user can do" 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"

Web browser



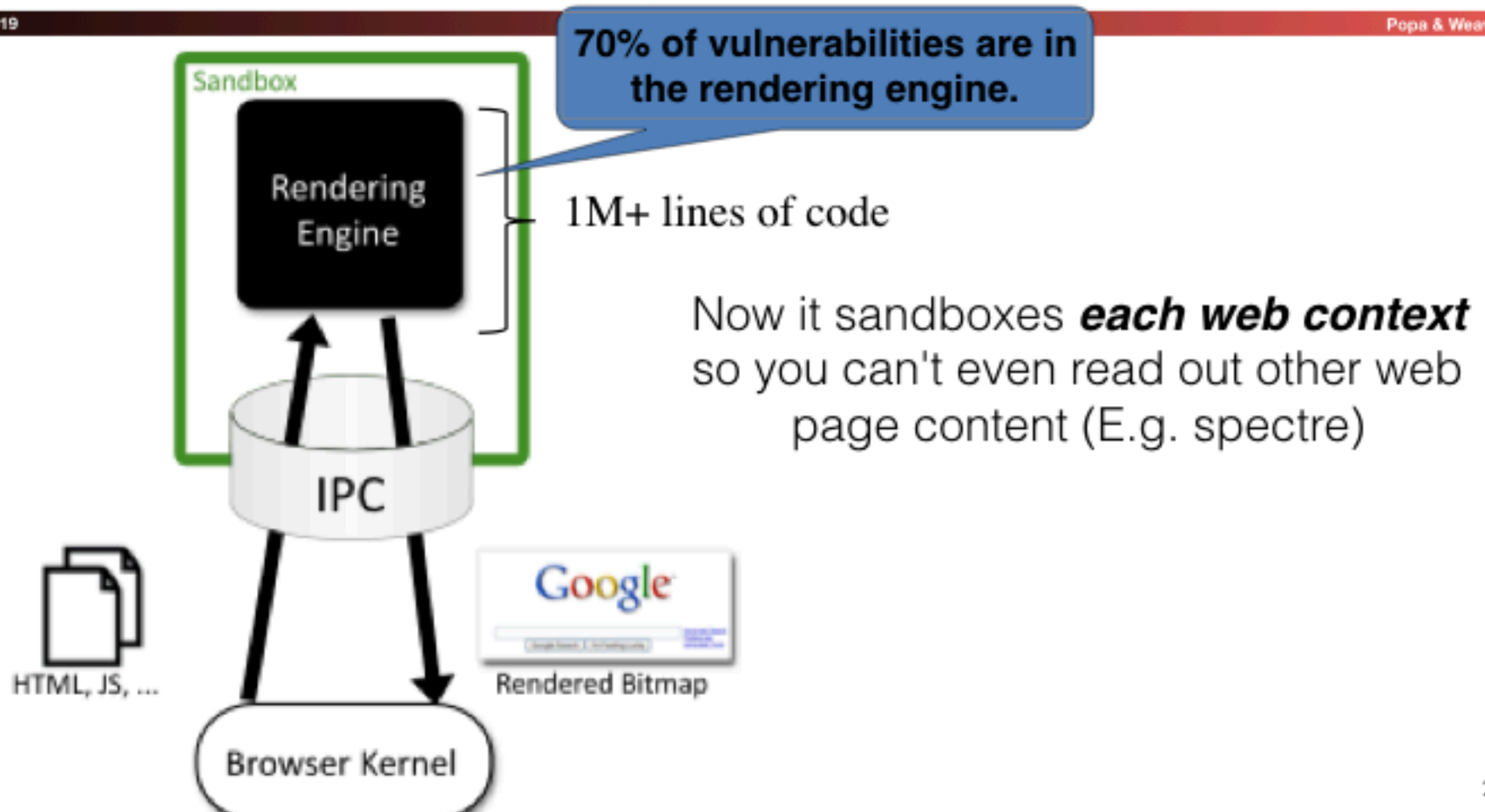
“Drive-by malware”: malicious web page exploits browser bug to infect local files

The Chrome browser



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

The Chrome browser



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 sound
- 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...



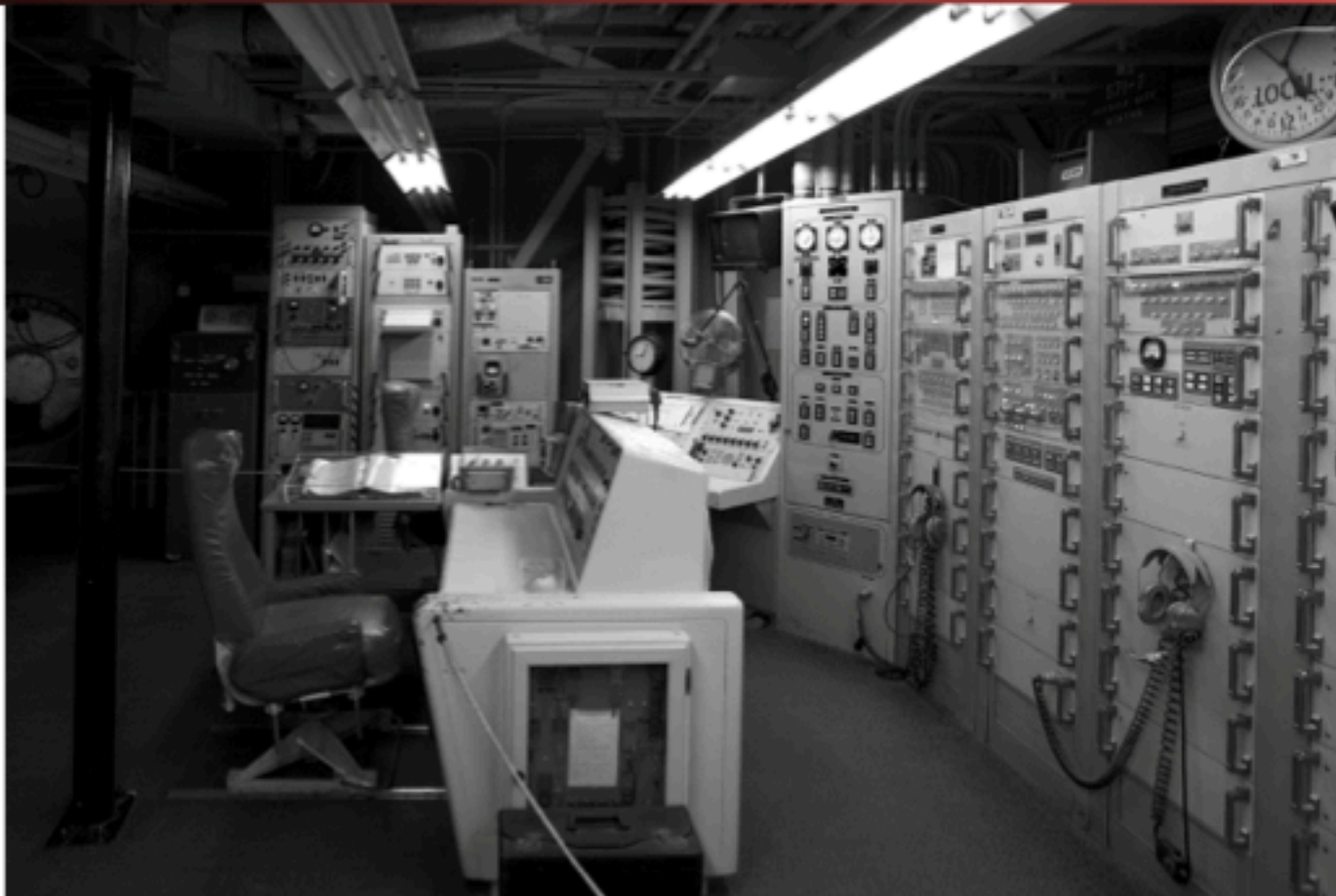
```
public void buyItem(Account buyer, Item item) {  
    if (item.cost > buyer.balance)  
        return; /* they can't afford it */  
  
    buyer.possessions.put(item); /* provide item */  
  
    buyer.possessionsUpdated(); /* freshen screen */  
  
    buyer.balance -= item.cost; /* deduct cost */  
  
    buyer.balanceUpdated(); /* freshen screen */  
}
```

What if an **uncaught exception** happens *here*?

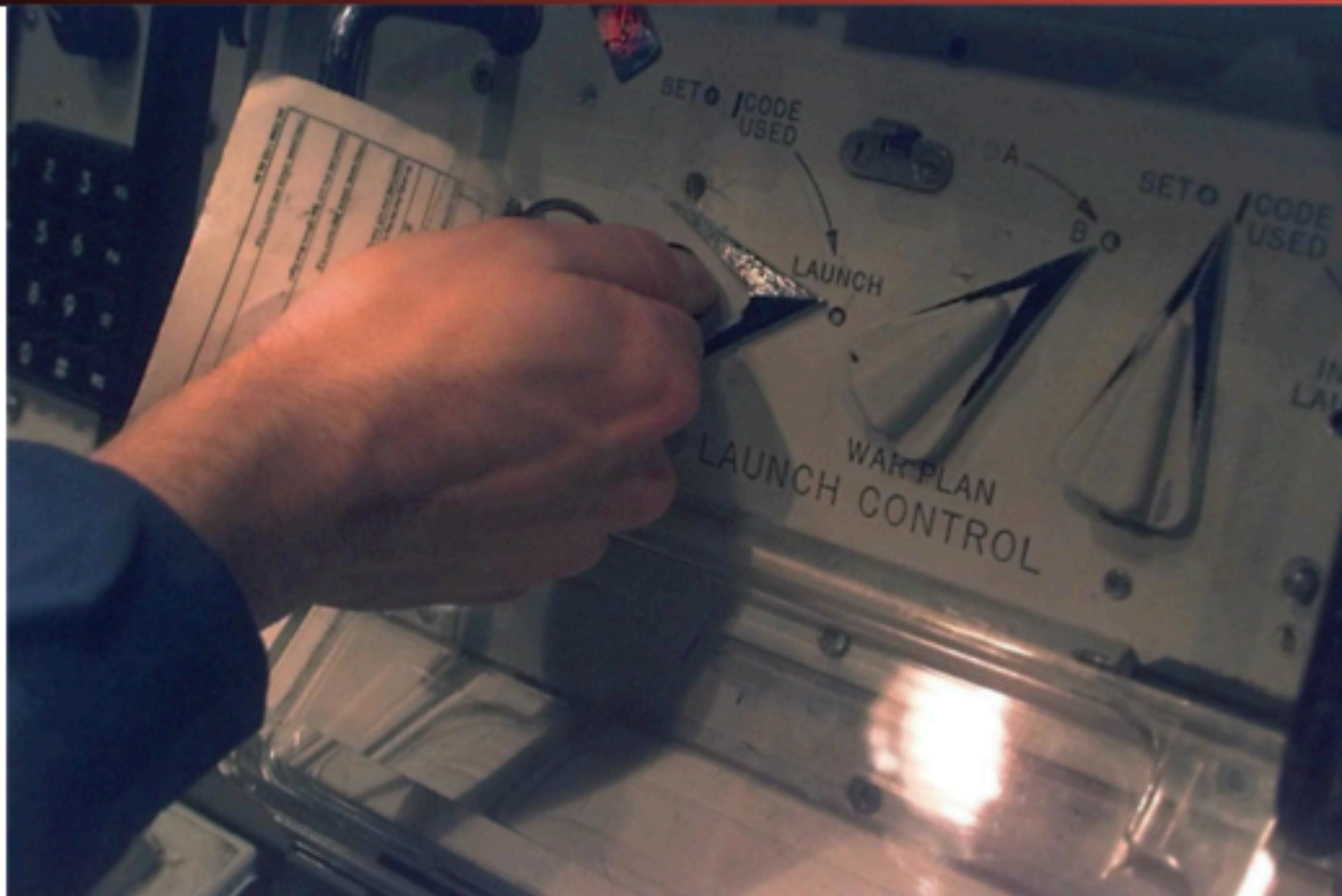
Welcome to a Nuclear Bunker

Computer Science 161 Spring 2019

Papa & Weaver



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

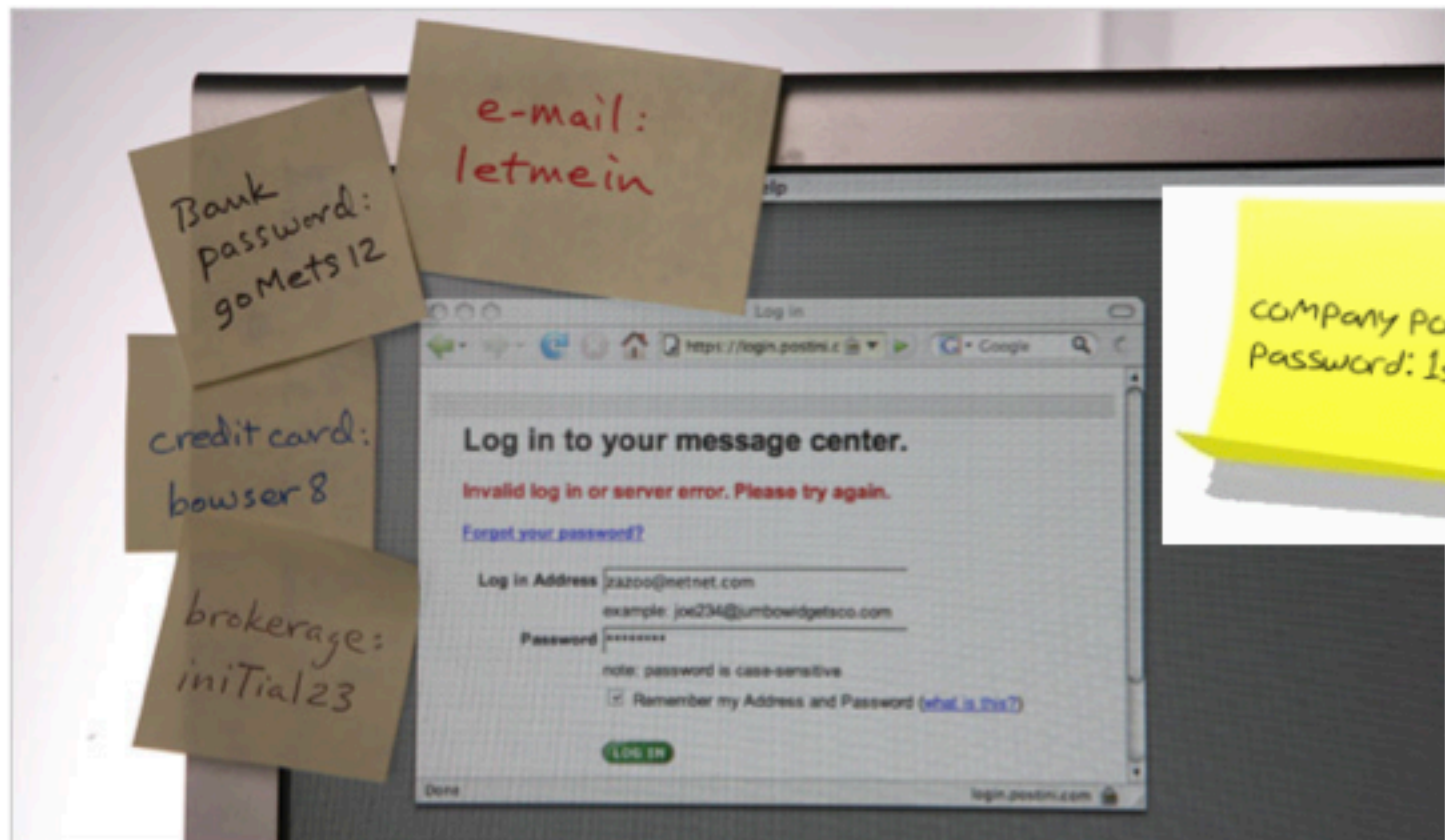


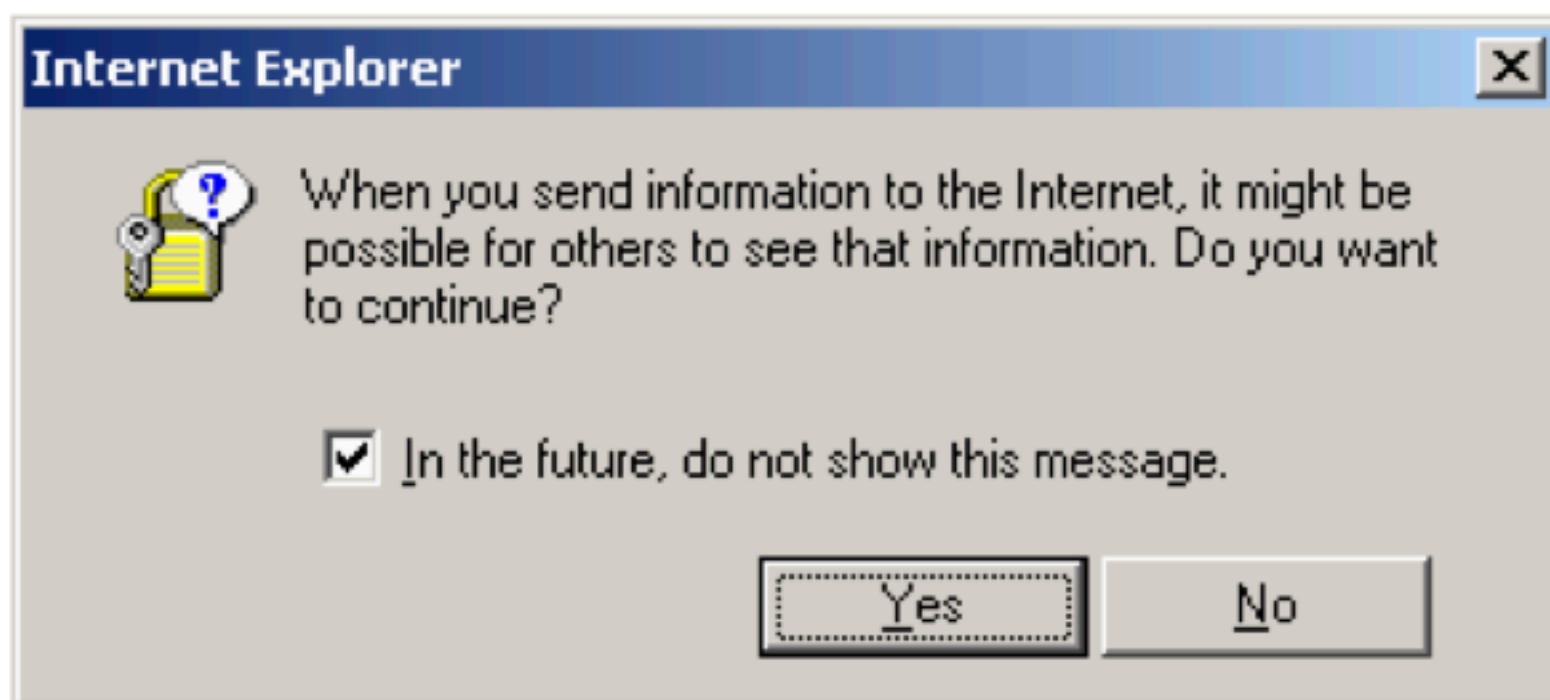
Summary:

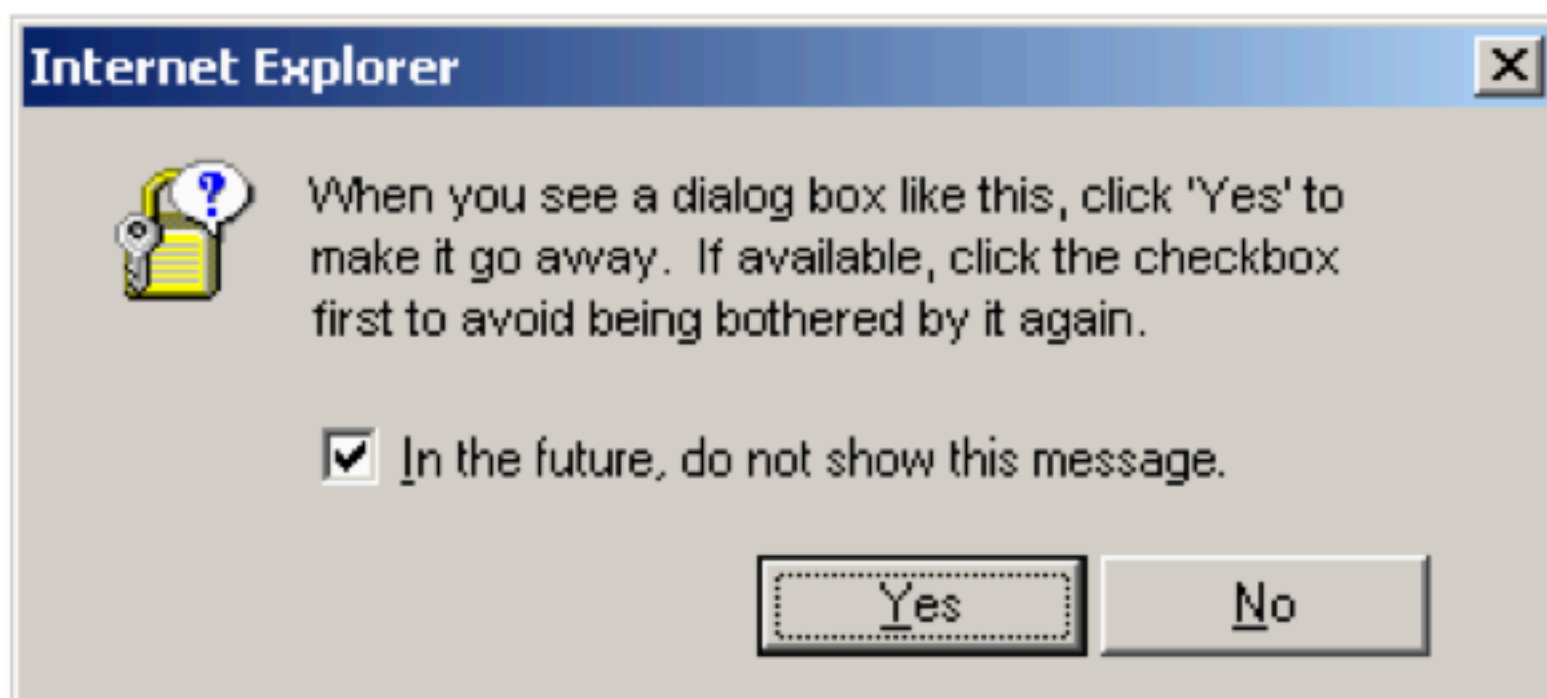
Notions Regarding Managing Privilege

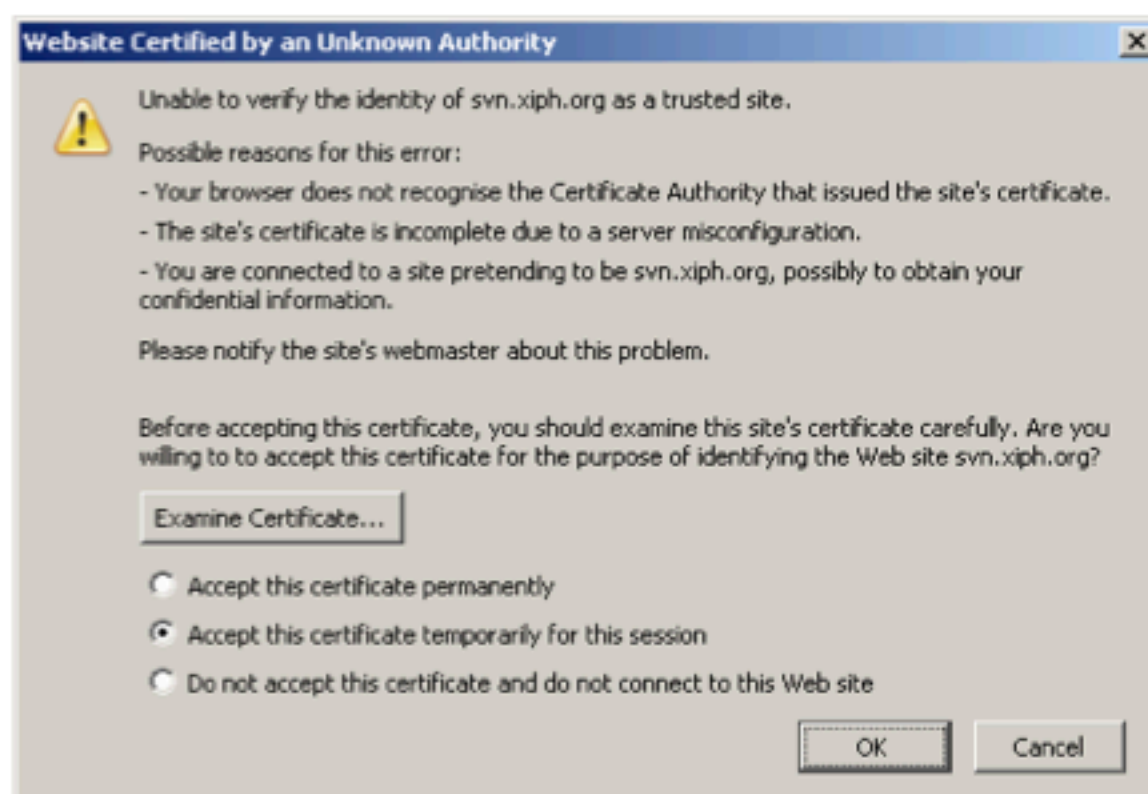
- **Least privilege**
 - The notion of avoiding having unnecessary privileges
- **Privilege separation**
 - A way to achieve least privilege by isolating access to privileges to a small Trusted Computing Base (TCB)
- **Separation of responsibility**
 - If you need to have a privilege, consider requiring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

Impact of a Password Policy











Security Keys and Human Factors

- This is a security key for storing key material for an encrypted military phone



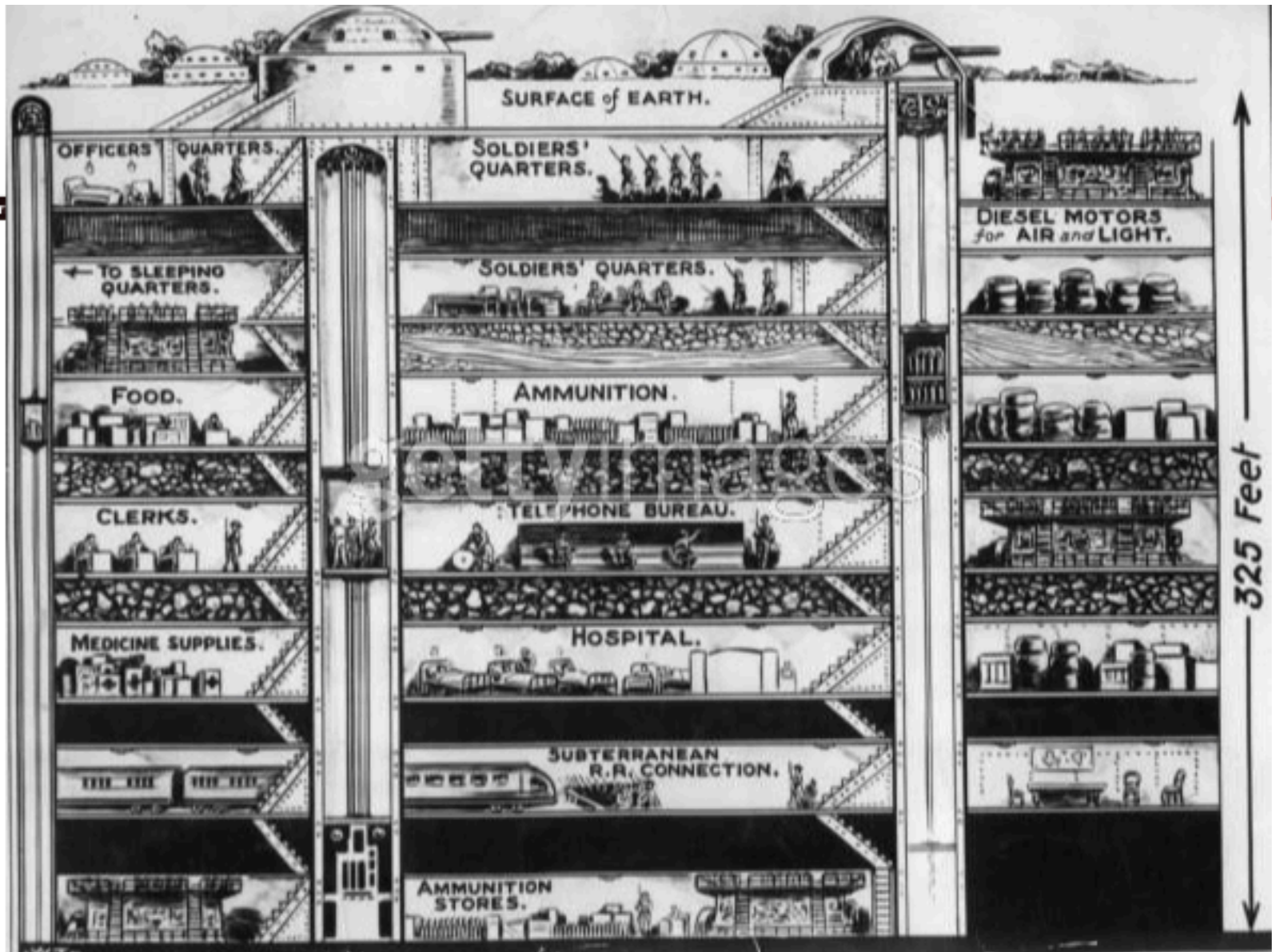
Summary:

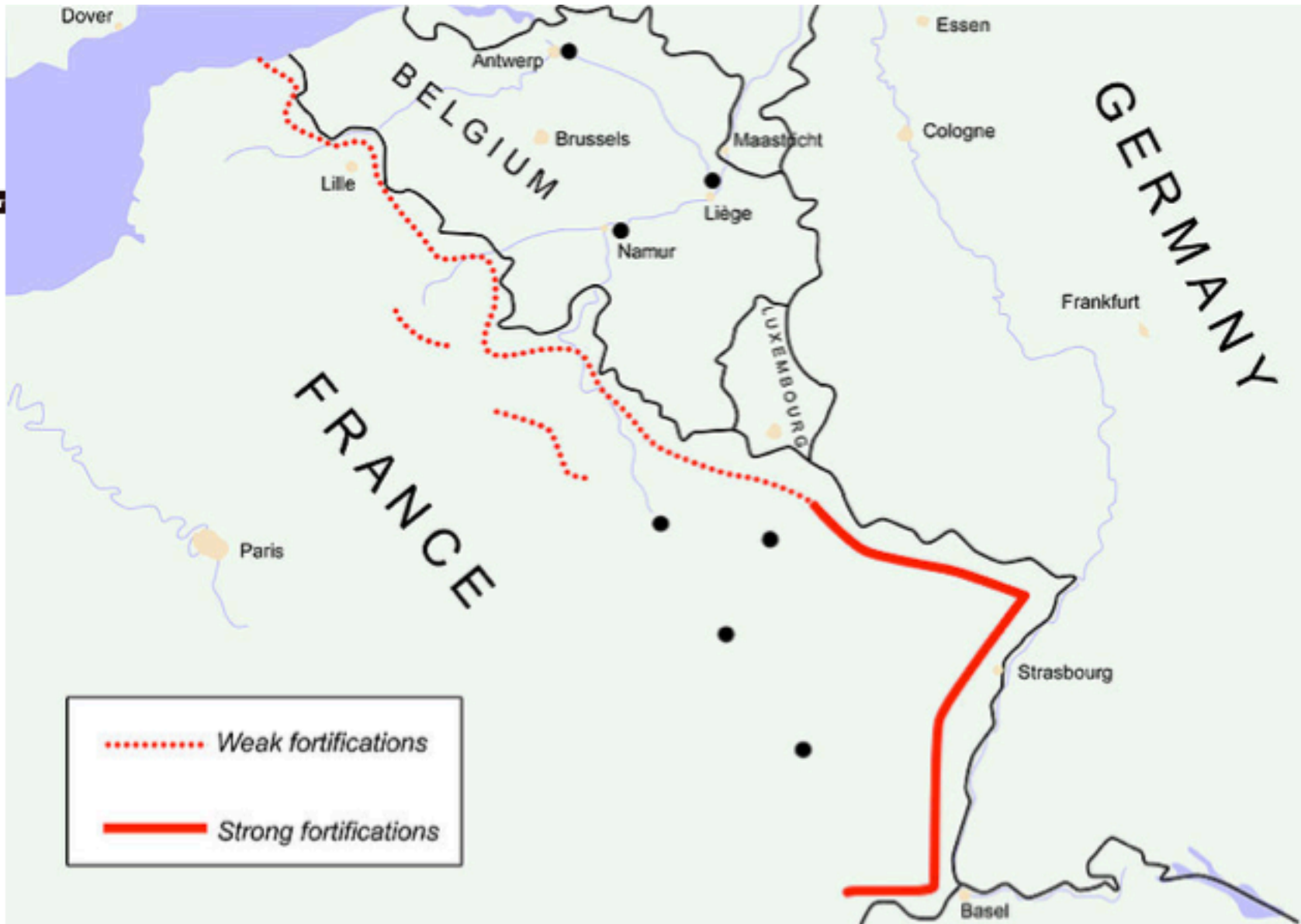
Dealing with Users

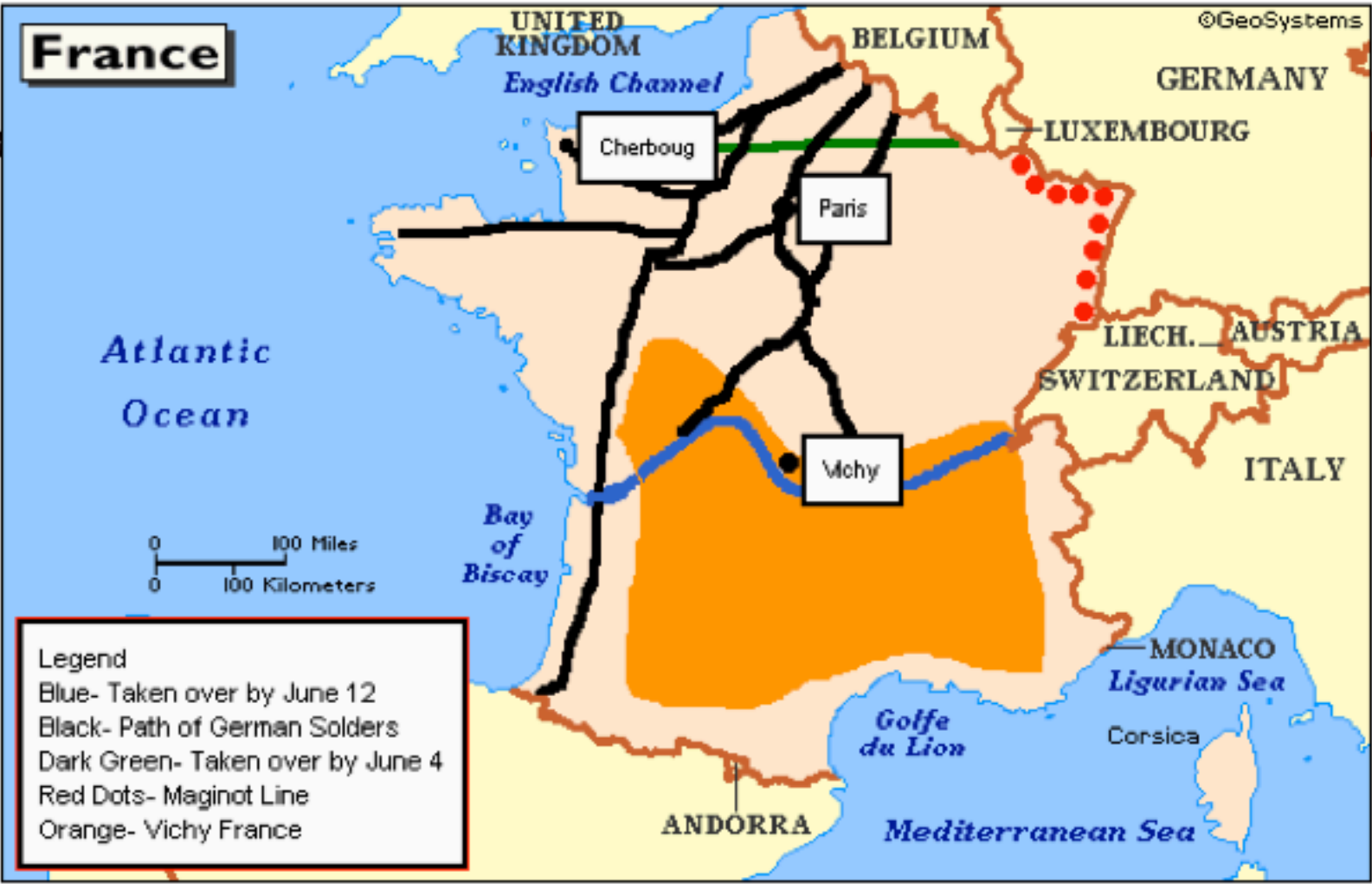
- Psychological acceptability
 - Will users abide a security mechanism, or decide to subvert it?

- 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









“Only as secure as the weakest link.”

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













“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













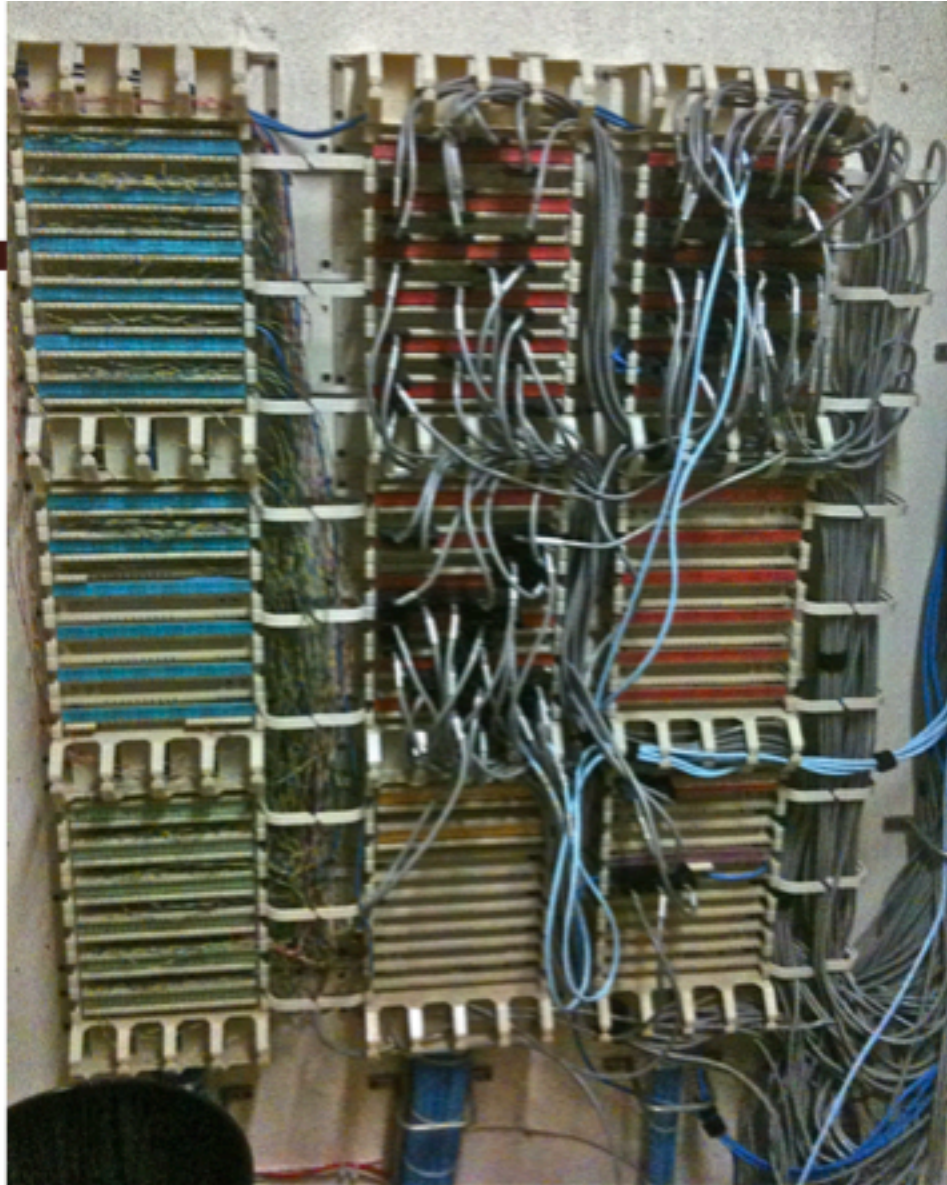
www.widelec.org

“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?



“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
- 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