Security Principles (continued)

Adapted from CS 161 Spring 2022 - Lecture 2

Next: Security Principles (continued)

- Security principles
  - Know your threat model
  - Consider human factors
  - Security is economics
  - Detect if you can’t prevent
  - Defense in depth
  - Least privilege
  - Separation of responsibility
  - Ensure complete mediation
  - Don’t rely on security through obscurity
  - Use fail-safe defaults
  - Design in security from the start

Know Your Threat Model

- **Threat model**: A model of who your attacker is and what resources they have
- It all comes down to people: The attackers
  - No attackers = No problem!
  - One of the best ways to counter an attacker is to attack their reasons
- Why do people attack systems?

Money

Advertisers

Politics

Anonymous

NSA

Fun

To watch the world burn

Threat Model: Common Assumptions for Attackers

- Assume the attacker…
  - Can interact with systems without notice
  - Knows general information about systems (operating systems, vulnerabilities in software, usually patterns of activity, etc.)
  - Can get lucky
    - If an attack only succeeds 1/1,000,000 times, the attacker will try 1,000,000 times!
  - May coordinate complex attacks across different systems
  - Has the resources required to mount the attack
    - This can be tricky depending on who your threat model is
  - Can and will obtain privileges if possible

Trusted Computing Base

- **Trusted computing base** (**TCB**): The components of a system that security relies upon
- Question: What would you want from a TCB?
- Properties of the TCB:
  - Correctness
  - Completeness (can't be bypassed)
  - Security (can't be tampered with)
- Generally made to be as small as possible
  - A smaller, simpler TCB is easier to write and audit.
  - **KISS principle**: Keep It Simple, Stupid

Security is Economics

Textbook Chapter 1.3

Security is Economics

- Cost/benefit analyses often appear in security
  - The cost of your defense should be less than the cost of attacks happening
  - More security (usually) costs more
  - If the attack costs more than the reward, the attacker probably won't do it
- Example: You don't put a $10 lock on a $1 rock…
  - … unless a $1 rock can be used to attack something even more valuable
- Example: You have a brand-new, undiscovered attack that will work on anybody's computer. You wouldn't expose it on a random civilian.
  - iPhone security vulnerabilities are often sold for ~$1M on the market

Physical Safes

- We want our safes to stop people from breaking in, so let's measure them by how long it takes an expert to break into one:

TL-15 ($3,000)
15 minutes with common tools

TL-30 ($4,500)
30 minutes with common tools

TRTL-30 ($10,000)
30 minutes with common tools and a cutting torch

TXTL-60 (>$50,000)
60 minutes with common tools, a cutting torch, and up to 4 oz of explosives

Takeaway: Security is economics

Burglar Alarms

- Security companies are supposed to detect home break-ins
  - Problem: Too many false alarms. Many alarms go unanswered
  - Why is it useful to place a sign?
  - Placing a sign helps deter burglars from entering at risk of being caught…
    - … even if you don't have an alarm installed!
  - An attacker might prefer the neighbor without a sign

Detect If You Can’t Prevent

Textbook Chapter 1.4

Detect if You Can’t Prevent

- **Deterrence**: Stop the attack before it happens by making the attacker prefer to do something else
- **Prevention**: Stop the attack before it happens by making the attack fail
- **Detection**: Learn that there was an attack (after it happened)
  - If you can’t stop the attack from happening, you should at least be able to know that the attack has happened.
- **Response**: Do something about the attack (after it happened)
  - Once you know the attack happened, you should respond
  - Detection without response is pointless!

Response: Mitigation and Recovery

- Assume that bad things will happen! You should plan security in way that lets you to get back to a working state.
- Example: Earthquakes
  - Have resources for 1 week of staying put
  - Have resources to travel 50 miles from my current location
- Example: Ransomware
  - Ransomware: An attacker steals your data and demands payment in exchange for recovering your data
  - Keep offsite backups!
  - If your computer and house catch on fire, it should be no big deal.

Detection but no Response

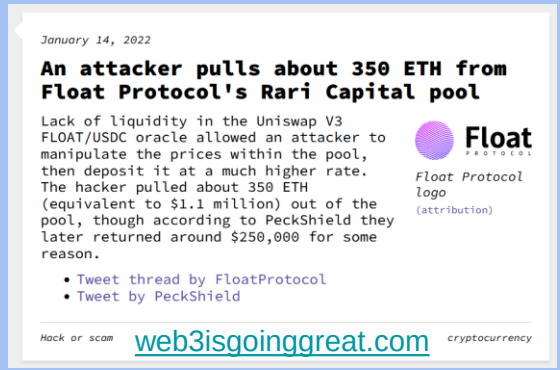
- Cryptocurrency transactions are irreversible. If you are hacked, you can never recover your Bitcoins.
  - $68M stolen from NiceHash exchange in December 2017
  - Four multi-million-dollar attacks on Ethereum in July 2018
  - Coinbase: One *detected* theft per day
  - Keep track of the fun at web3isgoinggreat.com
- **Takeaway:** Prevention is great, but depending only on prevention can be *brittle*: When prevention fails, the system fails catastrophically

Bloomberg

Link

Hacked Bitcoin Exchange Says Users May Share $68 Million Loss

Lulu Yilun Chen and Yuji Nakamura *August 5, 2016*



Defense in Depth

Textbook Chapter 1.5

Defense in Depth

- Multiple types of defenses should be layered together
- An attacker should have to breach all defenses to successfully attack a system
  - Ideally the strength of the defenses compounds somehow
- However, remember: security is economics
  - Defenses are not free.
  - Diminishing returns: Defenses are often less than the sum of their parts
    - 2 walls is much better than 1 wall
    - 101 walls is not much better than 100 walls

The Theodosian Walls of Constantinople

- The ancient capital of the Byzantine empire had a wall…
  - Well, they had a moat…
  - then a wall…
  - then a depression…
  - … and then an even bigger wall
- It also had towers to rain fire and arrows upon the enemy…
- Lasted until the Ottoman empire came along with cannons in 1453…
  - And now it's Istanbul not Constantinople
- **Takeaway**: Defense in depth: An attacker needed to breach all the walls
- **Takeaway**: Changing attacker technology changes defensive requirements

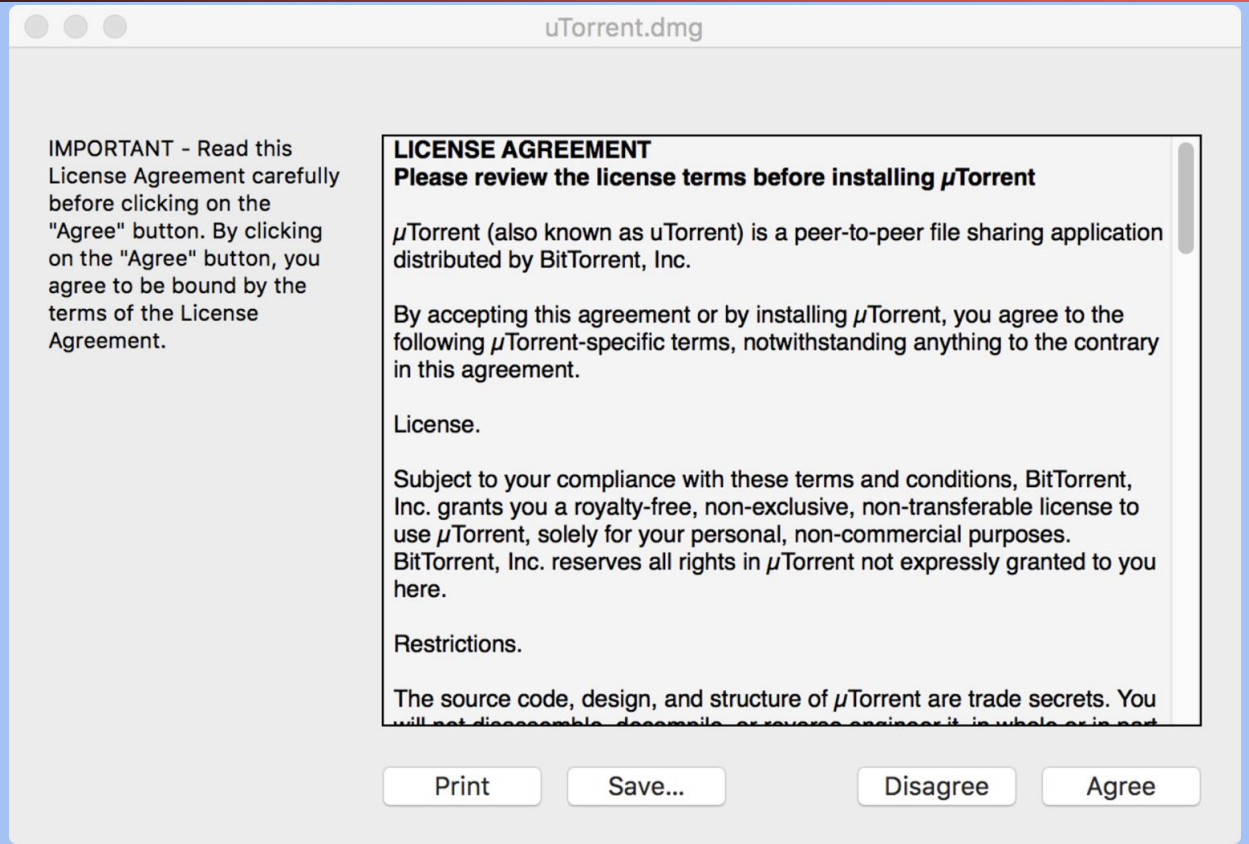
Least Privilege

Textbook Chapter 1.6

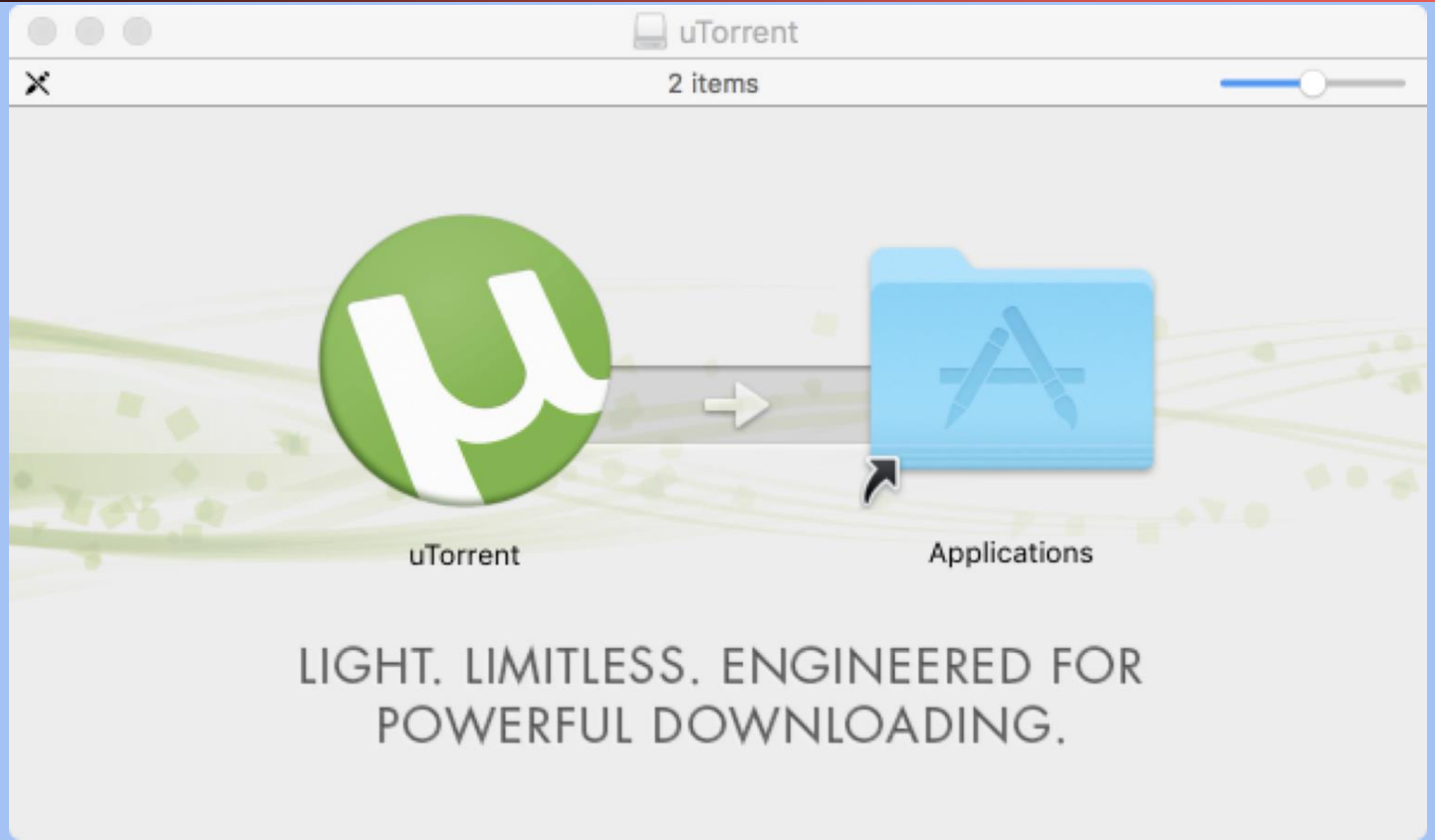
Least Privilege

- Consider the minimum permissions an entity or program *needs* to be able to do its job correctly, and grant only those permissions
  - If you grant unnecessary permissions, a malicious or hacked program could use those permissions against you

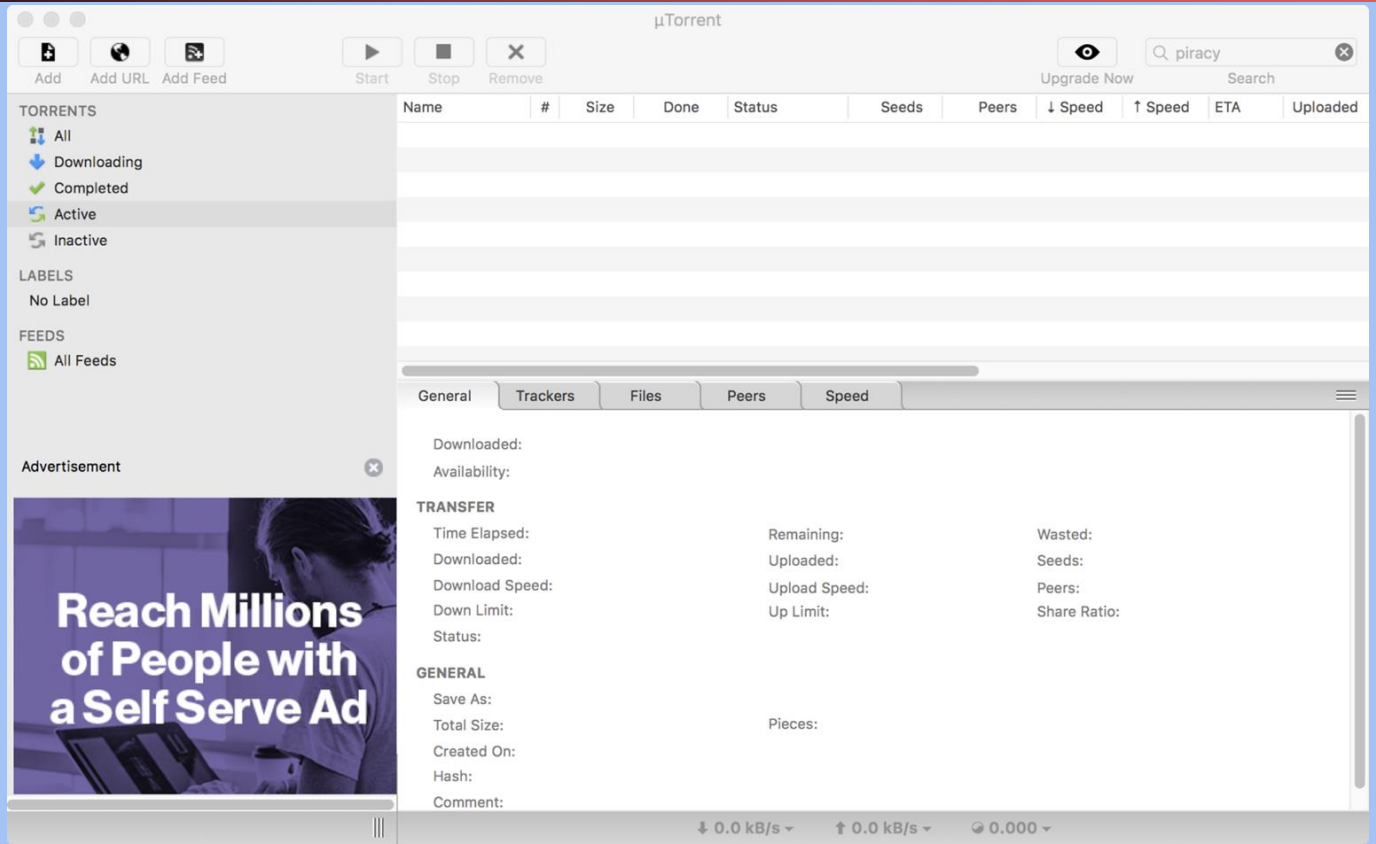
uTorrent


uTorrent.dmg
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
Print
Save...
Disagree
Agree

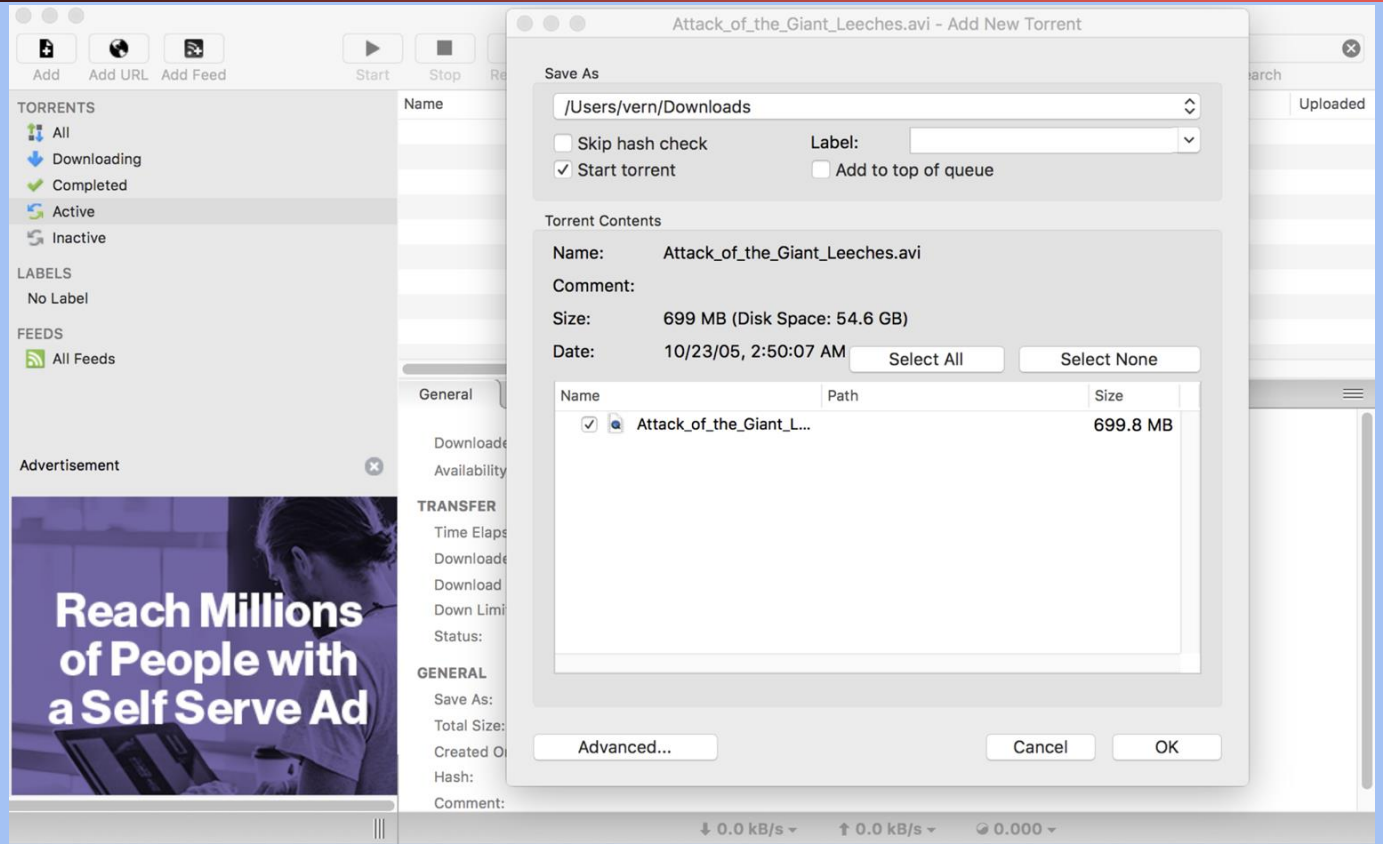
uTorrent


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

uTorrent

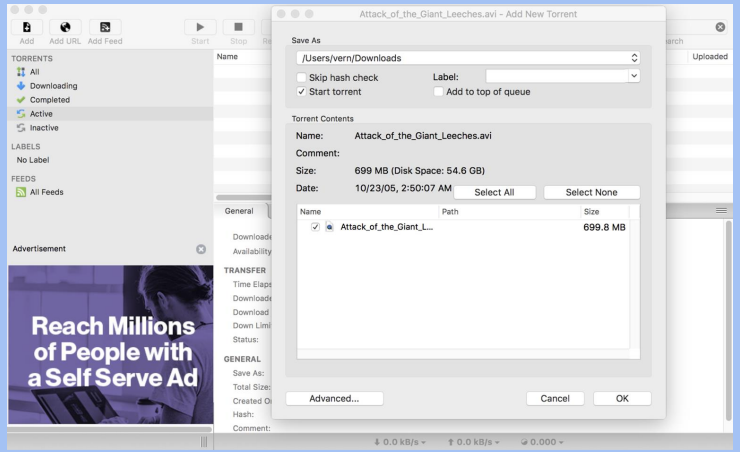

µTorrent
Add
Add URL
Add Feed
Start
Stop
Remove
Upgrade Now
piracy
Search
TORRENTS
All
Downloading
Completed
Active
Inactive
LABELS
No Label
FEEDS
All Feeds
Name
#
Size
Done
Status
Seeds
Peers
↓ Speed
↑ Speed
ETA
Uploaded
General
Trackers
Files
Peers
Speed
Downloaded:
Availability:
TRANSFER
Time Elapsed:
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:
Comment:
Advertisement
Reach Millions of People with a Self Serve Ad
0.0 kB/s
0.0 kB/s
0.000

uTorrent


Add
Add URL
Add Feed
Start
Stop
TORRENTS
All
Downloading
Completed
Active
Inactive
LABELS
No Label
FEEDS
All Feeds
Advertisement
Reach Millions of People with a Self Serve Ad
Name
Uploaded
General
TRANSFER
GENERAL
Save As:
Total Size:
Hash:
Comment:
Attack_of_the_Giant_Leeches.avi - Add New Torrent
Save As
/Users/vern/Downloads
Skip hash check
Start torrent
Label:
Add to top of queue
Torrent Contents
Name: Attack_of_the_Giant_Leeches.avi
Comment:
Size: 699 MB (Disk Space: 54.6 GB)
Date: 10/23/05, 2:50:07 AM
Select All
Select None
Name
Path
Size
Attack_of_the_Giant_L...
699.8 MB
Advanced...
Cancel
OK
0.0 kB/s
0.0 kB/s
0.000

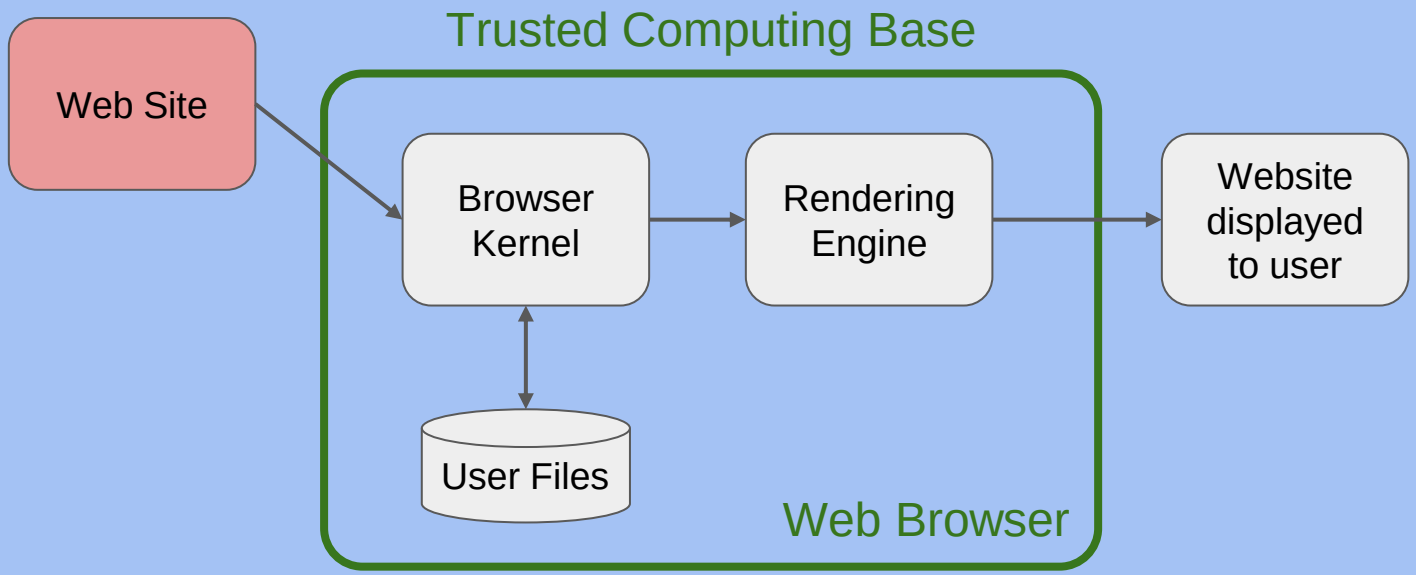
uTorrent

- What was this program able to do?
  - Leak your files
  - Delete your files
  - Send spam
  - Run another malicious program
- What does this program need to be able to do?
  - Access the screen
  - Manage some files (but not all files)
  - Make some Internet connections (but not all Internet connections)
- **Takeaway**: Least privilege



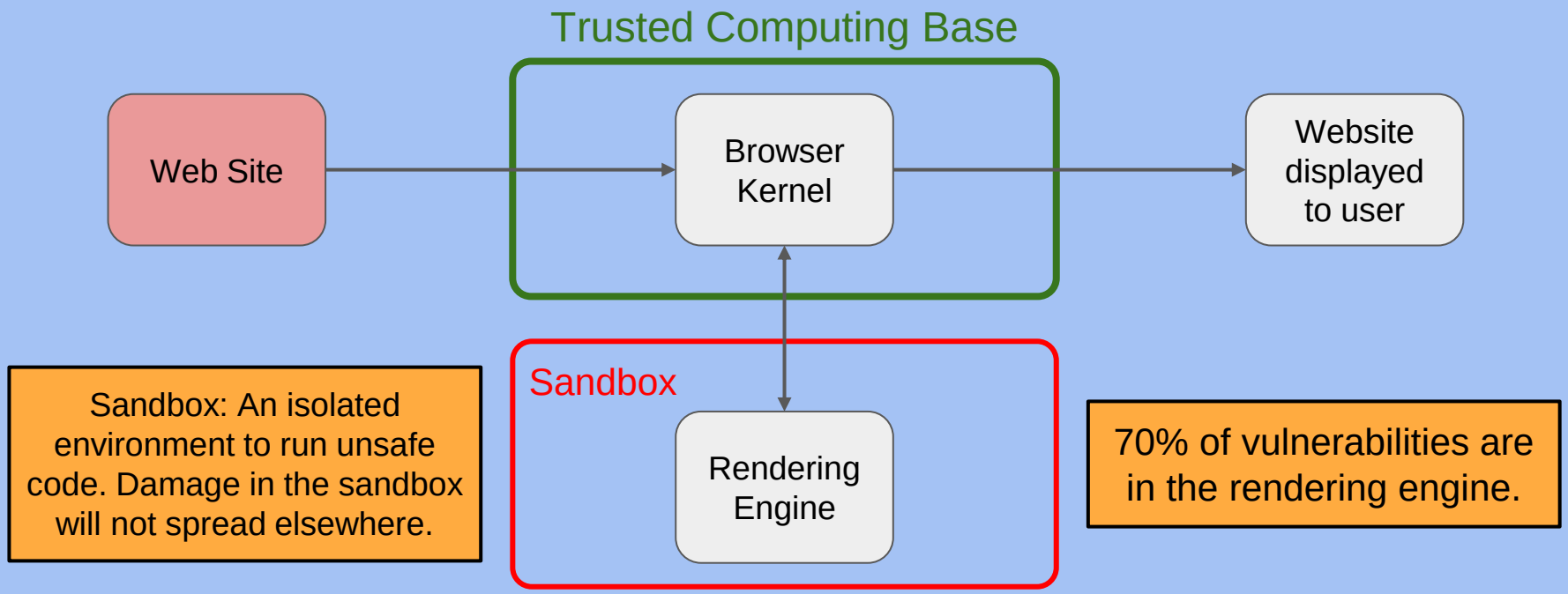
Browser Design with Poor Least Privilege

The contents of the web site come from an untrusted source (possibly malicious)



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

Google Chrome Design: Apply Least Privilege



Prevent "drive-by malware," where a malicious webpage exploits a browser bug to infect local files

Enabling Least Privilege: Access Control

- How to control who has access to particular data
  - Access needs to be *necessary* to accomplish the tasks
  - Least privilege: Excessive access can be a problem
- Example: Access control for DSP data in CS 161
  - Separate Google shared Drive from the main Drive
    - This allows independent access control *defaults*
  - *Limit* access to that Drive to those who have a need to know
    - Nick, head TAs, DSP TAs, course manager

Access Control for Systems: The Operating System

- The OS is the TCB of most modern systems, and it provides access controls to restrict the privileges of user programs
- The OS provides the following “guarantees”:
  - **Isolation**: A process can’t read or write the memory of any other process
  - **Permissions**: A process can only change files, interact with devices, etc. if it has permission to

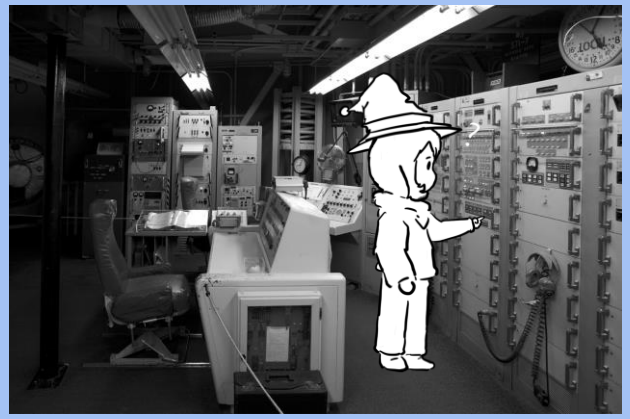
Separation of Responsibility

Textbook Chapter 1.7

Separation of Responsibility

- Also known as distributed trust
- If you need to have a privilege, consider requiring multiple parties to work together (collude) to exercise it
  - It’s much more likely for a single party to be malicious than for all multiple parties to be malicious and collude with one another

Welcome to a Nuclear Bunker



Welcome to a Movie Theater

Ensure Complete Mediation

Textbook Chapter 1.8 & 1.13

Security Principle: Ensure Complete Mediation

- Ensure that every access point is monitored and protected
- **Reference monitor**: Single point through which all access must occur
  - Example: A network firewall, airport security, the doors to the dorms
- Desired properties of reference monitors:
  - Correctness
  - Completeness (can't be bypassed)
  - Security (can't be tampered with)
  - Should be part of the TCB

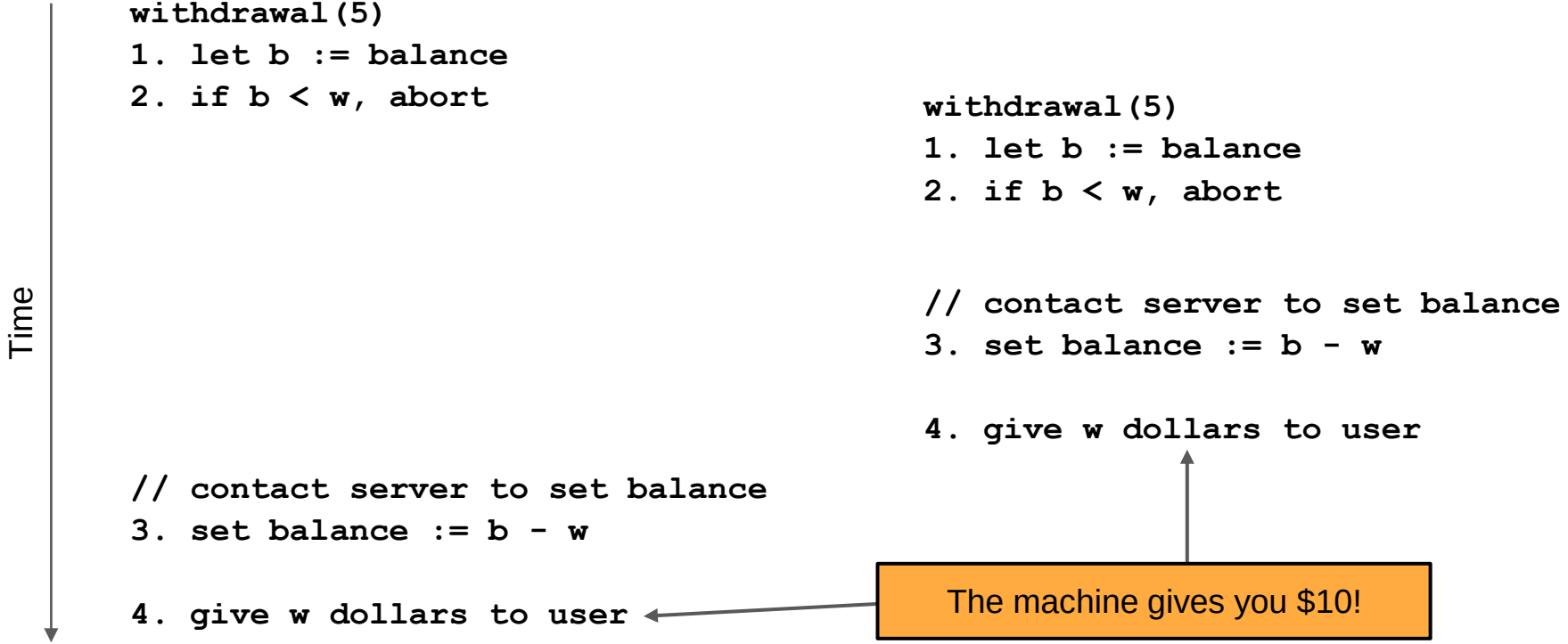
Time-of-Check to Time-of-Use

- A common failure of ensuring complete mediation involving race conditions
- Consider the following code:

```
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. give w dollars to user
```

Suppose you have $5 in your account. How can you trick this system into giving you more than $5?

Time-of-Check to Time-of-Use



Don’t Rely on Security Through Obscurity

Textbook Chapter 1.9

Don’t Rely on Security Through Obscurity

- Also known as **Shannon’s Maxim**
  - “The enemy knows the system”
- Also known as **Kerckhoff’s Principle**
  - “The only part of a cryptographic system unknown to the adversary is the cryptographic keys”

Highway Signs

Here's a highway sign.

Here's the hidden computer inside the sign.

Here's the control panel. Most signs use the default password, **`DOTS`**.

Highway Signs

Note/Takeaway: Do not **ever** do this. Yes, some former CS 161 students did it once.

Highway Signs

Takeaway: Don’t rely on security through obscurity

Don’t Rely on Security Through Obscurity

- Always assume that the attacker knows every detail about the system you are working with (algorithms, hardware, defenses, etc.)
  - Sometimes, obscurity *can help*
    - Example: We don’t tell you how we detect academic dishonesty
  - However, systems that *rely* on obscurity are brittle, since the attacker may find out!
- Don’t do security through obscurity!

Assume the attacker knows where the “secret” control panel is located, and knows the default password.

Use Fail-Safe Defaults

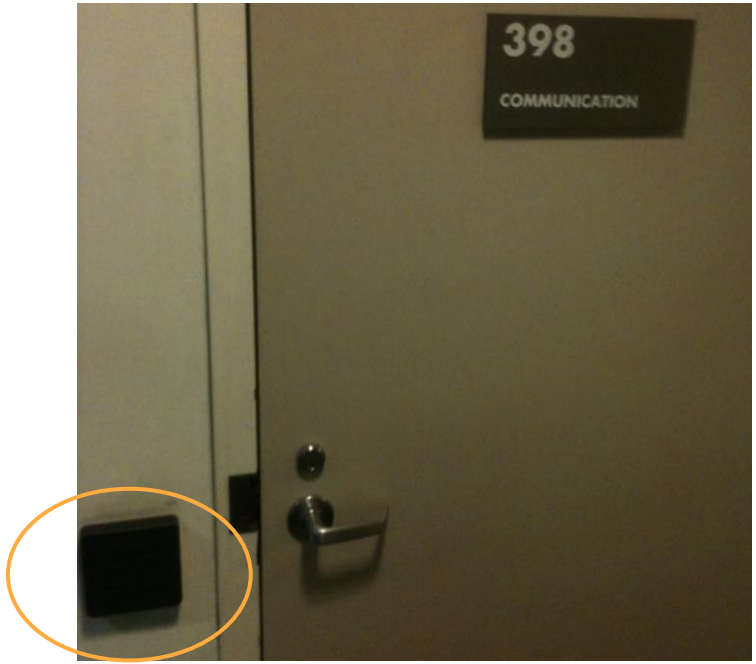
Textbook Chapter 1.10

Soda Hall

- Rooms in Berkeley's Soda Hall are guarded by electronic card keys
- What do you do if the power goes out?
  - Fail closed: No one can get in if the power is out
  - Fail open: Anyone can get in if the power goes out
- What's the best option to choose for closets with expensive equipment? What about emergency exit doors?
- **Takeaway**: Use fail-safe defaults… (if you can come up with one!)

Use Fail-Safe Defaults

- Choose default settings that “fail safe,” balancing security with usability when a system goes down
  - This can be hard to determine
- In the end, the right “default” often depends on context
  - Default open?
  - Default locked?



Design in Security from the Start

Textbook Chapter 1.11

Design in Security from the Start

- When building a new system, include security as part of the design considerations rather than patching it after the fact
  - A lot of systems today were not designed with security from the start, resulting in patches that don’t fully fix the problem!
- Keep these security principles in mind whenever you write code!

Summary: Security Principles

- **Know your threat model**: Understand your attacker and their resources and motivation
- **Consider human factors**: If your system is unusable, it will be unused
- **Security is economics**: Balance the expected cost of security with the expected benefit
- **Detect if you can't prevent**: Security requires not just preventing attacks but detecting and responding to them
- **Defense in depth**: Layer multiple types of defenses
- **Least privilege**: Only grant privileges that are needed for correct functioning, and no more
- **Separation of responsibility**: Consider requiring multiple parties to work together to exercise a privilege (code review and design review)
- **Ensure complete mediation**: All access must be monitored and protected, unbypassable
- **Don't rely on security through obscurity**: Assume the enemy knows the system
- **Use fail-safe defaults**: Construct systems that fail in a safe state, balancing security and usability.
- **Design in security from the start**: Consider all of these security principles when designing a new system, rather than patching it afterwards