Introduction and Security Principles

CS 161 Spring 2024 - Lecture 1

First Half of Today: Introductions and Logistics

- Staff introductions
- Course overview: What will you learn in this class?
- Course logistics
  - Lectures, discussions, office hours, and exams
  - Resources and communication platforms
  - Collaboration and academic honesty
  - DSP and extenuating circumstances
  - Stress management and mental health
  - Ethics
  - Case studies and blue slides
- What is security? Why is it important?

Staff Introductions

Who Am I? Peyrin (he/him)

- Three ways to save me some headaches:
- 1: Please call me “Peyrin”
  - No “professor”, “Mr.”, “sir”, “doctor”, etc. I’m not paid enough for that
- 2: Email me at cs161-staff@berkeley.edu
  - The head TAs and I all check this, so you'll get a faster response
  - If it's a personal matter, personal email is okay, but staff email is usually better
- 3: I am not a security researcher
  - My research focus was CS education, not security
  - If you have research-related questions, I'll try to point you in the right direction, but can't help directly

Actual real picture of me.

Who Am I? Raluca Ada Popa (she/her)

- Associate professor in computer security
- MIT PhD in security
- Leads the cryptographic systems research group, and co-runs SkyLab at UC Berkeley
- Research topics
  - Broadly: Building secure systems with the help of cryptography
  - More specifically: Secure computation, security and privacy aspects of gen AI; decentralized security
  - Co-founder of two cybersecurity companies, PreVeil (CTO) and Opaque (President)
- Taught this class 7 times
- 2 fun facts …

Our team of talented TAs!

Come back in a later lecture to see this slide. Sorry.

Course Overview

Learning Objectives

- How to think adversarially about computer systems
- How to assess threats for their significance
- How to build computer systems with robust security properties
- How to gauge the protections and limitations provided by today's technology
- How attacks work in practice

- What mistakes *not to make!*

Course Outline

- Introduction to Security
  - What are some general philosophies when thinking about security?
- Memory Safety
  - How do attackers exploit insecure software? How do we defend against these attacks?
- Cryptography
  - How do we securely send information over an insecure channel?
- Web Security
  - What are some attacks on the web, and how do we defend against them?
- Network Security
  - What are some attacks on the Internet, and how do we defend against them?
- Miscellaneous Topics
  - Useful, interesting, or fun applications of topics. Maybe some guest lectures!

Extra Tools and Skills

- Some extra non-security-related skills you can take away from this class:
- Memory safety
  - x86 assembly: A commonly-used assembly language
  - Using GDB: Debugging C code
- Cryptography
  - Becoming a better consumer: Be able to analyze security products and pick the right security tools for your software
- Web Security
  - Software engineering: Understanding how websites are built and how your web browser interacts with remote web servers (CS 169 preview)
- Network Security
  - Networking: How the Internet works (CS 168 preview)

Prerequisites

- CS 61B: Ability to work with large and complex codebases, data structures
  - Relevant for Project 2 (500–1000 lines of Go code)
- CS 61C: Familiarity with low-level memory layouts and assembly
  - We'll have a lecture reviewing all the 61C material you need to succeed
  - Relevant for the memory safety unit (Project 1, first four weeks of class only)
- CS 70: Familiarity with basic mathematical notation and proof structures
  - Relevant for the cryptography unit
  - We'll review CS 70 material as we encounter it during the cryptography lectures
- An ability to pick up new programming languages quickly
  - Project 2 will be in Go

Course Logistics

Enrollment

- Course staff does not control enrollment; we have to follow department policy
  - Only CS majors will be able to enroll this fall
  - If you are waiting to be declared and plan to enroll later, please email cs161-staff@berkeley.edu
- We are trying to clear the waitlist and enroll all interested students, but no promises, so please have a backup ready
- Concurrent enrollment students are awaiting approval from the department to proceed
  - We'll add you to class platforms at the end of the week if you still aren't approved

Course Structure: Lectures

- You are here!
- Monday/Wednesday, 6:30–8:00 PM PT
- Attendance is not taken

In-person	Synchronous online	Asynchronous online
● Live lectures in Dwinelle 155	● Live lectures over Zoom	● Lecture recordings posted on the website

Course Structure: Discussions

- Smaller sections led by a TA to practice with material
- Discussion schedule available at https://sp24.cs161.org/calendar
- CS 61C review session Thurs, Jan 18 from 2-3pm in VLSB 2066 + zoom
- All regular discussions start next week (week of January 22)
- You can attend any discussion section you want (no need to enroll in a section)
- Attendance is not taken

In-person	Synchronous online	Asynchronous online
• Discussion rooms posted on Ed/calendar	• Discussion zoom links posted on Ed	• Some sections will be recorded • Discussion walkthrough videos will be posted

Course Structure: Office Hours

- Space to ask questions about content, get help with projects, raise concerns with the course, etc. with a TA or instructor
- Office hours schedule available at https://sp24.cs161.org/calendar
- Office hours queue at https://oh.cs161.org/
- All office hours start next week (week of January 22)
- We'll take online help requests if absolutely necessary, but will prioritize in-person tickets
    - In our experience, in-person office hours are more efficient for staff and students

Course Structure: Exams

- Midterm: Thursday, February 29, 7–9pm PT.
- Final: Friday, May 10, 2024, 3–6pm PT
- Remote exam offered only at the same time as the regular exam
  - Will need to consent to our video proctoring policy if you plan to opt into the remote exam
  - We reserve the right to revoke remote exam privileges in instances of misconduct
- Alternate exam offered in-person only, immediately after the regular exam
- Forms to request online/alternate exams will be released closer to the exam

Resources

- Textbook: https://textbook.cs161.org/
  - Free! There's no textbook you need to pay for.
  - Readings are optional, but past students have said the textbook is helpful
- Course website: https://sp24.cs161.org/
  - Course schedule, lecture slides, assigned readings, and other resources are all posted here

Platforms

- Ed (https://edstem.org)
  - Discussion forum
  - Course-related communication should take place in Ed or happen in office hours
  - For private matters, you can make a private post
  - Please don't post publicly about project spoilers!
- Gradescope
  - All assignments are submitted and graded on Gradescope
- Email
  - cs161-staff@berkeley.edu for private matters. Staff with email access are tagged as "Staff Email Access" on the staff page of the website.
  - Ed response is faster, but staff will monitor the email if you're more comfortable with that

Grading Structure

- Homework: 10%
  - Completed individually
  - 7 homeworks in total, weighted equally
  - Gradescope with instant feedback: You can keep trying until you get the answer right
  - No credit for submitting late, unless you have an extension
- Projects: 40%
  - P1 = 10%, P2 = 20%, P3 = 10%
  - Completed individually or in groups of 2
  - No credit for submitting late, unless you have an extension
- Midterm: 20%
- Final: 30%

Class Policies: Extensions

- We do not have slip days or assignment drops in this course!
- However, you can request extensions on any assignment for any reason:
  - Extensions form linked at the top of the website (will be live soon)
  - Extensions ≤3 days will be automatically approved if submitted before the deadline
  - Longer extensions may be approved, but we'll need to check in first to make sure you're on pace to finish the class
- It is okay to request an extension if things come up! We're here to support you!
  - Life happens.
  - You can always request an extension for any reason (e.g. stress, other classes, etc.)
  - We want to keep deadlines to make sure you're on track to finish, while reducing the associated stress
- For the first few weeks, due to technical difficulties, don't expect a confirmation email unless your extension was **not** approved

Class Policies: DSP

- Disabled Students' Program (DSP)
  - There's a variety of accommodations UC Berkeley can help us set up for you in this class
  - https://dsp.berkeley.edu/
- Are you facing barriers in school due to a disability?
  - Apply to DSP!
  - We maintain proper access controls on this information: Only the staff members with the "DSP Data" tag on the website can access any DSP-related info
  - Our goal is to teach you the material in our course. The more accessible we can make it, the better.
- If you're registered/registering with DSP:
  - Please send us your Letter of Accommodations (LOA) through AIM (the DSP portal) ASAP!
  - We will be sending out onboarding materials to all DSP students who have sent us their LOA in AIM sometime this weekend, please keep an eye out

Class Policies: Collaboration

- Asking questions and helping others is encouraged
  - Discussing course topics with other is welcome!
- Limits of collaboration
  - Don’t share solutions with each other (except project partners)
  - You should never see or have possession of anyone else’s solutions—including from past semesters
  - If you're not sure, see the policies page on the website, or ask us first
- Staff is not responsible for partnerships
  - If you work in a group, the penalties apply for all group members, even if only one group member engaged in misconduct, or if the other group member was unaware of the misconduct.

Class Policies: Academic Honesty

- We’re here to help! There are plenty of staff and resources available for you
  - You can always talk to a staff member if you’re feeling stressed or tempted to cheat
- Academic dishonesty policies
  - Negative points on the assignment (e.g. if the midterm is worth 150 points, you'd receive a score of -150 on the midterm)
  - Referral to the Center for Student Conduct

Class Policies: Academic Honesty

- As a computer security class, we view potential cheaters as “attackers.”
- Our threat model assumes “rational” attackers.
  - A rational attacker will only launch an attacker if (expected benefit) > (expected cost)
    - (expected cost) = (cost of launching attack) + *(cost of getting caught)* * (probability of getting caught)
- Two-fold approach to academic integrity:
  - Detection: Use our tools to analyze and detect instance of academic dishonesty.
    - You will learn that “security through obscurity” is bad, but *obscurity can help*. We have ways.
  - Response: *At minimum*, you will receive negative points on the assignment.

Stress Management and Mental Health

- We want to reduce your stress where we can
  - Project 2 (mid-semester) is going to be the most intensive part of this class, but we've made things lighter towards the end (when every other class has stuff due)
- **Your health is more important than this course**
- If you feel overwhelmed, there are options
  - Academically: Ask on Ed, talk to staff in office hours, set up a meeting with staff to make a plan for your success this semester → Communication is key!
  - Non-academic:
    - Counselling and Psychological Services (CAPS) has multiple free, confidential services
      - Casual consultations: https://uhs.berkeley.edu/counseling/lets-talk
      - Crisis management: https://uhs.berkeley.edu/counseling/urgent
    - Check out UHS's resources: https://uhs.berkeley.edu/health-topics/mental-health

Ethics

- In this class, you will learn a lot about attacks out of necessity
  - To be able to defend against the attacker, you must learn the techniques that attackers use
- It is usually okay to break into your own systems
  - This is a great way to evaluate your own systems
- It is usually okay to break into someone else's systems with their explicit permission
  - For this class, per the misconduct policy, please don't share test cases to break into others' systems
- It is *grossly unethical* and *exceedingly criminal* to break into someone else's systems without their permission

Course Climate

- Please respect each other! This includes but is not limited to:
  - Using the preferred pronouns of your fellow students & staff members
  - Posting appropriate content on Ed
  - Respecting the identities of others
  - Being kind
  - Facilitating open and respectful discourse and upholding academic freedom, even for topics that may be political or controversial
- Staff reserves the right to lower your grade for exceptionally rude or disrespectful behavior
  - You do not need to be concerned with this policy if you treat others with even the bare minimum of respect.
- If you feel as though you have been impacted by a poor course climate, we have an anonymous feedback form on the website.

Case Studies and Blue Slides

- Security is often best taught through real-world case studies and stories
  - Lectures will sometimes use real-world examples to demonstrate concepts
  - Slides with a blue background are case study slides
- Content on blue slides are not tested on exams
  - You *do not* need to remember the exact details of the story
- Some blue slides will end in a **takeaway** that describes the moral of the story
  - You *do* need to understand the takeaway from the story

Example Case Study: Stuxnet

- **Stuxnet**: Discovered July 2010, released approximately March 2010
- Malicious code designed to only active under very specific circumstances
  - For most computers, do nothing
  - For computers connected to the Iranian nuclear program, tamper with centrifuges
- Researchers later deduced that Stuxnet was created by the United States and Israel to target Iran’s nuclear program
  - Is it ethical that the US and Israel are the "attackers" in this scenario?
  - Who decides what is "legal" and ethical in this scenario?

What is security?

What is security?

Enforcing a desired property *in the presence of an attacker*

data confidentiality

user privacy

data and computation integrity

authentication

availability

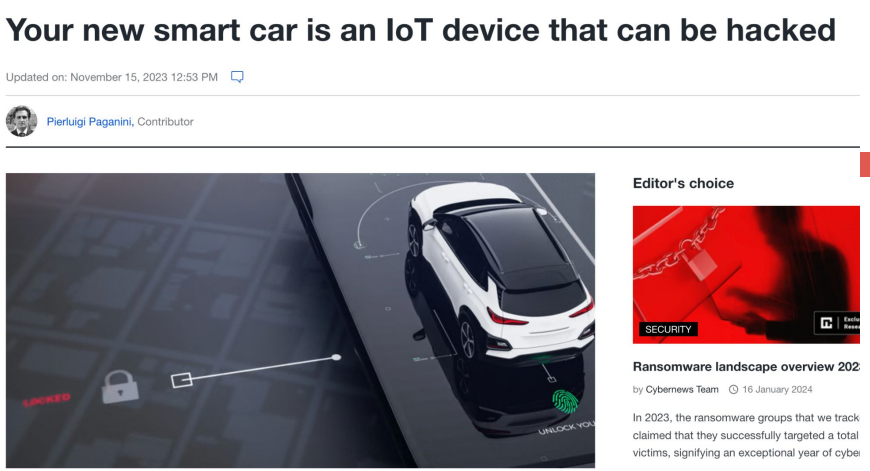
…

Why is security important?

- It is important for our
  - physical safety
  - confidentiality/privacy
  - functionality
  - protecting our assets
  - successful business
  - a country’s economy and safety
  - and so on…

Why is security important?

- Consider: Physical Safety



The Washington Post

FBI probe of alleged plane hack sparks worries over flight safety

Drew Harwell

PCWorld

Pacemaker hack can kill via laptop

Jeremy Kirk

Why is security important?

- Consider: Privacy/Confidentiality

Money

Link

Data Breach Tracker: All the Major Companies That Have Been Hacked

Karavbrandeisky *October 30, 2014*

DATA BREACHES

Norton Healthcare Ransomware Hack: 2.5 Million Personal Records Stolen

Compromised data includes names, dates of birth, Social Security numbers, health and insurance information, and driver's license numbers.

By Ionut Arghire
December 11, 2023

Why is security important?

- Consider: National security

THE WALL STREET JOURNAL.

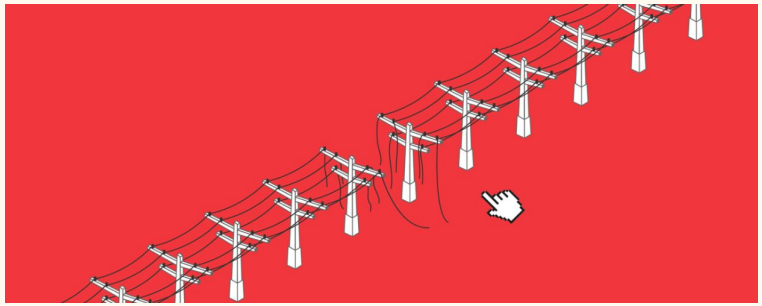
Link

America's Electric Grid Has a Vulnerable Back Door—and Russia Walked Through It

Rebecca Smith and Rob Barry

January 10, 2019

A Wall Street Journal reconstruction of the worst known hack into the nation's power system reveals attacks on hundreds of small contractors



What is hackable?

- Every software!
  - Especially things connected to the Internet
  - Assume that every system is a target
  - A casino was hacked because a fish-tank thermometer was hacked within the network

SLATE

Link

For the First Time, Hackers Have Used a Refrigerator to Attack Businesses

Julie Bort *January 17, 2014*

Security Principles

Textbook Chapter 1

Second Half of Today: Security Principles

- 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

Textbook Chapter 1.1 & 1.12

The Parable of the Bear Race

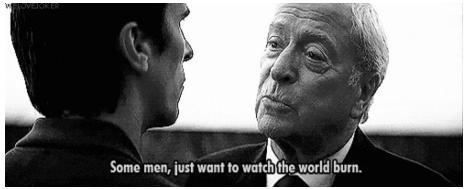
Reminder: blue slides are case studies. Remember the takeaway, not the story!

"I don't have to outrun the bear. **I just have to outrun you.**"

Takeaway: You often just need to have "good enough" defense to make attackers turn somewhere else.

Security Principle: 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
  - Politics
  - Retaliation
  - Fun, watching the world burn



Security Principle: Know Your Threat Model

- Consider: Personal security
- Who and why might someone attack *you*?
  - Criminals might attack you for money
  - Teenagers might attack you for laughs or to win online games
  - Governments might spy on you to collect intelligence
  - Intimate partners might spy on you
    - This is a surprisingly dangerous threat model!

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

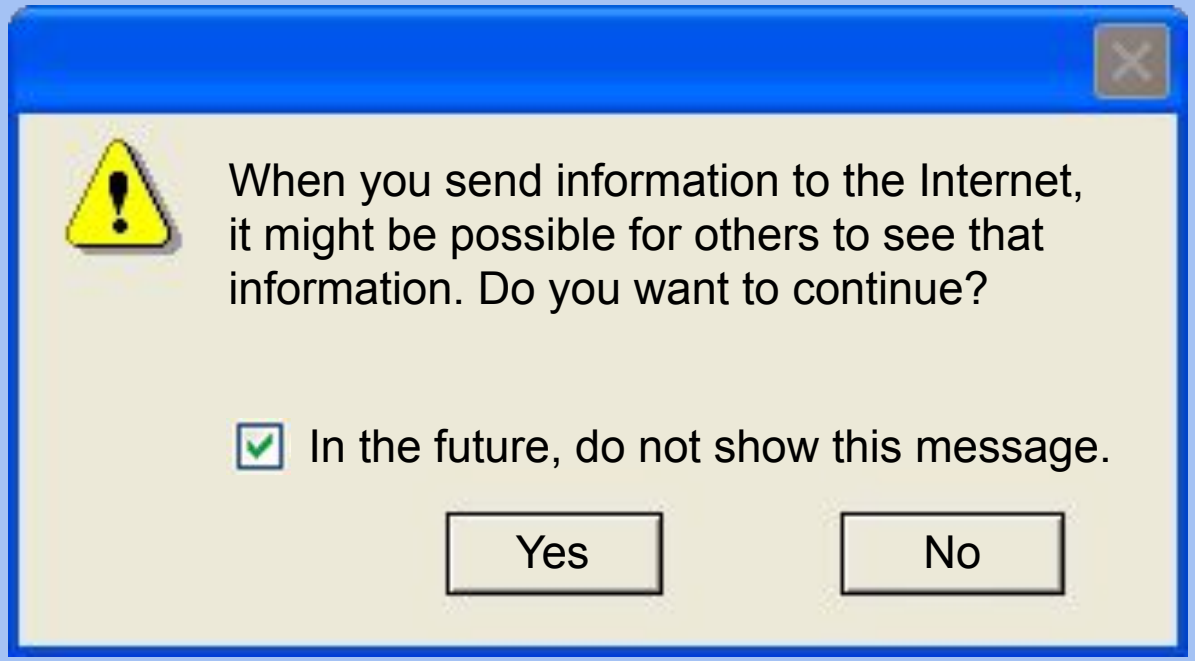
Trusted Computing Base

- **Trusted computing base (TCB)**: The components of a system that security relies upon
- 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

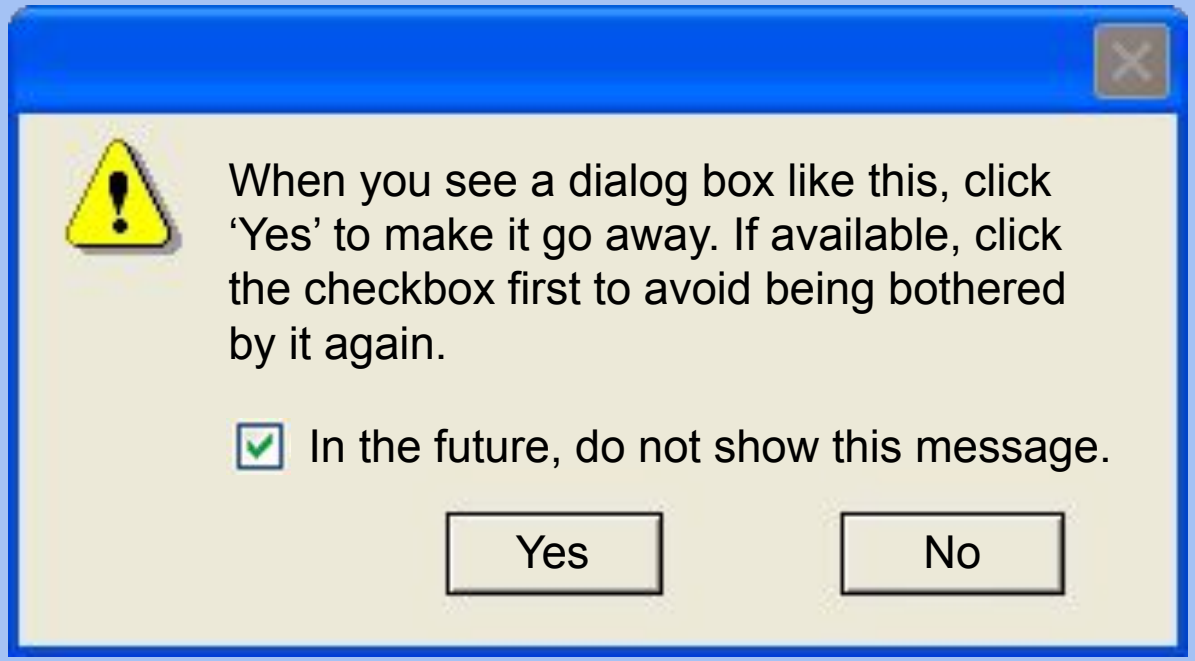
Consider Human Factors

Textbook Chapter 1.2

Warning Dialogs

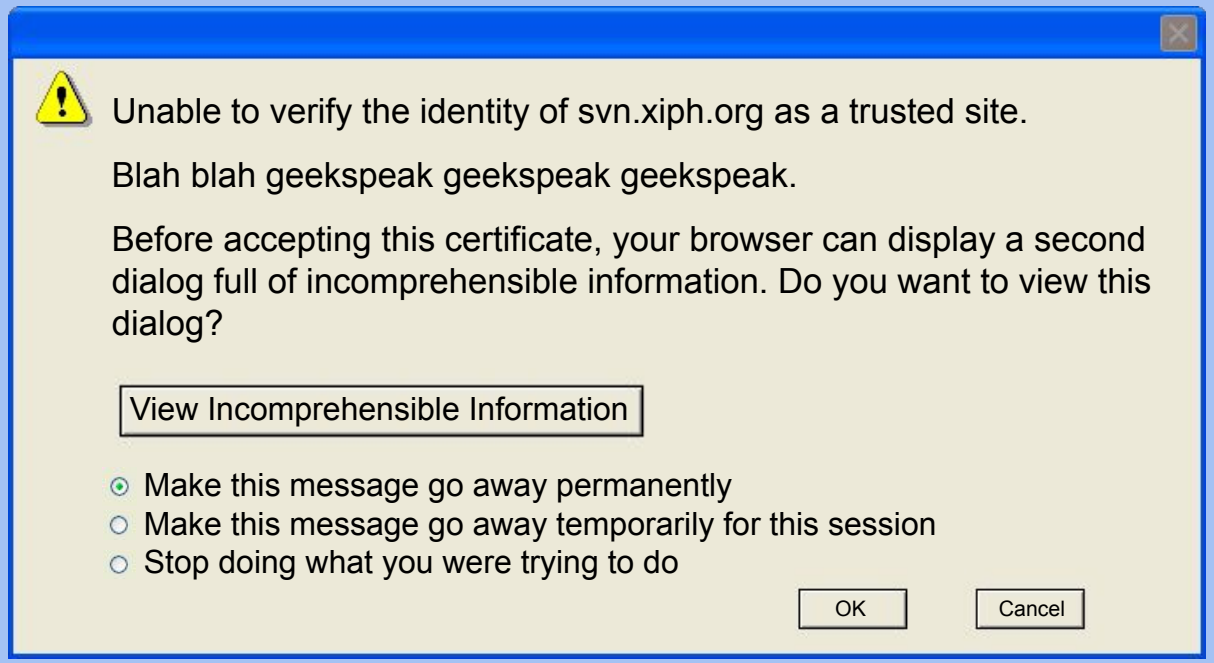


Warning Dialogs



Warning Dialogs

Warning Dialogs



The presence of warning dialogs often represent a failure: How is the user supposed to know what to do?

Takeaway: Consider human factors

Security Principle: Consider Human Factors

- It all comes down to people: The users
  - Users like convenience (ease of use)
  - If a security system is unusable, it will be unused
  - Users will find way to subvert security systems if it makes their lives easier
- It all comes down to people: The programmers
  - Programmers make mistakes
  - Programmers use tools that allow them to make mistakes (e.g. C and C++)
- It all comes down to people: Everyone else
  - Social engineering attacks exploit other people's trust and access for personal gain
- Consider the tools presented to users, and make them **fool**-proof

Physical security keys use the fact that humans are trained to safeguard keys

Security is Economics

Textbook Chapter 1.3

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

Security Principle: Security is Economics

- Cost/benefit analyses often appear in security: The expected benefit to the attacker should ideally be smaller than the expected cost of attack
  - 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 item…
  - … unless a $1 item 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, so it's probably safe to use an iPhone on a hostile network if you aren't a $1M target

Detect If You Can’t Prevent

Textbook Chapter 1.4

Burglar Alarms

- Security companies are supposed to detect home break-ins
  - Problem: Too many false alarms. Many alarms go unanswered
  - Placing a sign helps deter burglars from entering at risk of being caught…
    - … even if you don't have an alarm installed!
- **Takeway**: Prevent attacks when you can, but detect them if you can't

Security Principle: Detect if You Can’t Prevent

- **Deterrence**: Stop the attack before it happens
- **Prevention**: Stop the attack as it happens
- **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
  - Keep offsite backups!
  - If your computer and house catch on fire, it should be no big deal.

Detection but no Response

- Bitcoin 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
- **Takeaway:** Prevention is great, but you must not *only* depend on prevention; you must also respond

Bloomberg

Link

Hacked Bitcoin Exchange Says Users May Share $68 Million Loss

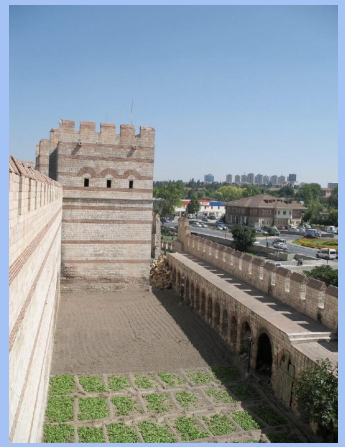
Lulu Yilun Chen and Yuji Nakamura — *August 5, 2016*

Defense in Depth

Textbook Chapter 1.5

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…
- **Takeaway**: Defense in depth



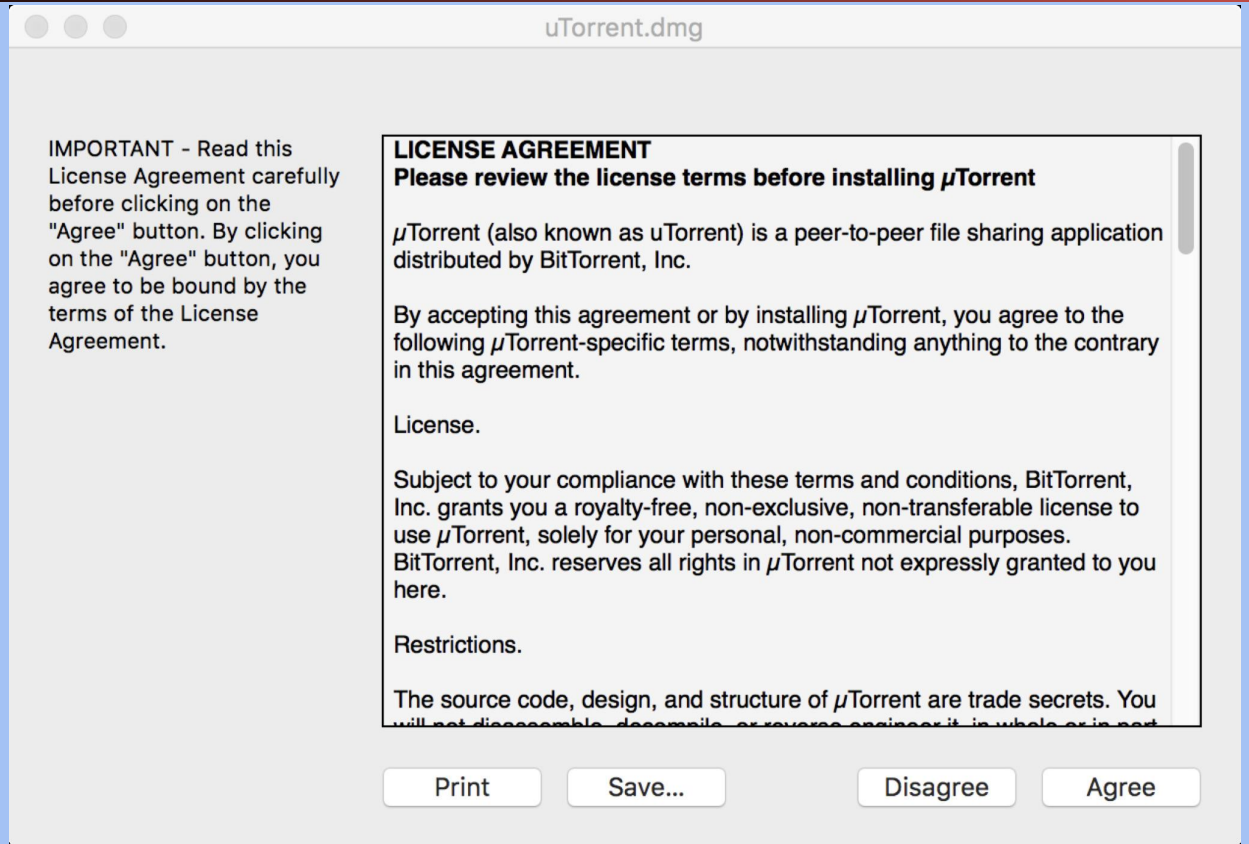
Security Principle: Defense in Depth

- Multiple types of defenses should be layered together
- An attacker should have to breach all defenses to successfully attack a system
- However, consider **security is economics**
  - Defenses are not free.
  - Defenses are often less than the sum of their parts

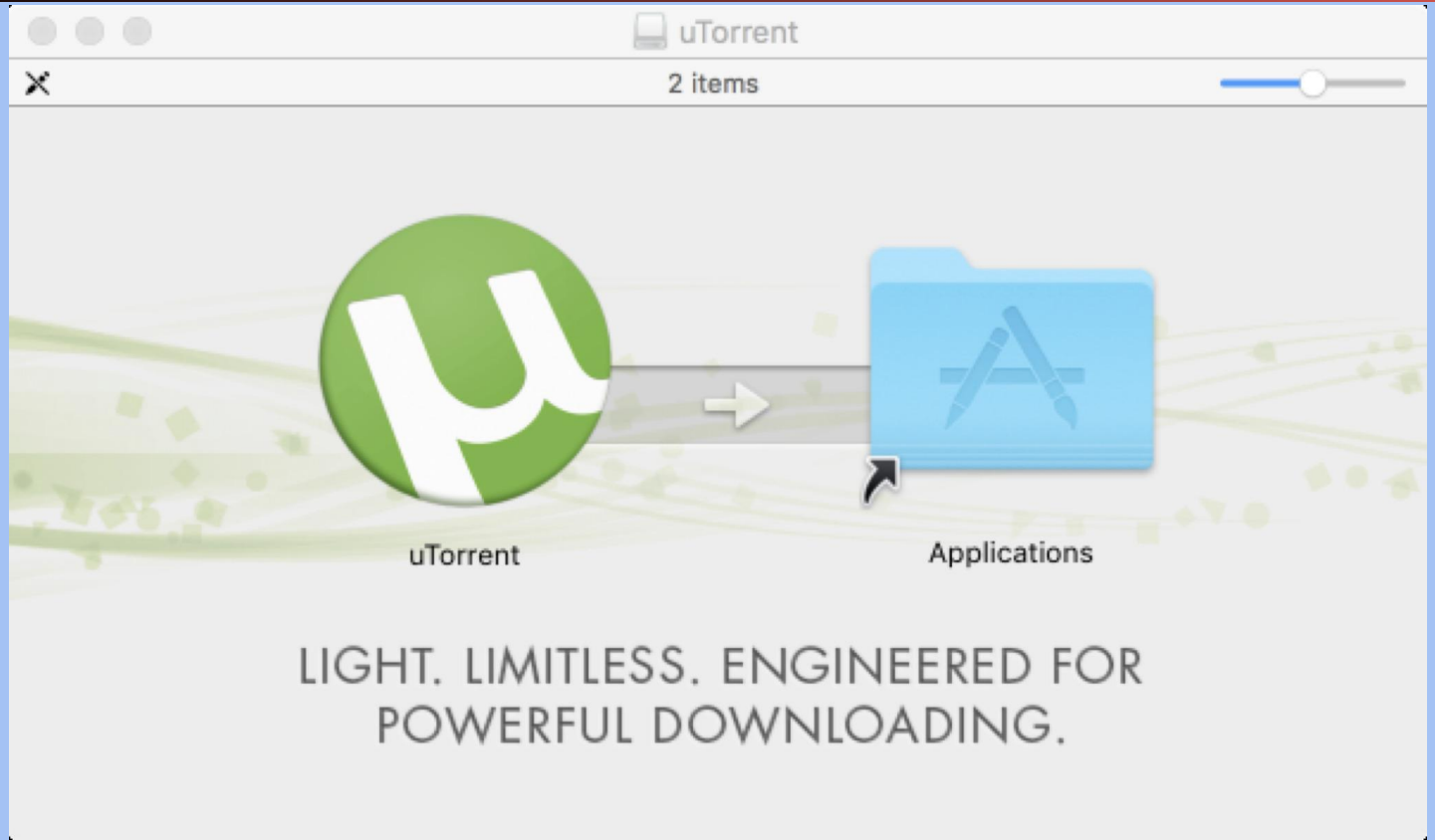
Least Privilege

Textbook Chapter 1.6

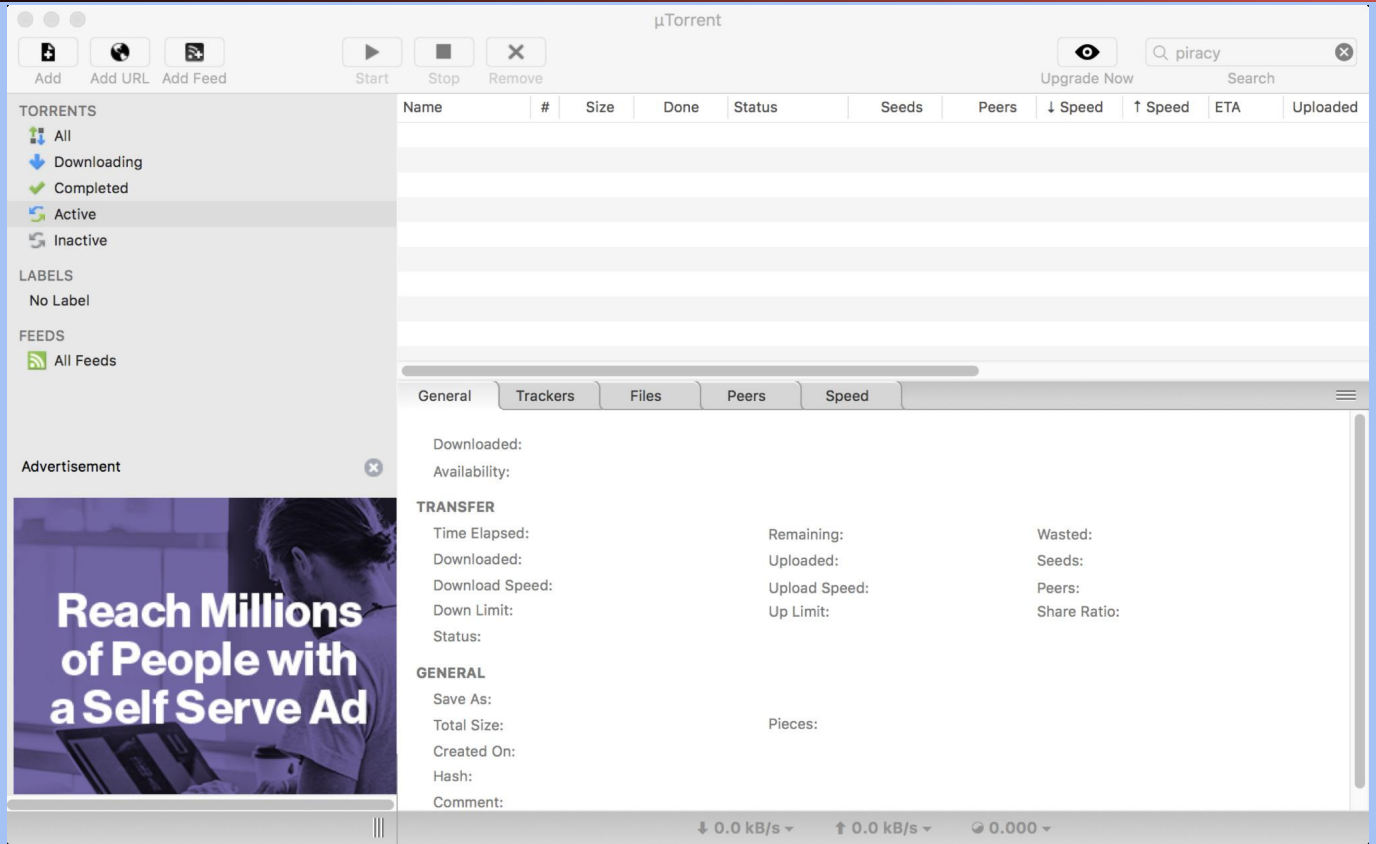
uTorrent


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

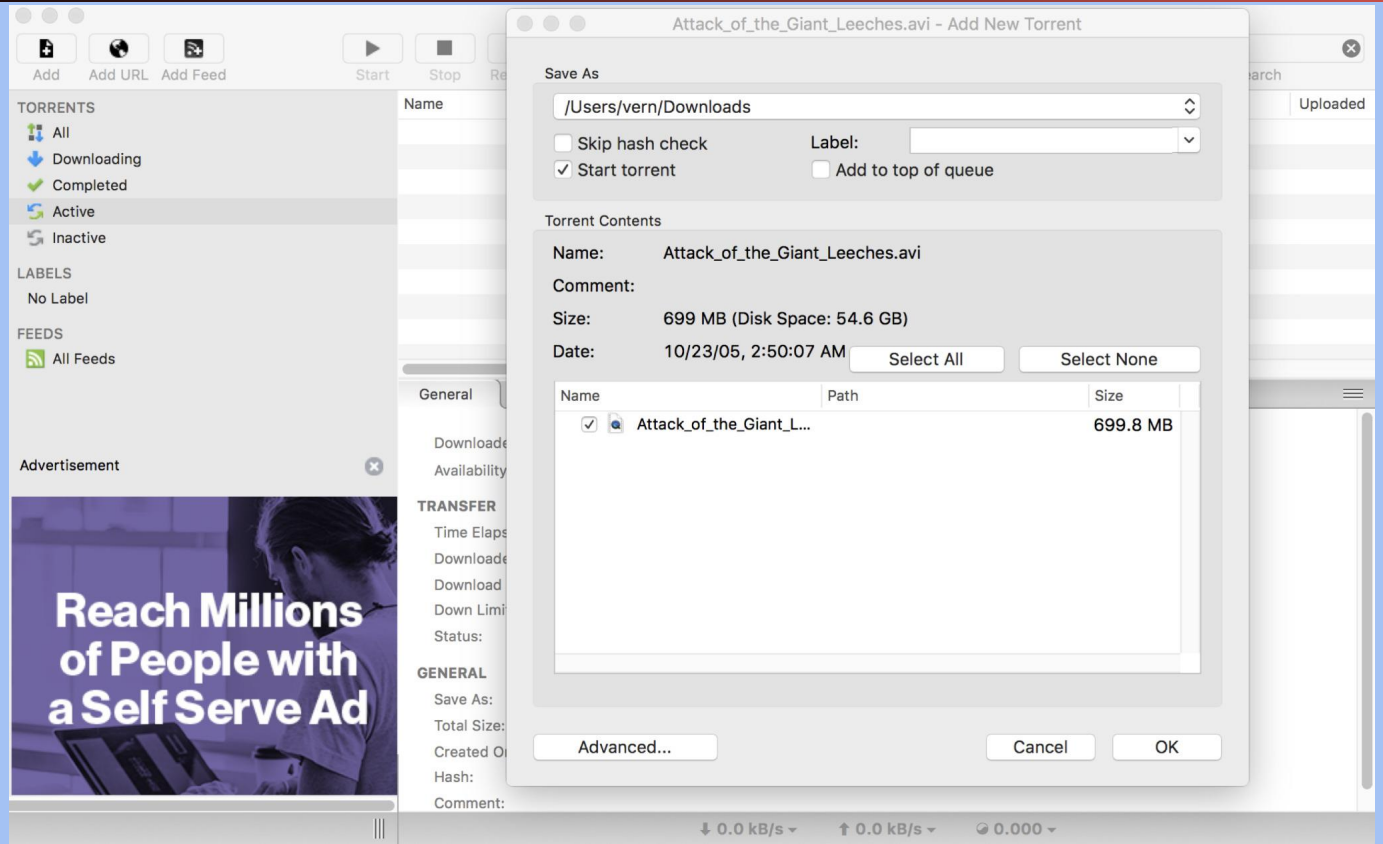
uTorrent


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

uTorrent

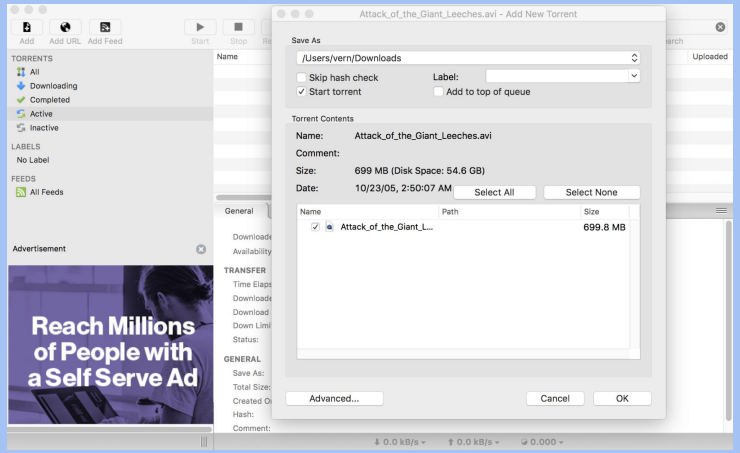

µTorrent
Add
Add URL
Add Feed
Start
Stop
Remove
Upgrade Now
piracy
Search
TORRENTS
All
Downloading
Completed
Active
Inactive
LABELS
No Label
FEEDS
All Feeds
Name
#
Size
Done
Status
Seeds
Peers
↓ Speed
↑ Speed
ETA
Uploaded
General
Trackers
Files
Peers
Speed
Downloaded:
Availability:
TRANSFER
Time Elapsed:
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



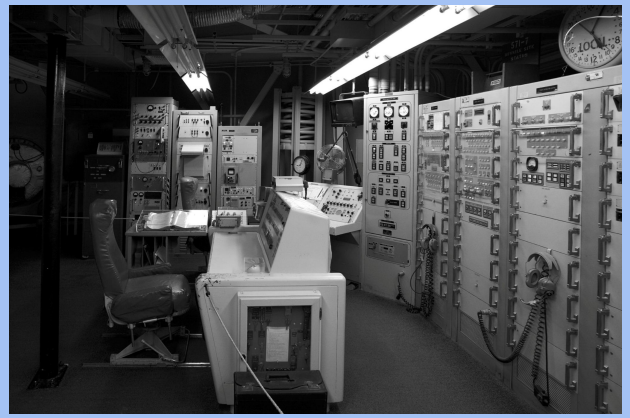
Security Principle: Least Privilege

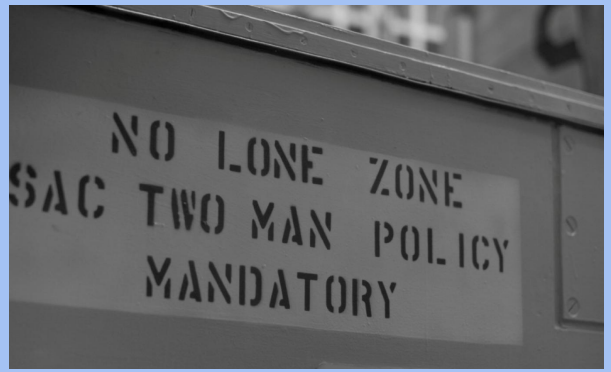
- Consider what permissions an entity or program *needs* to be able to do its job correctly
  - If you grant unnecessary permissions, a malicious or hacked program could use those permissions against you

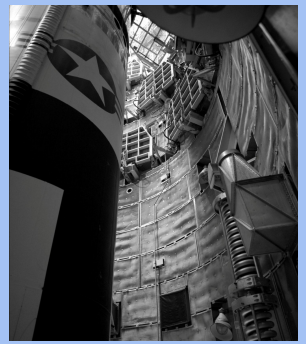
Separation of Responsibility

Textbook Chapter 1.7

Welcome to a Nuclear Bunker







Security Principle: Separation of Responsibility

- 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

Ensure Complete Mediation

Textbook Chapter 1.8 & 1.13

Spot the Issue

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

The cars drove around the barrier

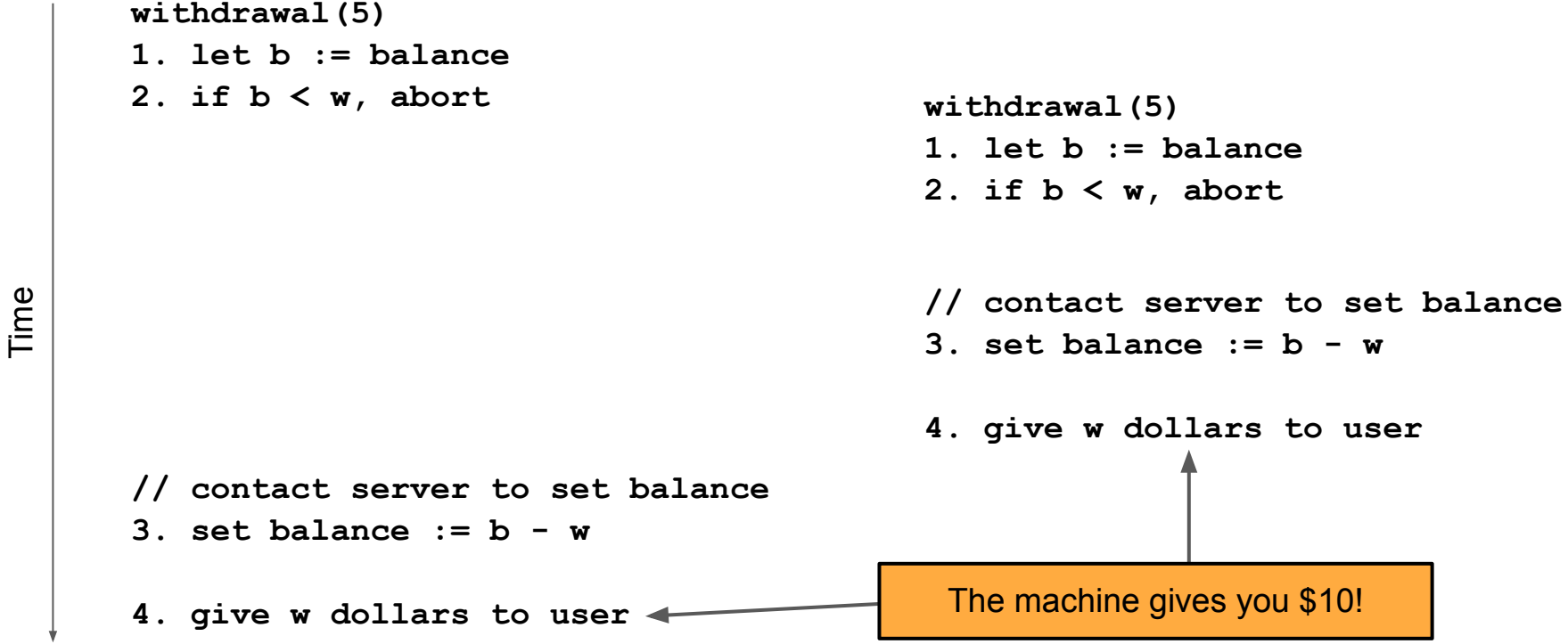
Time-of-Check to Time-of-Use

- 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

Accident on Motorway

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`**.

Caution! Zombies Ahead!!!

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

Trapped in Sign Factory! Send Help!

Takeaway: Shannon's maxim/Don't rely on security through obscurity

Security Principle: Shannon's Maxim

- **Shannon's maxim**: "The enemy knows the system"
- You should never rely on obscurity as part of your security. Always assume that the attacker knows every detail about the system you are working with (algorithms, hardware, defenses, etc.).

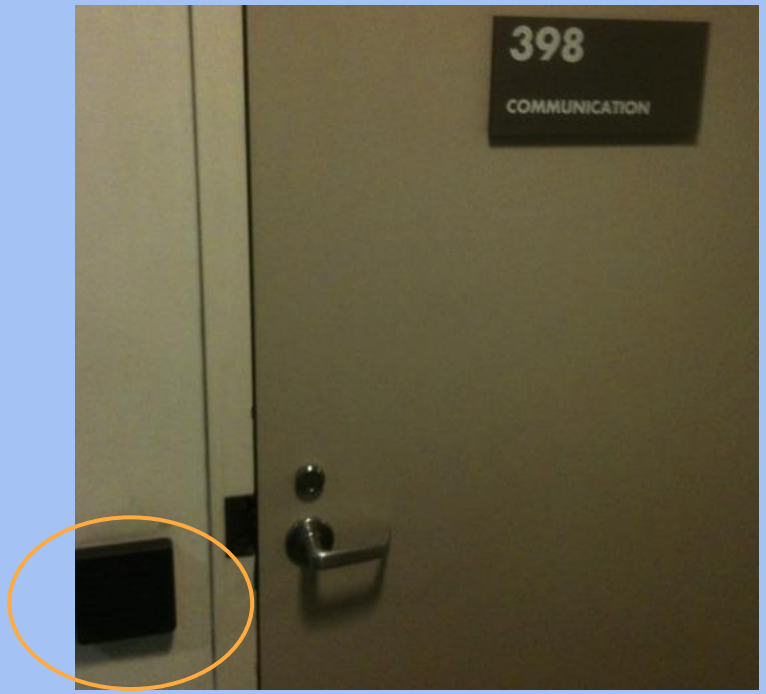
Assume the attacker knows where the "secret" control panel is located, and has read the manual with instructions on resetting the password.

Use Fail-Safe Defaults

Textbook Chapter 1.10

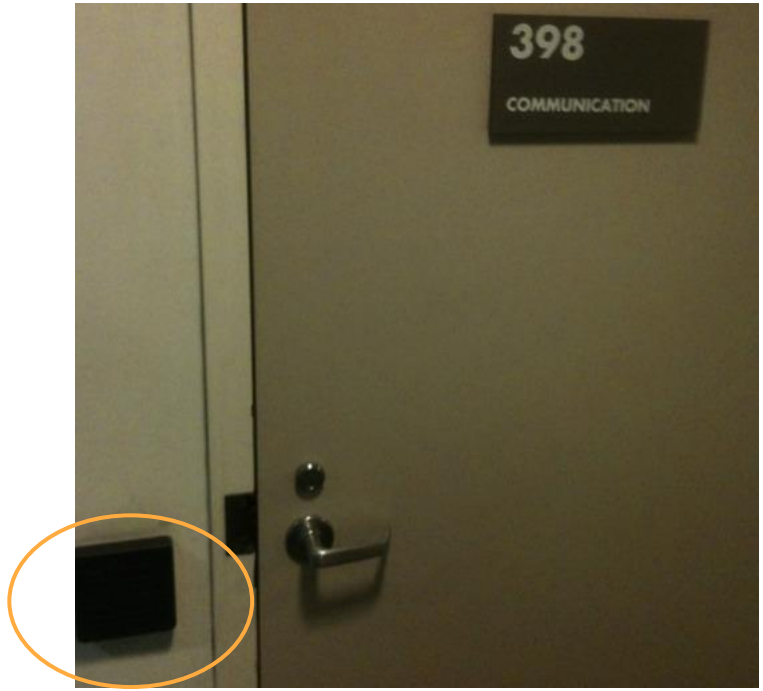
Soda Hall

- 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



Security Principle: 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



Design in Security from the Start

Textbook Chapter 1.11

Security Principle: 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!

Security Principles: Summary

- **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
- **Ensure complete mediation**: All access must be monitored and protected, unbypassable
- **Shannon's maxim**: 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