
PLANT
HART'S
SEEDS
RULE 777
IF YOU DON'T MAKE IT USABLE & SECURE
USERS WILL MAKE IT USABLE & INSECURE
imgflip.com

Lecture 2:
Some Security Principles

https://cs161.org

ObAdvertisement...

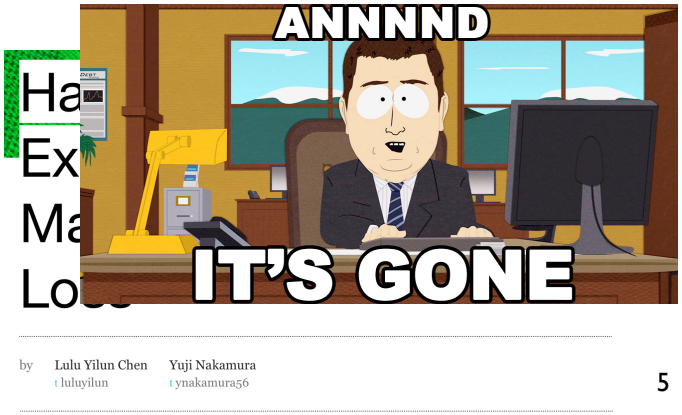
- I'm a lecturer, not a professor...
  - So I don't have graduate student
- But I do have some interesting research problems...
- So I have a URAP page...
  - One is debugging my drone control board
  - One (not yet listed) is seeing if a RPi-4 is capable of controlling an autonomous drone
  - One is banging on the Great Firewall of China

Security often comes down comes down to money...

- "You don't put a $10 lock on a $1 rock...
  - Unless the attacker can ***leverage*** that $1 rock to attack something more important
- "You don't risk exposing a $1M zero-day on a nobody"
  - So I'm quite content to use my iPhone in a hostile network: free market cost of a zero-day (unknown/unpatchable) exploit chain for iOS is somewhere between $500k to $1.5M
- Cost/benefit analyses appear all throughout security

Prevention

- The goal of prevention is to stop the "bad thing" from happening at all
- On one hand, if prevention works its great
  - E.g. if you don't write in an unsafe language (like C) you will ***never*** worry about buffer overflow exploits
- On the other hand, if you can ***only*** depend on prevention...
  - You get Bitcoin and Bitcoin thefts
  - E.g. $68M stolen from a Bitcoin exchange
  - Or Ethereum's July 2018: four separate multi-million-dollar theft incidents
  - Or Coinbase accounts: Averaging a ***known*** theft a day!



Detection & Response

- Detection: See that something is going wrong
- Response: Actually ***do*** something about it
  - Without some response, what is the point of detecting something being wrong?

Burglar Alarms Cops Won't Answer

Jacquie Simms, left, leader of the Watts neighborhood council, and fellow Watts residents Milton Smith and his wife, Bernece, are seen outside the Smith's home, which is equipped with a burglar alarm, in Los Angeles, Friday, Feb. 7, 2003. / **AP**

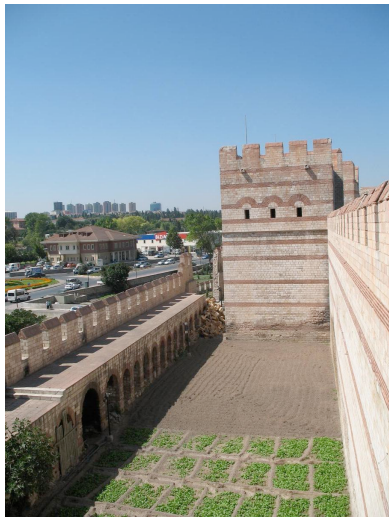
Comment / Share / Tweet / Stumble / Email

False Positive and False Negatives

- False positive:
  - You alert when there is nothing there
- False negative:
  - You fail to alert when something is there
- This is the real cost of detection:
  - Responding to false positives ***is not free***
    - And too many false positives and alarms get removed
  - False negatives mean a failure

Defense in Depth

- The notion of layering multiple types of protection together
  - EG, the Theodesian Walls of Constantinople:
    Moat -> wall -> depression -> even bigger wall
    - And some towers to rain down an eclectic mix of flaming and pointy death on those caught up in the defenses
- Hypothesis is that attacker needs to breech all the defenses
  - At least until something comes along to make the defense irrelevant like, oh, say siege cannons
- But defense in depth ***isn't free***:
  - You are throwing more resources at the problem
  - And although it can be better, it is less than the sum of the parts...



Composing Detectors for Defense In Depth... TINSTAAFL

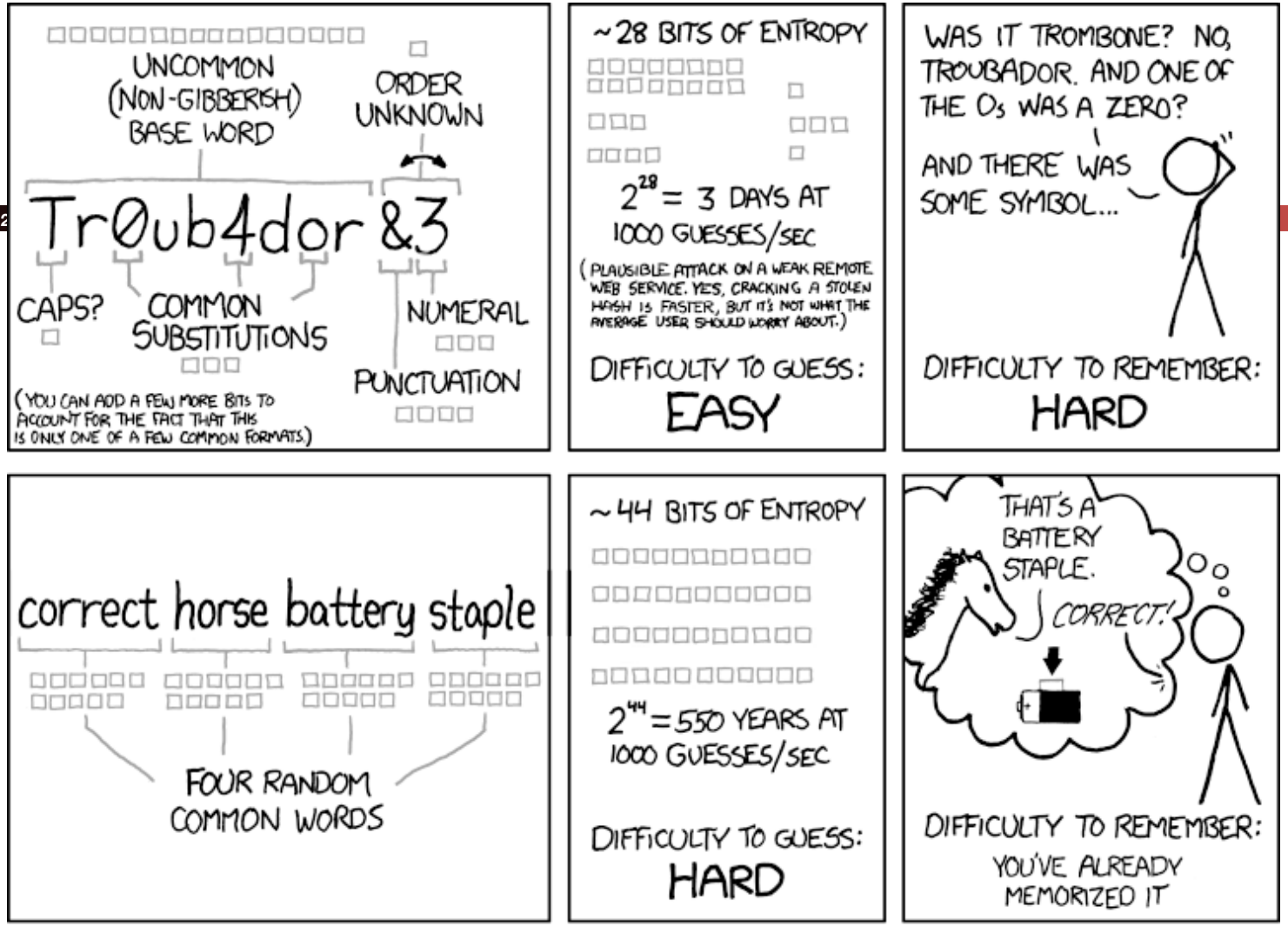
- There Is No Such Thing As A Free Lunch!
- The best case: the two detectors are ***independent***
  - With FP1 and FP2 false positive rates and FN1 and FN2 false negative rates
    - Rate is 0-1:
      0 is it never has a false positive/negative,
      1 is it is always a false positive/negative...
- Parallel composition: ***either*** detector may alert to trigger a response
  - ***Reduces*** false negatives: new rate is FN1*FN2
  - ***Increases*** false positive rate: new rate is FP1 + (1-FP1)*FP2
- Serial composition: ***both*** detectors must alert
  - ***Reduces*** false positives: new rate is FP1*FP2
  - ***Increases*** false negatives: new rate is FN1 + (1-FN1) * FN2

Mitigation & Recovery...

- OK, something bad happened...
  - Now what?
- Assumption: bad things ***will*** happen in the system
  - So can we design things so we can get back working?
- So how do I plan for earthquakes?
  - "1 week of stay put and 50+ miles of get outta town"
- So how do I plan for ransomware?
  - "If my computer and house catches on fire, I have backups"... AKA, "If you love it, ***back it up!***"

Real World Security...
How is your account breached?

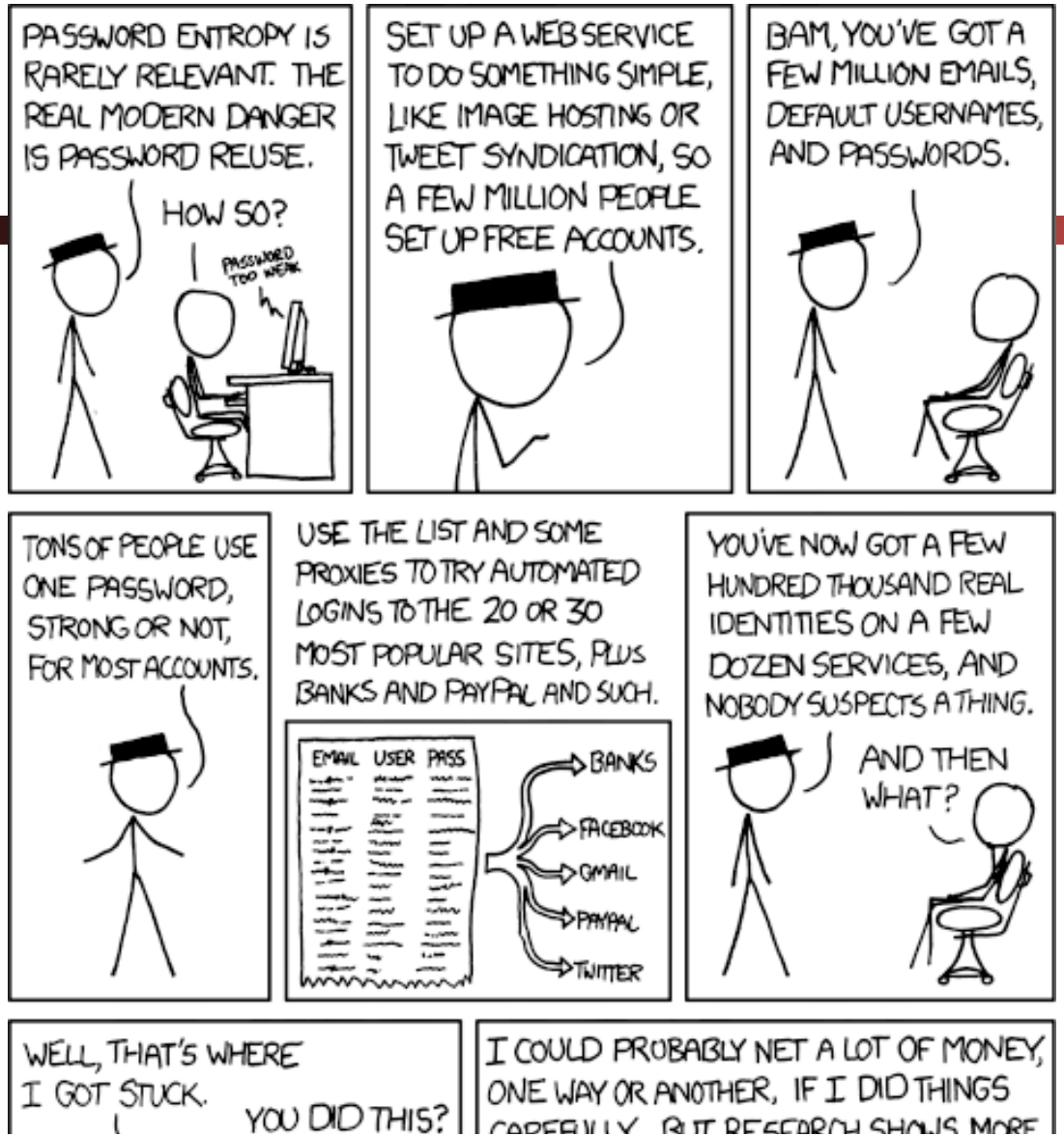
- Humans can't remember good passwords...
  - Well, we can remember a couple good passwords, but that's about it

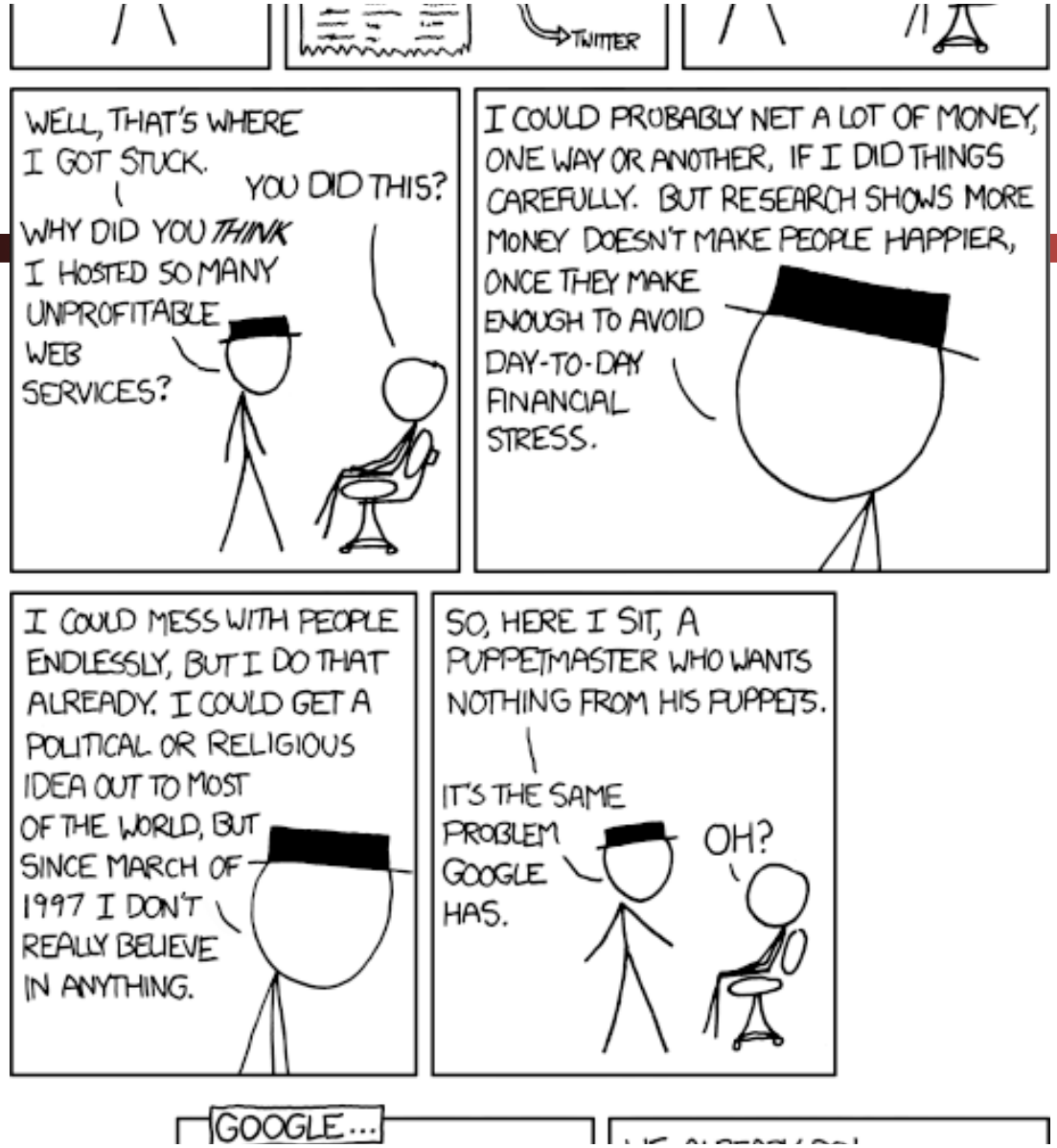


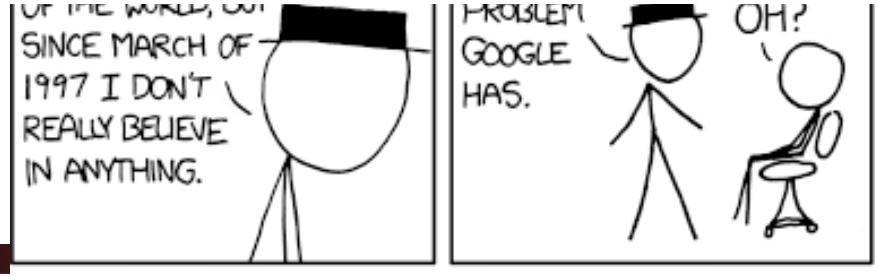
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Real World Security...
How is your account breached?

- So we compensate with password ***reuse***
  - You use the same lame password on a large number of sites that ***hopefully*** don't matter
- One of those sites gets breeched...
  - And now the bad guy has your password
  - And can now log into all those other sites where you used the same password...


PASSWORD ENTROPY IS RARELY RELEVANT. THE REAL MODERN DANGER IS PASSWORD REUSE.
HOW SO?
PASSWORD TOO WEAK
SET UP A WEB SERVICE TO DO SOMETHING SIMPLE, LIKE IMAGE HOSTING OR TWEET SYNDICATION, SO A FEW MILLION PEOPLE SET UP FREE ACCOUNTS.
BAM, YOU'VE GOT A FEW MILLION EMAILS, DEFAULT USERNAMES, AND PASSWORDS.
TONS OF PEOPLE USE ONE PASSWORD, STRONG OR NOT, FOR MOST ACCOUNTS.
USE THE LIST AND SOME PROXIES TO TRY AUTOMATED LOGINS TO THE 20 OR 30 MOST POPULAR SITES, PLUS BANKS AND PAYPAL AND SUCH.
EMAIL USER PASS
BANKS
FACEBOOK
GMAIL
PAYPAL
TWITTER
YOU'VE NOW GOT A FEW HUNDRED THOUSAND REAL IDENTITIES ON A FEW DOZEN SERVICES, AND NOBODY SUSPECTS A THING.
AND THEN WHAT?
WELL, THAT'S WHERE I GOT STUCK.
YOU DID THIS?
I COULD PROBABLY NET A LOT OF MONEY, ONE WAY OR ANOTHER, IF I DID THINGS


TWITTER
WELL, THAT'S WHERE I GOT STUCK.
YOU DID THIS?
WHY DID YOU THINK I HOSTED SO MANY UNPROFITABLE WEB SERVICES?
I COULD PROBABLY NET A LOT OF MONEY, ONE WAY OR ANOTHER, IF I DID THINGS CAREFULLY. BUT RESEARCH SHOWS MORE MONEY DOESN'T MAKE PEOPLE HAPPIER, ONCE THEY MAKE ENOUGH TO AVOID DAY-TO-DAY FINANCIAL STRESS.
I COULD MESS WITH PEOPLE ENDLESSLY, BUT I DO THAT ALREADY. I COULD GET A POLITICAL OR RELIGIOUS IDEA OUT TO MOST OF THE WORLD, BUT SINCE MARCH OF 1997 I DON'T REALLY BELIEVE IN ANYTHING.
SO, HERE I SIT, A PUPPETMASTER WHO WANTS NOTHING FROM HIS PUPPETS.
IT'S THE SAME PROBLEM GOOGLE HAS.
OH?
GOOGLE...


SINCE MARCH OF 1997 I DON'T REALLY BELIEVE IN ANYTHING.
GOOGLE HAS.
OH?


GOOGLE...
OKAY, EVERYONE, WE CONTROL THE WORLD'S INFORMATION. NOW IT'S TIME TO TURN EVIL. WHAT'S THE PLAN?
MAKE BOATLOADS OF MONEY?
Google
WE ALREADY DO!
SET UP A COMPANYWIDE CoD4: MODERN WARFARE TOURNAMENT EACH WEEK?
THAT'S NOT EVIL!
OOH, DIBS ON THE LOBBY TV!
OKAY, WE SUCK AT THIS.

So what to do?
Password Managers

- A program which runs on your computer or phone
  - You enter a master password to unlock an encrypted store
  - It can then enter passwords for you in websites
  - It can also generate strong, unique, random passwords
- Often include cloud syncing as well
  - So you ***better*** make sure your master password is good
  - But now means you have your master password everywhere
- Several options, I personally like 1password but there are others as well
  - EG, others like Keepass

1password

And Fido U2F Security Keys

- A very powerful second-factor for 2-factor authentication
  - Touch to cryptographically prove that you hold the key...
- We will use this as a case study when we get to cryptography...
- But takeaway for now: This ***can not be phished***:
  - The security key itself knows which site it is talking to through the browser:
    it knows the difference between www.google.com and www.g00gle.com

So For Account Security...

- Use a password manager
  - So you have a unique password for each site and a bad guy can't do "credential stuffing"
- Always enable 2-factor
  - So that even if a bad guy gets your password they have to get the second factor
    - Even SMS is better than nothing!
  - Even if you are successfully phished the bad guy only gets temporary access
- When possible, use a security key
  - Bad guys can't phish it at all!

So Homework -1:
Real World Security...

- Decide on a password manager and get it
  - If your CalNet password is shared with anything else, change it!
- Get yourself a security key
  - I like the Yubico ones, either a basic "security key" for $20 or a full Yubikey 5 for $50... But anything supporting U2F/FIDO2 will do
- Enable security key authentication on your CalNet and Google accounts
  - And all other key email accounts & social media accounts
- Now silently laugh at phishers and password stuffers!

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

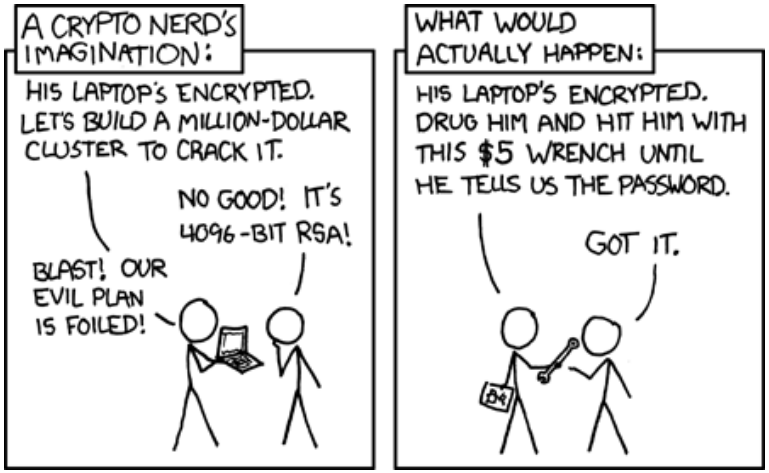
- TL-30:
  - The same expert and tools now takes 30 minutes

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



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!

Google

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 ...

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.

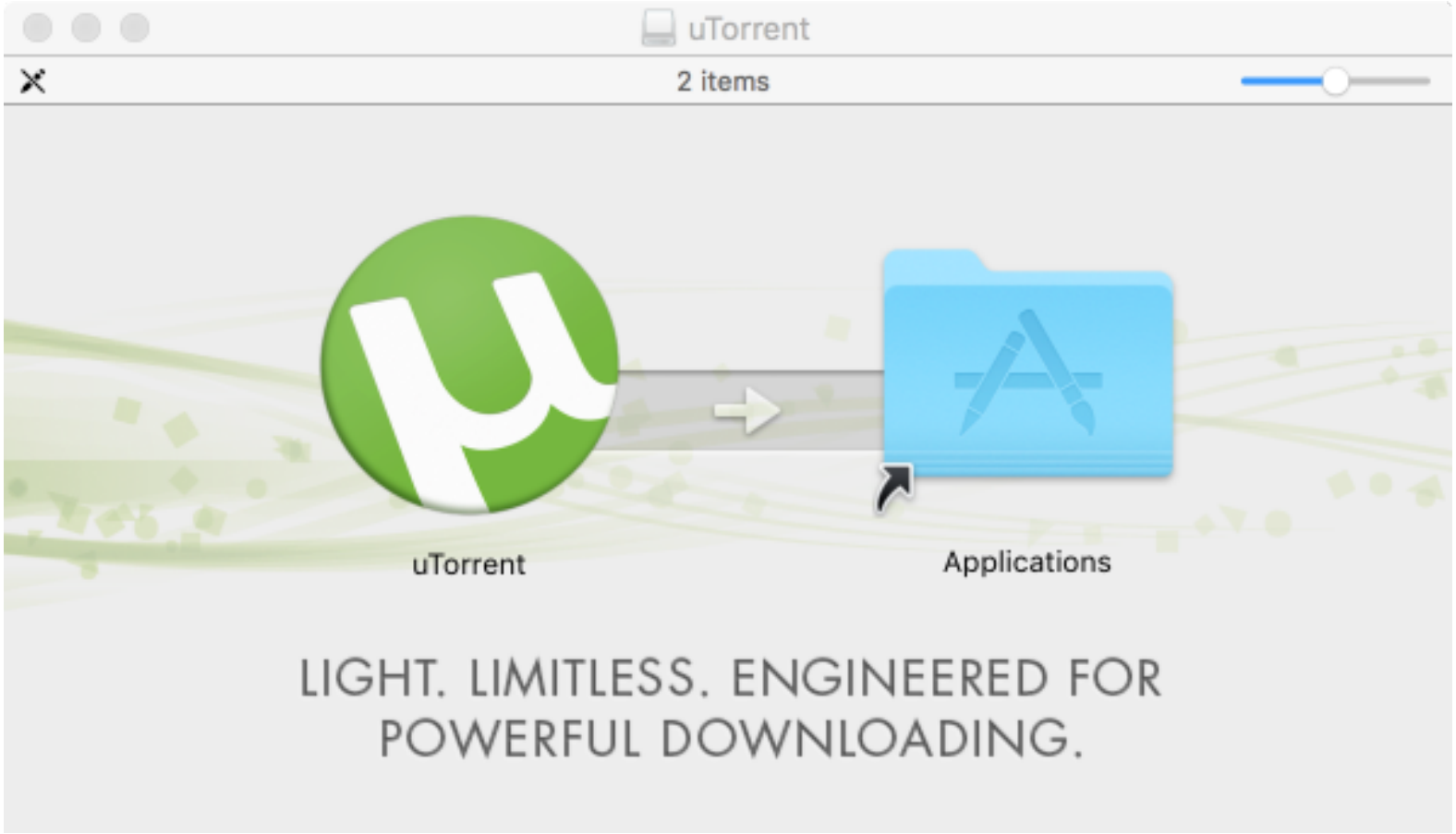
License.

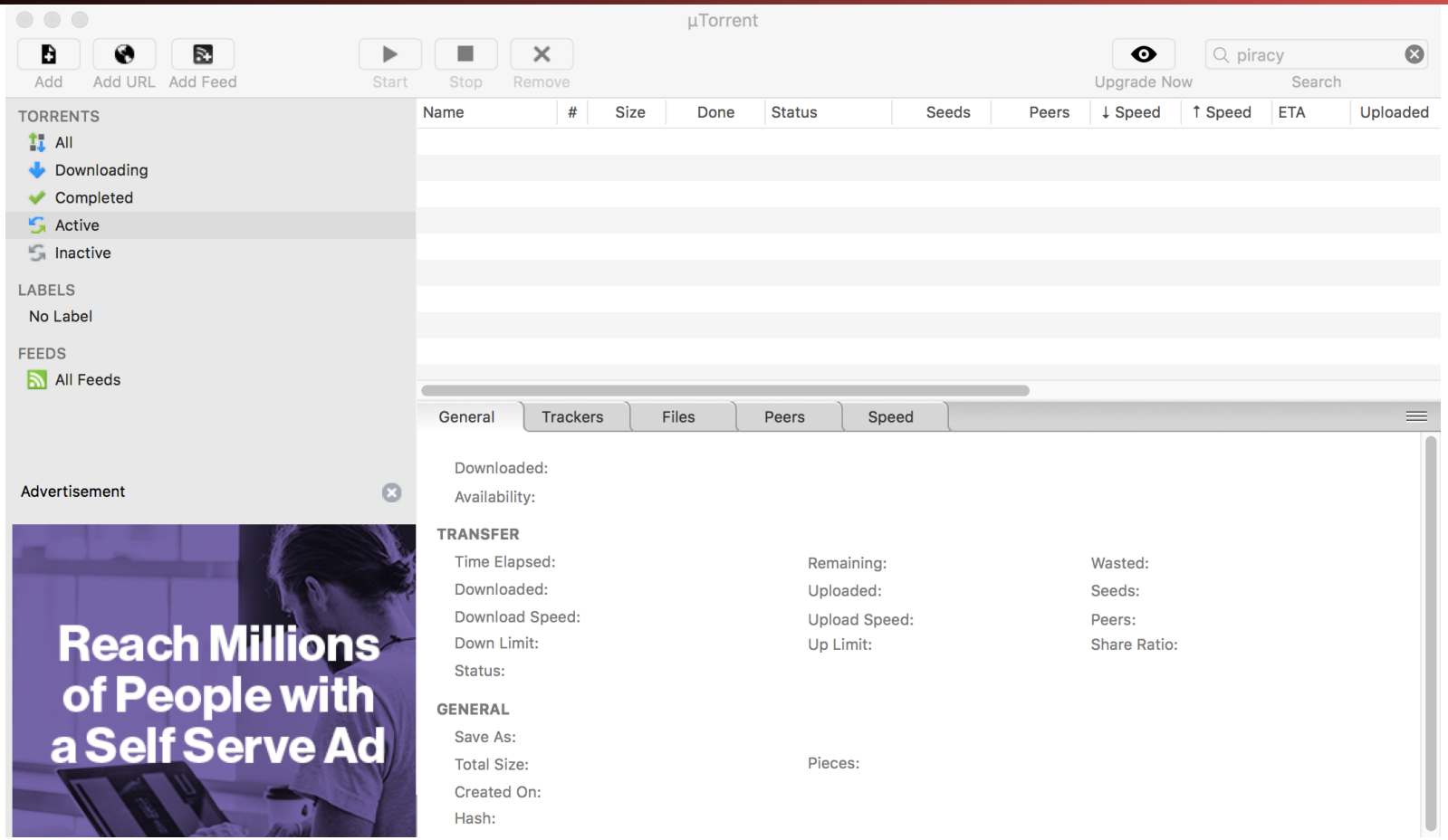
Subject to your compliance with these terms and conditions, BitTorrent, Inc. grants you a royalty-free, non-exclusive, non-transferable license to use µTorrent, solely for your personal, non-commercial purposes. BitTorrent, Inc. reserves all rights in µTorrent not expressly granted to you here.

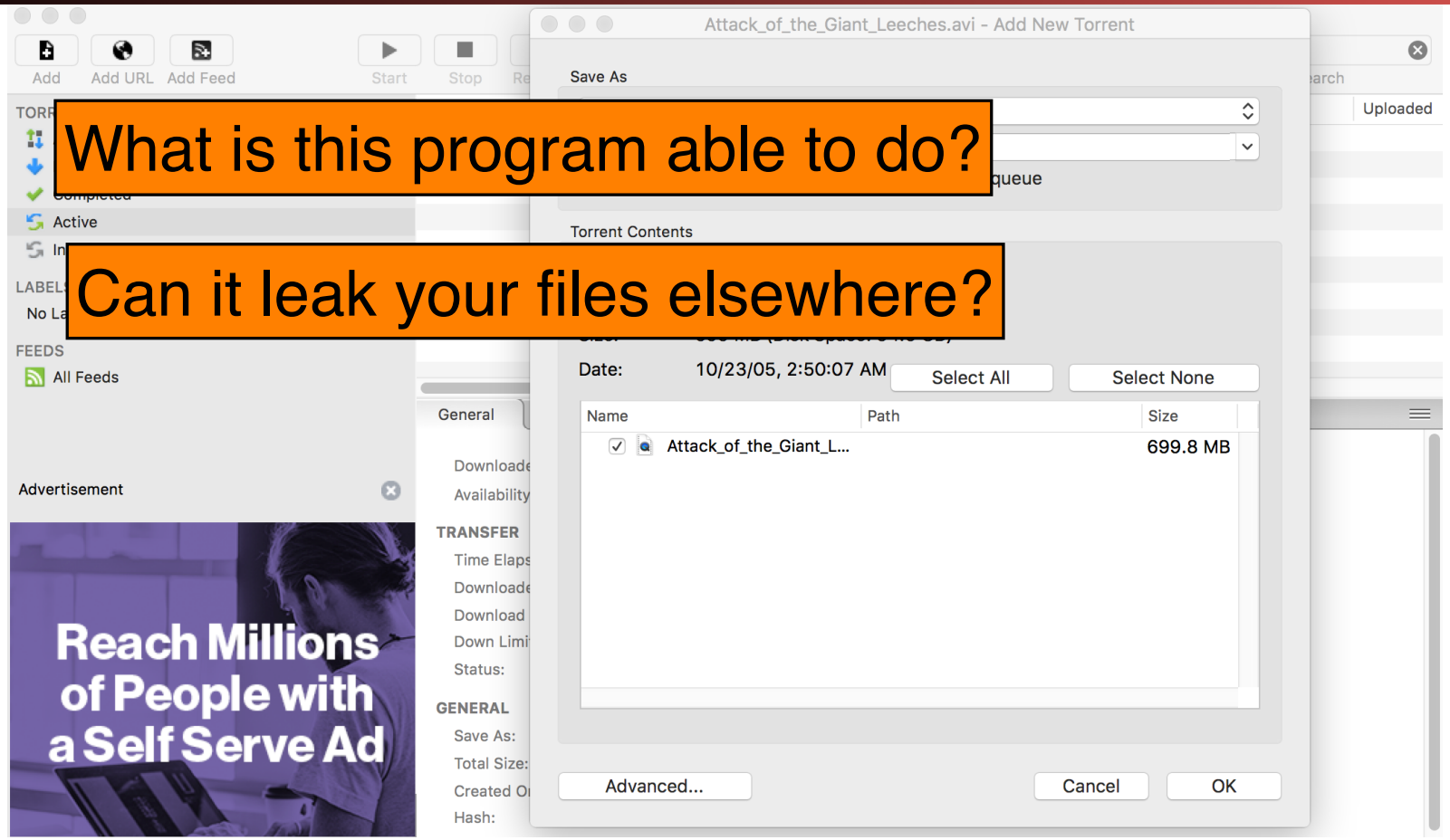
Restrictions.

The source code, design, and structure of µTorrent are trade secrets. You

Print Save... Disagree Agree

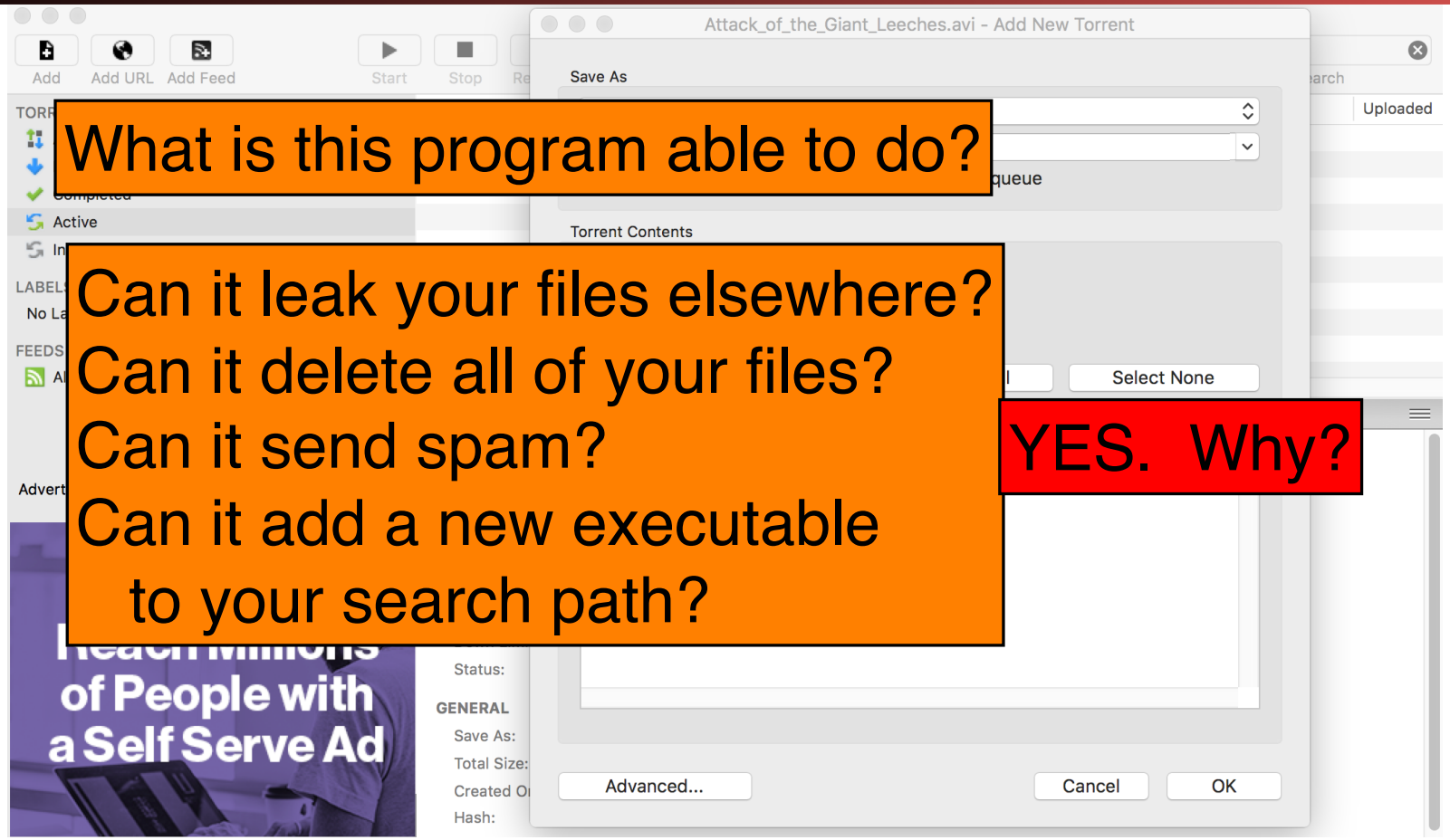

uTorrent
2 items
uTorrent
Applications
LIGHT. LIMITLESS. ENGINEERED FOR
POWERFUL DOWNLOADING.


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:
Advertisement
Reach Millions of People with a Self Serve Ad



What is this program able to do?

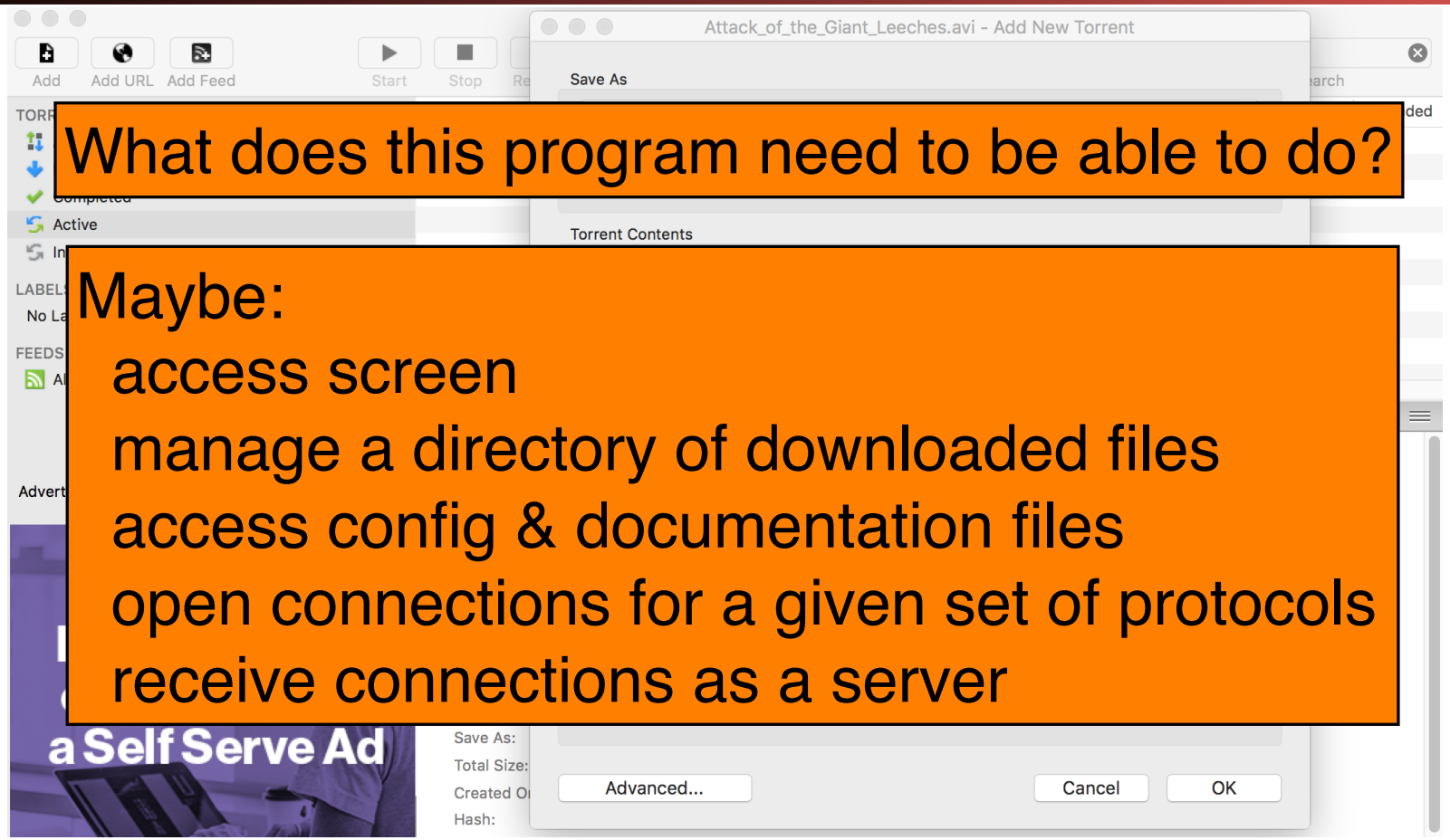
Can it leak your files elsewhere?



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

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?

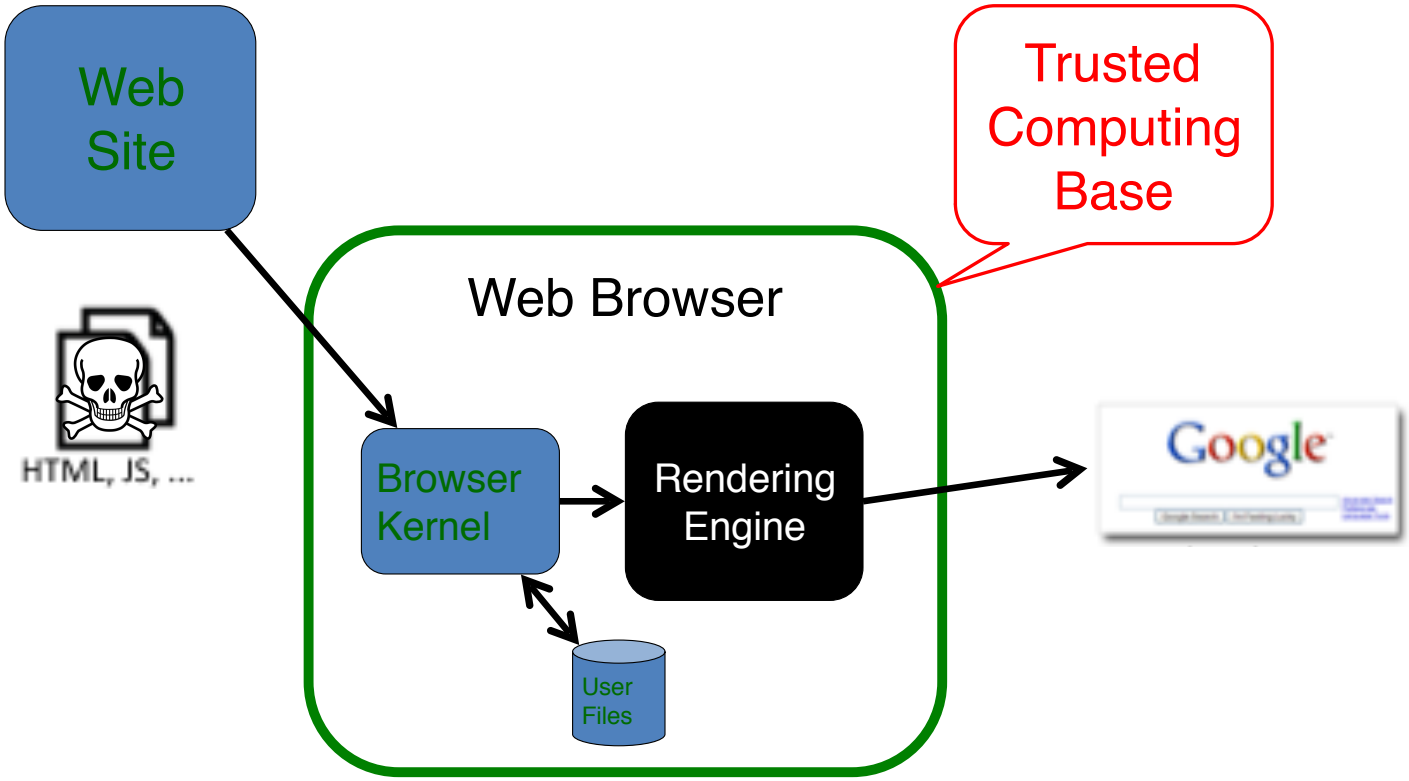
Thinking About Least Privilege

- 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...

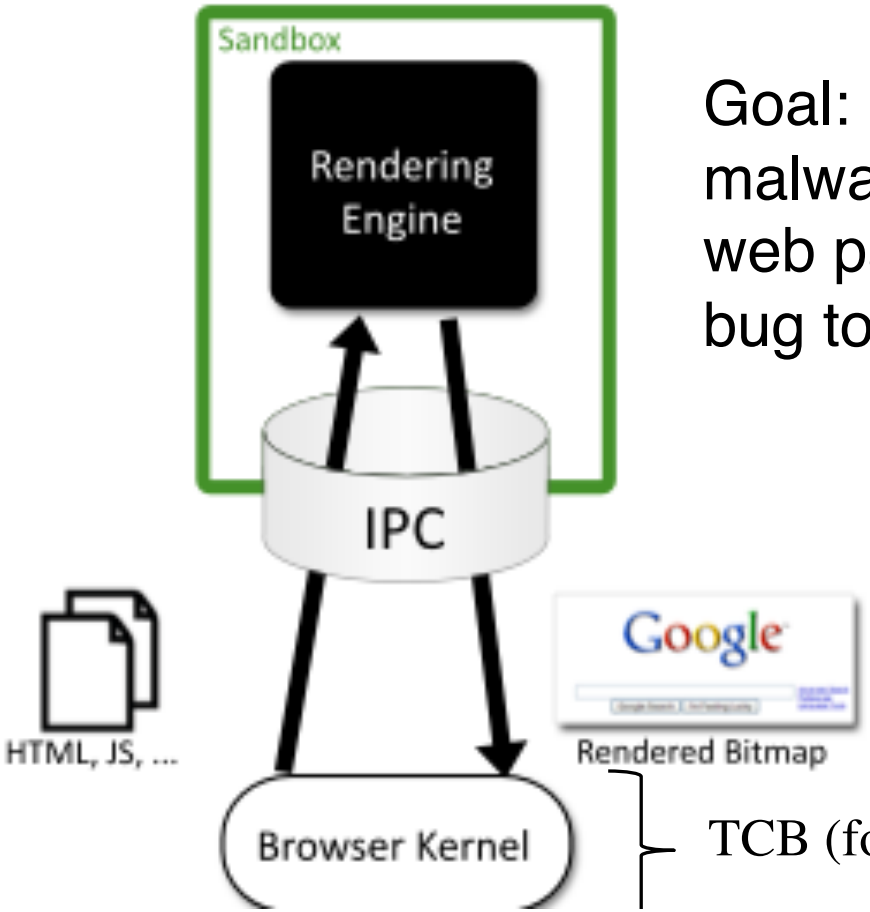
- The operating system ***process*** provide the following "guarentees" (you hope)
  - Isolation: A process can not access (read OR write) the memory of any other process
  - Permissions: A process can only change files etc if it has permission to
    - This ***usually*** means "Anything that the user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

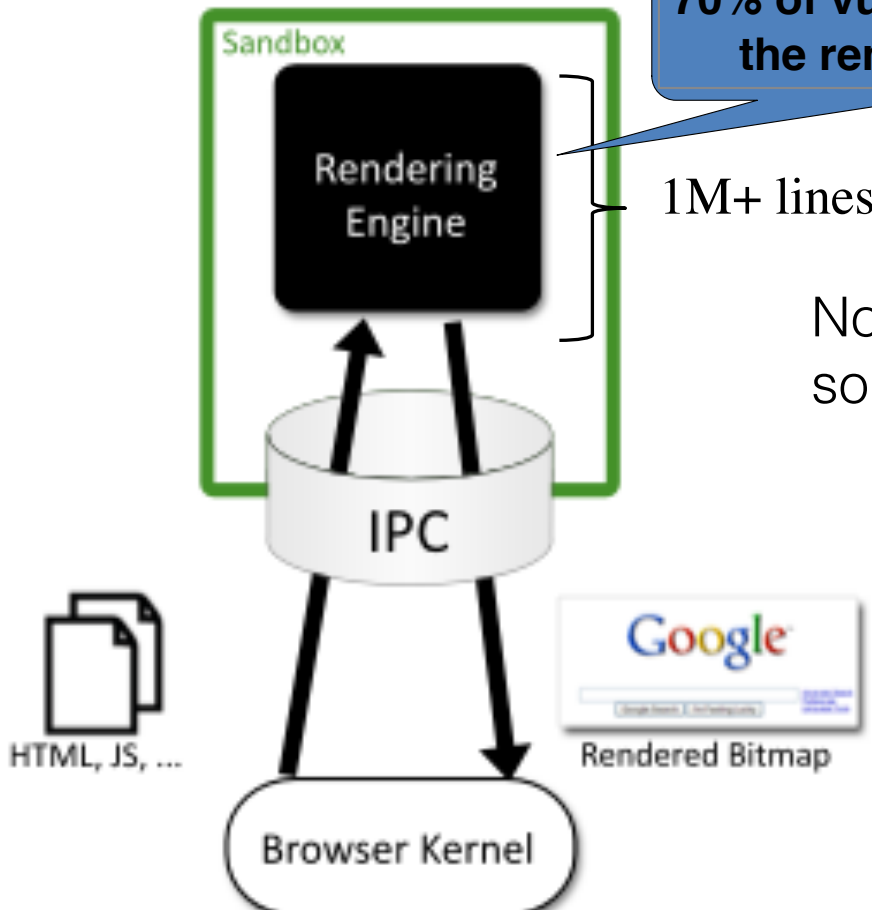


Goal: prevent "drive-by malware", where a malicious web page exploits a browser bug to infect local files

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

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...