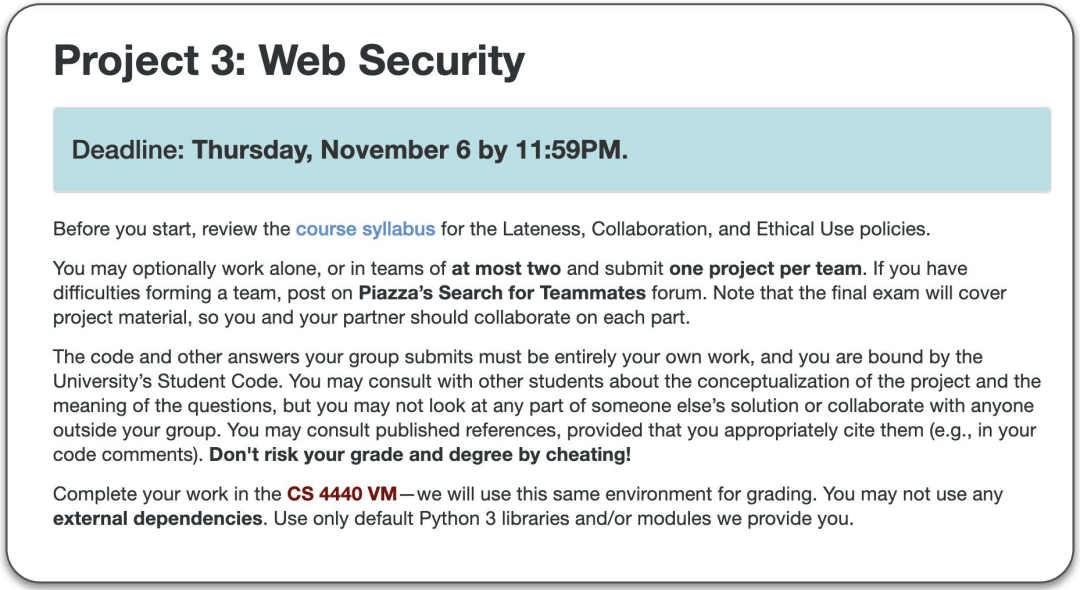
Week 11: Lecture B

Security in Practice: Tor

Thursday, November 6, 2025

Announcements

- **Project 3: WebSec** released
  - **Deadline: tonight** by 11:59PM



Project 3: Web Security

Deadline: Thursday, November 6 by 11:59PM.

Before you start, review the course syllabus for the Lateness, Collaboration, and Ethical Use policies.

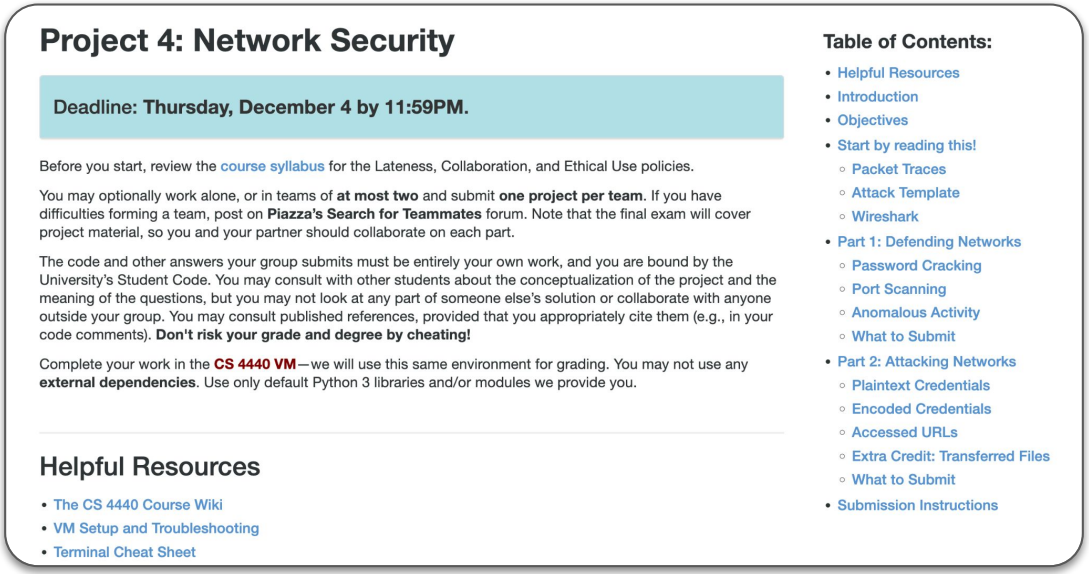
You may optionally work alone, or in teams of **at most two** and submit **one project per team**. If you have difficulties forming a team, post on **Piazza's Search for Teammates** forum. Note that the final exam will cover project material, so you and your partner should collaborate on each part.

The code and other answers your group submits must be entirely your own work, and you are bound by the University's Student Code. You may consult with other students about the conceptualization of the project and the meaning of the questions, but you may not look at any part of someone else's solution or collaborate with anyone outside your group. You may consult published references, provided that you appropriately cite them (e.g., in your code comments). **Don't risk your grade and degree by cheating!**

Complete your work in the **CS 4440 VM**—we will use this same environment for grading. You may not use any **external dependencies**. Use only default Python 3 libraries and/or modules we provide you.

Announcements

- **Project 4: NetSec** released
  - **Deadline:** Thursday, December 4th by 11:59PM



Project 4: Network Security

Deadline: Thursday, December 4 by 11:59PM.

Before you start, review the course syllabus for the Lateness, Collaboration, and Ethical Use policies.

You may optionally work alone, or in teams of **at most two** and submit **one project per team**. If you have difficulties forming a team, post on **Piazza's Search for Teammates** forum. Note that the final exam will cover project material, so you and your partner should collaborate on each part.

The code and other answers your group submits must be entirely your own work, and you are bound by the University's Student Code. You may consult with other students about the conceptualization of the project and the meaning of the questions, but you may not look at any part of someone else's solution or collaborate with anyone outside your group. You may consult published references, provided that you appropriately cite them (e.g., in your code comments). **Don't risk your grade and degree by cheating!**

Complete your work in the **CS 4440 VM**—we will use this same environment for grading. You may not use any **external dependencies**. Use only default Python 3 libraries and/or modules we provide you.

Helpful Resources

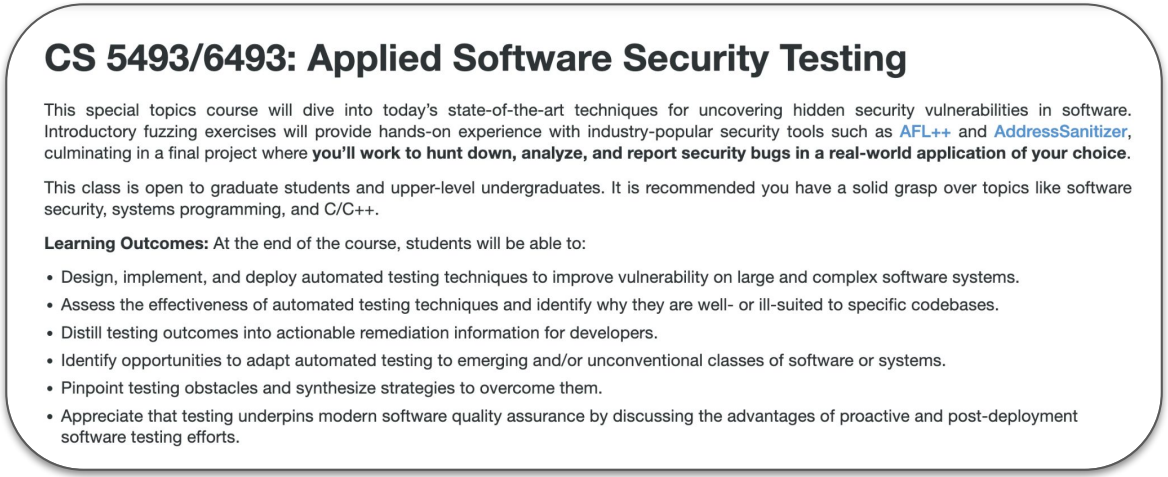
- The CS 4440 Course Wiki
- VM Setup and Troubleshooting
- Terminal Cheat Sheet

Table of Contents:

- Helpful Resources
- Introduction
- Objectives
- Start by reading this!
  - Packet Traces
  - Attack Template
  - Wireshark
- Part 1: Defending Networks
  - Password Cracking
  - Port Scanning
  - Anomalous Activity
  - What to Submit
- Part 2: Attacking Networks
  - Plaintext Credentials
  - Encoded Credentials
  - Accessed URLs
  - Extra Credit: Transferred Files
  - What to Submit
- Submission Instructions

Interested in fuzzing?

- **Spring 2026: CS 5493/6493: Applied Software Security Testing**
  - **Everything you'd ever want to know about fuzzing for finding security bugs!**
  - Course project: team up to fuzz **a real program** (of your choice), and find and report its bugs!
  - http://cs.utah.edu/~snagy/courses/cs5493/



CS 5493/6493: Applied Software Security Testing

This special topics course will dive into today's state-of-the-art techniques for uncovering hidden security vulnerabilities in software. Introductory fuzzing exercises will provide hands-on experience with industry-popular security tools such as AFL++ and AddressSanitizer, culminating in a final project where **you'll work to hunt down, analyze, and report security bugs in a real-world application of your choice**.

This class is open to graduate students and upper-level undergraduates. It is recommended you have a solid grasp over topics like software security, systems programming, and C/C++.

Learning Outcomes: At the end of the course, students will be able to:

- Design, implement, and deploy automated testing techniques to improve vulnerability on large and complex software systems.
- Assess the effectiveness of automated testing techniques and identify why they are well- or ill-suited to specific codebases.
- Distill testing outcomes into actionable remediation information for developers.
- Identify opportunities to adapt automated testing to emerging and/or unconventional classes of software or systems.
- Pinpoint testing obstacles and synthesize strategies to overcome them.
- Appreciate that testing underpins modern software quality assurance by discussing the advantages of proactive and post-deployment software testing efforts.

Kahlert School of Computing

Graduate Program Open House

Why PhD?

Information session for prospective graduate students

- What to expect from graduate school
- Reasons to pursue graduate career
- Perspective of alumni and current students
- How to prepare your application (and a statement of purpose)

RSVP / Zoom links:

November 14, 3:00pm – 5:00pm

MEB 3147 (LCR) and Zoom (free pizza—please RSVP)

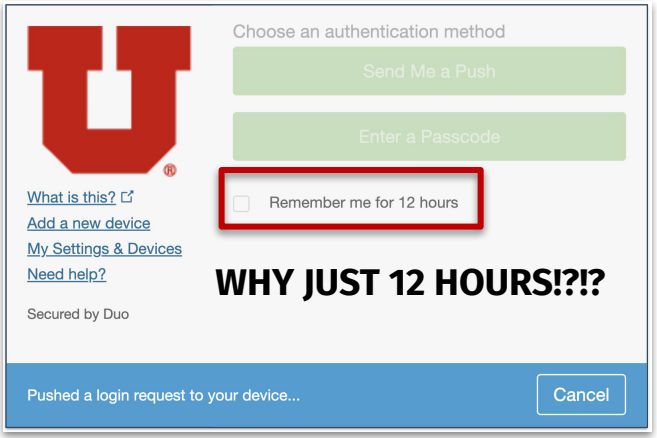
Questions?

Last time on CS 4440...

Authentication
Multi-factor Authentication
One-time Passwords
Secure Password Storage

What is authentication?

- **What is it?**
  - That password you re-use for every website
  - An ever-changing set of rules to frustrate you
  - The most annoying thing about attending UofU



What is authentication?

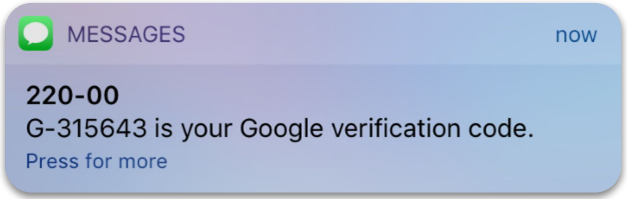
- **Goal: ???**
- **Problem: ???**
- **Challenge: ???**

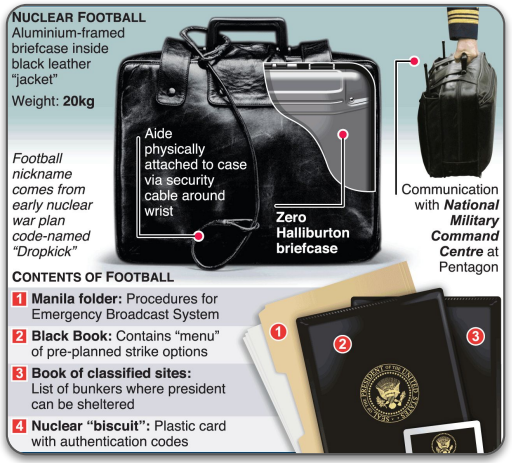
What is authentication?

- **Goal:** establish trust in the **identity** of another communicating party
- **Problem:** **cannot directly interact** with them to verify their identity
- **Challenge:** how can someone prove they are **who they say they are**?

The Three Factors of Authentication

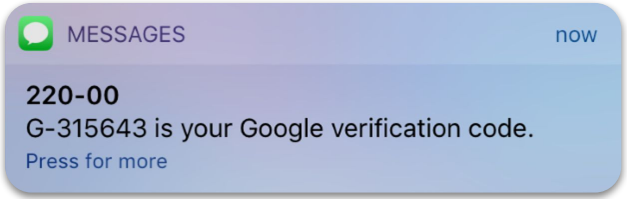
- Something you ???
- Something you ???
- Something you ???

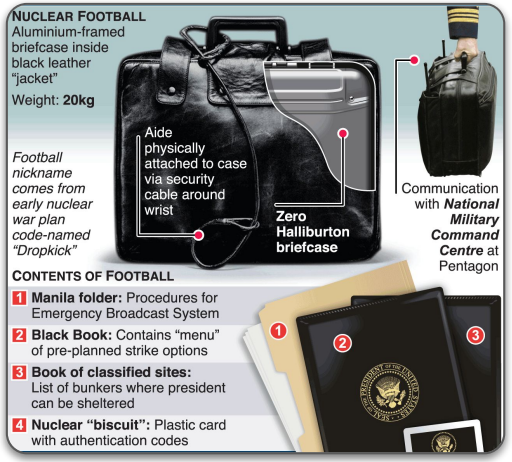




The Three Factors of Authentication

- Something you **have**
- Something you **are**
- Something you **know**



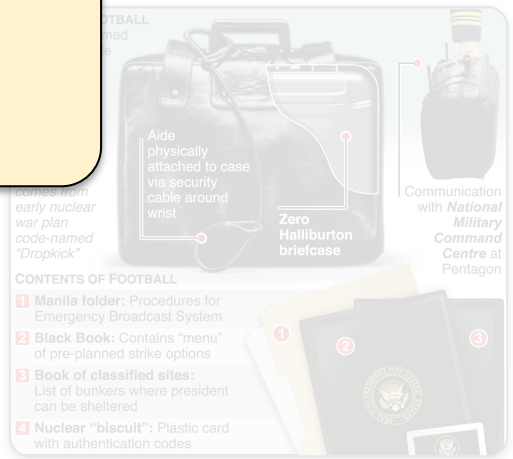


The Three Factors of Authentication

- Something you **have**
- Something you **are**
- Something you **know**

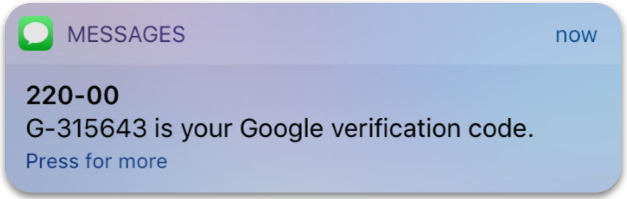
Examples?

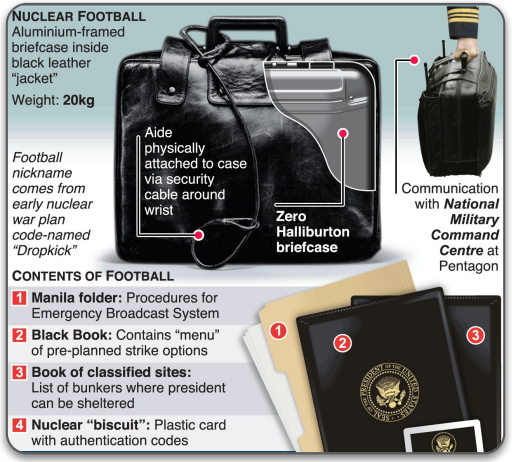
(**in-class poll**)



The Three Factors of Authentication

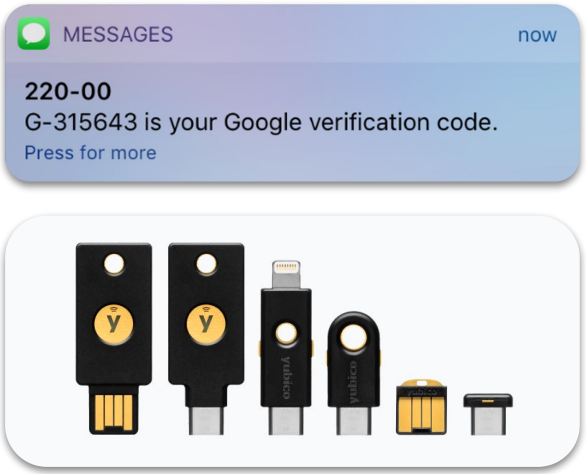
- Something you **have**
  - Smartphone
  - Laptop
  - Email account
- Something you **are**
  - Your fingerprint
  - Your DNA
  - Your iris, retina
- Something you **know**
  - Account password, banking PIN number
  - Nuclear strike challenge-response code

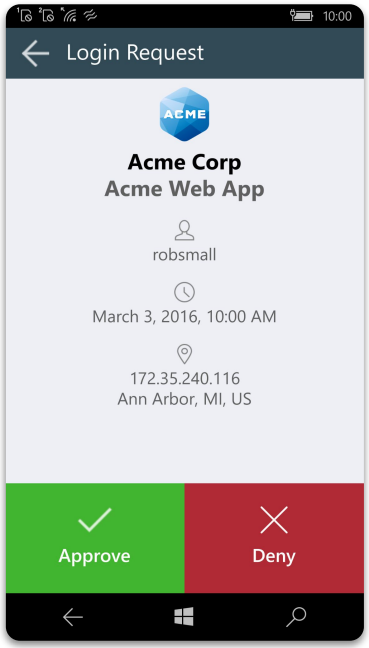




One-time PINs

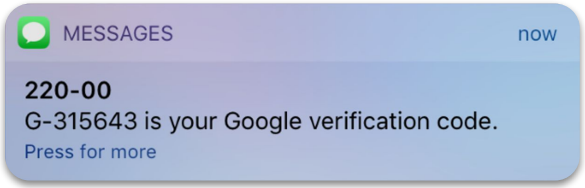
- **Provides proof of: ???**

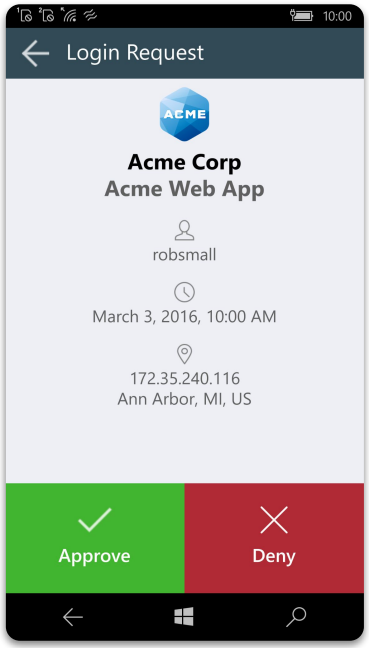




One-time PINs

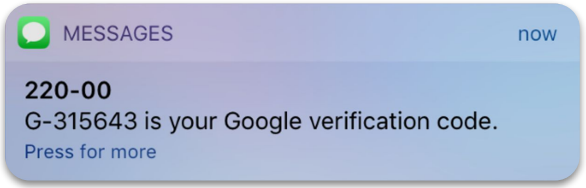
- **Provides proof of: possession**
  - A PIN/code valid for only **one** login session or transaction
- **Delivering** One-time PINs:
  - **???**

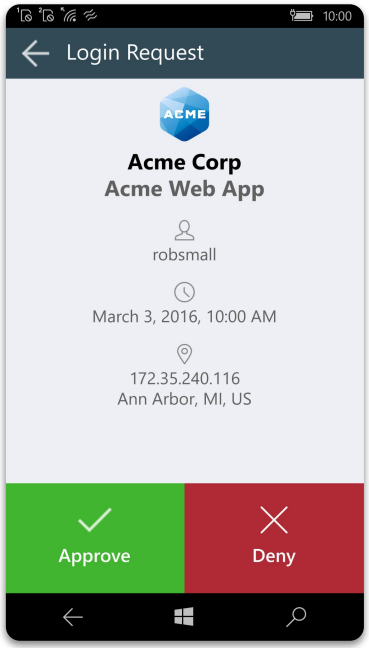




One-time PINs

- **Provides proof of: possession**
  - A PIN/code valid for only **one** login session or transaction
- **Delivering** One-time PINs:
  - **SMS**
    - Phone call
    - Text message
  - **Hardware**
    - Yubico YubiKey
    - RSA SecureID
  - **Application**
    - DUO Mobile
    - Google authenticator



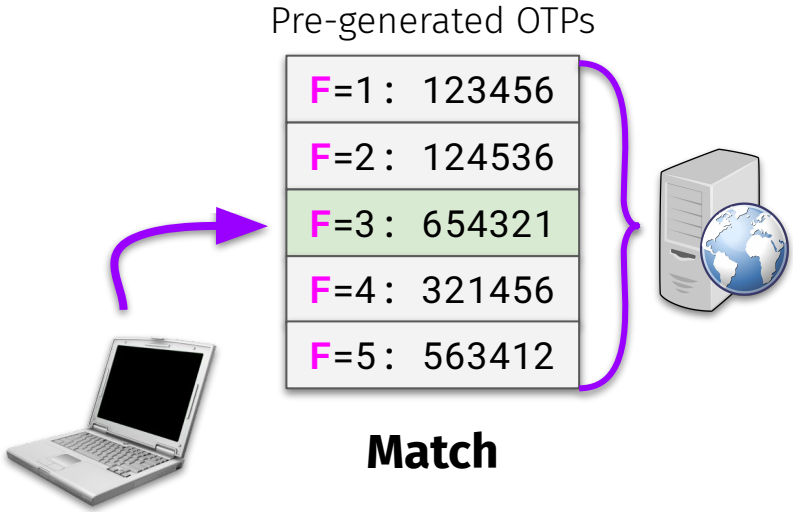


Implementing OTPs

- **Better idea:** independently generate OTP codes based on a **moving factor**
  - E.g., intervals of **time**, unique session **count**, etc.

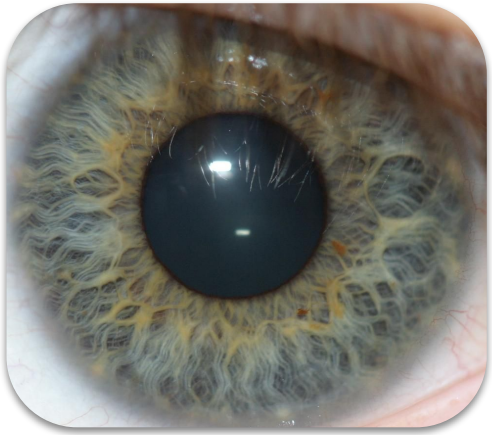
- **Common OTP protocols:**
  - HMAC-based OTP (**HOTP**)
    - Use **session count** as factor
  - Time-based OTP (**TOTP**)
    - Use **time interval** as factor

- **Problem: desynchronization**
  - E.g., user hits "login" one too many times
  - **Solution:** make a few OTPs; user matches once



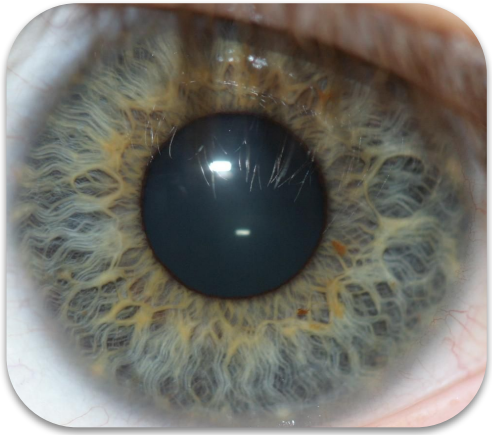
Biometrics

- **Provides proof of ???**



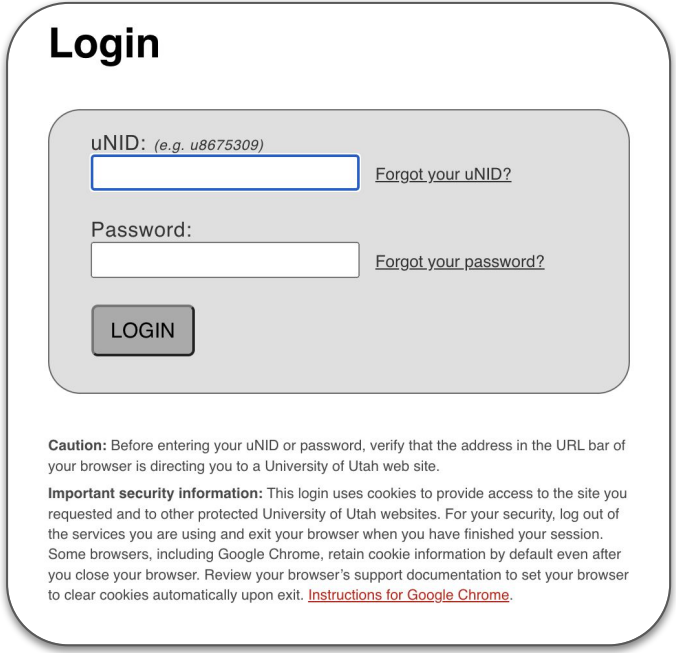
Biometrics

- **Provides proof of physical identity**
- **Something unique to you** (hopefully)
  - Fingerprint, iris, retina, DNA
- Security = **unlikely match probability**
  - Fingerprint match chance: **1 in $64 * 10^{13}$**
  - Iris pattern match chance: **1 in 10^{78}**



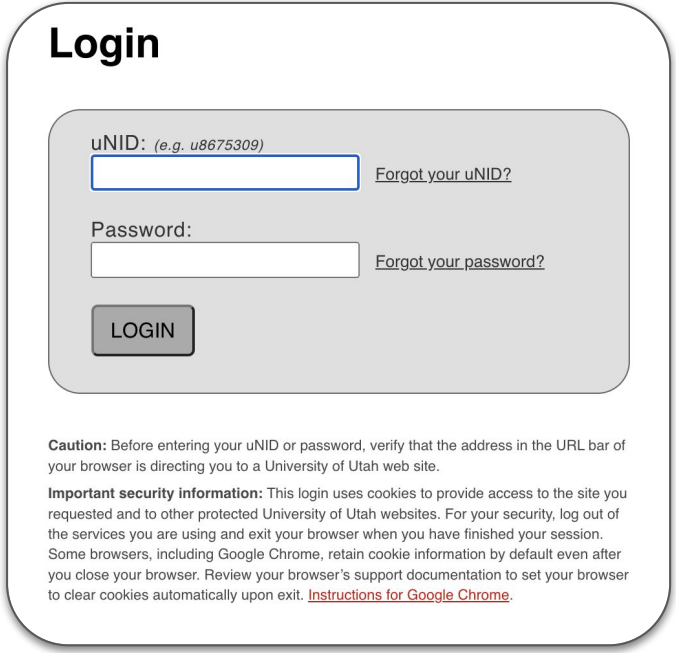
Passwords

- **Proof of something you ???**



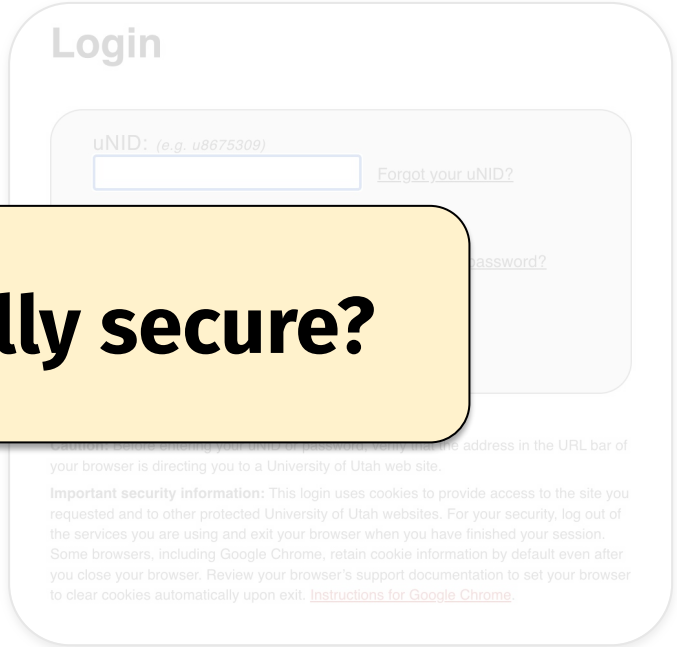
Passwords

- **Proof of something you know**
  - Something that you forget?
- A **secret** string of data that confirms a user's identity
  - **Letters** (ABCDEFGH)
  - **Digits** (0123456789)
  - **Other symbols** ($#%-_!)



Passwords

- **Proof of something you know**
  - Something that you forget?
- A **secret** string of data that confirms a
  - Letters
  - Digits
  - Other sy



Cryptographically secure?

Passwords

- **Proof of something you know**
  - Something that you forget?
- A **secret** string of data that confirms a user's identity
  - **Letters** (ABCDEFGH)
  - **Digits** (0123456789)
  - **Other symbols** ($#%-_!)
- **Cryptographically secure?**
  - **Not at all!**

Password Attacks

- **Passwords stored in plaintext**
  - ???
- **Passwords that are reused**
  - ???
- **Passwords that aren't random**
  - ???
- **Device-issued default passwords**
  - ???

Username	Password
666666	666666
888888	888888
admin	*(none)*
admin	1111
admin	1111111
admin	1234
admin	12345
admin	123456
admin	54321
admin	7ujMko0admin
admin	admin

1 in 3 U.S. Pet Parents Have Used Their Pet's Name as Their Password

Password Attacks

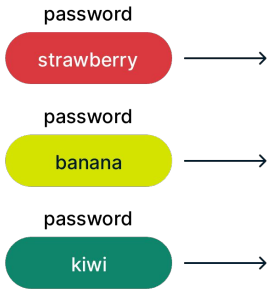
- **Passwords stored in plaintext**
  - Easily stolen if attacker breaches DB
- **Passwords that are reused**
  - Only takes one plaintext breach
- **Passwords that aren't random**
  - Easily guessable via info about you
- **Device-issued default passwords**
  - Attacker can make one big dictionary

Username	Password
666666	666666
888888	888888
admin	*(none)*
admin	1111
admin	1111111
admin	1234
admin	12345
admin	123456
admin	54321
admin	7ujMko0admin
admin	admin

1 in 3 U.S. Pet Parents Have Used Their Pet's Name as Their Password

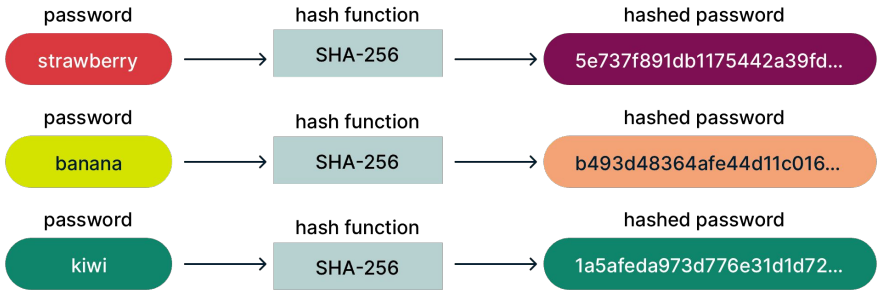
Better Server-side Password Storage

- **Hashing passwords:** increases security by **???**



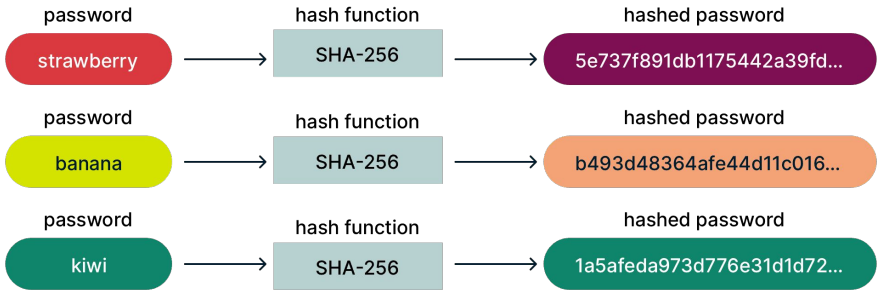
Better Server-side Password Storage

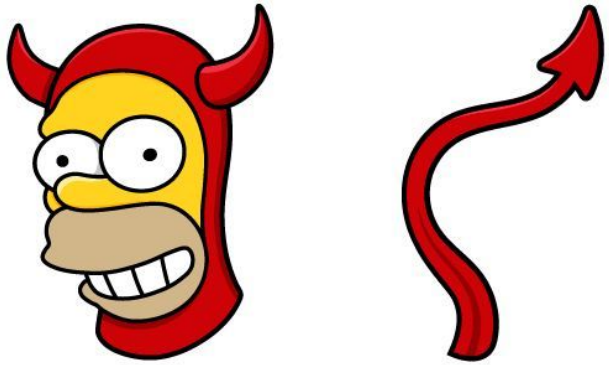
- **Hashing passwords:** increases security by **obfuscating passwords**
- Why are **weak** hash functions bad?
  - ???
- Why are **fast** hash functions bad?
  - ???



Better Server-side Password Storage

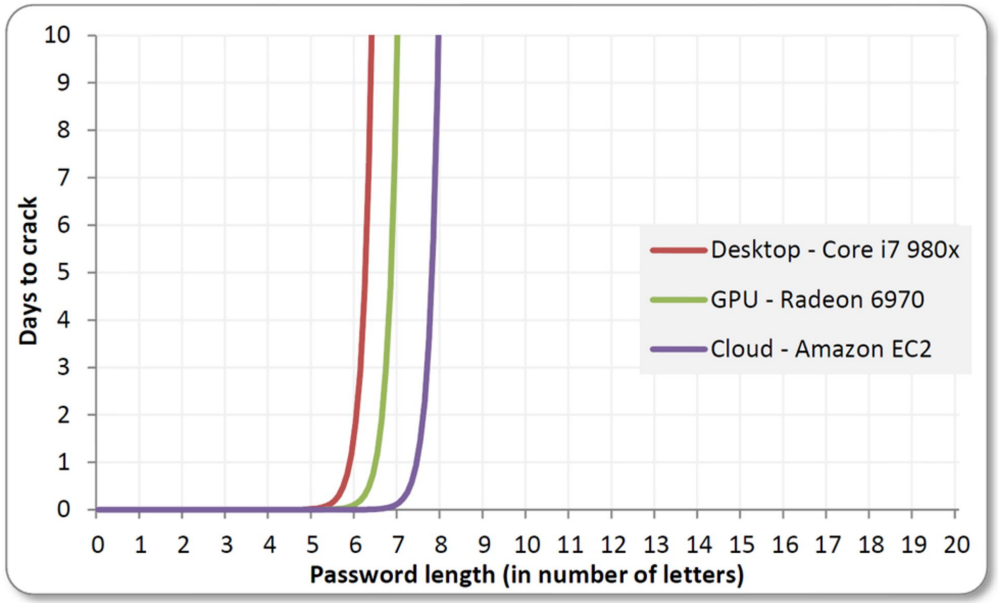
- **Hashing passwords:** increases security by **obfuscating passwords**
- Why are **weak** hash functions bad?
  - **Collision and pre-image attacks** = attacker easily finds working password
- Why are **fast** hash functions bad?
  - **Rainbow table attack** = attacker an efficiently pre-generate nearly all `(password, hash)` pairs





Attack: Password Cracking

- Assume attacker knows hash function and wants to **find a single password**
  - Rapidly **becoming more doable** with advances in hardware!



Better Server-side Password Storage

- **Slower hash functions**
  - **Why?**

Better Server-side Password Storage

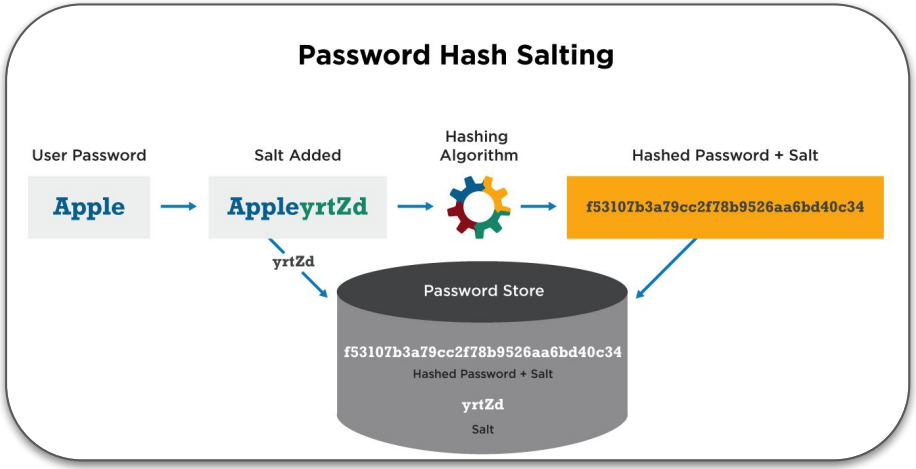
- **Slower hash functions**
  - Makes rainbow table generation **more computationally expensive** for attackers!
  - E.g., **Bcrypt, Scrypt**—perform multiple rounds of hashing (**much slower**)

- **Salted passwords:**
  - **Why?**

Better Server-side Password Storage

- **Slower hash functions**
  - Makes rainbow table generation **more computationally expensive** for attackers!
  - E.g., **Bcrypt, Scrypt**—perform multiple rounds of hashing (**much slower**)

- **Salted passwords:**
  - Add **extra data** when generating hash
  - **Goal:** same input = different output
  - Salting considerations:
    - **???**



Is "password || c$#4440!" a secure salt?

Yes!

0%

No!

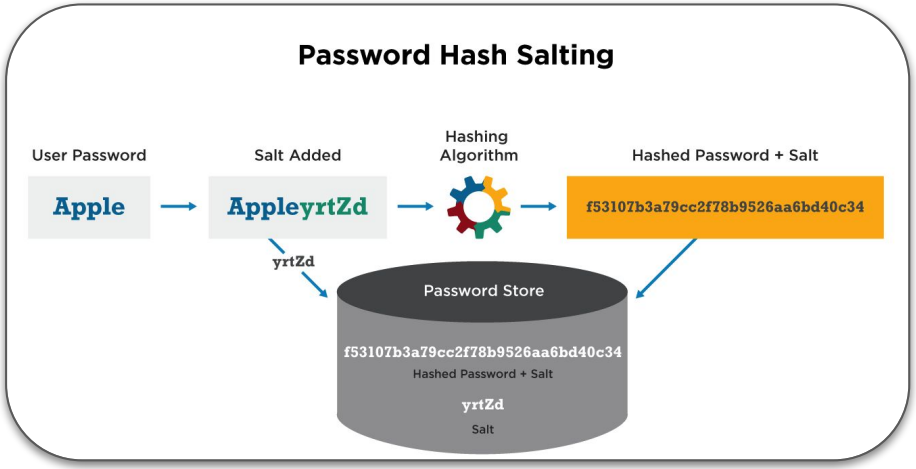
0%

Better Server-side Password Storage

- **Slower hash functions**
  - Makes rainbow table generation **more computationally expensive** for attackers!
  - E.g., **Bcrypt, Scrypt**—perform multiple rounds of hashing (**much slower**)

- **Salted passwords:**
  - Add **extra data** when generating hash
  - **Goal:** same input = different output
  - Salting considerations:
    - Salt should **not be short**
    - Should be **unique** per user

- **Better: salting + slow hashing!**



Attack: Client-side Password Theft

- **How?**

Attack: Client-side Password Theft

- **How?**
  - Keyloggers, unencrypted transit, phishing, angry ex-partner

```
Hypertext Transfer Protocol
› GET /libs/qimessaging/1.0/qimessaging.js?v=1.2.0 HTTP/1.1\r\n
  Host: 10.0.0.6\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
  Accept: */*\r\n
  Accept-Language: en-US,en;q=0.5\r\n
  Accept-Encoding: gzip, deflate\r\n
  Referer: http://10.0.0.6/\r\n
  Connection: keep-alive\r\n
▾ Authorization: Basic bmFvOmNhcmVzc2VzLTIwMDE=\r\n
    Credentials: nao:[redacted]
  \r\n
```

From: Bank of America <crvdgi@comcast.net>
Subject: Notification Irregular Activity
Date: September 23, 2014 3:44:42 PM PDT
To: Undisclosed recipients: ;
Reply-To: crvdgi@comcast.net

Bank of America

Online Banking Alert

Would be capitalized

Dear member:

We detected unusual activity on your Bank of America debit card on **09/22/2014**.
For your protection, please verify this activity so you can continue making debit card transactions without interruption.

Please sign in to your account **at** https://www.bankofamerica.com

to review and verify your account activity, After verifying your debit card transactions we will take the necessary steps to protect your account from fraud.

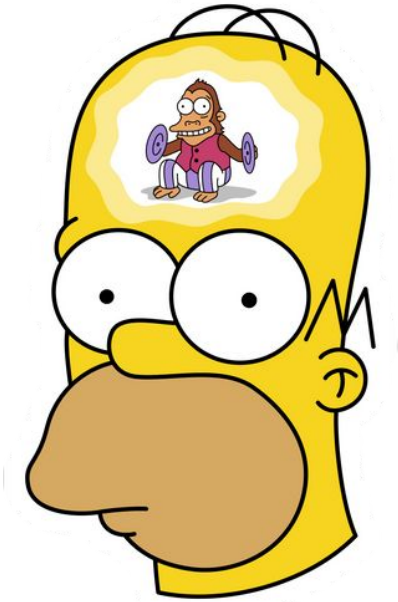
http://bit.do/ghsdfhgsd

If you do not contact us, certain limitations may be placed on your debit card.

Grammatical Error

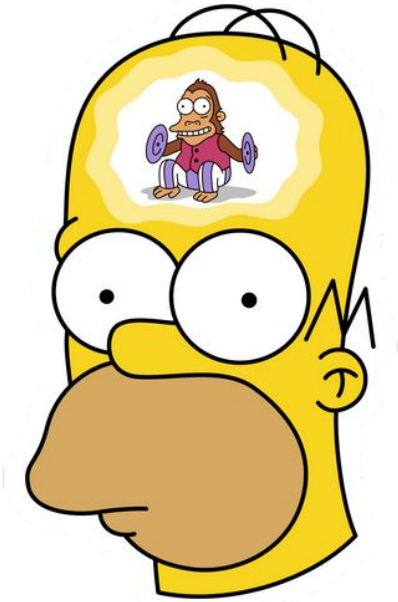
Forgetting & Recovering Passwords

- **Drawbacks of these mechanisms?**
  - Email plaintext password:
    - ???
  - Password recovery email:
    - ???
  - Fall-back security questions:
    - ???



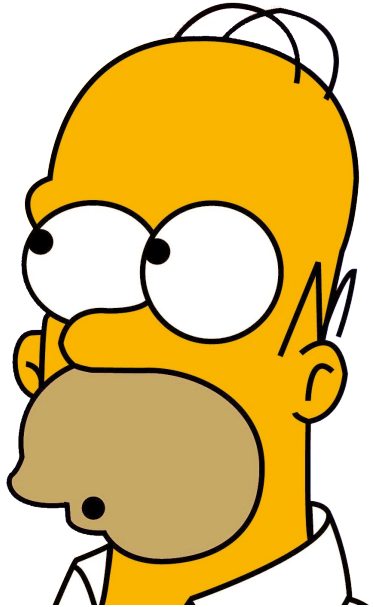
Forgetting & Recovering Passwords

- **Drawbacks of these mechanisms?**
  - Email plaintext password:
    - **Assume email compromised!**
  - Password recovery email:
    - **Same as above!**
  - Fall-back security questions:
    - **Hopefully not obvious!**
- **Better approaches:**
  - Deliver session-specific PIN via SMS or call
    - Assume **device harder to compromise**



Authentication trade-offs / challenges?

- **Replay attacks**
  - ???
- **Poisoning attacks**
  - ???
- **Noisy sensors**
  - ???
- **Change / loss of biometric**
  - ???



Authentication trade-offs / challenges?

- **Replay attacks**
  - Spoofs an enrolled user
- **Poisoning attacks**
  - Alter enrollment template
  - Alter one user's enrollment
- **Noisy sensors**
  - Gives attackers "leeway" in crafting adversarial inputs
- **Change / loss of biometric**
  - **Change:** cataracts surgery
  - **Loss:** losing your finger

After an initial analysis, the Indian and American scientists used three iris sensors and two commercial iris biometric matchers to check if the new irises passed biometric authentication. They found that the iris sensors' success rate dropped to 75% after surgery. The biometric matchers did better authenticating 93% of the irises.

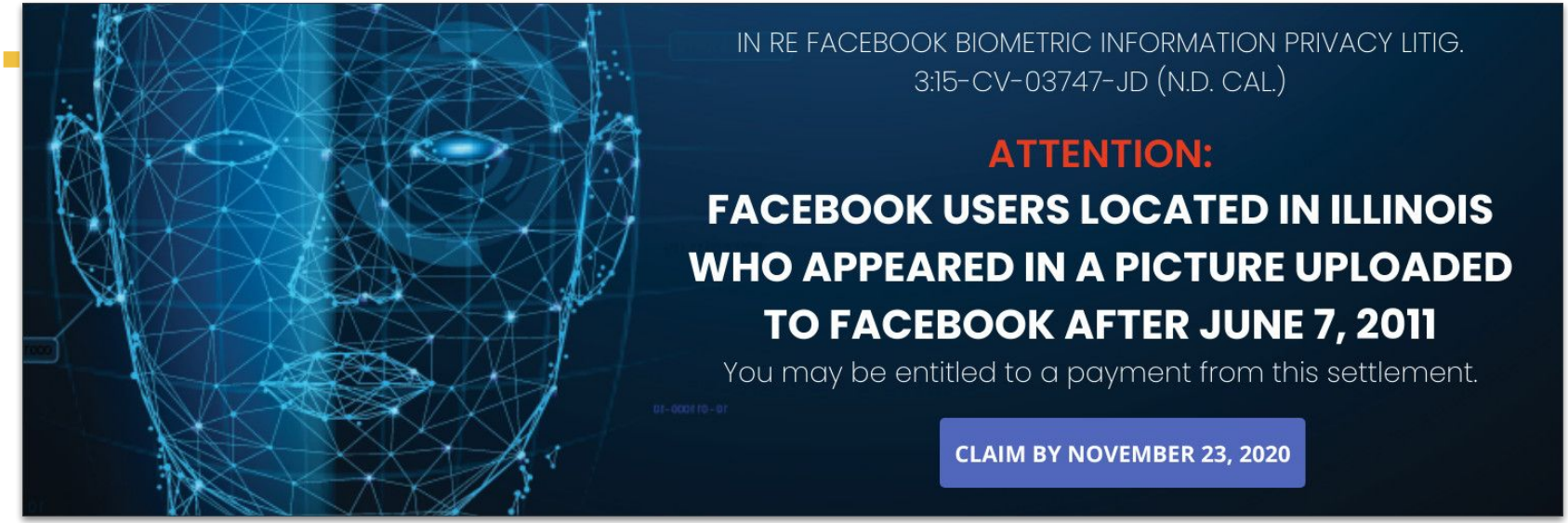
Crane horror *Reg* reader uses his severed finger to unlock Samsung Galaxy phone

On the other hand he was fine

Authentication trade-offs / challenges?

- Are biometrics **ethical**?

Authentication trade-offs / challenges?



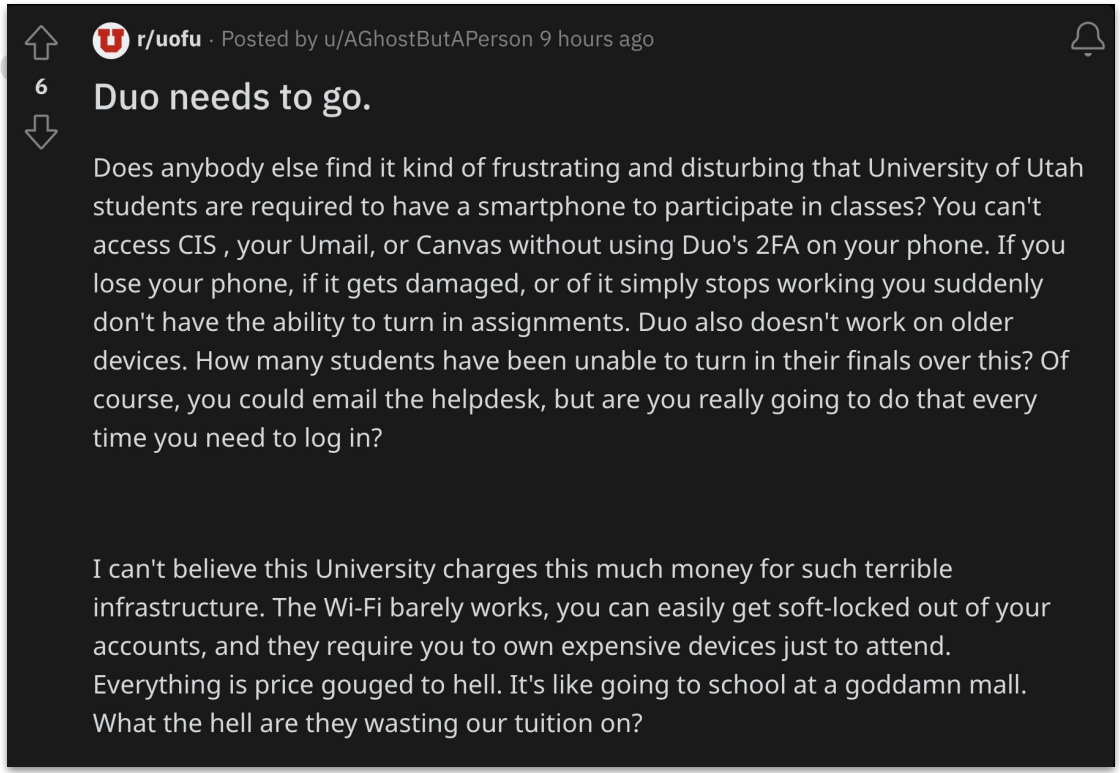
Facebook, Inc. has settled a class action that claimed Facebook collected and stored the biometric data of Facebook users in Illinois without the proper notice and consent in violation of Illinois law as part of its "Tag Suggestions" feature and other features involving facial recognition technology. Facebook denies it violated any law.

Authentication trade-offs / challenges?

- Is **convenience** a concern?

Authentication trade-offs / challenges?

- Is convenie



r/uofu · Posted by u/AGhostButAPerson 9 hours ago

6

Duo needs to go.

Does anybody else find it kind of frustrating and disturbing that University of Utah students are required to have a smartphone to participate in classes? You can't access CIS , your Umail, or Canvas without using Duo's 2FA on your phone. If you lose your phone, if it gets damaged, or of it simply stops working you suddenly don't have the ability to turn in assignments. Duo also doesn't work on older devices. How many students have been unable to turn in their finals over this? Of course, you could email the helpdesk, but are you really going to do that every time you need to log in?

I can't believe this University charges this much money for such terrible infrastructure. The Wi-Fi barely works, you can easily get soft-locked out of your accounts, and they require you to own expensive devices just to attend. Everything is price gouged to hell. It's like going to school at a goddamn mall. What the hell are they wasting our tuition on?

Authentication trade-offs / challenges?

- Whose **responsibility** is password security?

Authentication trade-offs / challenges?

GoDaddy Breached – Plaintext Passwords – 1.2M Affected

There is an update available here: GoDaddy Breach Widens to tsoHost, Media Temple, 123Reg, Domain Factory, Heart Internet, and Host Europe

This morning, GoDaddy disclosed that an unknown attacker had gained unauthorized access to the system used to provision the company's Managed WordPress sites, impacting up to 1.2 million of their WordPress customers. Note that this number does not include the number of custom
some GoDaddy customers hav

Facebook Stored Hundreds of Millions of User Passwords in Plain Text for Years

March 21, 2019

Hundreds of millions of **Facebook** users had their account passwords stored in plain text and searchable by thousands of Facebook employees — in some cases going back to 2012, KrebsOnSecurity has
ion has so far found no
s to this data.

Why Was Equifax So Stupid About Passwords?

Massive Credit Bureau Stored Users' Plaintext Passwords in Testing Environment

Mathew J. Schwartz (euroinfosec) · September 24, 2018

Authentication trade-offs / challenges?

Password was ‘Louvre’: weak surveillance code exposed after $102 million Louvre heist

The revelation comes as French lawmakers press the museum’s leadership for answers over how one of the world’s most secure cultural landmarks was breached so easily.

By: EXPRESS WEB DESK

New Delhi , November 6, 2025 08:31 PM IST

Share

Comments

Print

As Trusted Source on

A police car parks in the courtyard of the Louvre museum, one week after the robbery. (AP Photo/Thomas Padilla)

When thieves broke into the Louvre in Paris last month and made off with $102 million worth of jewels, they didn’t just expose a hole in France’s most famous museum, they laid bare a shocking lapse in its digital defences.

TODAY’S EPAPER

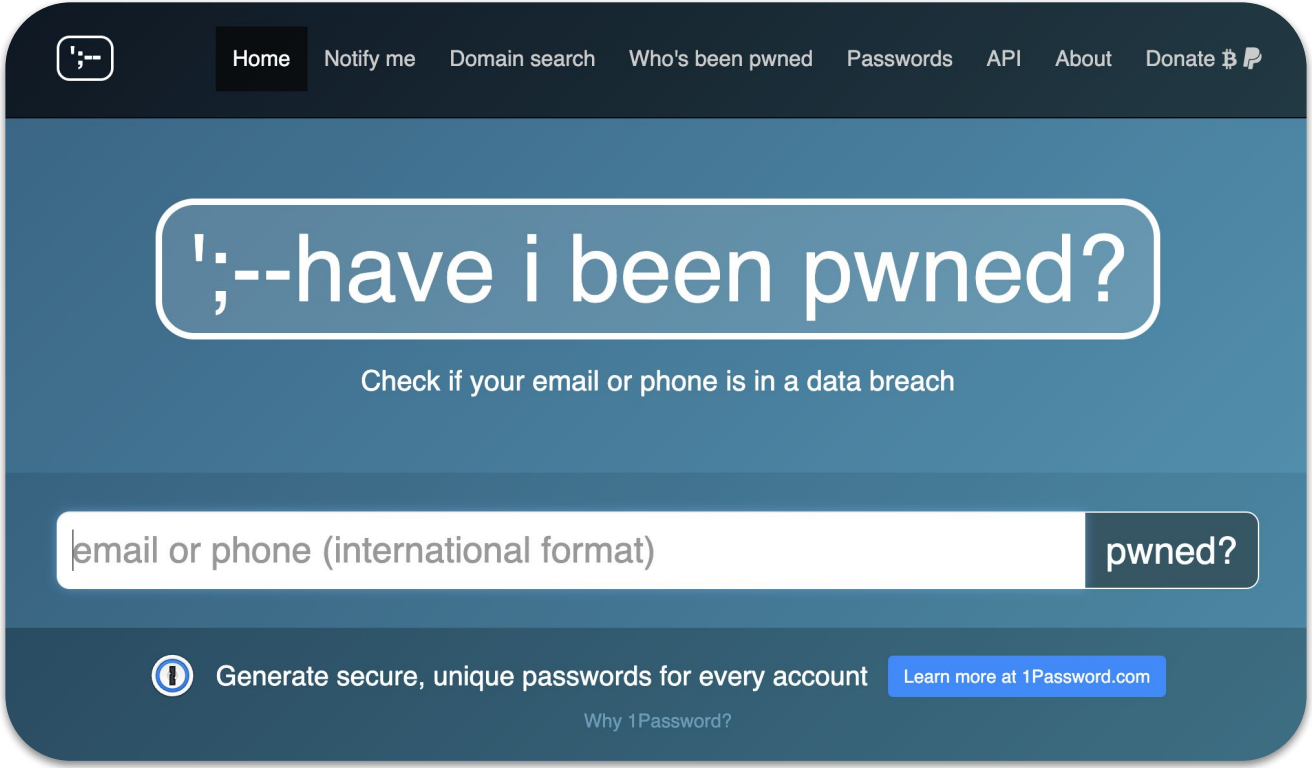
Read today’s ePaper

TOP STORY

1 Brazilian model reacts after Rahul Gandhi’s claim of Haryana electoral fraud

2 Meet Maya Handa, the campaign manager behind Zohran Mamdani’s historic win in New York’s mayoral polls

Always be vigilant!



Always be vigilant!

Questions?

This time on CS 4440...

Tor: The Onion Router
Internet Anonymity
Attacks on Tor
Project 4 Tips

What is Tor?

"Tor protects you by bouncing your communications around a distributed network of relays run by volunteers all around the world: it prevents somebody watching your Internet connection from learning **what sites you visit**, it prevents the sites you visit from learning **your physical location**, and it lets you access **sites which are blocked**."

Tor's Goal: Anonymity

- What is **anonymity**?
  - ???
- Versus **confidentiality**?
  - ???

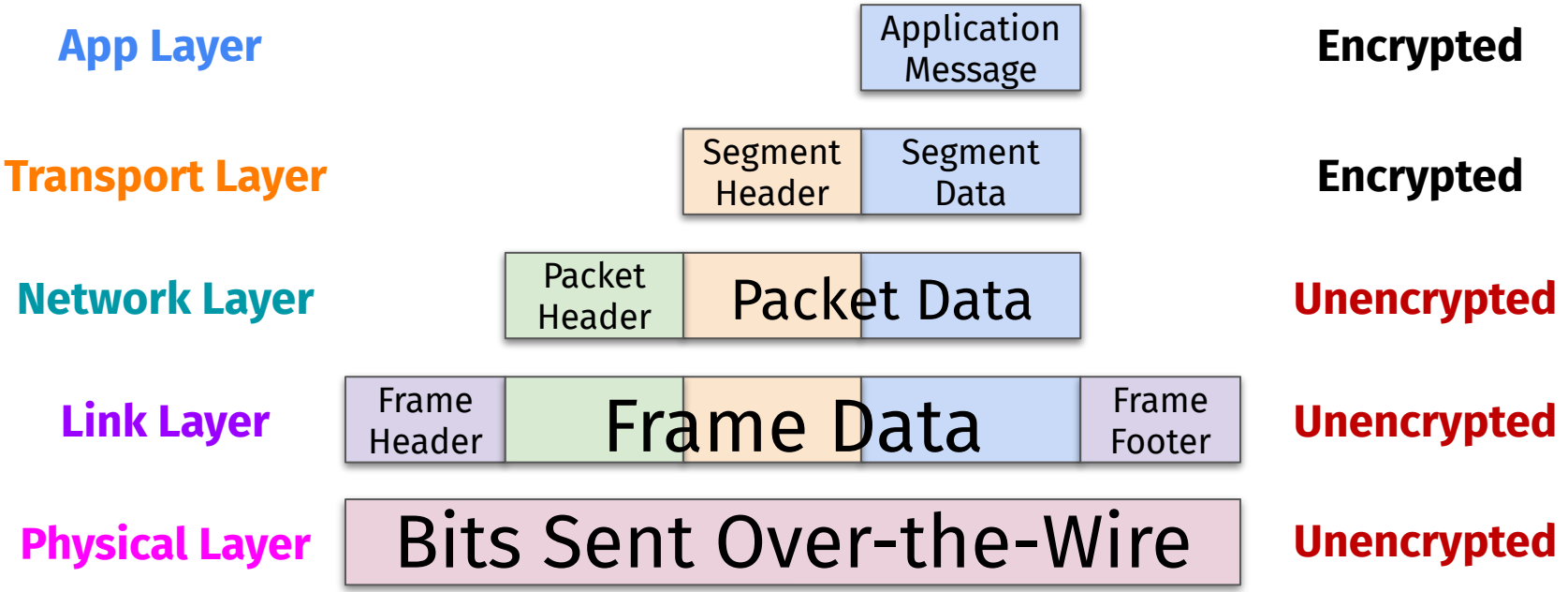
Tor's Goal: Anonymity

- What is **anonymity**?
  - I want to **say or do something** without the adversary knowing **that it was me** who said/did it
- Versus **confidentiality**?
  - **Confidentiality** = the contents
  - **Anonymity** = the identities

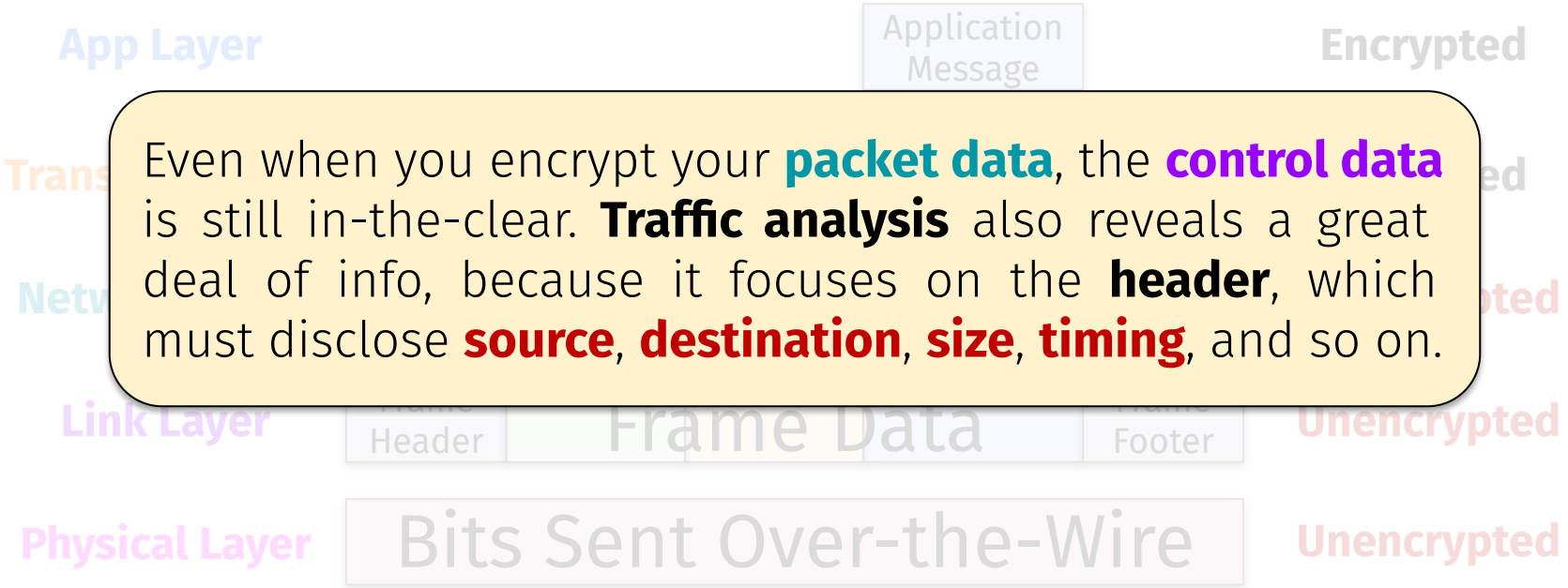
How/why does **anonymity** matter to **you**?

Why does internet anonymity matter?

How do the internet/web provide anonymity?

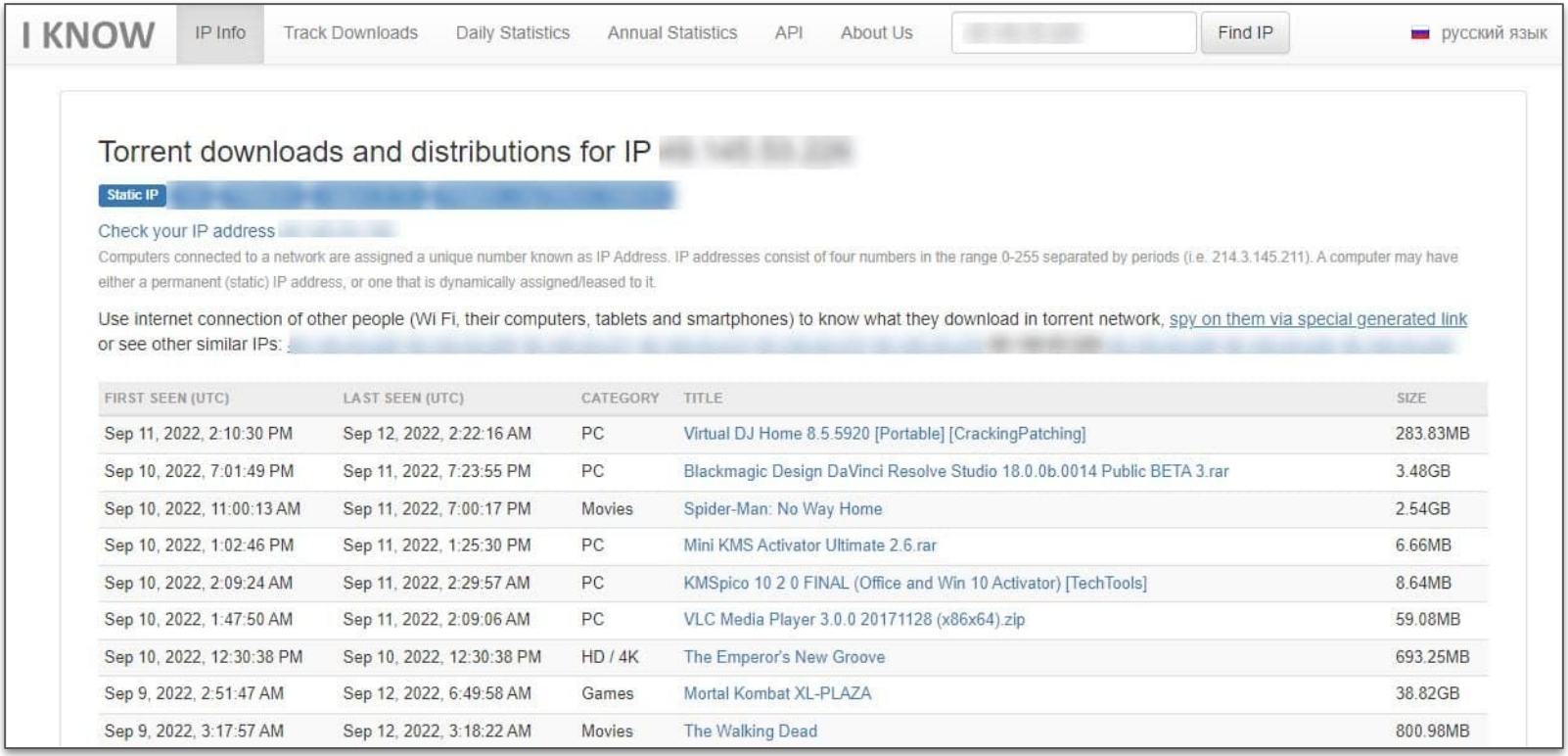


How do the internet/web provide anonymity?



Even when you encrypt your **packet data**, the **control data** is still in-the-clear. **Traffic analysis** also reveals a great deal of info, because it focuses on the **header**, which must disclose **source**, **destination**, **size**, **timing**, and so on.

How do the internet/web provide anonymity?



I KNOW | IP Info | Track Downloads | Daily Statistics | Annual Statistics | API | About Us | Find IP | русский язык

Torrent downloads and distributions for IP [redacted]

Static IP [redacted]

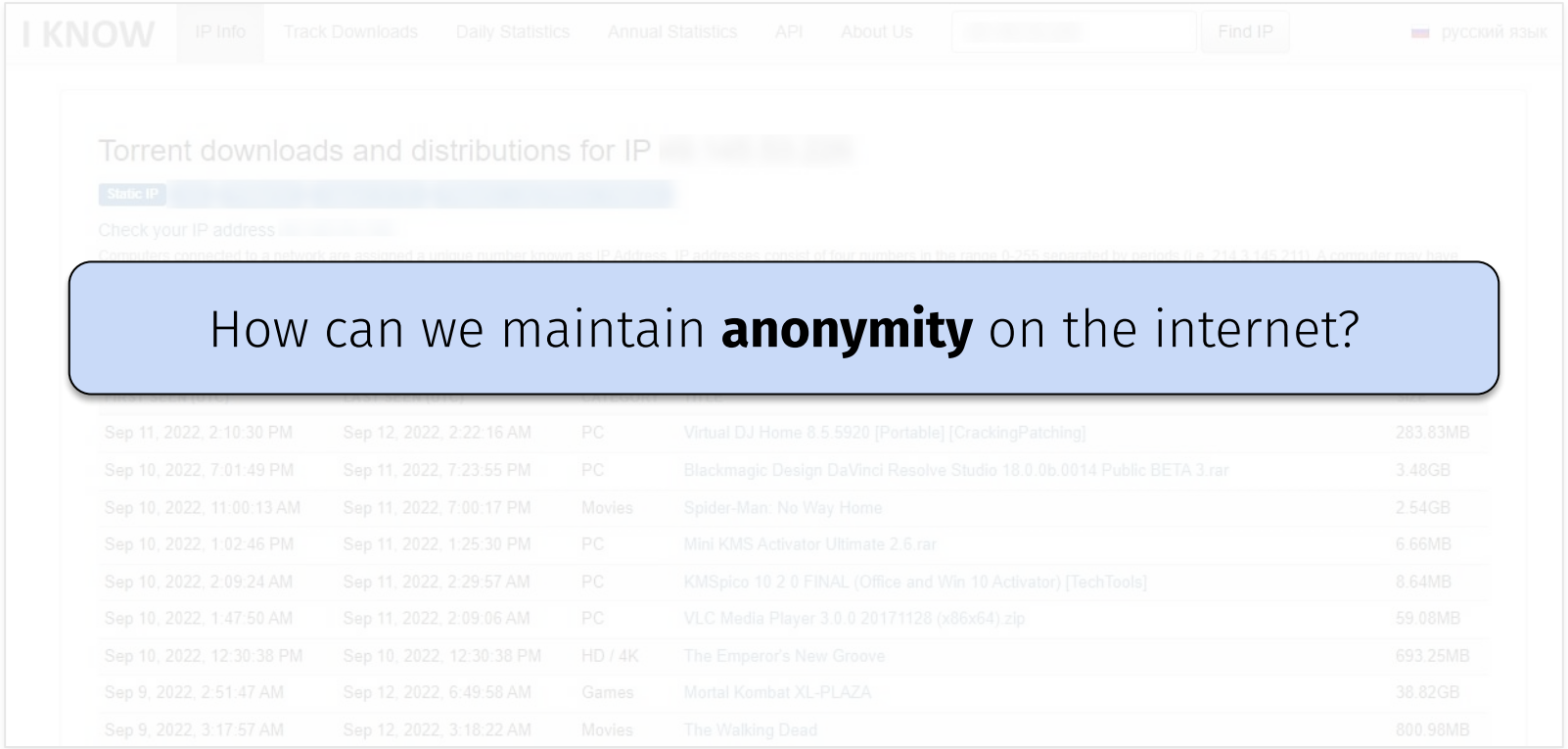
Check your IP address [redacted]

Computers connected to a network are assigned a unique number known as IP Address. IP addresses consist of four numbers in the range 0-255 separated by periods (i.e. 214.3.145.211). A computer may have either a permanent (static) IP address, or one that is dynamically assigned/leased to it.

Use internet connection of other people (Wi Fi, their computers, tablets and smartphones) to know what they download in torrent network, spy on them via special generated link or see other similar IPs: [redacted]

FIRST SEEN (UTC)	LAST SEEN (UTC)	CATEGORY	TITLE	SIZE
Sep 11, 2022, 2:10:30 PM	Sep 12, 2022, 2:22:16 AM	PC	Virtual DJ Home 8.5.5920 [Portable] [CrackingPatching]	283.83MB
Sep 10, 2022, 7:01:49 PM	Sep 11, 2022, 7:23:55 PM	PC	Blackmagic Design DaVinci Resolve Studio 18.0.0b.0014 Public BETA 3.rar	3.48GB
Sep 10, 2022, 11:00:13 AM	Sep 11, 2022, 7:00:17 PM	Movies	Spider-Man: No Way Home	2.54GB
Sep 10, 2022, 1:02:46 PM	Sep 11, 2022, 1:25:30 PM	PC	Mini KMS Activator Ultimate 2.6.rar	6.66MB
Sep 10, 2022, 2:09:24 AM	Sep 11, 2022, 2:29:57 AM	PC	KMSpico 10 2 0 FINAL (Office and Win 10 Activator) [TechTools]	8.64MB
Sep 10, 2022, 1:47:50 AM	Sep 11, 2022, 2:09:06 AM	PC	VLC Media Player 3.0.0 20171128 (x86x64).zip	59.08MB
Sep 10, 2022, 12:30:38 PM	Sep 10, 2022, 12:30:38 PM	HD / 4K	The Emperor's New Groove	693.25MB
Sep 9, 2022, 2:51:47 AM	Sep 12, 2022, 6:49:58 AM	Games	Mortal Kombat XL-PLAZA	38.82GB
Sep 9, 2022, 3:17:57 AM	Sep 12, 2022, 3:18:22 AM	Movies	The Walking Dead	800.98MB

How do the internet/web provide anonymity?

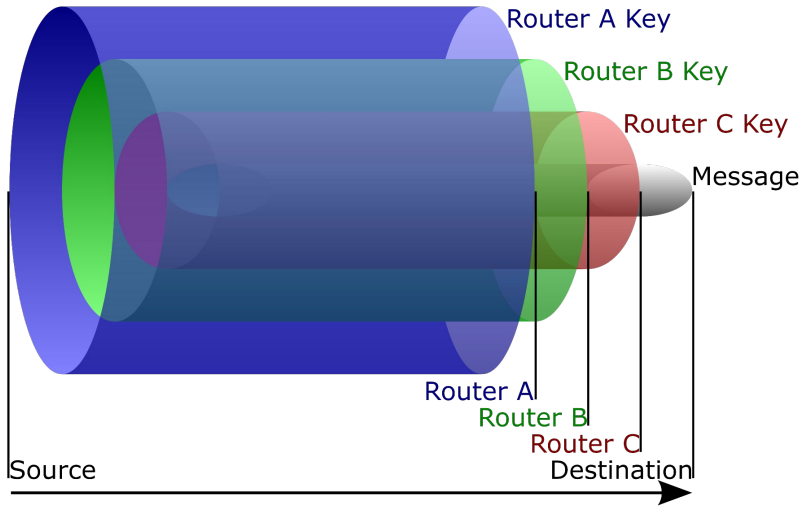


How can we maintain **anonymity** on the internet?

Tor: The Onion Router

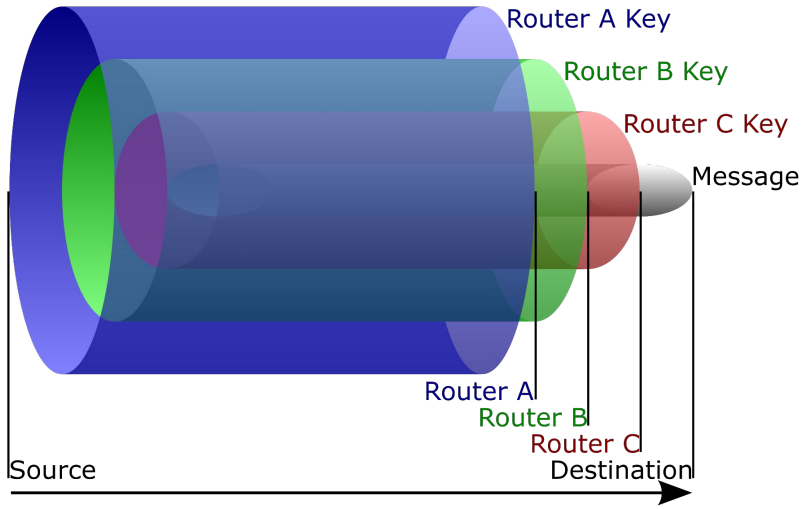
Anonymity Primitive: Onion Routing

- Each message is **repeatedly encrypted**
  - **Analogy:** multiple layers of an onion



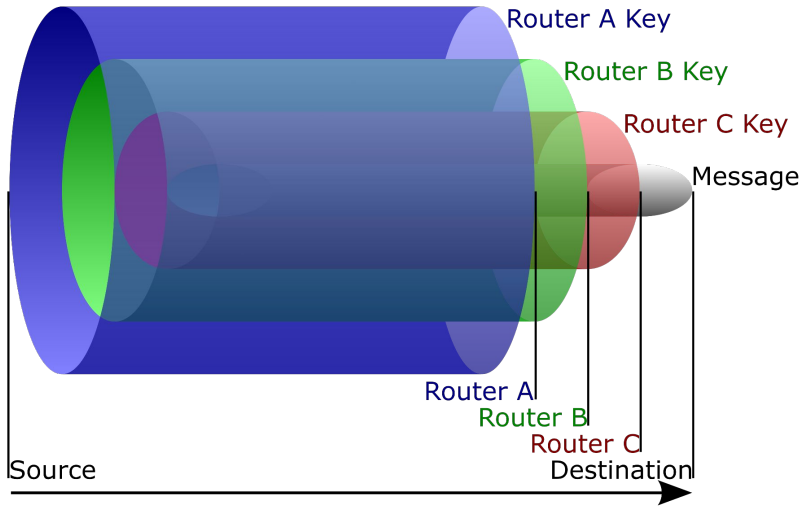
Anonymity Primitive: Onion Routing

- Each message is **repeatedly encrypted**
  - **Analogy:** multiple layers of an onion
- Sent through **multiple network nodes**
  - These nodes are called **onion routers**
  - Each node removes an encryption layer to uncover the message **routing instructions**
  - Process repeats when sent to next router

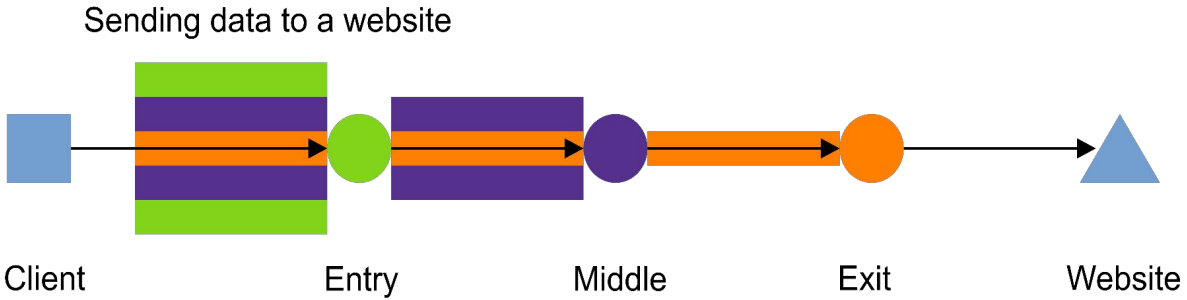


Anonymity Primitive: Onion Routing

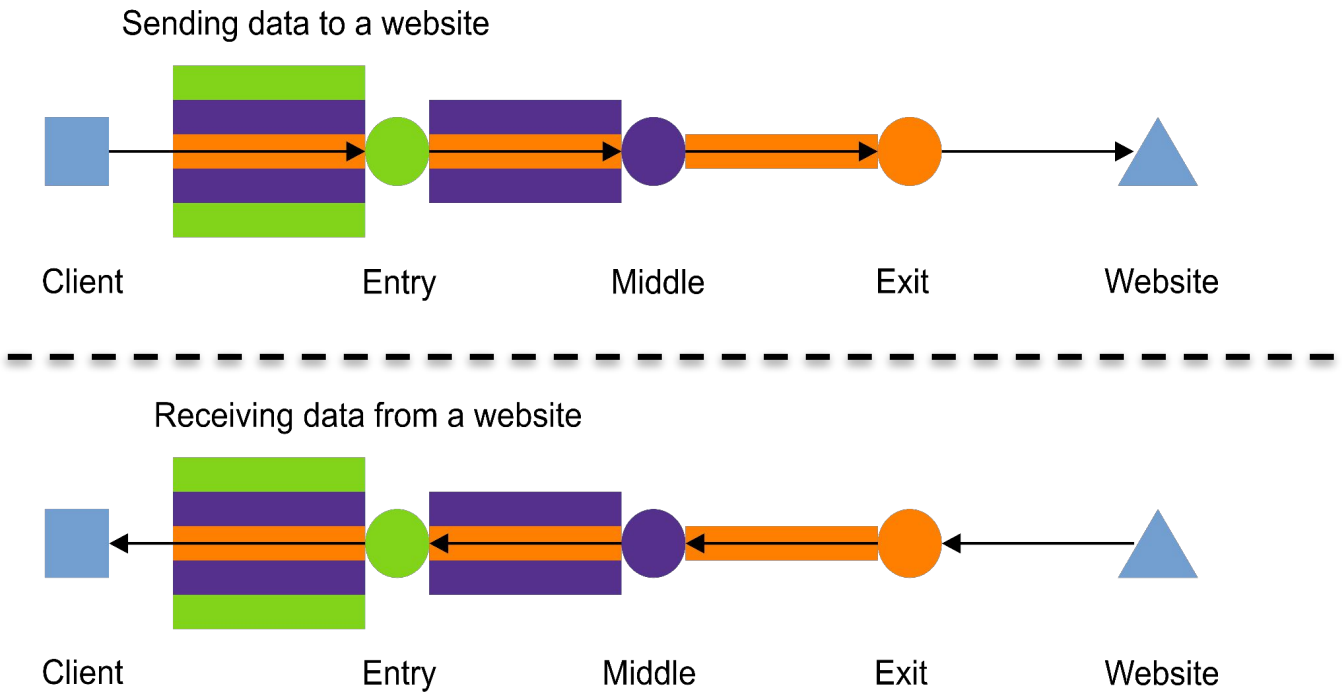
- Each message is **repeatedly encrypted**
  - **Analogy:** multiple layers of an onion
- Sent through **multiple network nodes**
  - These nodes are called **onion routers**
  - Each node removes an encryption layer to uncover the message **routing instructions**
  - Process repeats when sent to next router
- **Anonymity:** prevents any intermediary nodes from knowing message **origin**, **destination**, and **contents**



Onion Routing Visualized

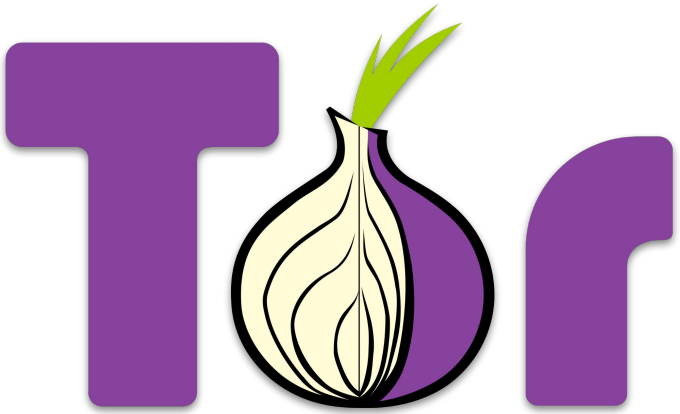


Onion Routing Visualized



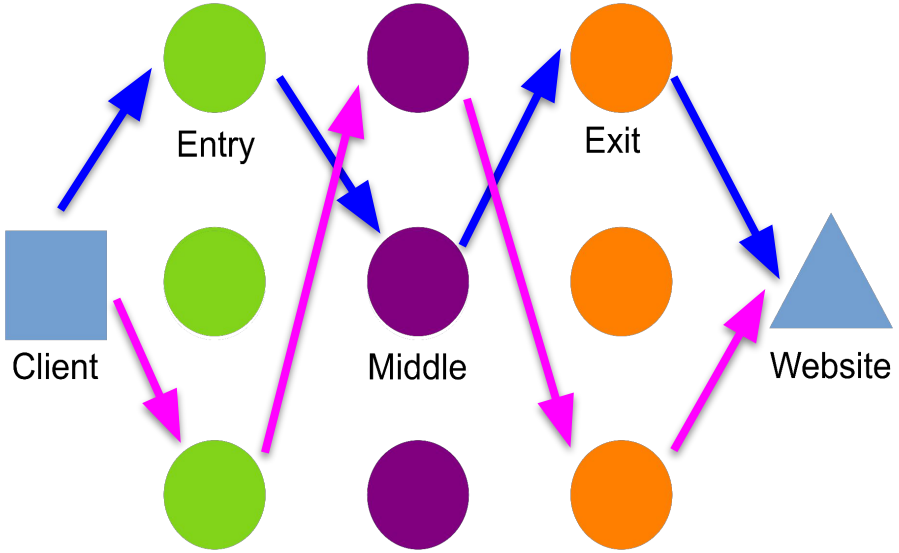
Tor: The Onion Router

- **Tor:** a distributed overlay network
  - Anonymizes TCP-based applications
    - Secure shell
    - Web browsing
    - Instant messaging



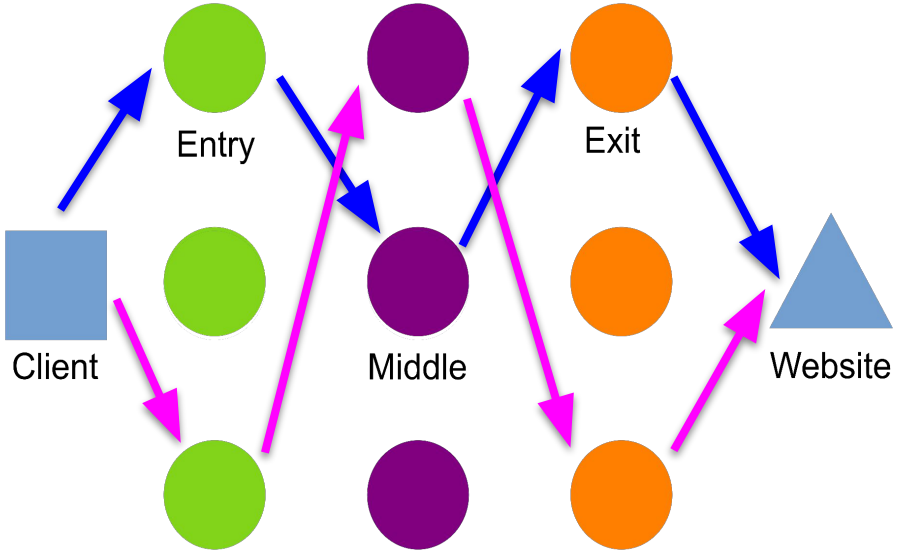
Tor: The Onion Router

- **Tor:** a distributed overlay network
  - Anonymizes TCP-based applications
    - Secure shell
    - Web browsing
    - Instant messaging
- Clients choose the **circuit paths**
  - Messages unwrapped at each onion router using a symmetric key

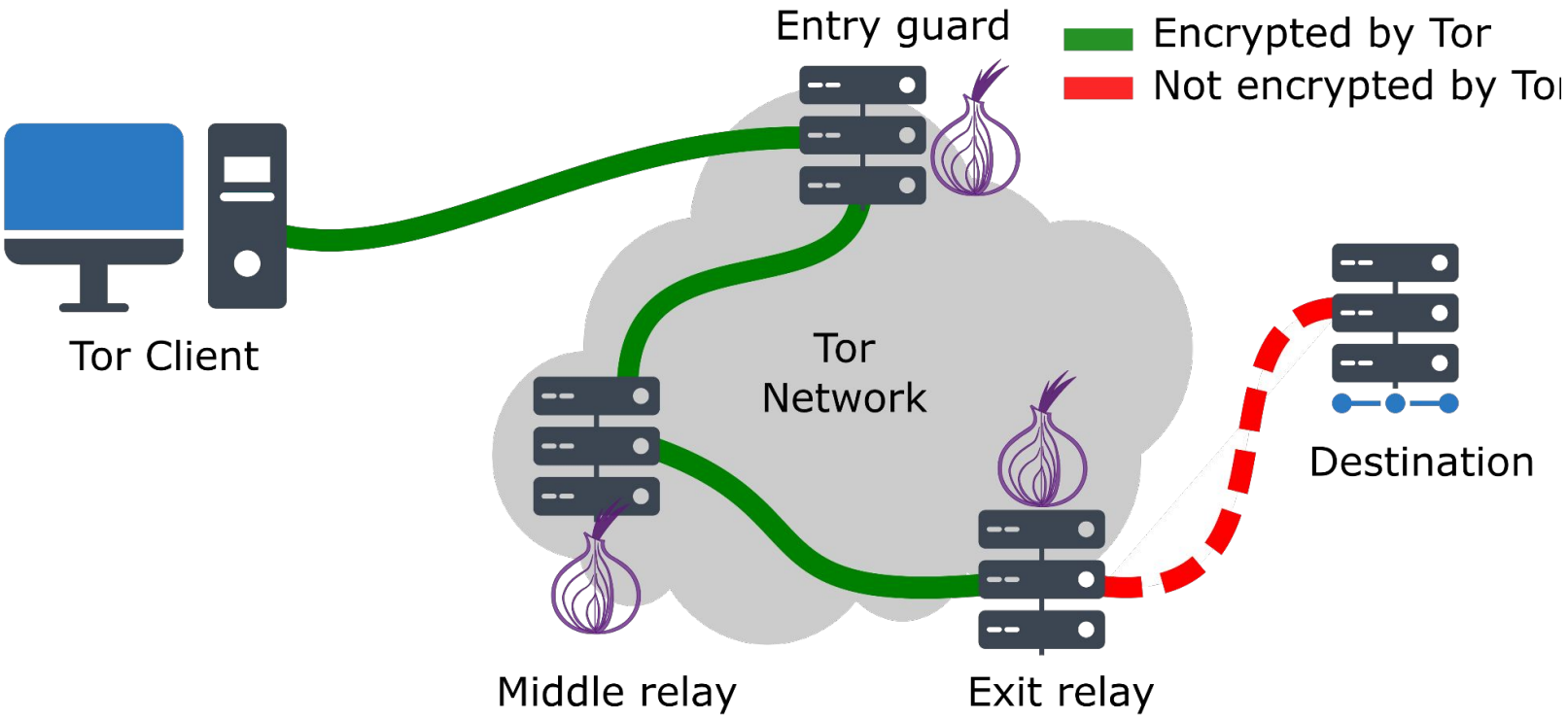


Tor: The Onion Router

- **Tor:** a distributed overlay network
  - Anonymizes TCP-based applications
    - Secure shell
    - Web browsing
    - Instant messaging
- Clients choose the **circuit paths**
  - Messages unwrapped at each onion router using a symmetric key
- Onion routers only know their **successor** or **predecessor** nodes
  - They don't know of any other nodes



How Tor Works

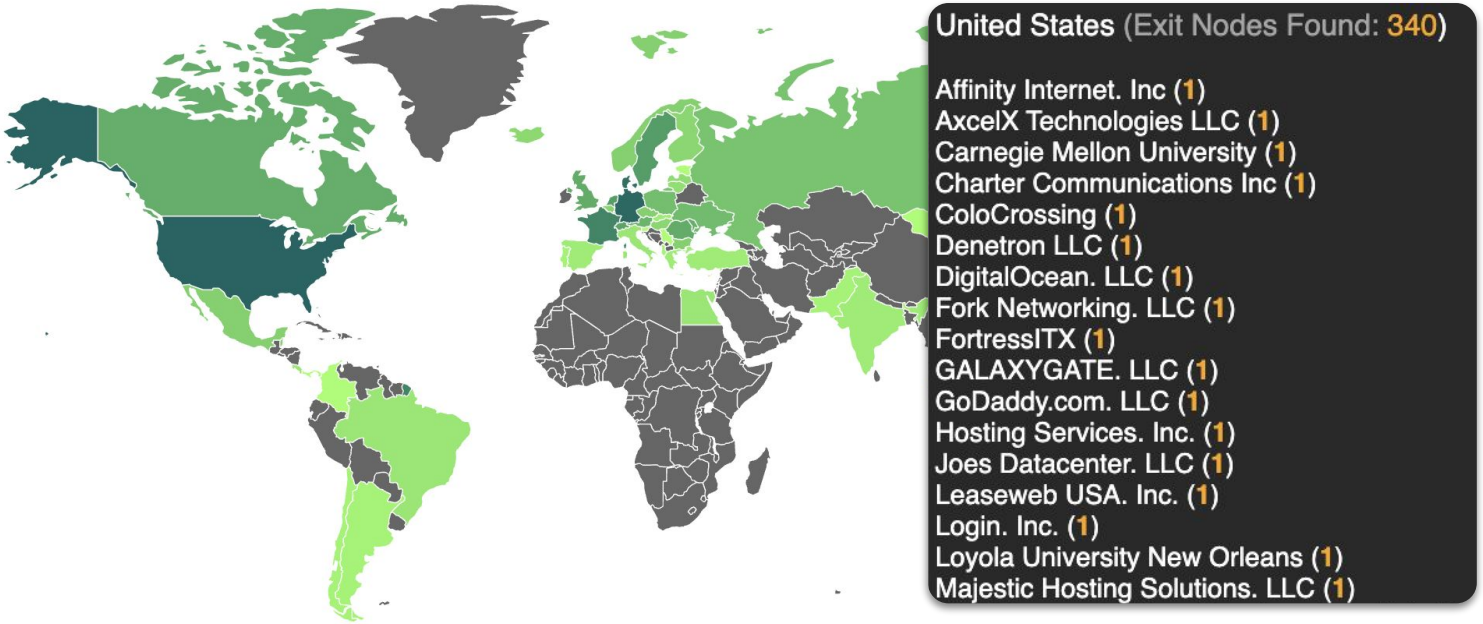


Trust in Tor

- **Entry node:** knows that Alice is using Tor as well as the identity of **middle node**
  - Does not know the destination!
- **Exit node:** knows a Tor user is connecting to the destination, but not **which** user
- **Destination:** knows that some Tor user is connecting to it via the exit node
- Tor does **not** provide encryption between the **exit node** and **message destination**
  - **That is what HTTPS is for!**

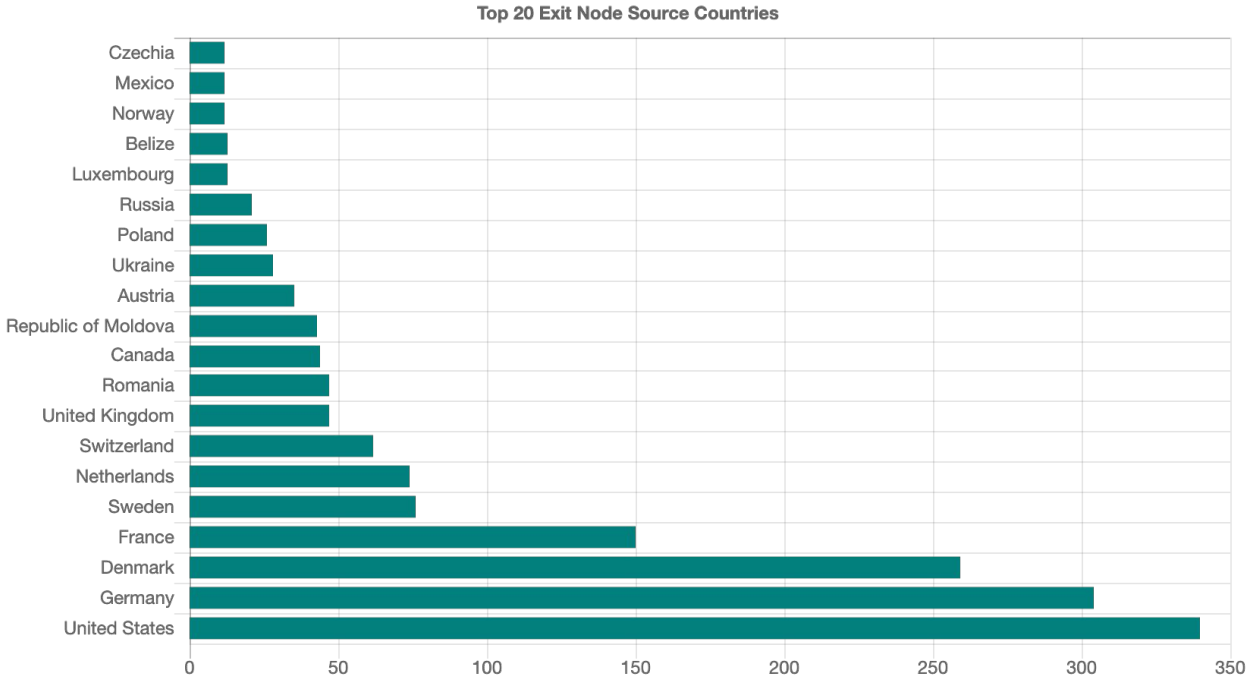
The Tor Network

- Lots of nodes spread out around the world



The Tor Network

- Lots of nodes spread out around the world

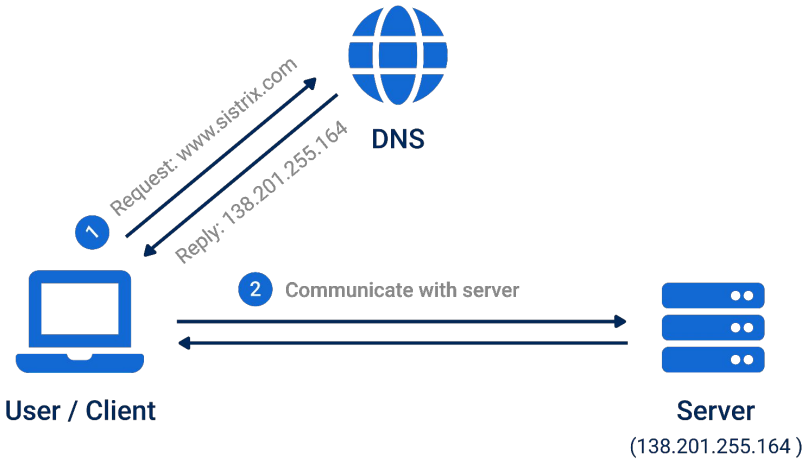


Questions?

Attacking Tor

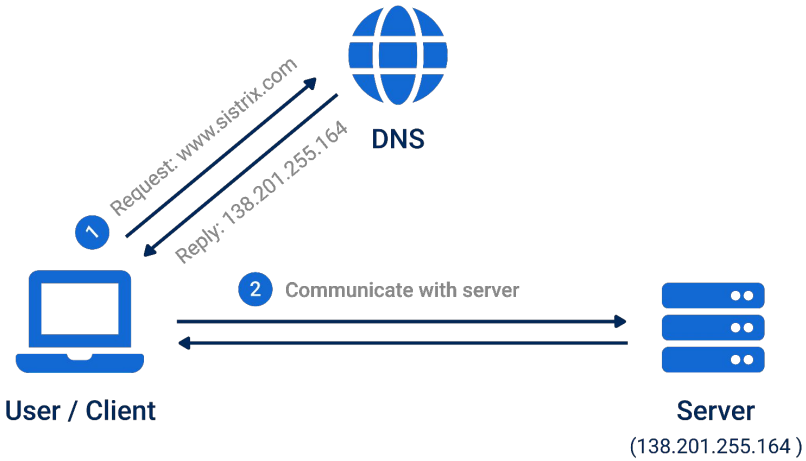
Recap: The Domain Name System

- ???



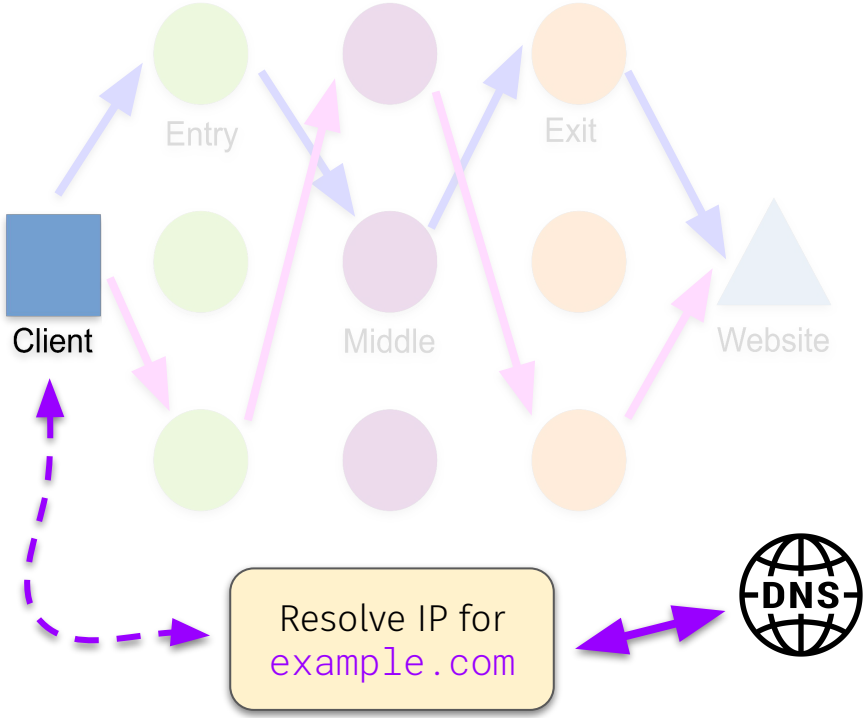
Recap: The Domain Name System

- **Distributed database** implemented in hierarchy of many name servers
- **Application-layer protocol:**
  - Hosts and domain name servers communicate to resolve **domain names**
    - Address–name translation
- **Result:** user requests **domain name**
  - But their host really gets its **IP address**
  - Convenient!



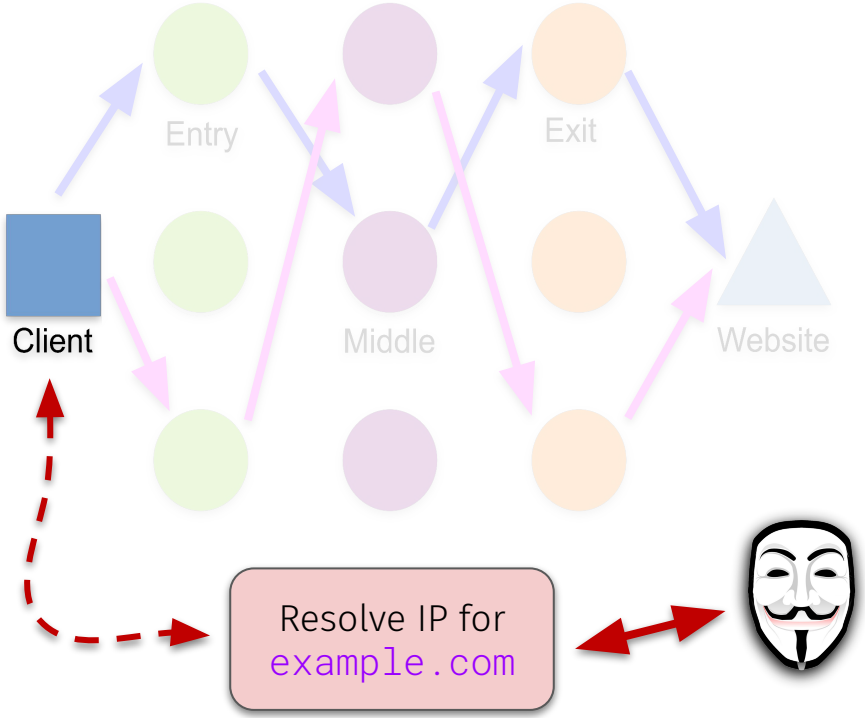
Attack 1: DNS Leaks

- **DNS requests** are **not** sent through Tor by default



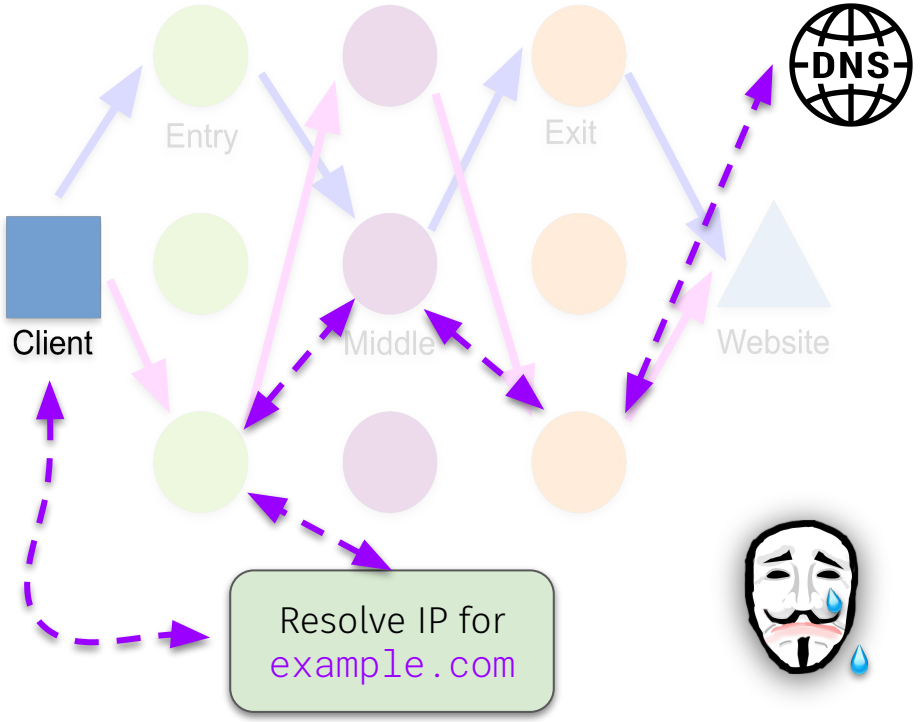
Attack 1: DNS Leaks

- **DNS requests** are **not** sent through Tor by default
- Attackers could see what **websites** are being visited



Attack 1: DNS Leaks

- **DNS requests** are **not** sent through Tor by default
- Attackers could see what **websites** are being visited
- **Fix:** external software can be used to reroute DNS via Tor
  - This is **not** default behavior
  - **Examples:** FoxyProxy, Privoxy



Attack 1: DNS Leaks

Brave browser's Tor feature found to leak .onion queries to ISPs

Jessica Haworth 19 February 2021 at 14:27 UTC
Updated: 01 July 2021 at 16:27 UTC

Privacy | Dark Web | Browsers

Developers are issuing hotfix

UPDATED Brave, the privacy-focused web browser, is exposing users' activity on Tor's hidden servers – aka the 'dark web' – to their internet service providers, it has been confirmed.

Brave is shipped with a built-in feature that integrates the Tor anonymity network into the browser, providing both security and privacy features that can help obscure a user's activity on the web.

Tor is also used to access .onion websites, which are hosted on the dark net.

Earlier today (February 19), a blog post from 'Rambler' claimed that Brave was leaking DNS requests made in the Brave browser to a user's ISP.

Attack 2: Traffic Analysis

- ???

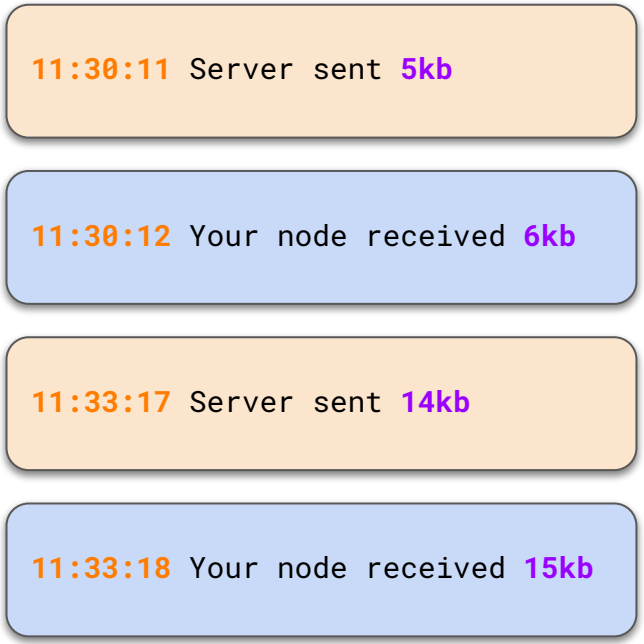
Attack 2: Traffic Analysis

- **Volume and Timing Analysis**:
  - Measure **traffic going in/out** of Tor network
  - Identify patterns to aid in reconnaissance
  - Identify likelihood you are accessing a page

Attack 2: Traffic Analysis

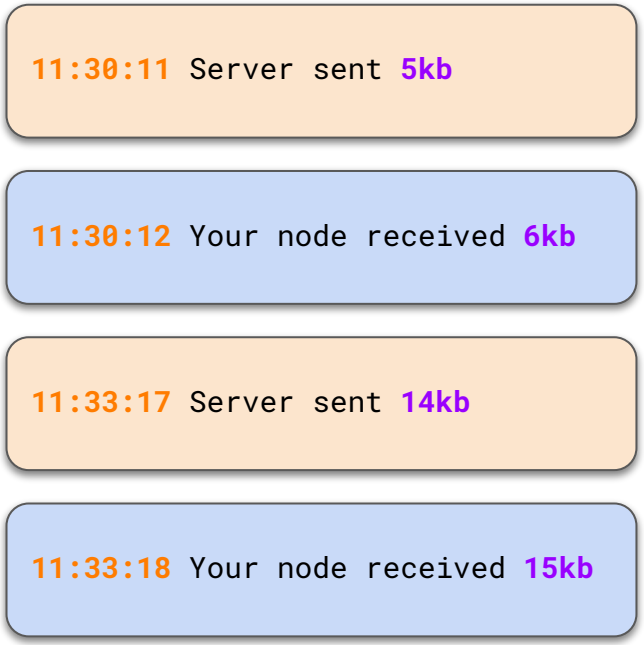
- **Volume and Timing Analysis:**
  - Measure **traffic going in/out** of Tor network
  - Identify patterns to aid in reconnaissance
  - Identify likelihood you are accessing a page

- **Examples:**
  - **Volume:** watch video vs. reading webpage
  - **Timing:** when you sent/received packets



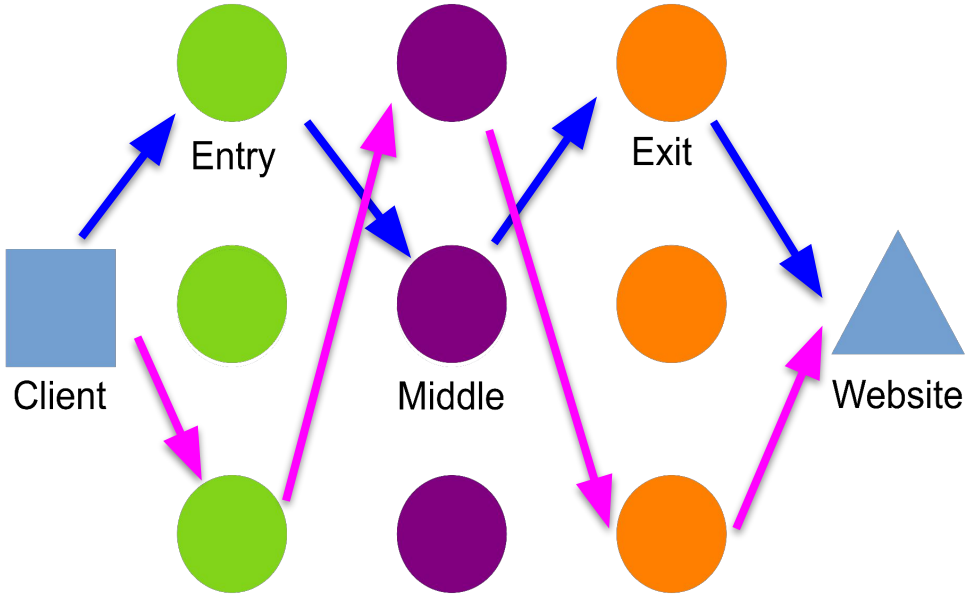
Attack 2: Traffic Analysis

- **Volume and Timing Analysis:**
  - Measure **traffic going in/out** of Tor network
  - Identify patterns to aid in reconnaissance
  - Identify likelihood you are accessing a page
- **Examples:**
  - **Volume:** watch video vs. reading webpage
  - **Timing:** when you sent/received packets
- **Defenses:**
  - Intentionally adding noisy traffic
    - Cons: latency atop of latency



Attack 3: Malicious Nodes

- **Traffic leaving exit nodes** (e.g., a request to a website) is **unencrypted**



Attack 3: Malicious Nodes

- Traffic ... ted

"Honey Onions" probe the Dark Web: at least 3% of Tor nodes are rogues

"If you control **enough** of the Tor network, it's possible to get a kind of **bird's eye view** of the traffic being routed through it."

>25% of the Tor network's exit capacity has been attacking Tor users

Figure 1: Malicious Tor exit fraction (measured in % of the entire available Tor network exit capacity) over time by this particular malicious entity between July 2020 and April 2021. Peak value: The attacker did manage approx. 27.5% of the Tor networks exit capacity on 2021–02–02. Graph by nusenu (raw data source: Tor Project/onionoo)

Attack 3: Malicious Nodes

- Traffic leaving [illegible] unencrypted

Police Go on Fishing Expedition, Search the Home of Seattle Privacy Activists Who Maintain Tor Network

ANSEL HERZ

Seattle police descended on the Queen Anne condo of **two outspoken privacy activists** with a search warrant early this morning, leaving them shaken and upset.

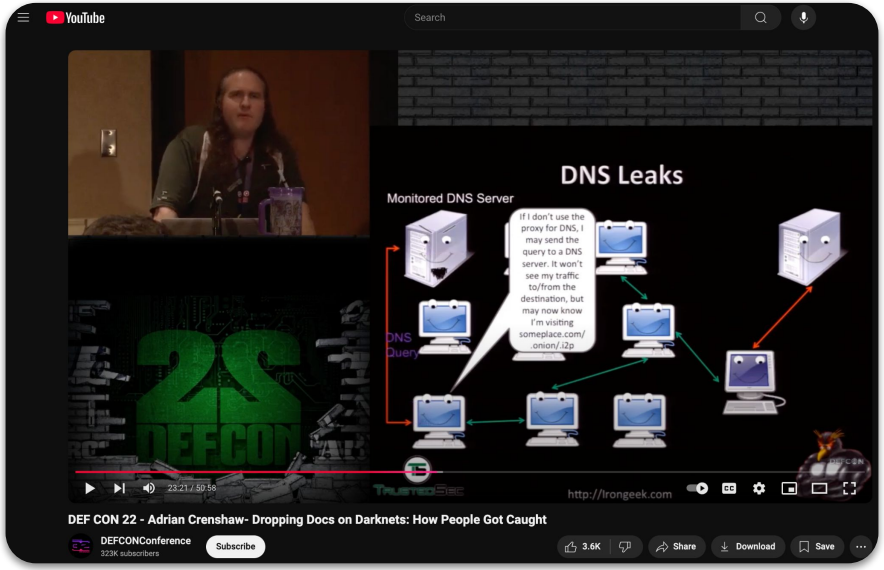
Jan Bultmann and David Robinson, a married couple and co-founders of the Seattle Privacy Coalition, said they were awakened at 6:15 a.m. by a team of six detectives from the SPD knocking on the door. Bultmann said were made to sit outside as the officers, who had a search warrant, examined their equipment. They claimed to be looking for child pornography.

The SPD acknowledged this morning that no child porn was found, no assets were seized, and **no arrests were made.**

Questions?

Supplemental: Dropping Docs on Darknets

- Dan Crenshaw's awesome DEF CON talk on ToR attacks—**check it out!**



https://www.youtube.com/watch?v=eQ2OZKitRwc

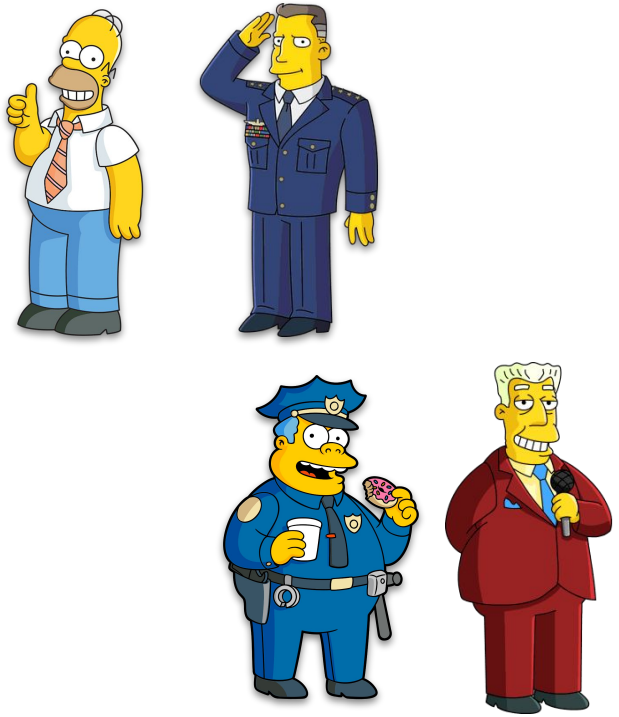
Tor Users and Websites

Who uses Tor?

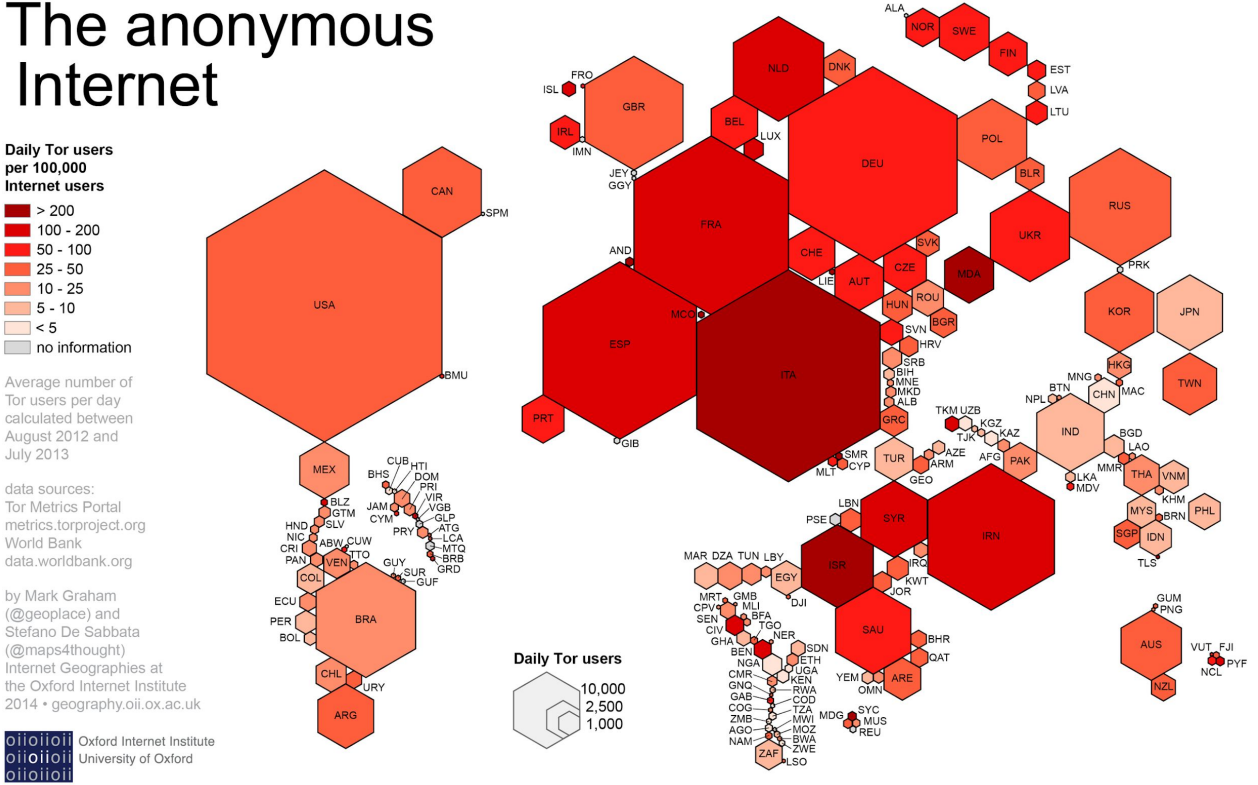
- ???

Who uses Tor?

- **Normal People**
  - Privacy-conscious folks
- **Intelligence Agencies**
  - Secret agents in the field
- **Law Enforcement**
  - Online “undercover” operations
- **Journalists and Bloggers**
  - Citizen journalists inspiring social change
- **Activists and Whistleblowers**
  - Raising their voice and avoiding persecution
- **White-hat and Black-hat Hackers**
  - And everyone in between!



Who uses Tor?


The anonymous Internet
Daily Tor users per 100,000 Internet users
> 200
100 - 200
50 - 100
25 - 50
10 - 25
5 - 10
< 5
no information
Average number of Tor users per day calculated between August 2012 and July 2013
data sources:
Tor Metrics Portal
metrics.torproject.org
World Bank
data.worldbank.org
by Mark Graham (@geoplace) and Stefano De Sabbata (@maps4thought)
Internet Geographies at the Oxford Internet Institute
2014 • geography.oii.ox.ac.uk
Oxford Internet Institute
University of Oxford
Daily Tor users
10,000
2,500
1,000

Who uses Tor?

Internet censorship in the Arab Spring

1 language

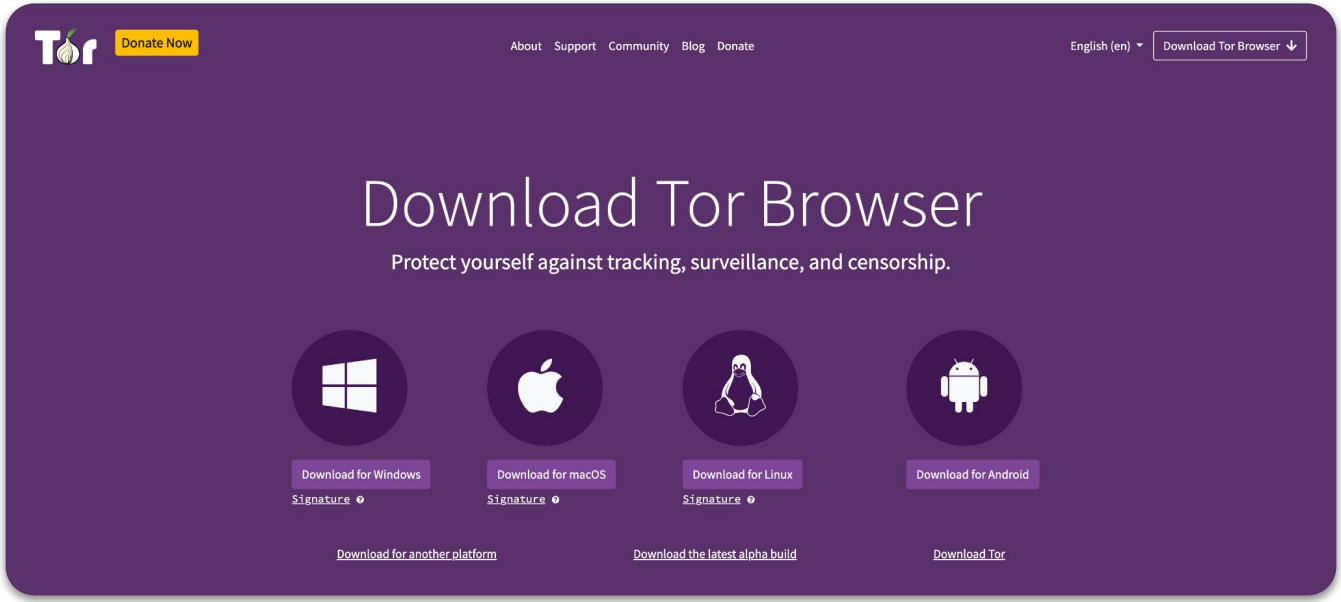
Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

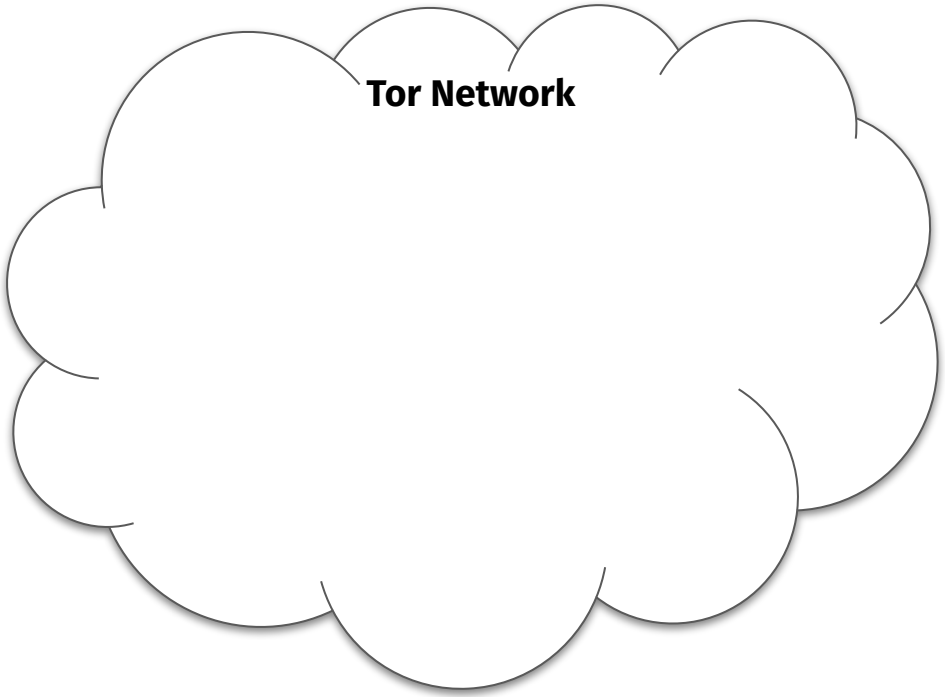
Main articles: Arab Spring and Internet censorship

The level of **Internet censorship in the Arab Spring** was escalated. Lack of Internet freedom was a tactic employed by authorities to quell protests. Rulers and governments across the Arab world utilized the law, technology, and violence to control what was being posted on and disseminated through the Internet. In Egypt, Libya, and Syria, the populations witnessed full Internet shutdowns as their respective governments attempted to quell protests. In Tunisia, the government of Zine El Abidine Ben Ali hacked into and stole passwords from citizens' Facebook accounts. In Saudi Arabia and Bahrain, bloggers and "netizens" were arrested and some are alleged to have been killed. The developments since the beginning of the Arab Spring in 2010 have raised the issue of Internet access as a human right and have revealed the type of power certain authoritarian governments retain over the people and the Internet.

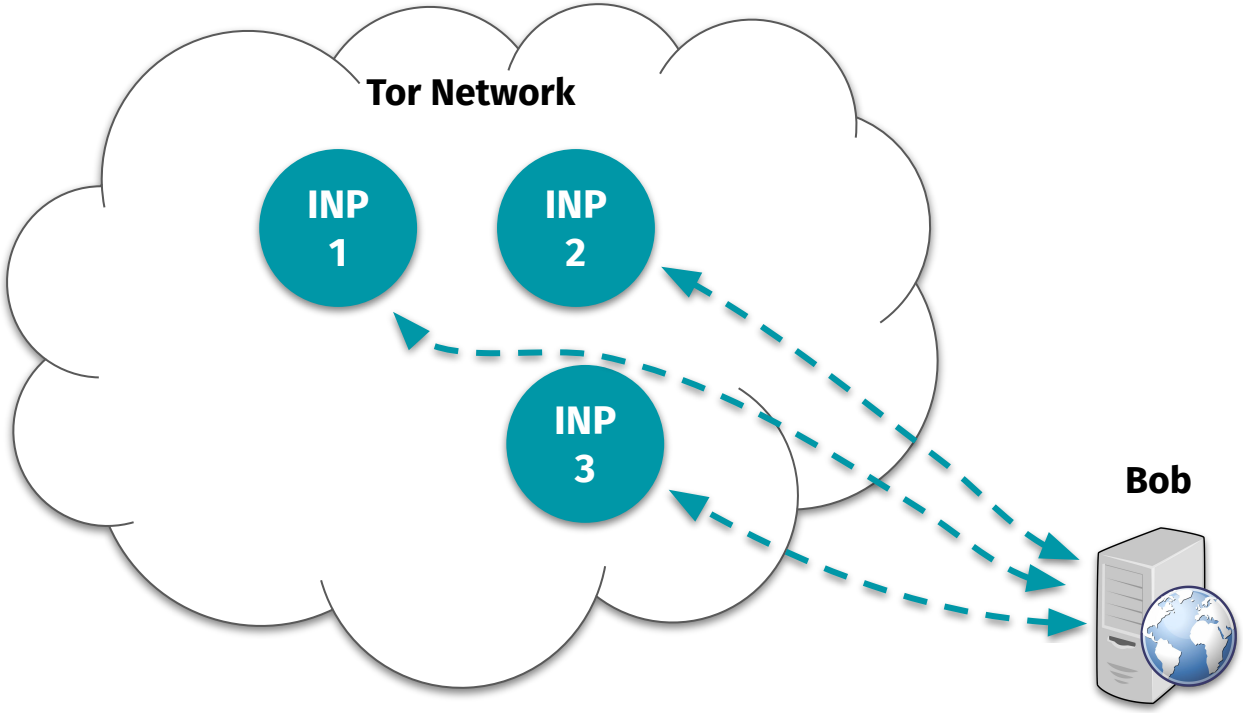
How can you use Tor?



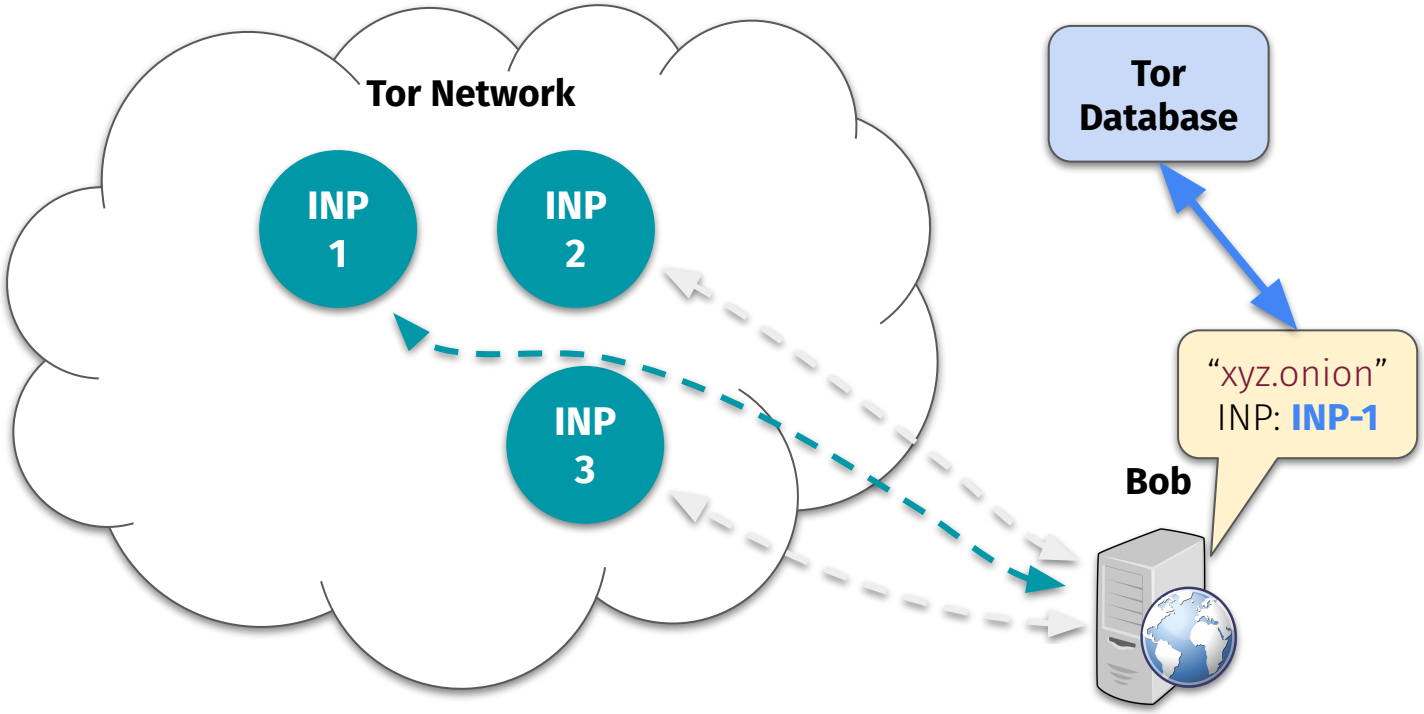
Hidden Services



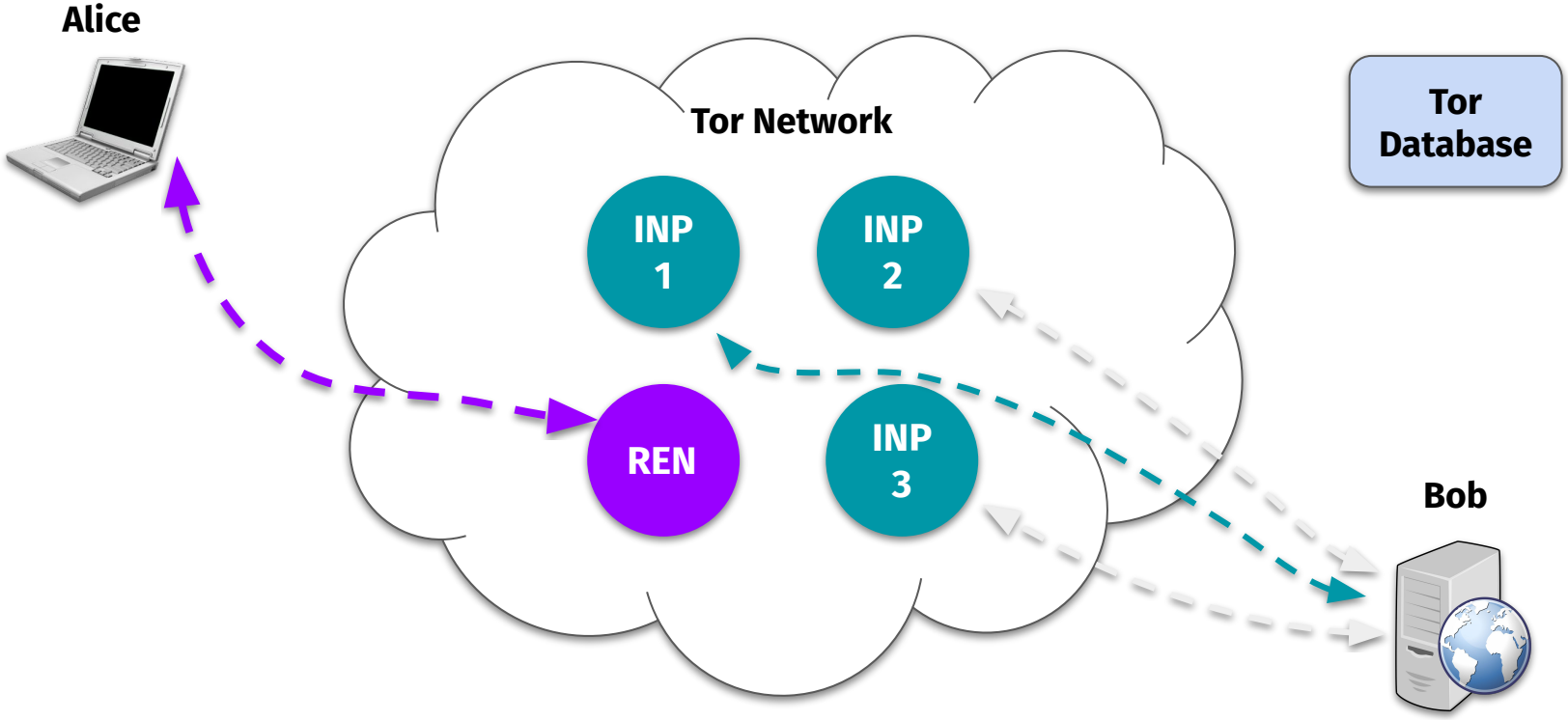
Hidden Services



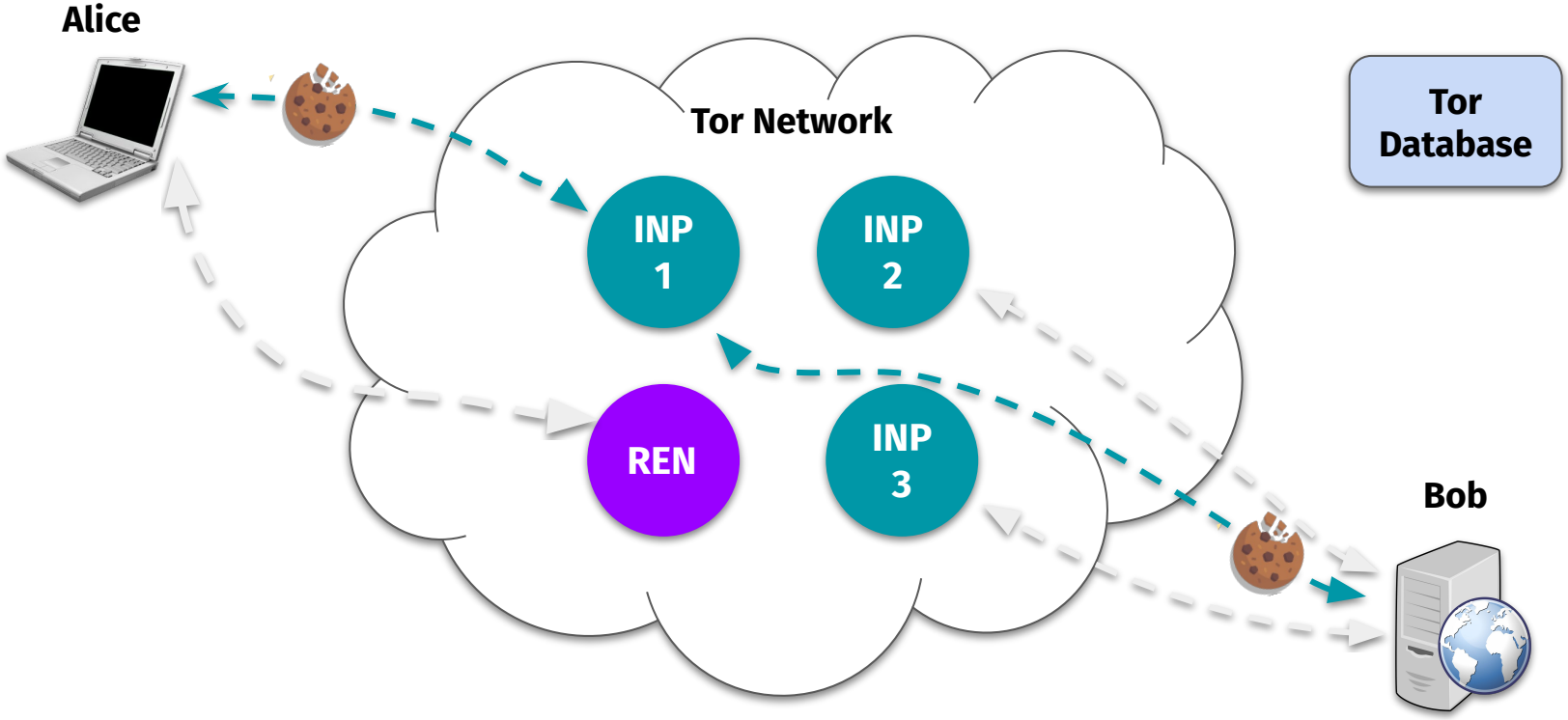
Hidden Services



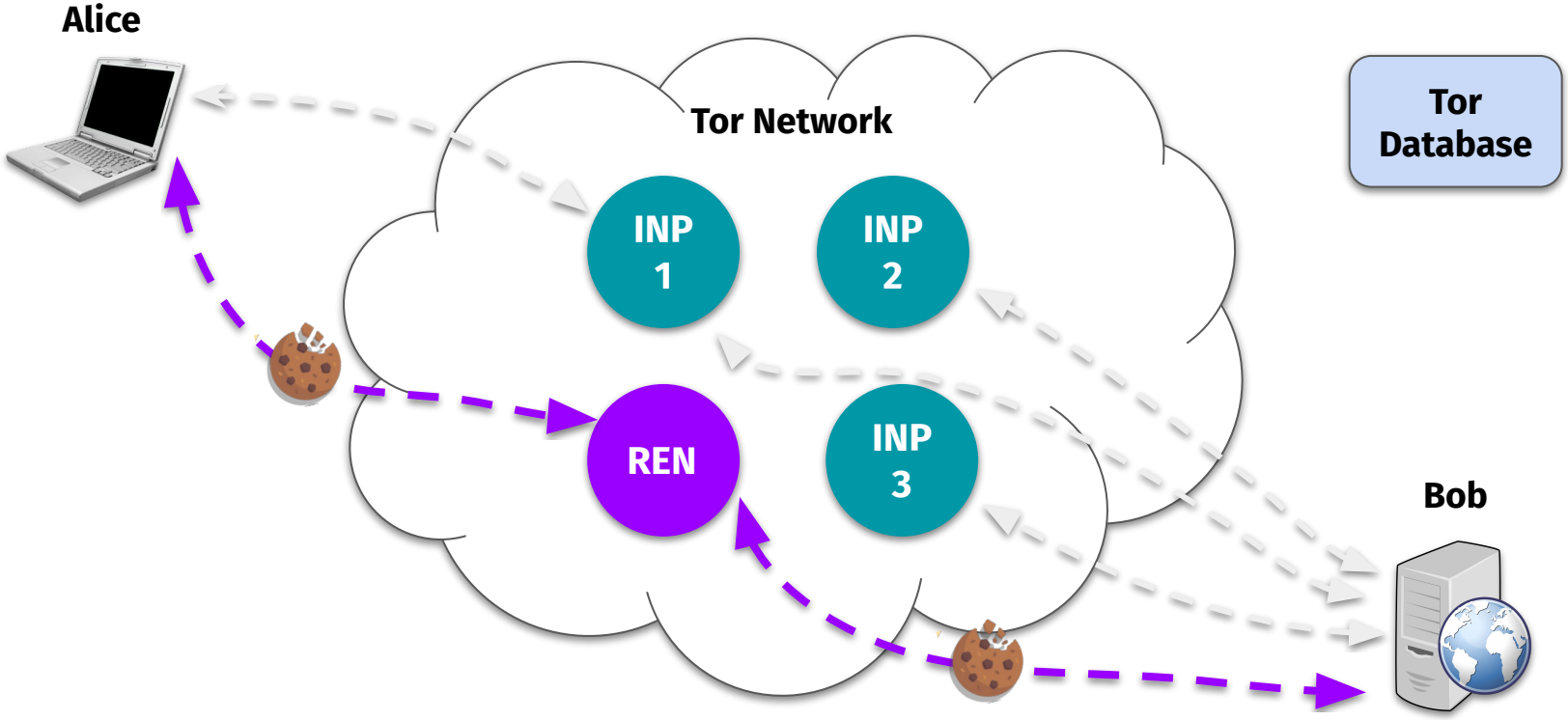
Hidden Services



Hidden Services



Hidden Services



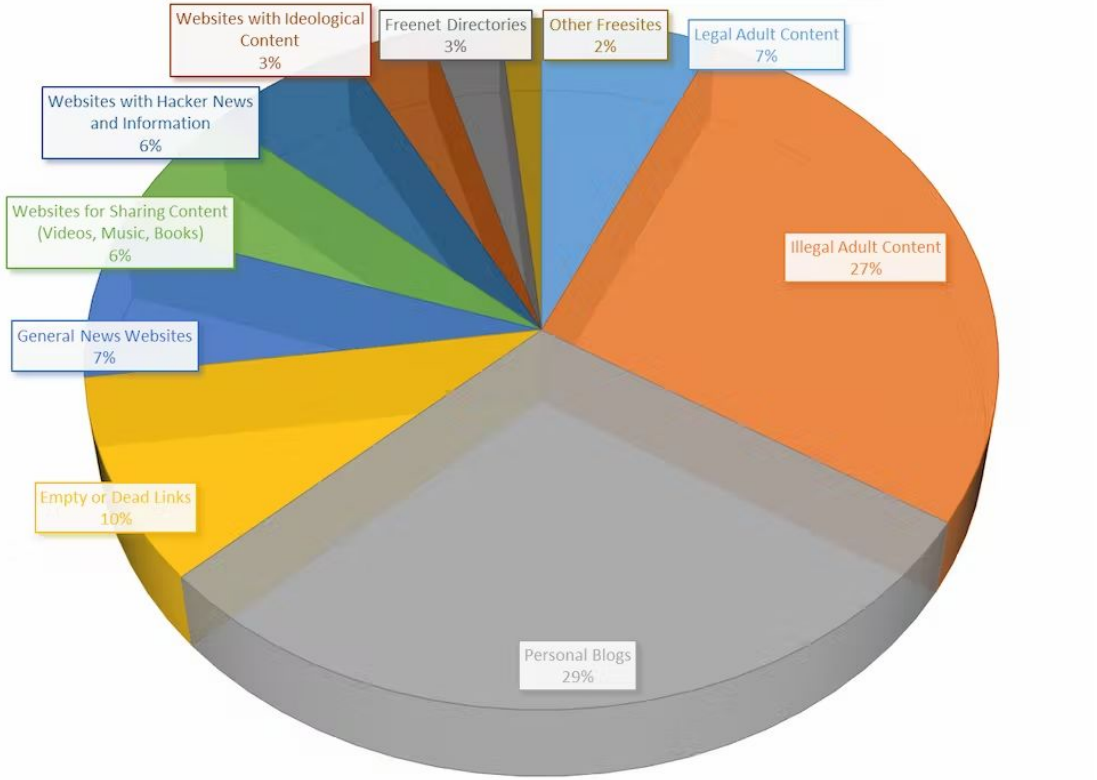
Hidden Services

Alice

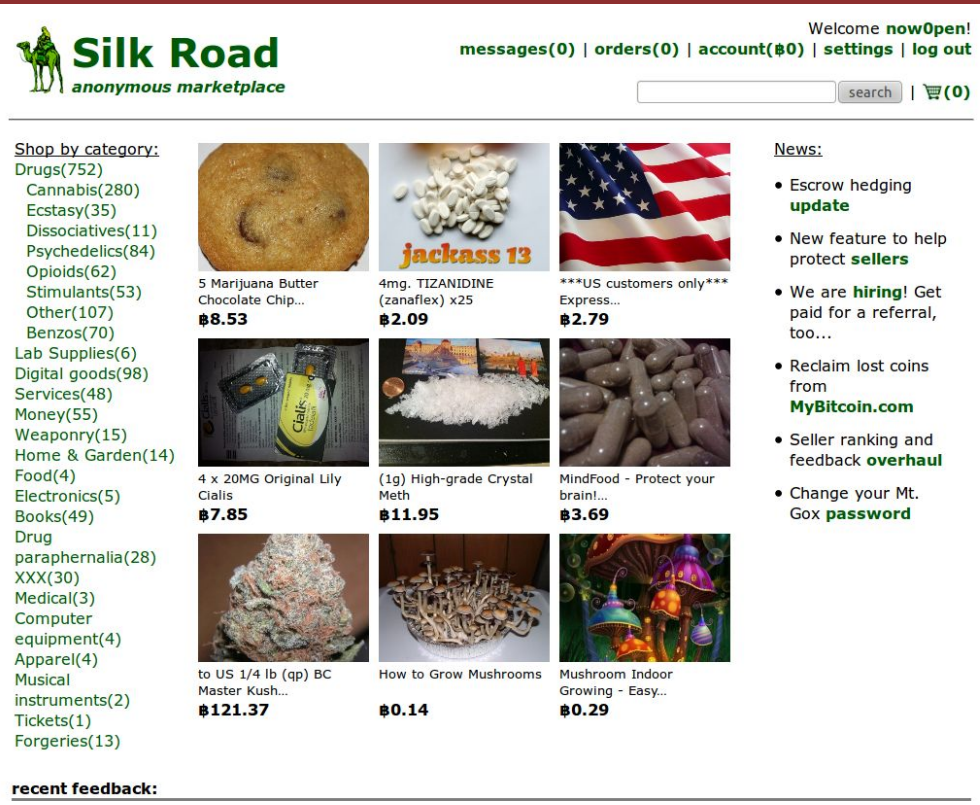
Tor Database

Bob

What services get hidden?



What services get hidden?



What services get hidden?

Positive Tor Use Cases

Introducing DNS Resolver for Tor

06/05/2018

Mahrud Sayrafi

In case you haven't heard yet, Cloudflare launched a privacy-first DNS resolver service on April 1st. It was no joke! The service, which was our first consumer-focused service, supports emerging DNS standards such as DNS over HTTPS:443 and TLS:853 in addition to traditional protocols over UDP:53 and TCP:53, all in one easy to remember address: 1.1.1.1.

Positive Tor Use Cases

Questions?

Project 4 Tips

Project 4 Overview

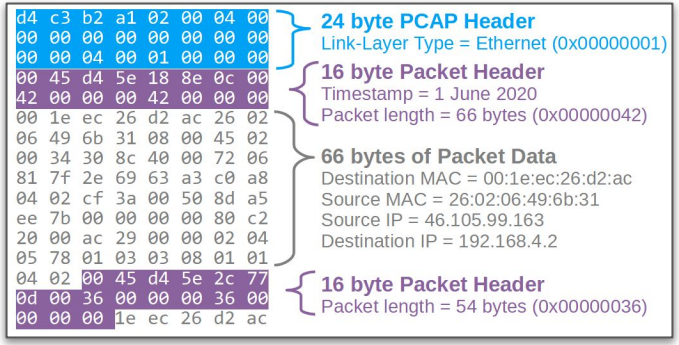
- Focuses on **network packet analysis**
  - Leveraging data contained within packets to achieve network defenses and attacks

Project 4 Overview

- Focuses on **network packet analysis**
  - Leveraging data contained within packets to achieve network defenses and attacks
- **Scenario:** helping a fictional university secure its enterprise campus network
  - Detect and characterizing likely attacks
  - Demonstrate how info can be intercepted

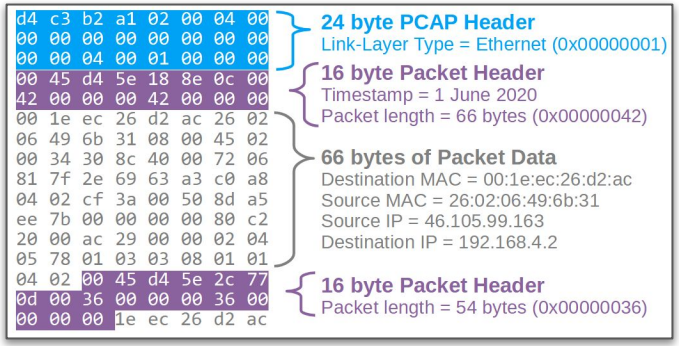
Project 4 Overview

- We provide a series of network packet traces (**pcaps**)
  - **Your job:** write scripts to analyze them!



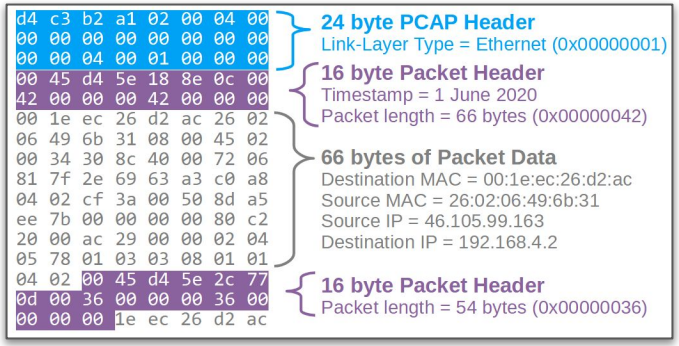
Project 4 Overview

- We provide a series of network packet traces (**pcaps**)
  - **Your job:** write scripts to analyze them!
- **Part 1:** detecting **network attacks**
  - Password cracking, port scanning, SYN floods
- **Part 2:** stealing **sensitive information**
  - Unencrypted credentials, browsing history
  - **Extra credit:** stealing transfered files



Project 4 Overview

- We provide a series of network packet traces (**pcaps**)
  - **Your job:** write scripts to analyze them!
- **Part 1:** detecting **network attacks**
  - Password cracking, port scanning, SYN floods
- **Part 2:** stealing **sensitive information**
  - Unencrypted credentials, browsing history
  - **Extra credit:** stealing transfered files
- You will use Python 3's **Scapy** library
  - A huge and powerful packet analysis API…
  - But we'll really only use **a few parts** of it



Scapy Fundamentals

- Python API for programmatic packet capture and analysis
  - Think of it as **“Wireshark in API form”**

Scapy Fundamentals

- Python API for programmatic packet capture and analysis
  - Think of it as **"Wireshark in API form"**
- We provide **skeleton code** template
  - Sets-up the packet parsing workflow

```
#!/usr/bin/python3
import logging
logging.getLogger("scapy.runtime").setLevel(logging.ERROR)
from scapy.all import *
import re

def parsePacket(packet):
    if not packet.haslayer("TCP"): return
    # ------------------------------------------
    # TODO: finish implementing parsePacket()!
    # ------------------------------------------
    return

if __name__ == "__main__":
    for packet in rdpcap(sys.argv[1]):
        parsePacket(packet)
```

Scapy Fundamentals

- Python API for programmatic packet capture and analysis
  - Think of it as **"Wireshark in API form"**
- We provide **skeleton code** template
  - Sets-up the packet parsing workflow
  - **Your job:** finish implementing the function `parsePacket()`

```
#!/usr/bin/python3
import logging
logging.getLogger("scapy.runtime").setLevel(logging.ERROR)
from scapy.all import *
import re

def parsePacket(packet):
    if not packet.haslayer("TCP"): return
    # ----------------------------------------
    # TODO: finish implementing parsePacket()!
    # ----------------------------------------
    return

if __name__ == "__main__":
    for packet in rdpcap(sys.argv[1]):
        parsePacket(packet)
```

Scapy Fundamentals

- Python API for programmatic packet capture and analysis
  - Think of it as **"Wireshark in API form"**
- We provide **skeleton code** template
  - Sets-up the packet parsing workflow
  - **Your job:** finish implementing the function `parsePacket()`
- You may also add **additional code**
  - E.g., global variables or data structures
  - E.g., printing functionality in `main()`

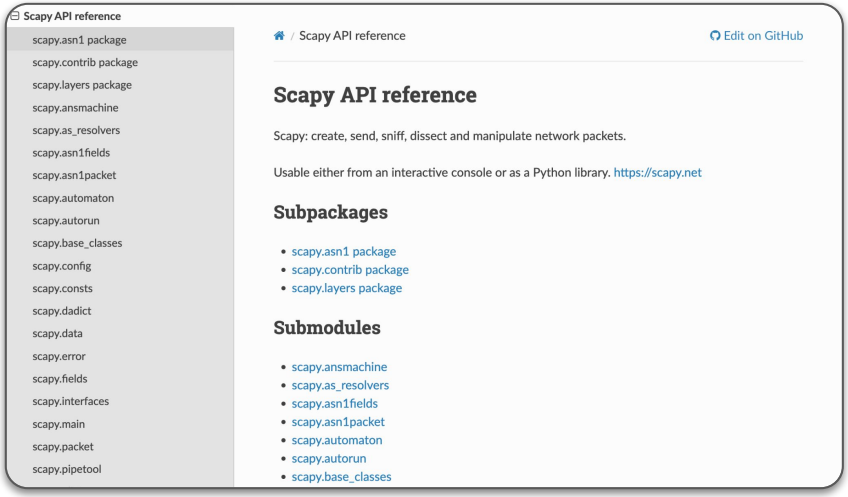
```
#!/usr/bin/python3
import logging
logging.getLogger("scapy.runtime").setLevel(logging.ERROR)
from scapy.all import *
import re

def parsePacket(packet):
    if not packet.haslayer("TCP"): return
    # -----------------------------------------
    # TODO: finish implementing parsePacket()!
    # -----------------------------------------
    return

if __name__ == "__main__":
    for packet in rdpcap(sys.argv[1]):
        parsePacket(packet)
```

Scapy Fundamentals

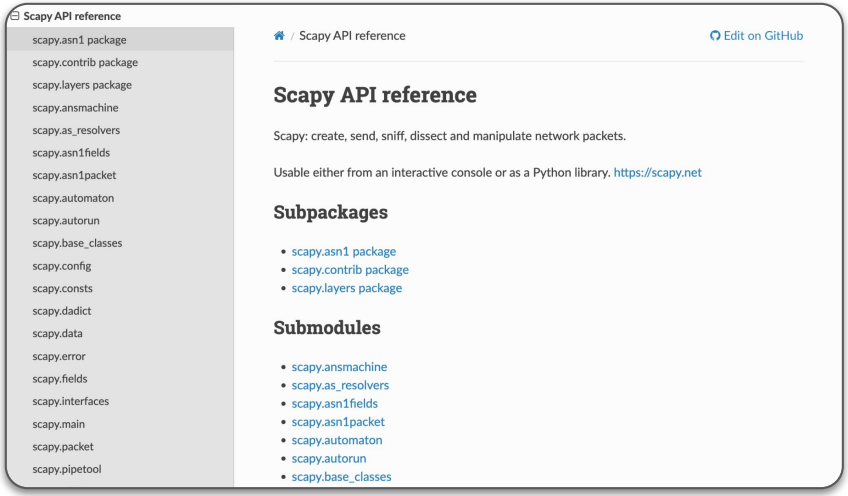
- Only a few things you'll need...



Scapy Fundamentals

- Only a few things you'll need…
  - Get a packet's **TCP flags**:

```
packet["TCP"].flags
```



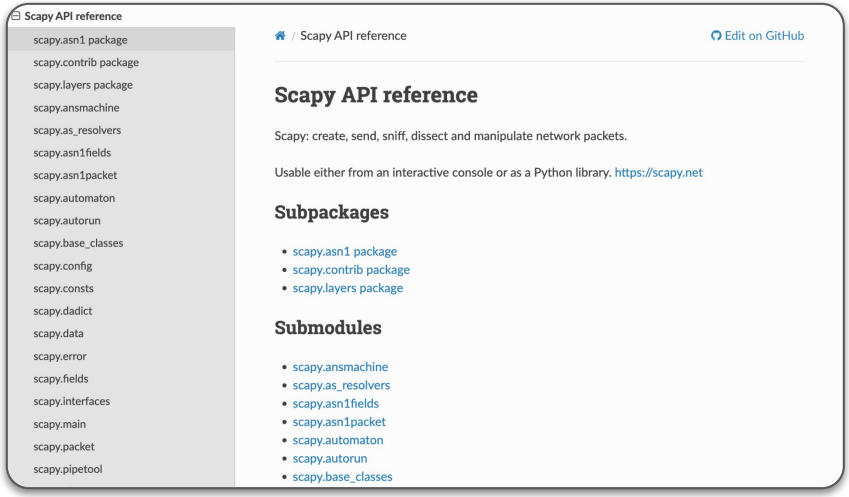
Scapy Fundamentals

- Only a few things you'll need...
  - Get a packet's **TCP flags**:

```
packet["TCP"].flags
```

  - Get a packet's **destination port**

```
packet["TCP"].dport
```



Scapy Fundamentals

- Only a few things you'll need…
  - Get a packet's **TCP flags**:

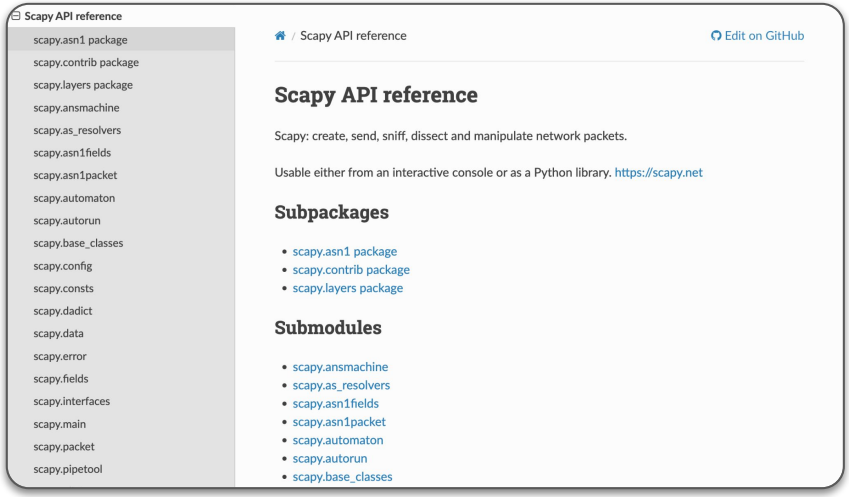
```
packet["TCP"].flags
```

  - Get a packet's **destination port**

```
packet["TCP"].dport
```

  - Get a packet's **source IP address**

```
packet["IP"].src
```



Scapy Fundamentals

- Only a few things you'll need...
  - Get a packet's **TCP flags**:

```
packet["TCP"].flags
```

  - Get a packet's **destination port**

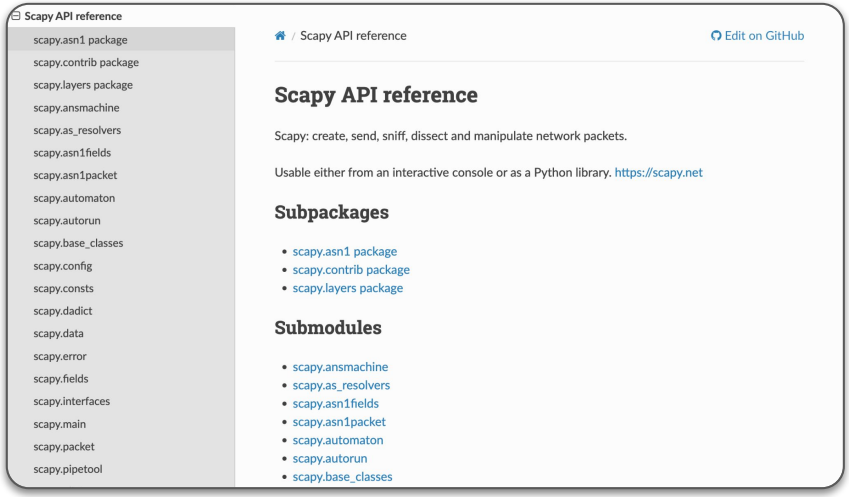
```
packet["TCP"].dport
```

  - Get a packet's **source IP address**

```
packet["IP"].src
```

  - Get a packet's TCP **payload**:

```
bytes(packet["TCP"].payload).decode('utf-8','replace')
```



Scapy Fundamentals

- Only a few things you'll need...
  - Get a packet's **TCP flags**:

```
packet["TCP"].flags
```

  - Get a pa

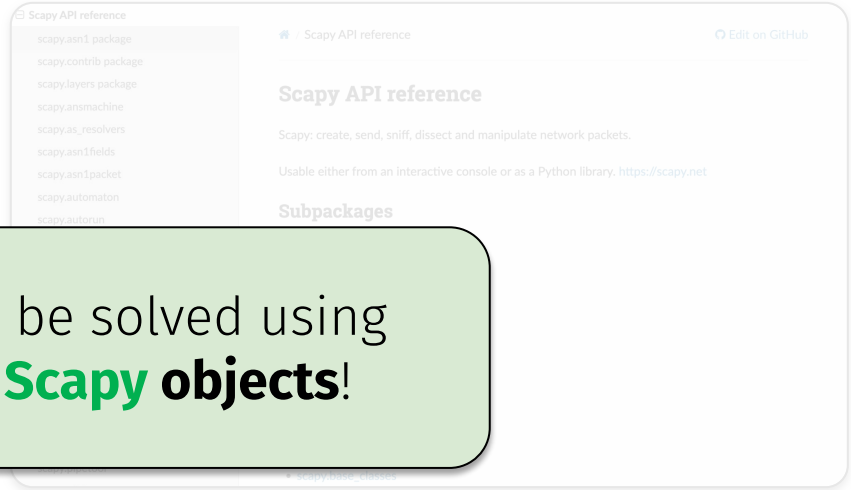
```
pack
```

  - Get a pa

```
packet["IP"].src
```

  - Get a packet's TCP **payload**:

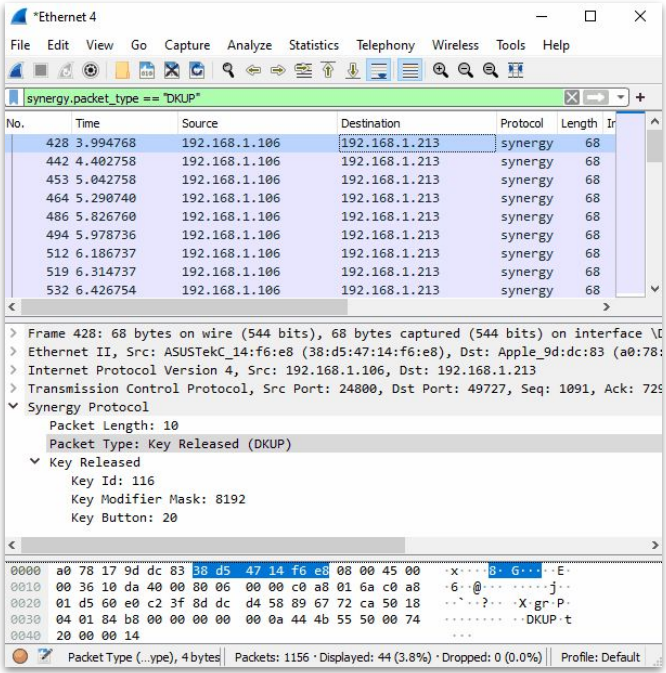
```
bytes(packet["TCP"].load).decode('utf-8','replace')
```



All of the targets can be solved using a few **fundamental Scapy objects**!

Suggested Workflow

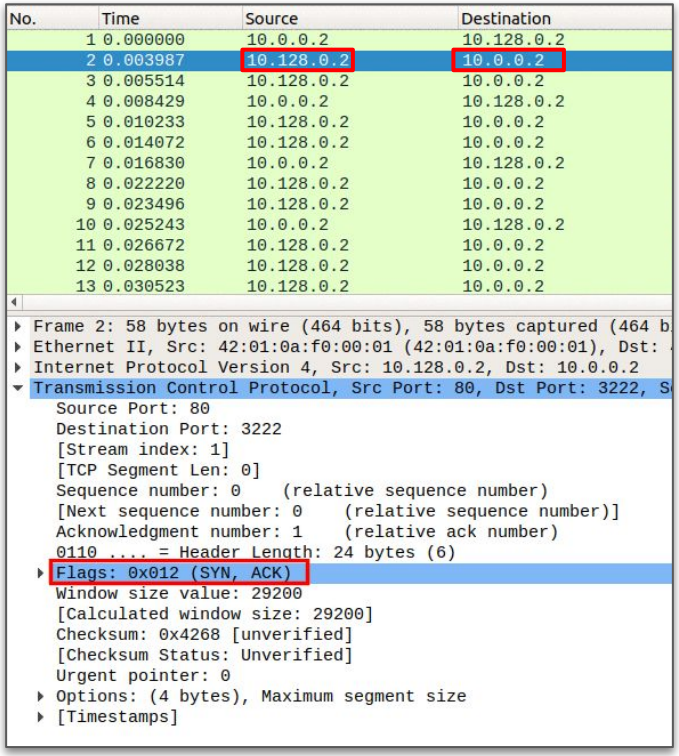
- Before you start writing a **Scapy** script, inspect the trace *manually* via **Wireshark**
  - Super helpful for viewing a packet's contents
  - Use this to bootstrap your script's approach!



Suggested Workflow

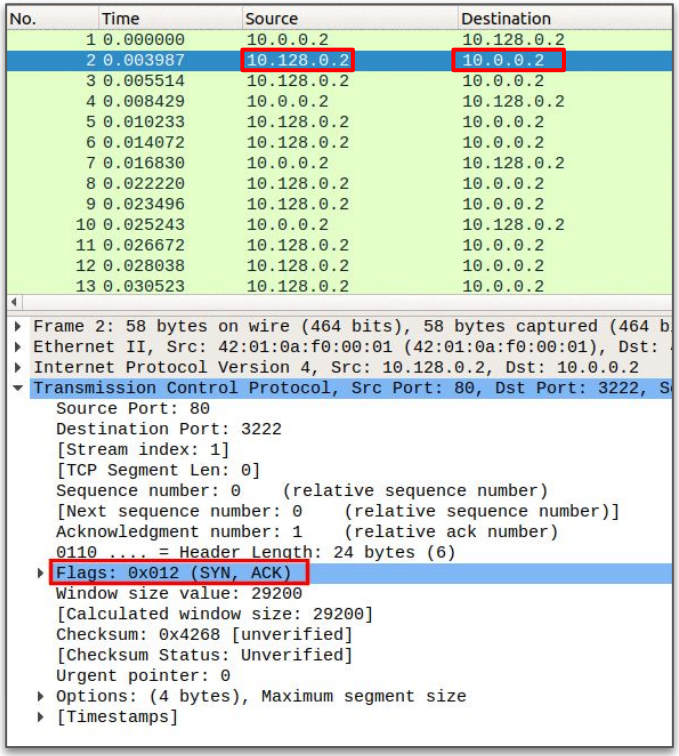
- Before you start writing a **Scapy** script, inspect the trace *manually* via **Wireshark**
  - Super helpful for viewing a packet's contents
  - Use this to bootstrap your script's approach!

- For each target, answer the following:
  - What **packet fields** matter?
  - How to **extract** relevant data?
  - How to **store and process** this data?



Suggested Workflow

- Before you start writing a **Scapy** script, inspect the trace *manually* via **Wireshark**
  - Super helpful for viewing a packet's contents
  - Use this to bootstrap your script's approach!
- For each target, answer the following:
  - What **packet fields** matter?
  - How to **extract** relevant data?
  - How to **store and process** this data?
- Finalize your **high-level game plan** first!
  - Then start developing your solution scripts!



Questions?

Next time on CS 4440...

Election Cybersecurity