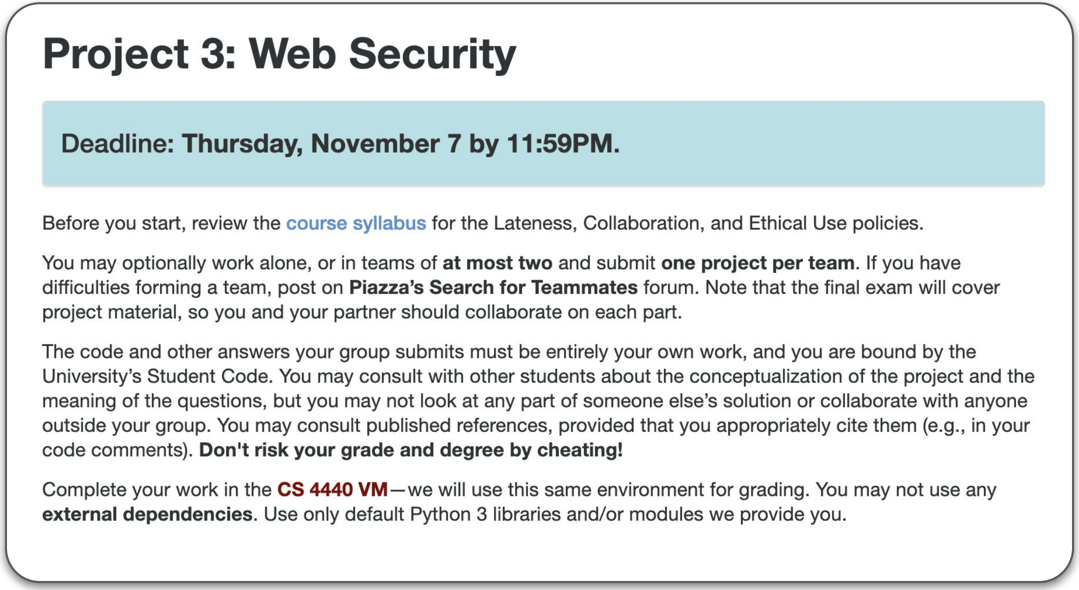
Week 11: Lecture B

Security in Practice: Tor

Thursday, November 7, 2024

Announcements

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



Project 3: Web Security

Deadline: Thursday, November 7 by 11:59PM.

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

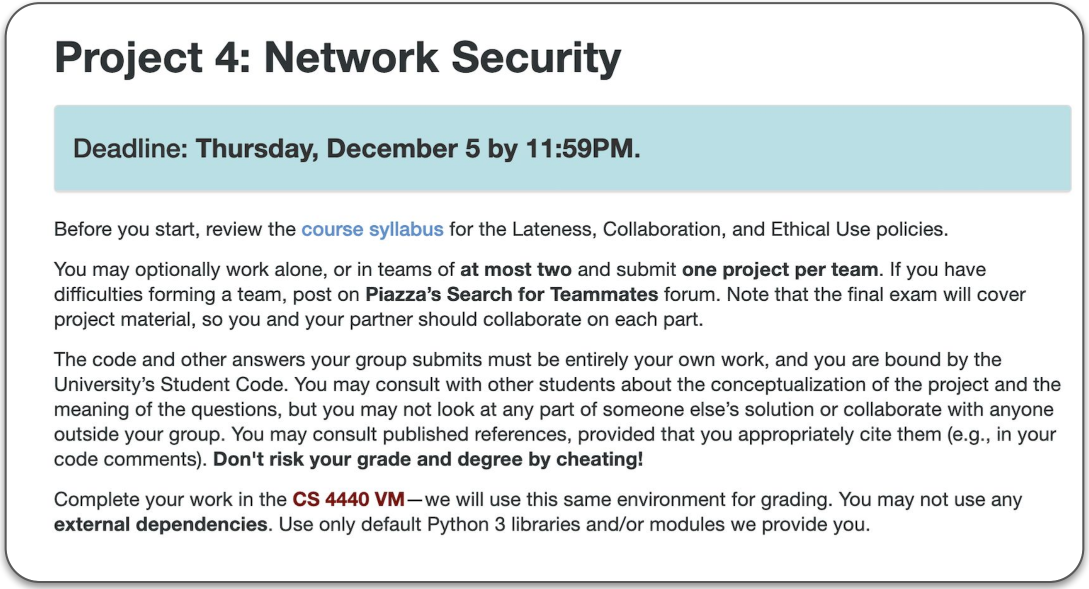
You may optionally work alone, or in teams of **at most two** and submit **one project per team**. If you have difficulties forming a team, post on **Piazza's Search for Teammates** forum. Note that the final exam will cover project material, so you and your partner should collaborate on each part.

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 5th by 11:59PM



Project 4: Network Security

Deadline: Thursday, December 5 by 11:59PM.

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

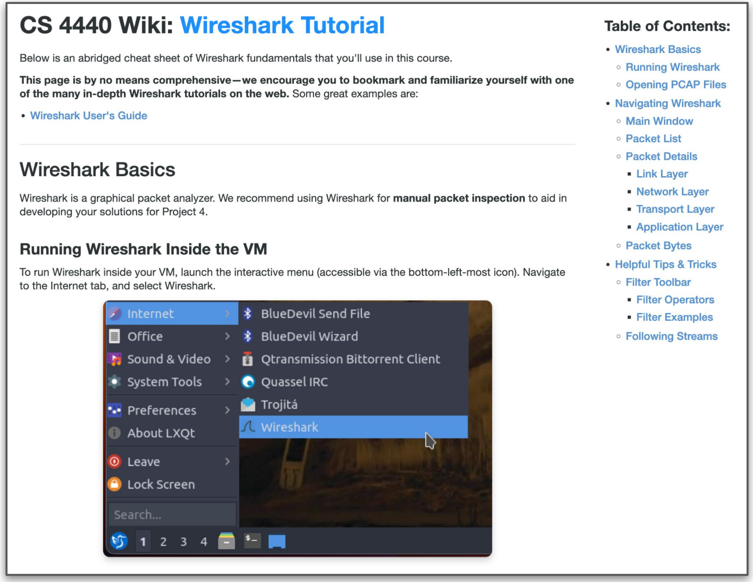
You may optionally work alone, or in teams of **at most two** and submit **one project per team**. If you have difficulties forming a team, post on **Piazza's Search for Teammates** forum. Note that the final exam will cover project material, so you and your partner should collaborate on each part.

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

- New **Wiki pages** to help you on Project 4:



CS 4440 Wiki: Wireshark Tutorial

Below is an abridged cheat sheet of Wireshark fundamentals that you'll use in this course.

This page is by no means comprehensive—we encourage you to bookmark and familiarize yourself with one of the many in-depth Wireshark tutorials on the web. Some great examples are:

- Wireshark User's Guide

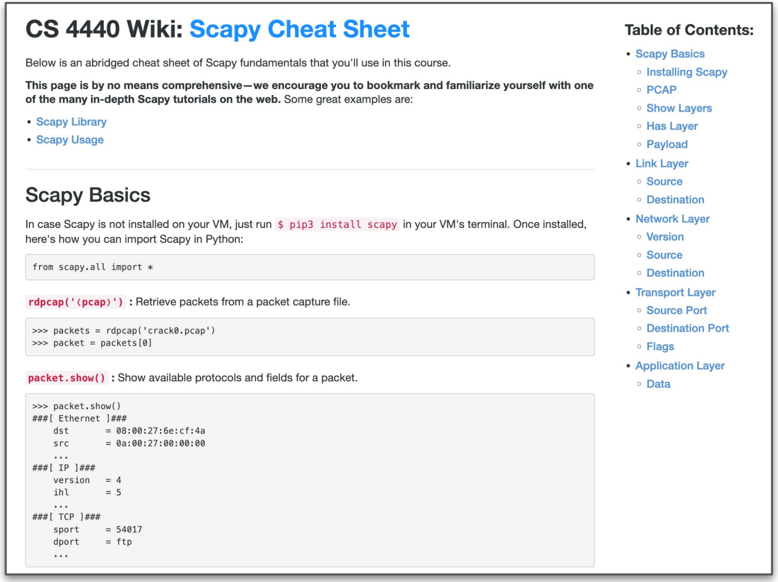
Wireshark Basics

Wireshark is a graphical packet analyzer. We recommend using Wireshark for **manual packet inspection** to aid in developing your solutions for Project 4.

Running Wireshark Inside the VM

To run Wireshark inside your VM, launch the interactive menu (accessible via the bottom-left-most icon). Navigate to the Internet tab, and select Wireshark.

Table of Contents:



CS 4440 Wiki: Scapy Cheat Sheet

Below is an abridged cheat sheet of Scapy fundamentals that you'll use in this course.

This page is by no means comprehensive—we encourage you to bookmark and familiarize yourself with one of the many in-depth Scapy tutorials on the web. Some great examples are:

- Scapy Library
- Scapy Usage

Scapy Basics

In case Scapy is not installed on your VM, just run `$ pip3 install scapy` in your VM's terminal. Once installed, here's how you can import Scapy in Python:

```
from scapy.all import *
```

`rdpcap('‹pcap›')` : Retrieve packets from a packet capture file.

```
>>> packets = rdpcap('crack0.pcap')
>>> packet = packets[0]
```

`packet.show()` : Show available protocols and fields for a packet.

```
>>> packet.show()
###[ Ethernet ]###
    dst       = 08:00:27:6e:cf:4a
    src       = 0a:00:27:00:00:00
    ...
###[ IP ]###
    version   = 4
    ihl       = 5
    ...
###[ TCP ]###
    sport     = 54017
    dport     = ftp
    ...
```

Table of Contents:

Announcements

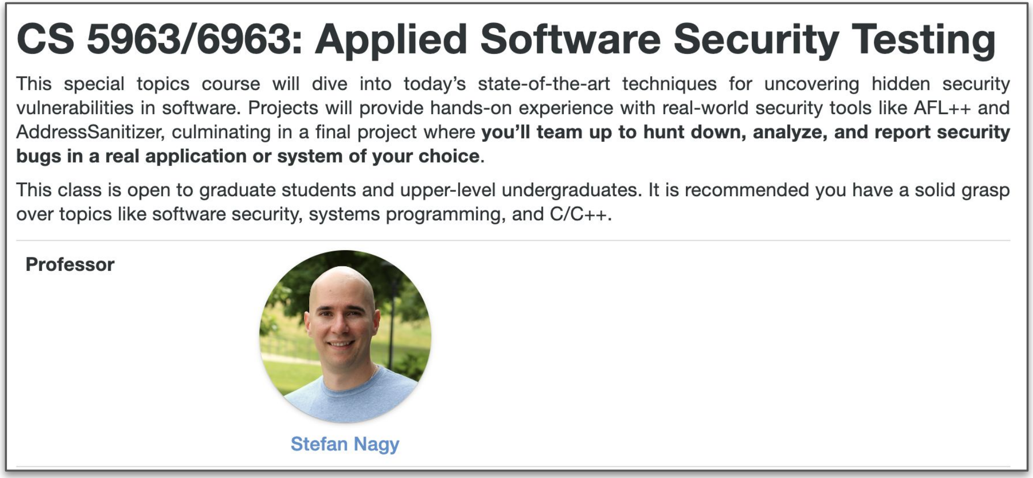
utahsec

See Discord for meeting info!

utahsec.cs.utah.edu

Interested in fuzzing?

- **Spring 2025: CS 5963/6963: 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!
  - https://cs.utah.edu/~snagy/courses/cs5963/



CS 5963/6963: 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. Projects will provide hands-on experience with real-world security tools like AFL++ and AddressSanitizer, culminating in a final project where **you'll team up to hunt down, analyze, and report security bugs in a real application or system 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++.

Professor

Stefan Nagy

Questions?

Last time on CS 4440…

Authentication

Multi-factor Authentication

One-time Passwords

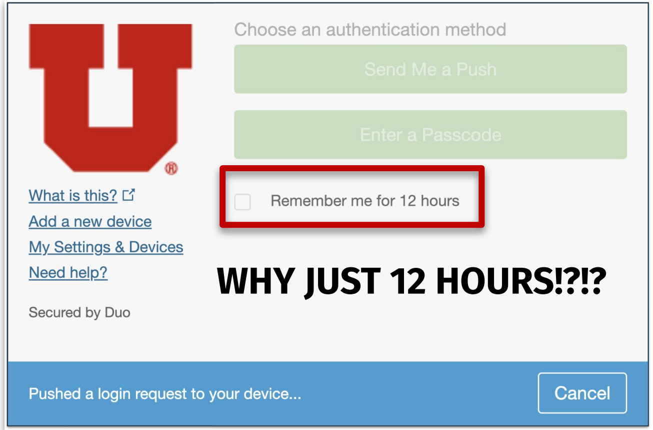
Secure Password Storage

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

I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.

Sorry, but your password must contain an uppercase letter, a number, a hieroglyph, a feather from a hawk and the blood of a unicorn.
someecards
user card


Choose an authentication method

Send Me a Push

Enter a Passcode

□ Remember me for 12 hours

What is this?

Add a new device

My Settings & Devices

Need help?

Secured by Duo

Pushed a login request to your device...

Cancel

WHY JUST 12 HOURS!?!?

What is authentication?

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

What is authentication?

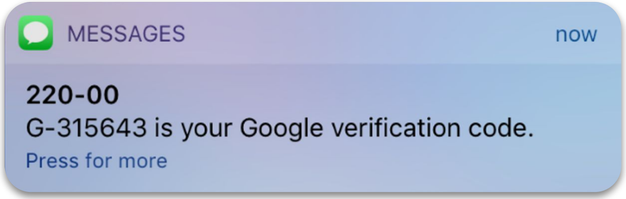
- **Goal:** establish trust in the **identity** of another communicating party

- **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 **???**

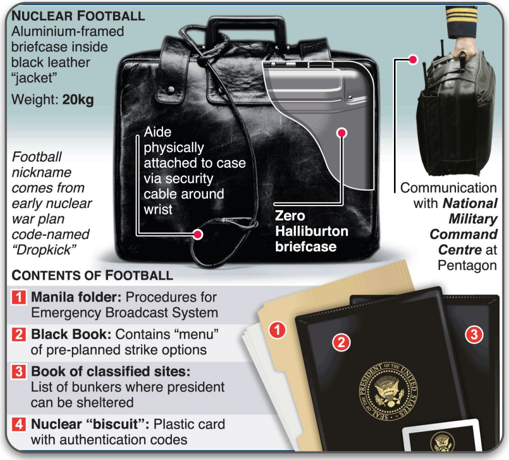


MESSAGES now

220-00

G-315643 is your Google verification code.

Press for more



NUCLEAR FOOTBALL Aluminium-framed briefcase inside black leather "jacket"

Weight: **20kg**

Football nickname comes from early nuclear war plan code-named "Dropkick"

Aide physically attached to case via security cable around wrist

Zero Halliburton briefcase

Communication with ***National Military Command Centre*** at Pentagon

CONTENTS OF FOOTBALL

1 **Manila folder:** Procedures for Emergency Broadcast System

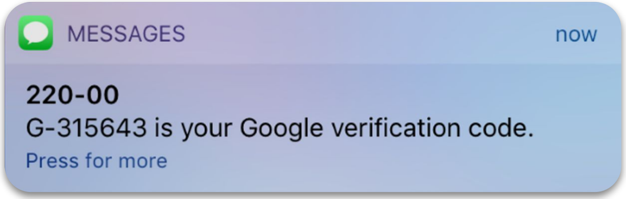
2 **Black Book:** Contains "menu" of pre-planned strike options

3 **Book of classified sites:** List of bunkers where president can be sheltered

4 **Nuclear "biscuit":** Plastic card with authentication codes

The Three Factors of Authentication

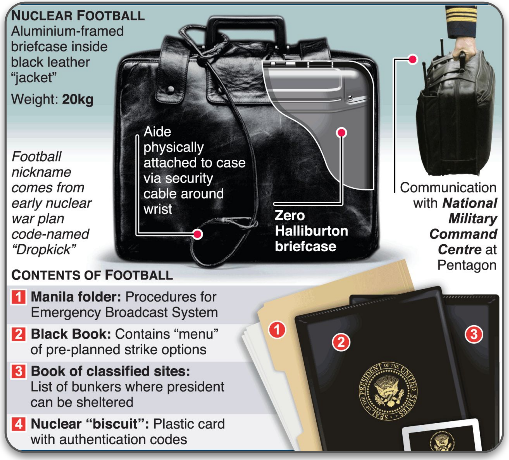
- Something you **have**
  - Smartphone
  - Laptop
  - Email account
- Something you **are**
  - Your fingerprint
  - Your DNA
  - Your iris, retina
- Something you **know**
  - Account password, banking PIN number
  - Nuclear strike challenge-response code


MESSAGES now

220-00

G-315643 is your Google verification code.

Press for more


NUCLEAR FOOTBALL Aluminium-framed briefcase inside black leather "jacket"

Weight: **20kg**

Football nickname comes from early nuclear war plan code-named "Dropkick"

Aide physically attached to case via security cable around wrist

Zero Halliburton briefcase

Communication with ***National Military Command Centre*** at Pentagon

CONTENTS OF FOOTBALL

1 **Manila folder:** Procedures for Emergency Broadcast System

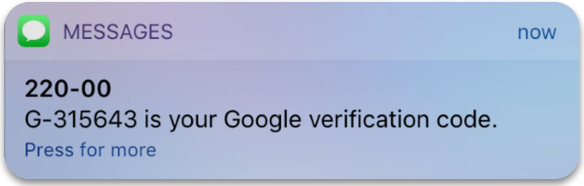
2 **Black Book:** Contains "menu" of pre-planned strike options

3 **Book of classified sites:** List of bunkers where president can be sheltered

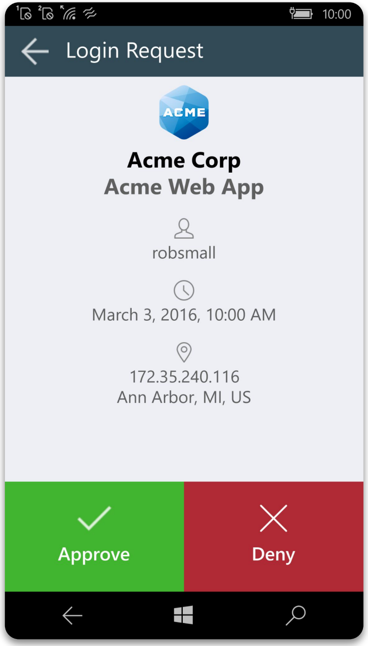
4 **Nuclear "biscuit":** Plastic card with authentication codes

One-time PINs

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


MESSAGES now

220-00
G-315643 is your Google verification code.
Press for more



Login Request

Acme Corp
Acme Web App

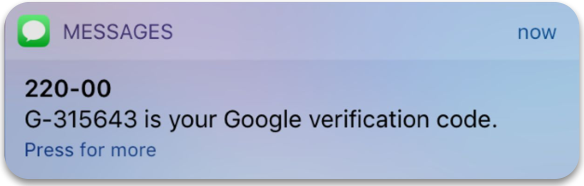
robsmall

March 3, 2016, 10:00 AM

172.35.240.116
Ann Arbor, MI, US

One-time PINs

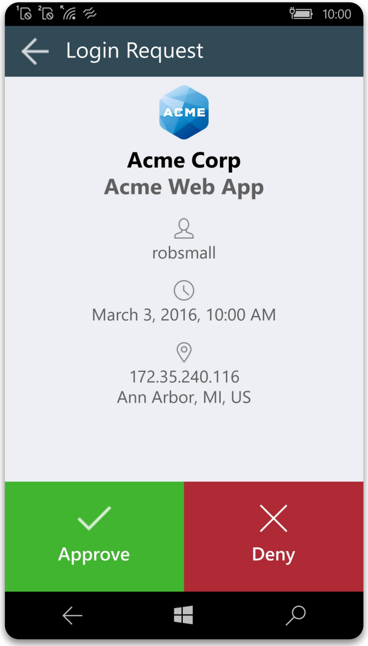
- **Provides proof of: possession**
  - A PIN/code valid for only **one** login session or transaction
- **Delivering** One-time PINs:
  - **???**


MESSAGES now

220-00

G-315643 is your Google verification code.

Press for more


10:00

Login Request

Acme Corp

Acme Web App

robsmall

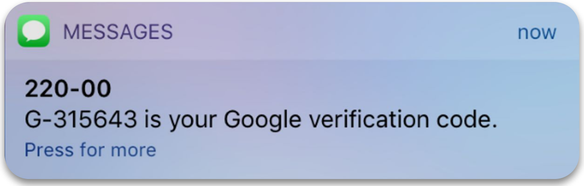
March 3, 2016, 10:00 AM

172.35.240.116

Ann Arbor, MI, US

One-time PINs

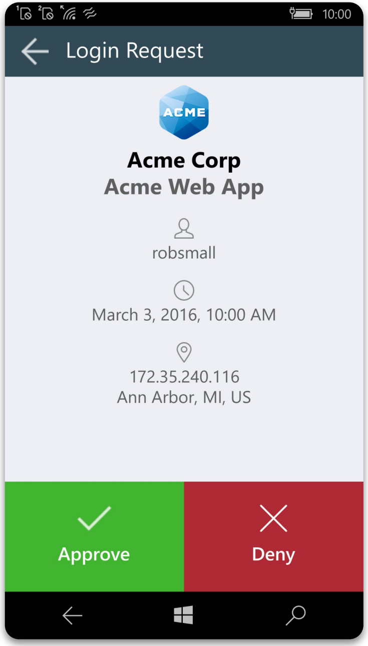
- **Provides proof of: possession**
  - A PIN/code valid for only **one** login session or transaction
- **Delivering** One-time PINs:
  - **SMS**
    - Phone call
    - Text message
  - **Hardware**
    - Yubico YubiKey
    - RSA SecureID
  - **Application**
    - DUO Mobile
    - Google authenticator


MESSAGES now

220-00

G-315643 is your Google verification code.

Press for more


10:00

Login Request

Acme Corp

Acme Web App

robsmall

March 3, 2016, 10:00 AM

172.35.240.116

Ann Arbor, MI, US

Approve

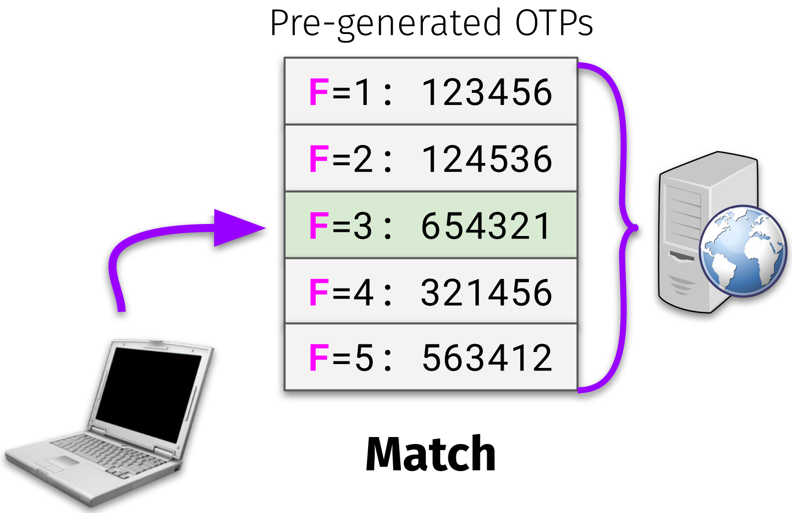
Deny

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



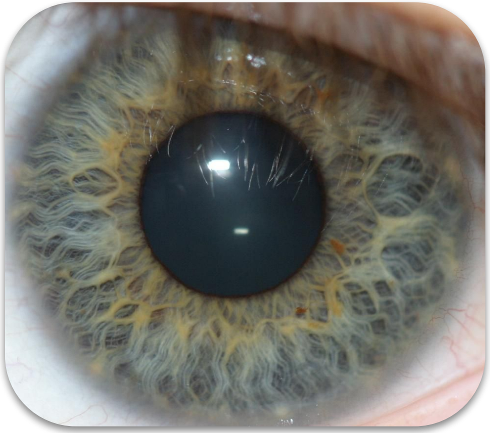
Pre-generated OTPs

F=1 :	123456
F=2 :	124536
F=3 :	654321
F=4 :	321456
F=5 :	563412

Match

Biometrics

- **Provides proof of ???**

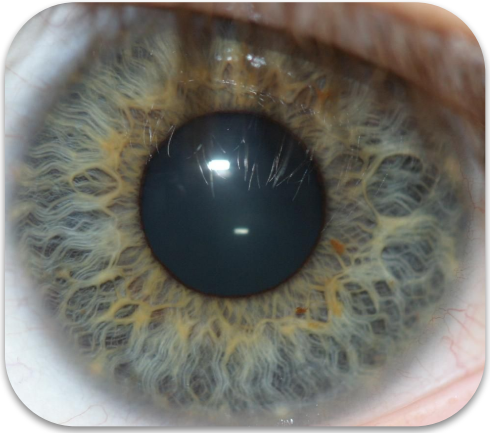


Biometrics

- **Provides proof of physical identity**

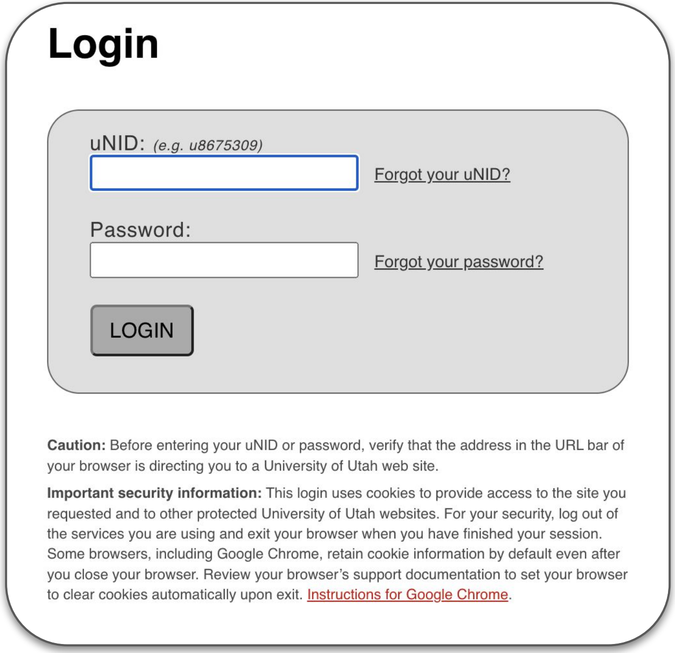
- **Something unique to you** (hopefully)
  - Fingerprint, iris, retina, DNA

- 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 ???**



Login

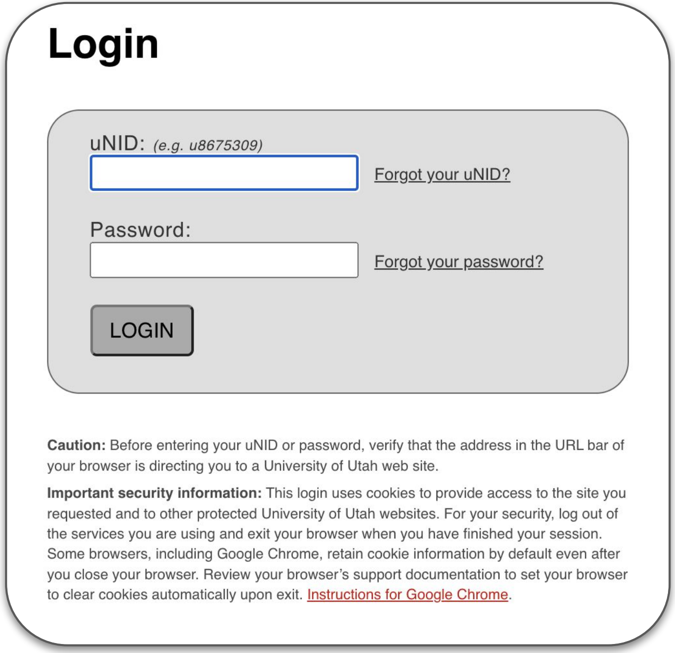
uNID: *(e.g. u8675309)*

Caution: Before entering your uNID or password, verify that the address in the URL bar of your browser is directing you to a University of Utah web site.

Important security information: This login uses cookies to provide access to the site you requested and to other protected University of Utah websites. For your security, log out of the services you are using and exit your browser when you have finished your session. Some browsers, including Google Chrome, retain cookie information by default even after you close your browser. Review your browser's support documentation to set your browser to clear cookies automatically upon exit. Instructions for Google Chrome.

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?**
  - ???

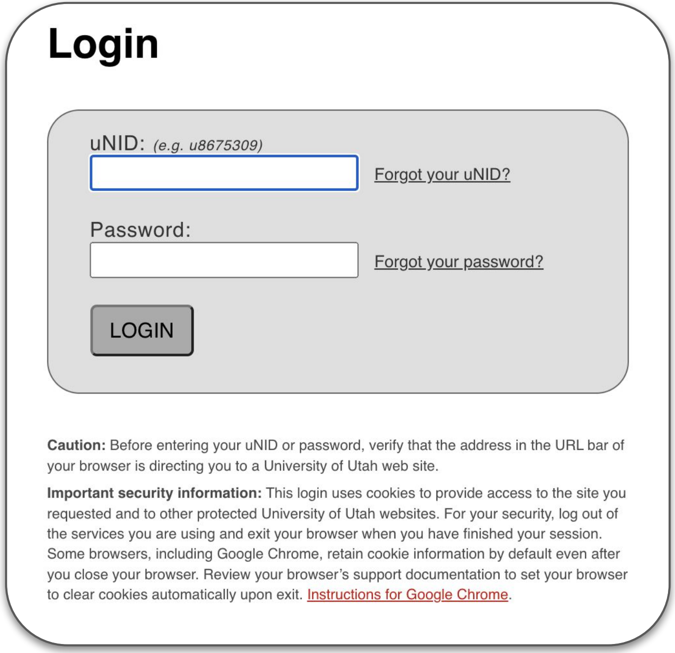


Caution: Before entering your uNID or password, verify that the address in the URL bar of your browser is directing you to a University of Utah web site.

Important security information: This login uses cookies to provide access to the site you requested and to other protected University of Utah websites. For your security, log out of the services you are using and exit your browser when you have finished your session. Some browsers, including Google Chrome, retain cookie information by default even after you close your browser. Review your browser's support documentation to set your browser to clear cookies automatically upon exit. Instructions for Google Chrome.

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!**



Login

uNID: *(e.g. u8675309)*

Forgot your uNID?

Password:

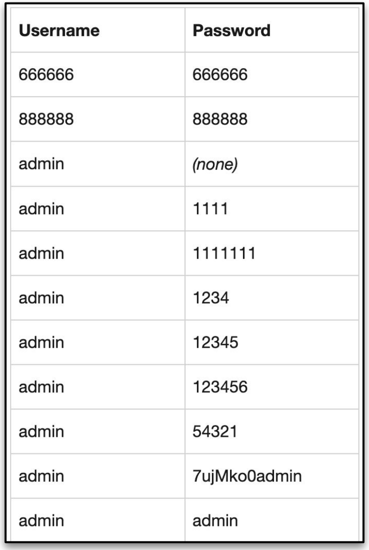
Forgot your password?

Caution: Before entering your uNID or password, verify that the address in the URL bar of your browser is directing you to a University of Utah web site.

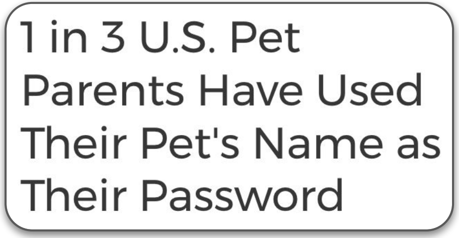
Important security information: This login uses cookies to provide access to the site you requested and to other protected University of Utah websites. For your security, log out of the services you are using and exit your browser when you have finished your session. Some browsers, including Google Chrome, retain cookie information by default even after you close your browser. Review your browser's support documentation to set your browser to clear cookies automatically upon exit. Instructions for Google Chrome.

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

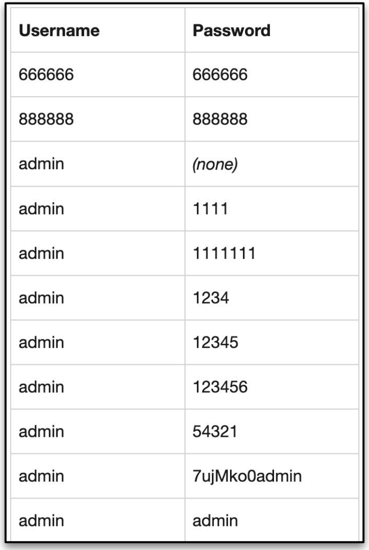


1 in 3 U.S. Pet Parents Have Used Their Pet's Name as Their Password

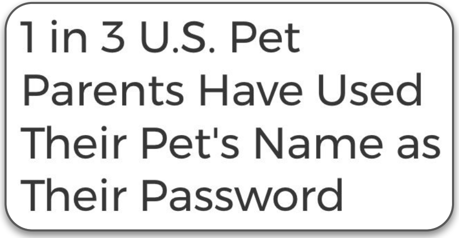
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.

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

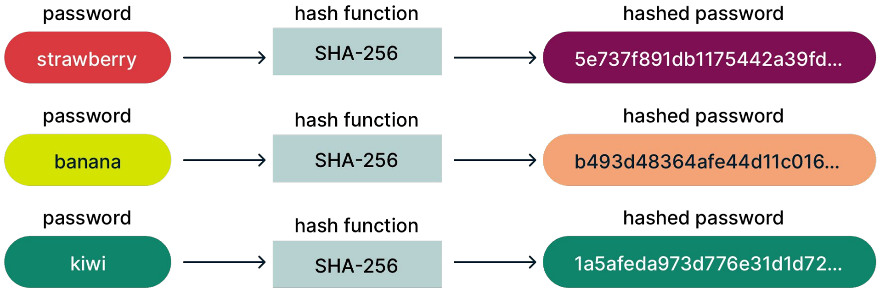
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.

Better Server-side Password Storage

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

- Why are **weak** hash functions bad?
  - **???**

- Why are **fast** hash functions bad?
  - **???**

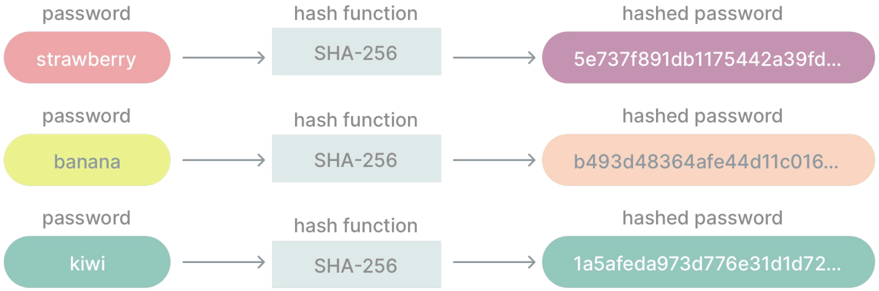


Better Server-side Password Storage

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

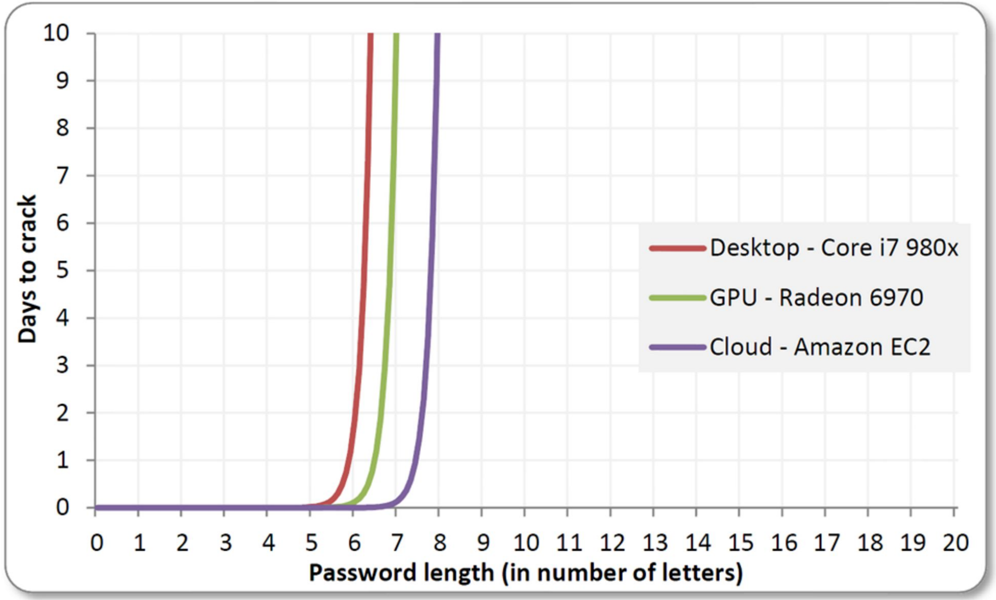
- Why are **weak** hash functions bad?
  - **Collision and pre-image attacks** = attacker easily finds working password

- 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!



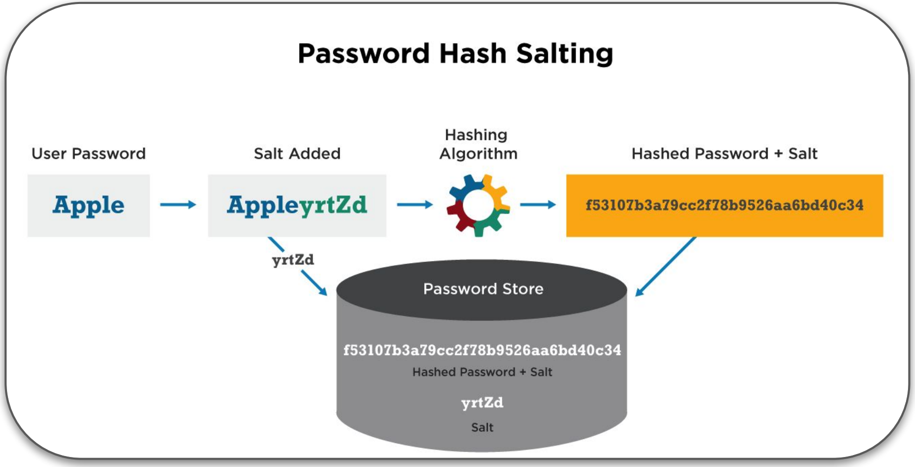
Better Server-side Password Storage

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

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**)

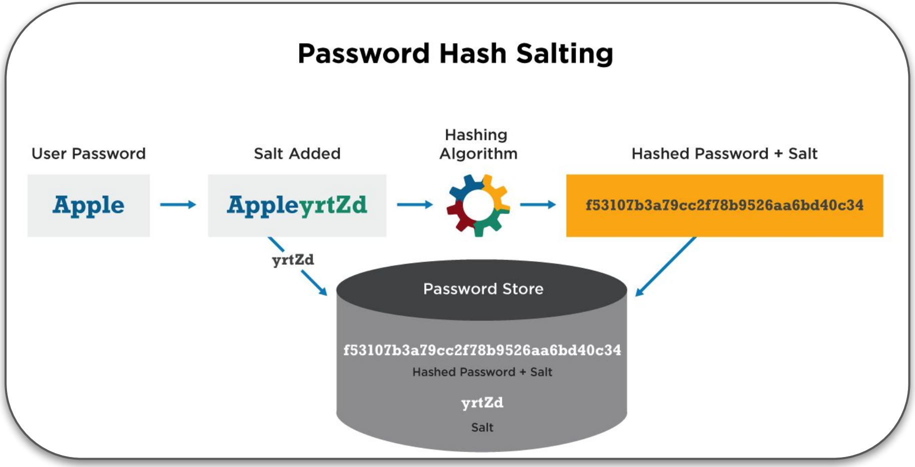
- **Salted passwords:**
  - Add **extra data** when generating hash
  - **Goal:** same input = different output



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

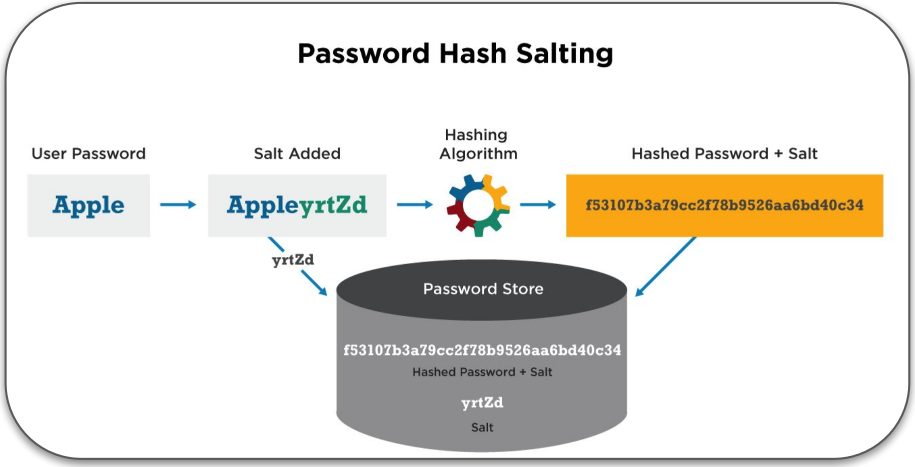


Better Server-side Password Storage

- **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!**

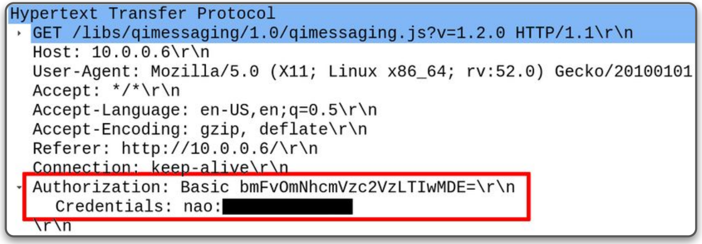


Attack: Client-side Password Theft

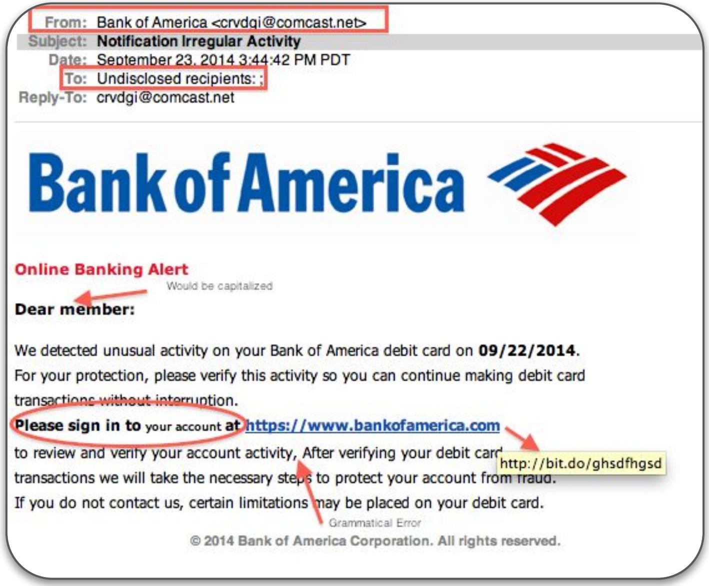
- **How?**

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

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

© 2014 Bank of America Corporation. All rights reserved.

Forgetting and Recovering Passwords

- Security questions:
    - What’s your childhood pet?
- Password recovery email
    - Click here to reset your password!
- Send in plaintext to email
    - Your password is “in$3cur3”

Good security?

Forgetting and Recovering Passwords

- Security questions:
  - What’s your childhood pet?
- Password recovery email
  - Click here to reset your password!
- Send in plaintext to email
  - Your password is “in$3cur3”

Bad security! Attacker might have control of the victim’s **email**!

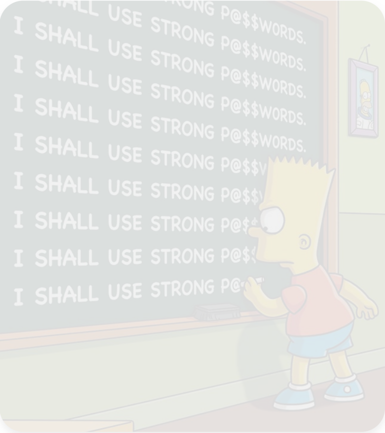
Forgetting and Recovering Passwords

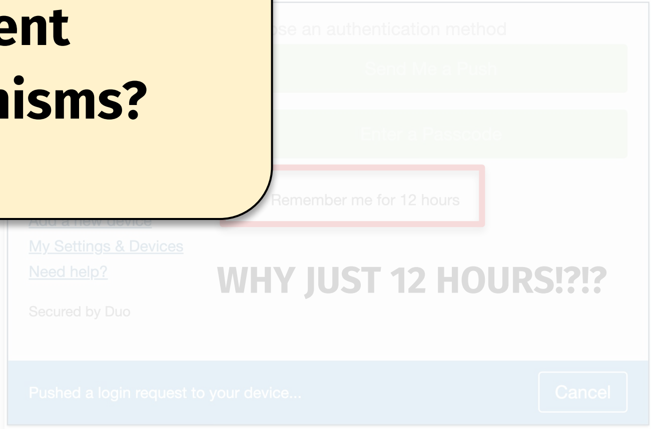
- Security questions:
  - What's your childhood pet?
- Password recovery email
  - Click here to reset your password!
- Send in plaintext to email
  - Your password is "in$3cur3"
- Other approaches:
  - Phone call
  - Session-specific PIN

Bad security! Attacker might have control of the victim's **email**!

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 ann


I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.
I SHALL USE STRONG P@$$WORDS.



se an authentication method

Remember me for 12 hours

Secured by Duo

Pushed a login request to your device...

WHY JUST 12 HOURS!?!?

Trade-offs of different authentication mechanisms?

Trade-offs / challenges of secure auth?

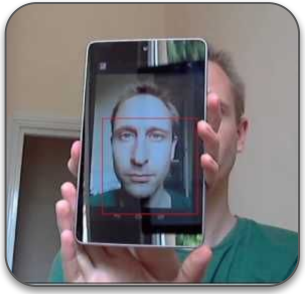
Nobody has responded yet.

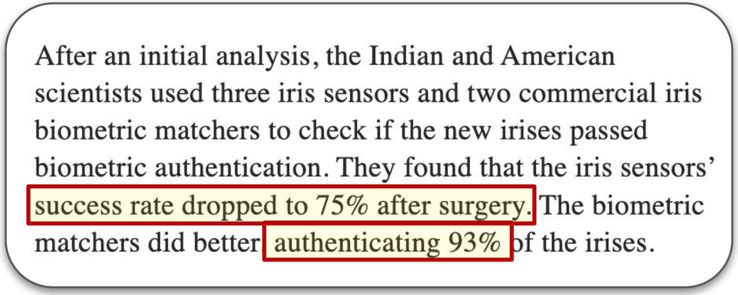
Hang tight! Responses are coming in.

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

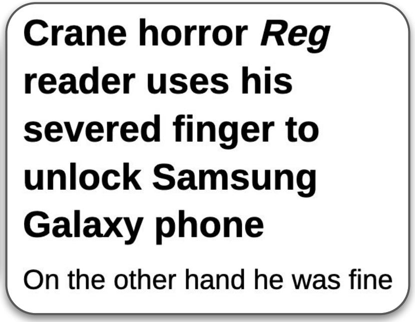
Authentication trade-offs / challenges?

- **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





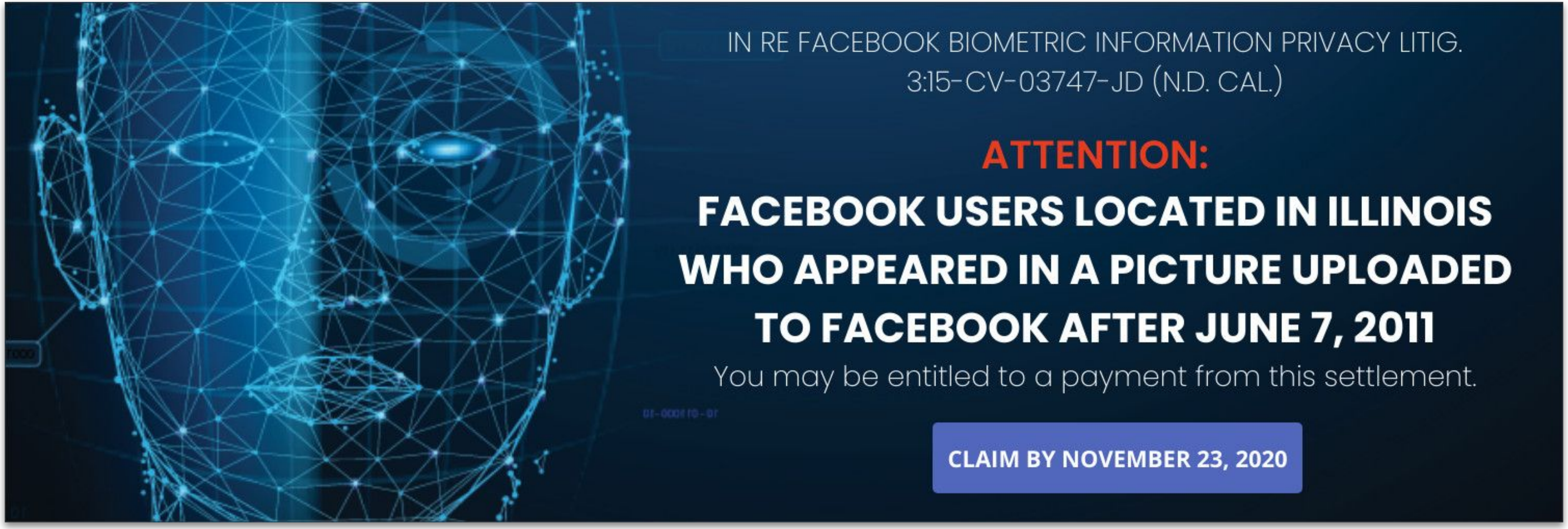
After an initial analysis, the Indian and American scientists used three iris sensors and two commercial iris biometric matchers to check if the new irises passed biometric authentication. They found that the iris sensors' success rate dropped to 75% after surgery. The biometric matchers did better authenticating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

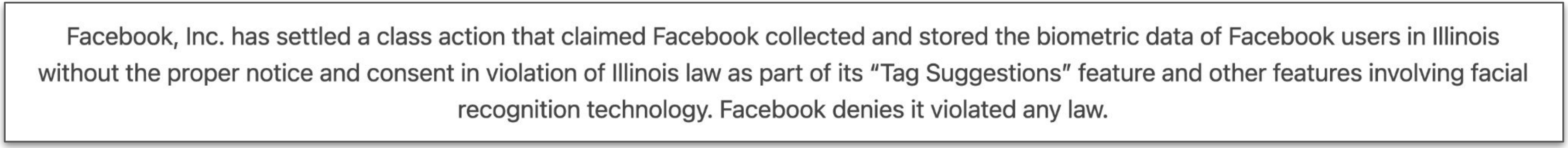


IN RE FACEBOOK BIOMETRIC INFORMATION PRIVACY LITIG.
3:15-CV-03747-JD (N.D. CAL.)

ATTENTION:

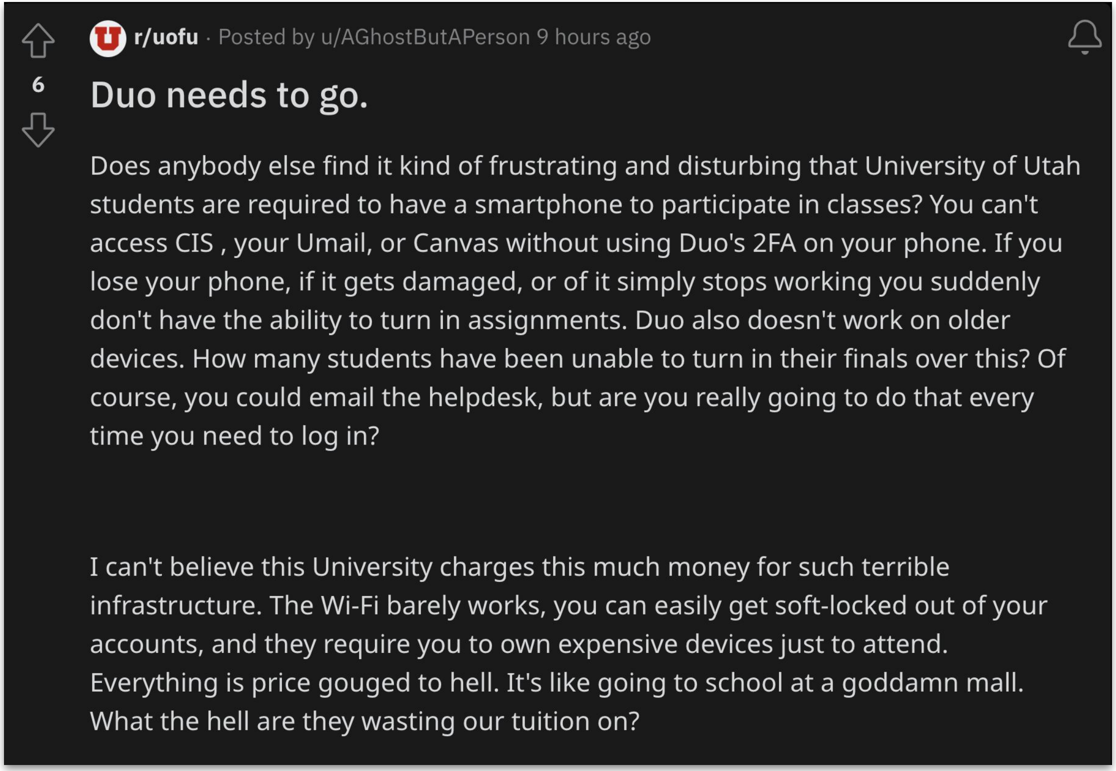
FACEBOOK USERS LOCATED IN ILLINOIS WHO APPEARED IN A PICTURE UPLOADED TO FACEBOOK AFTER JUNE 7, 2011

You may be entitled to a payment from this settlement.



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?



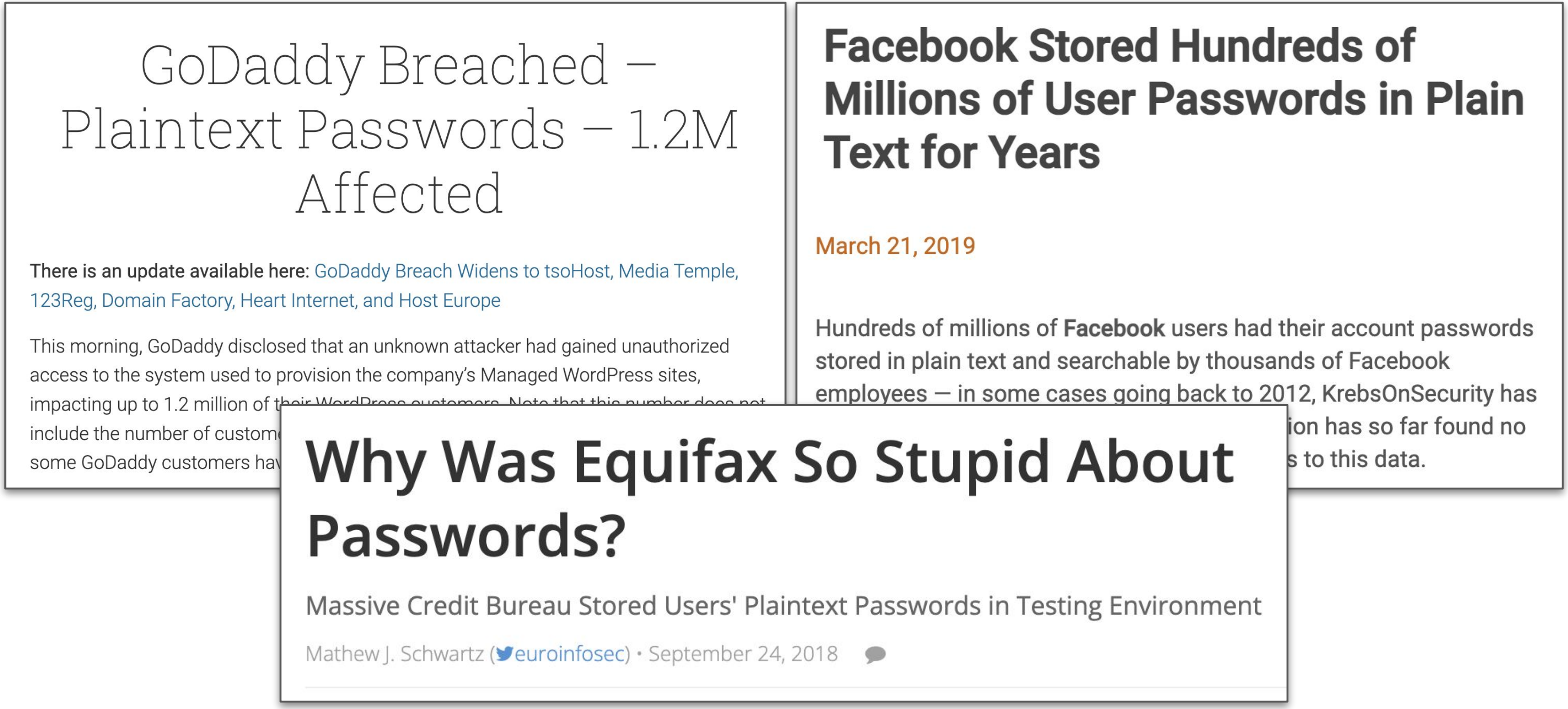
r/uofu · Posted by u/AGhostButAPerson 9 hours ago

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?

Always be vigilant!



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

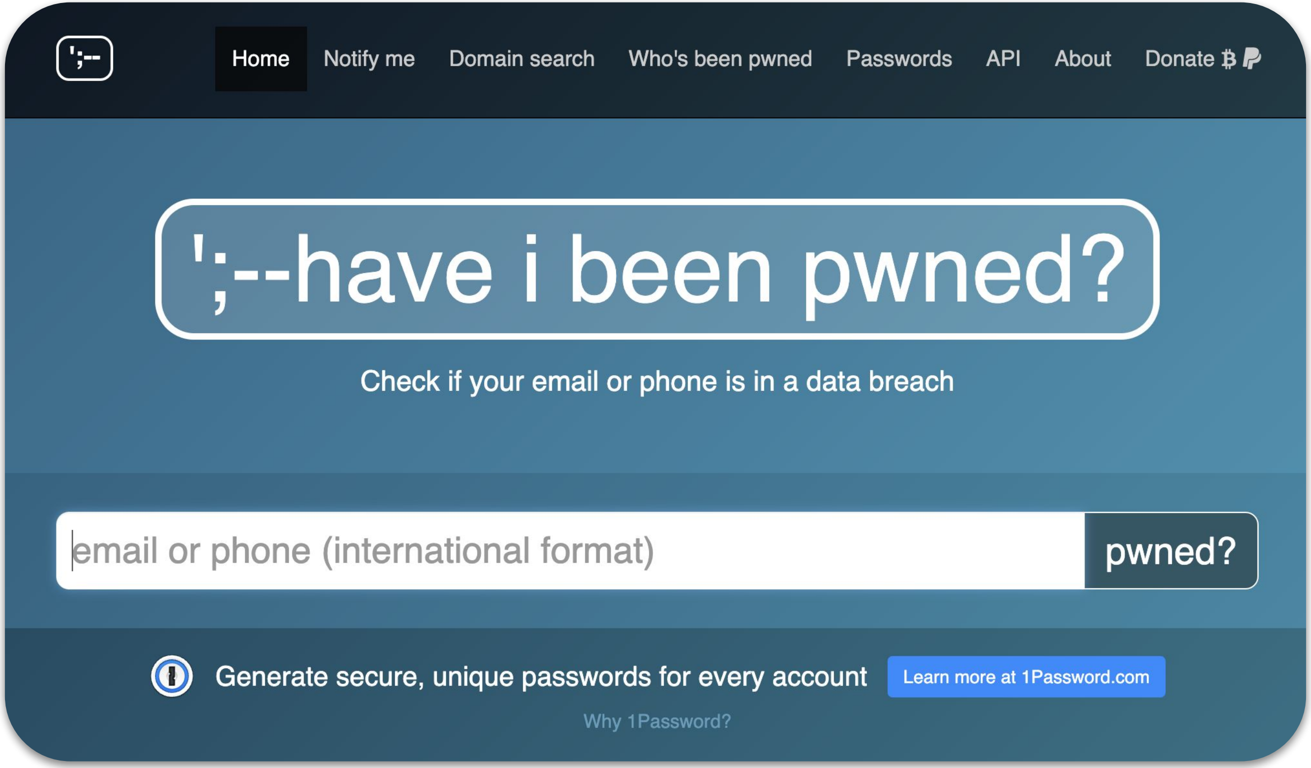
s to this data.

Why Was Equifax So Stupid About Passwords?

Massive Credit Bureau Stored Users' Plaintext Passwords in Testing Environment

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

Always be vigilant!



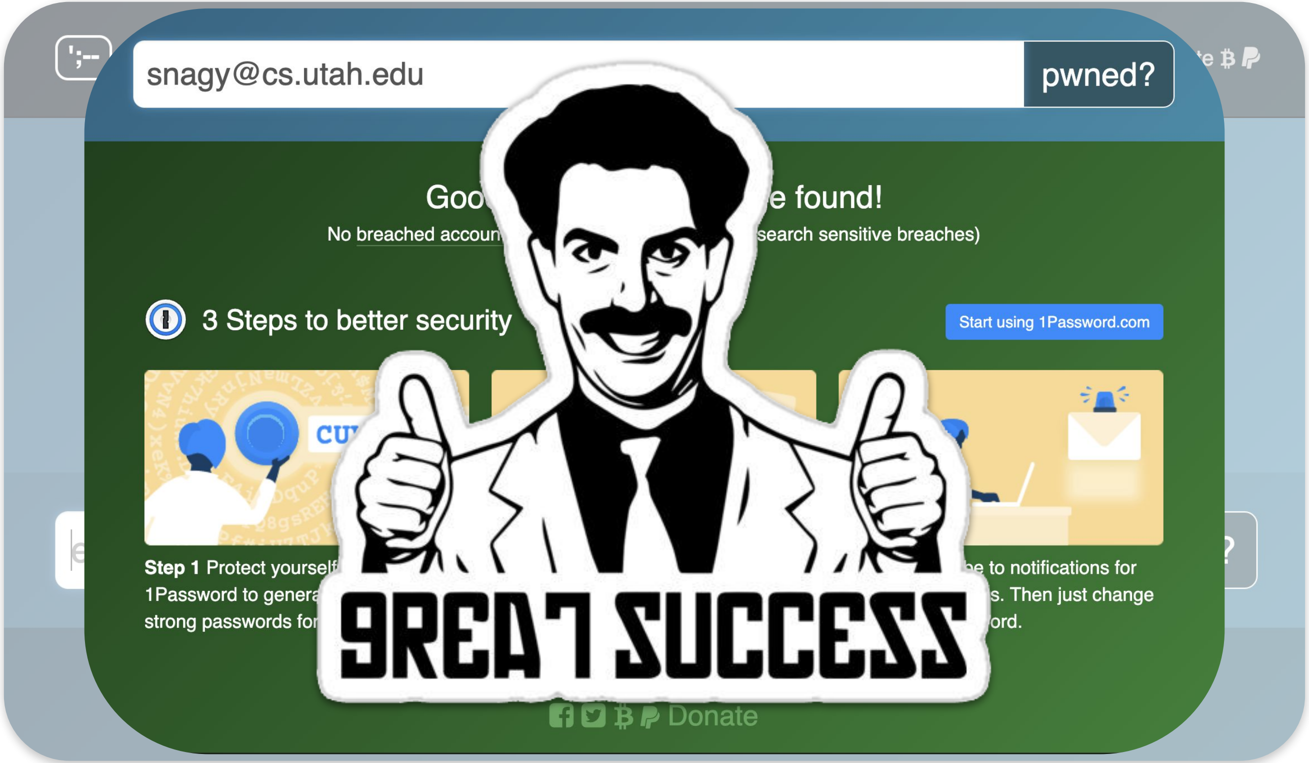
';--have i been pwned?

Check if your email or phone is in a data breach

email or phone (international format)

Generate secure, unique passwords for every account

Always be vigilant!



snagy@cs.utah.edu

pwned?

Goo e found!

No breached accoun search sensitive breaches)

3 Steps to better security

Start using 1Password.com

Step 1 Protect yourself
1Password to genera
strong passwords fo

e to notifications for
s. Then just change
ord.

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

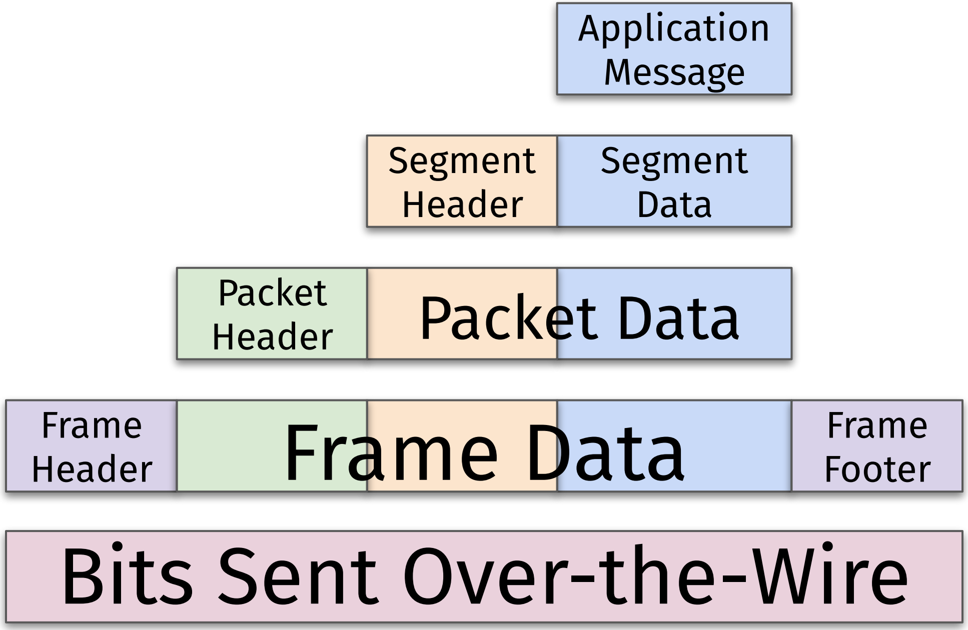
- What is **anonymity**?
  - I want to **say or do something** without the adversary knowing **that it was me** who said/did it

- Versus **confidentiality**?
  - **Confidentiality** = the contents
  - **Anonymity** = the identities

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

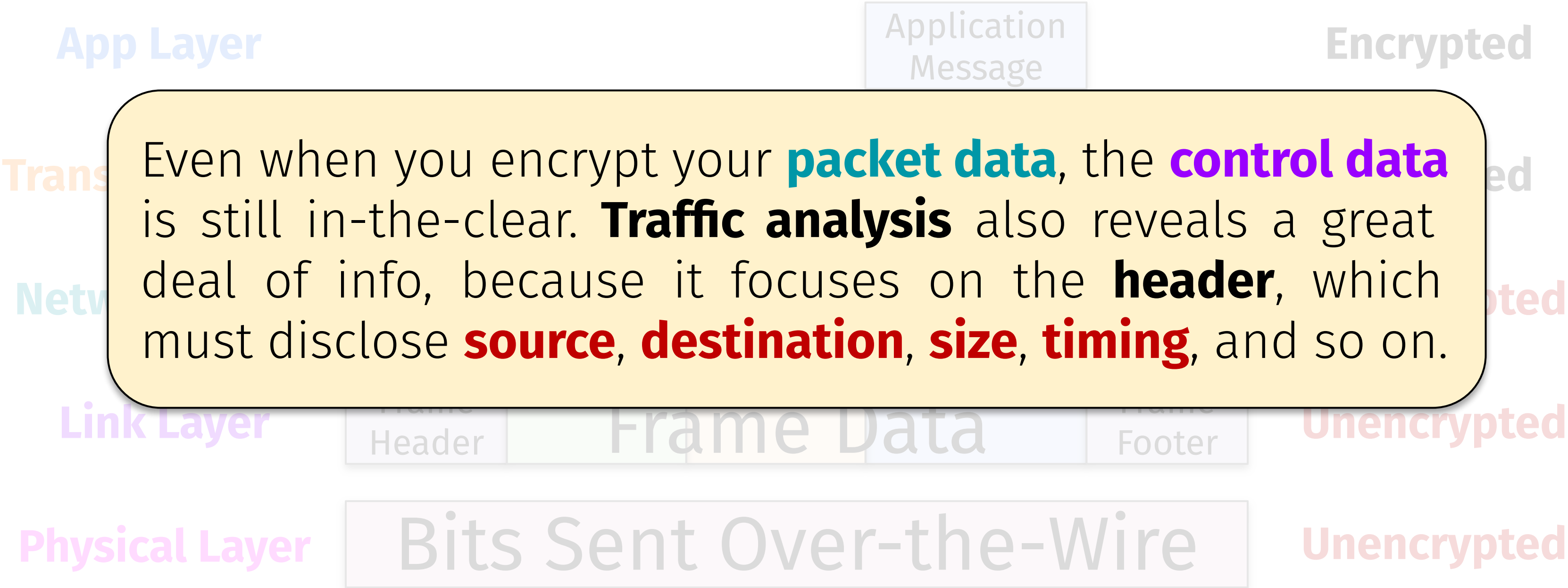
Why does internet anonymity matter?

How do the internet/web provide anonymity?



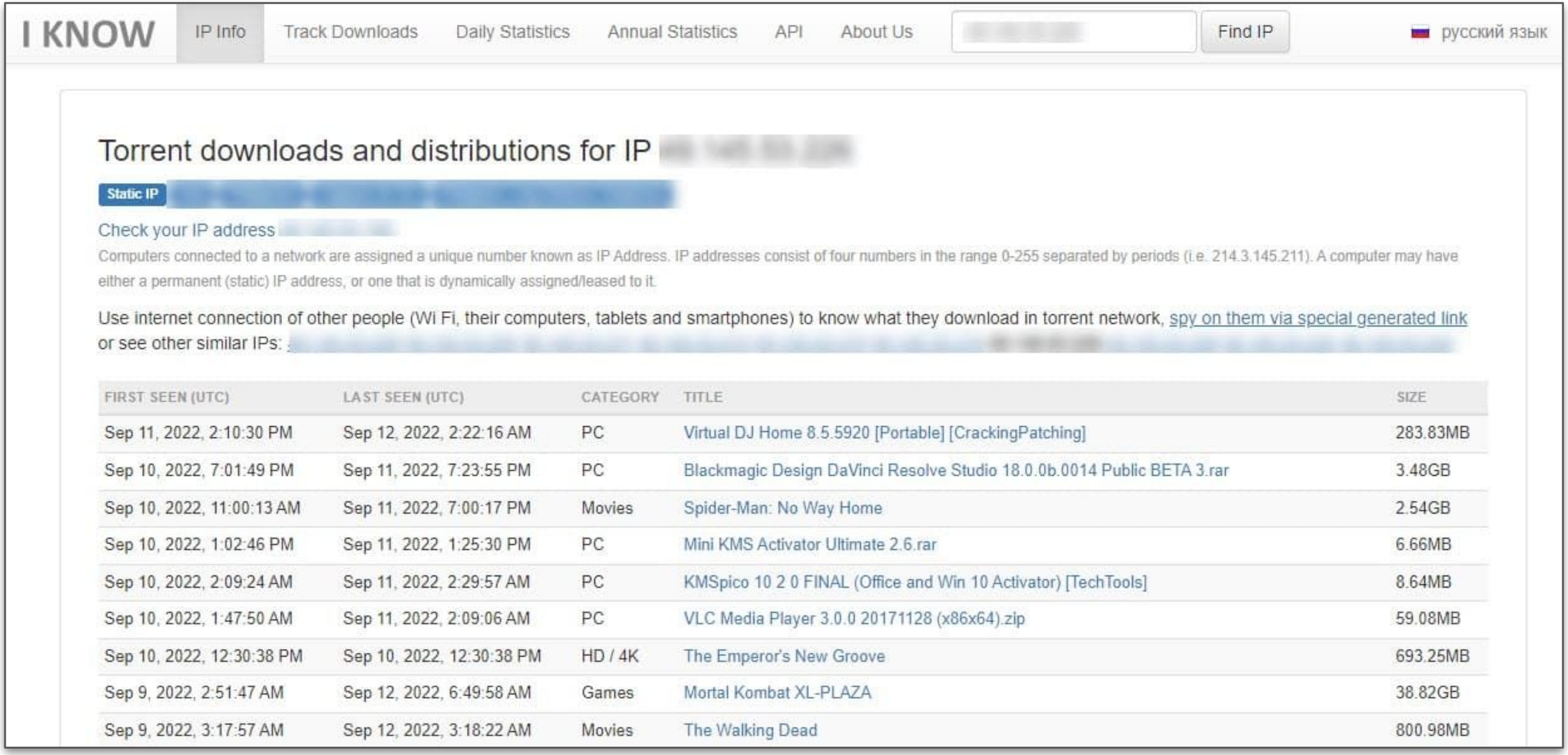
App Layer	Application Message	Encrypted
Transport Layer	Segment Header · Segment Data	Encrypted
Network Layer	Packet Header · Packet Data	Unencrypted
Link Layer	Frame Header · Frame Data · Frame Footer	Unencrypted
Physical Layer	Bits Sent Over-the-Wire	Unencrypted

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?



Torrent downloads and distributions for IP [redacted]

Static IP [redacted]

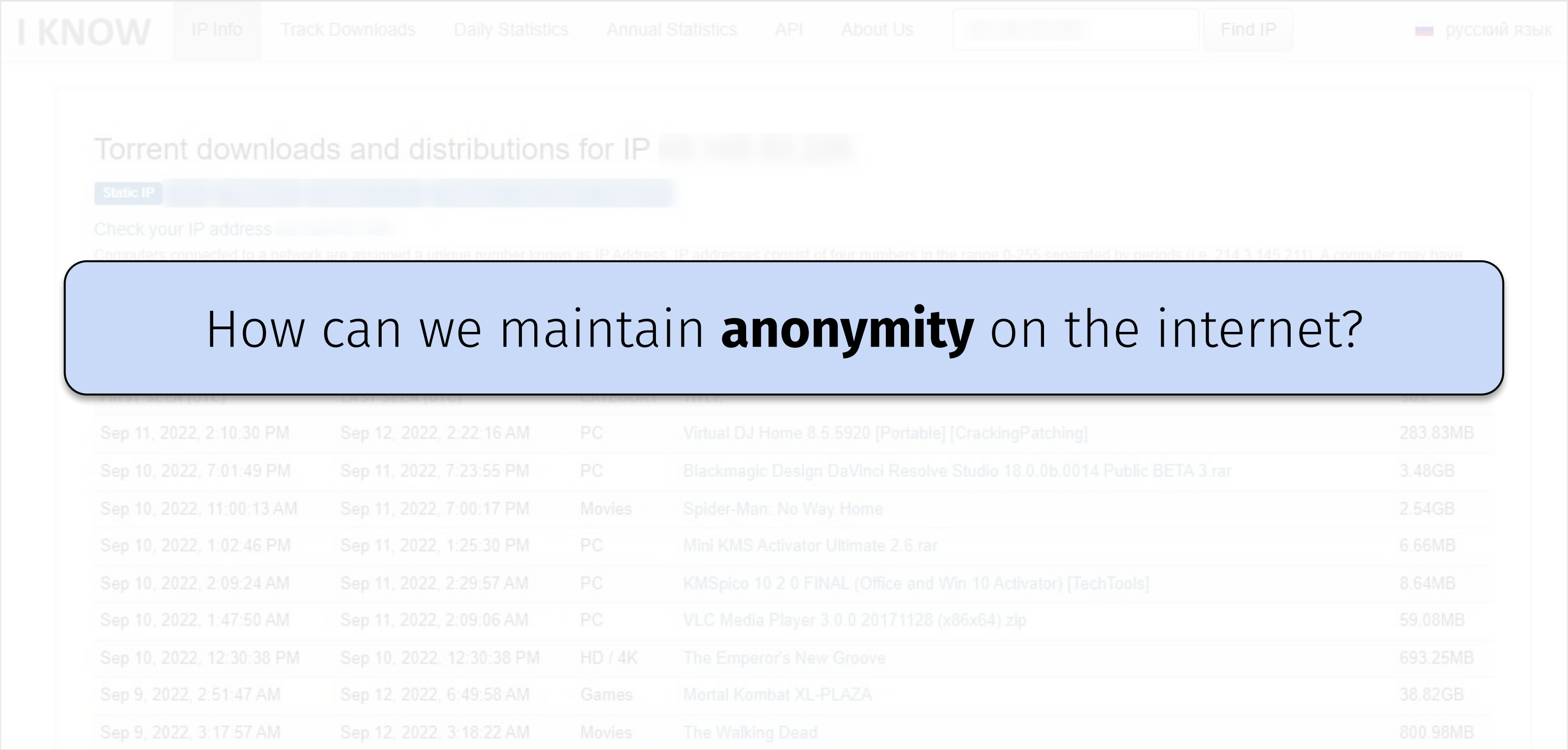
Check your IP address [redacted]

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?



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

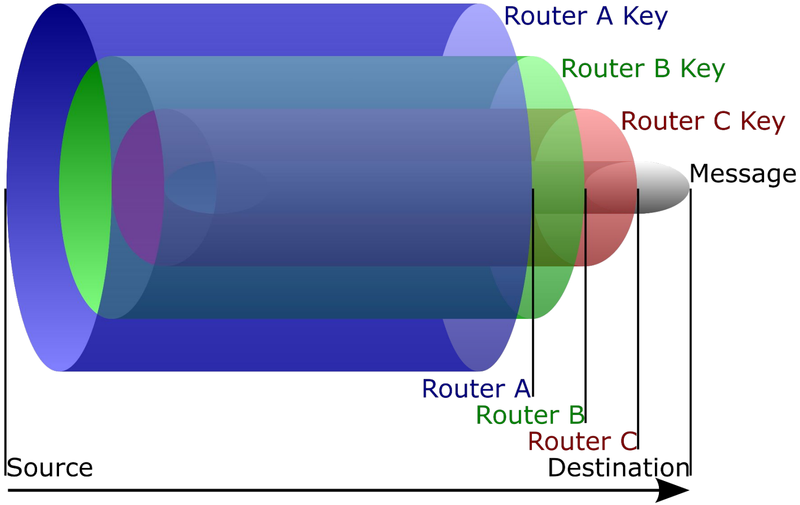
FIRST SEEN (UTC)	LAST SEEN (UTC)	CATEGORY	TITLE	SIZE
Sep 11, 2022, 2:10:30 PM	Sep 12, 2022, 2:22:16 AM	PC	Virtual DJ Home 8.5.5920 [Portable] [CrackingPatching]	283.83MB
Sep 10, 2022, 7:01:49 PM	Sep 11, 2022, 7:23:55 PM	PC	Blackmagic Design DaVinci Resolve Studio 18.0.0b.0014 Public BETA 3.rar	3.48GB
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 can we maintain **anonymity** on the internet?

Tor: The Onion Router

Anonymity Primitive: Onion Routing

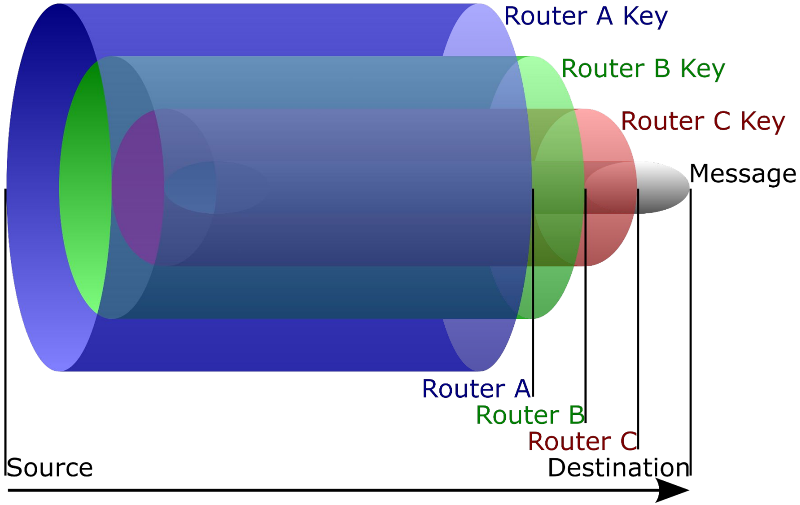
- Each message is **repeatedly encrypted**
    - **Analogy:** multiple layers of an onion



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

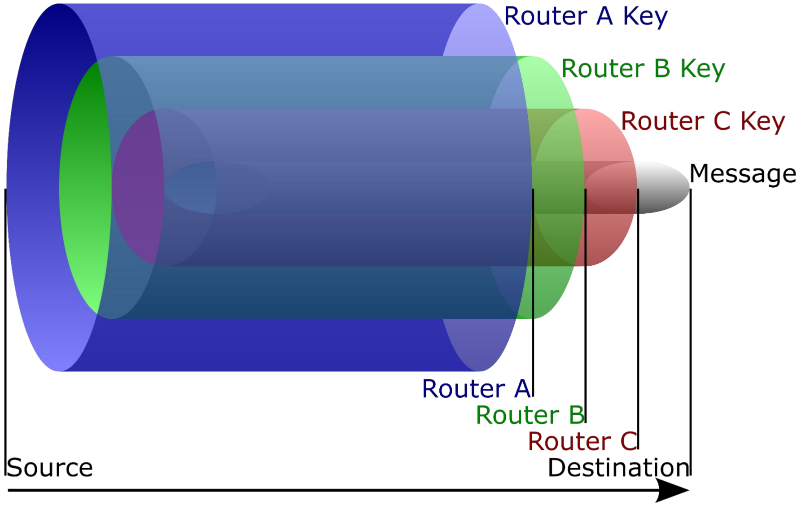


Anonymity Primitive: Onion Routing

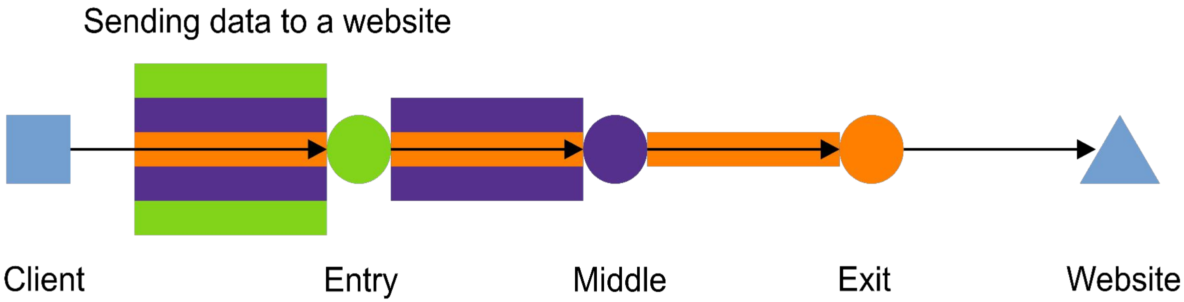
- Each message is **repeatedly encrypted**
  - **Analogy:** multiple layers of an onion

- 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**

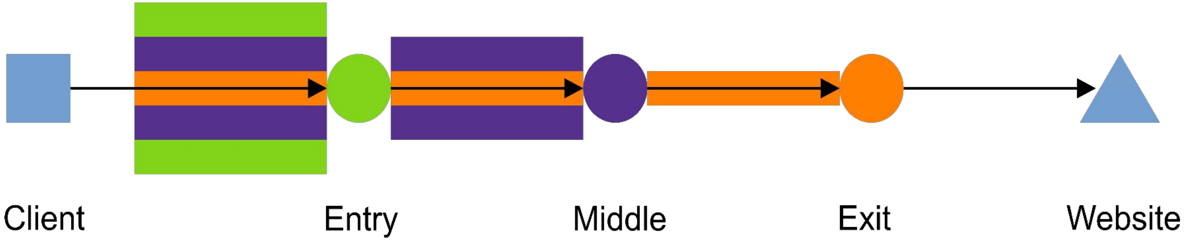


Onion Routing Visualized

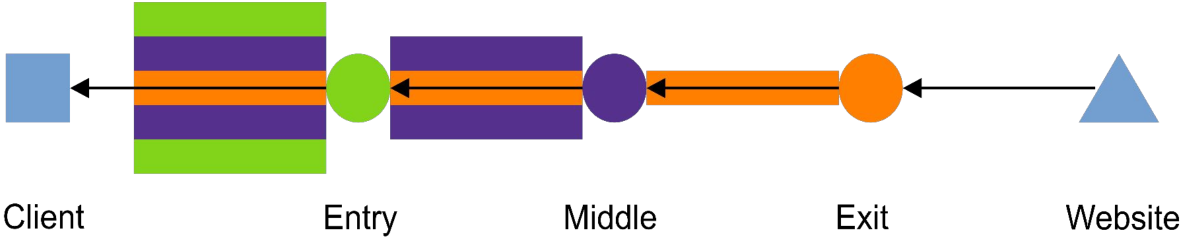


Onion Routing Visualized

Sending data to a website

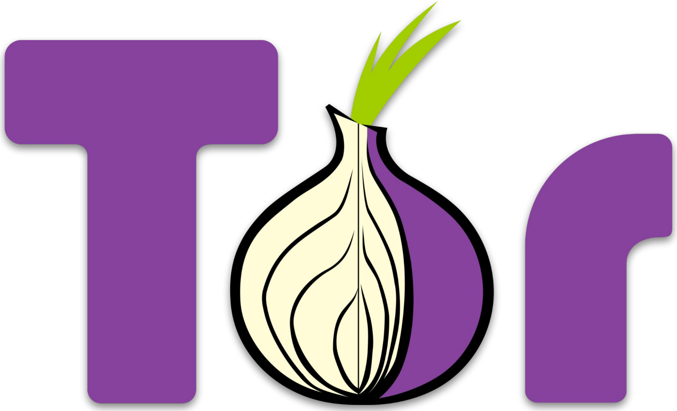


Receiving data from a website



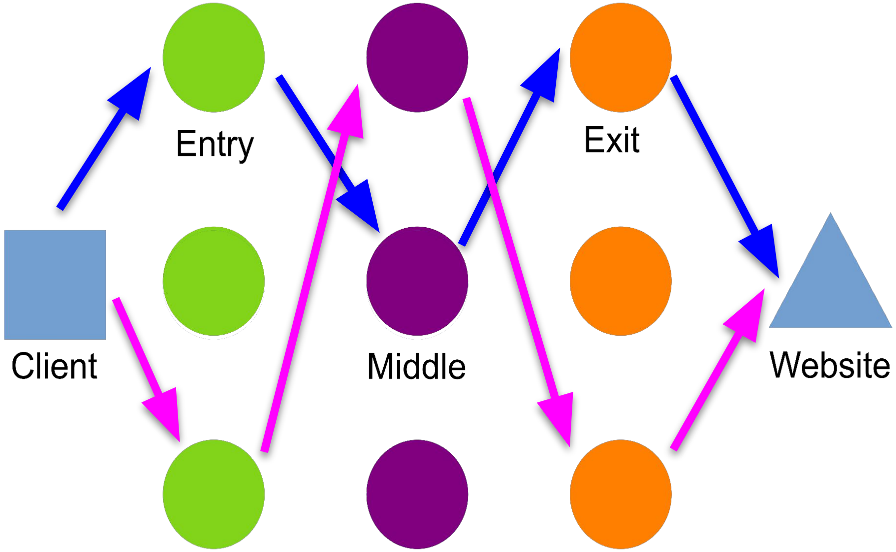
Tor: The Onion Router

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



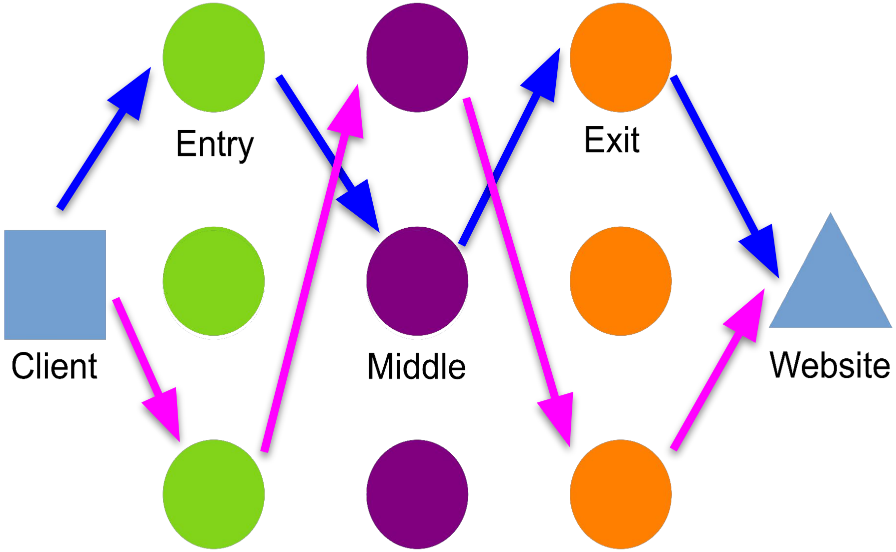
Tor: The Onion Router

- **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

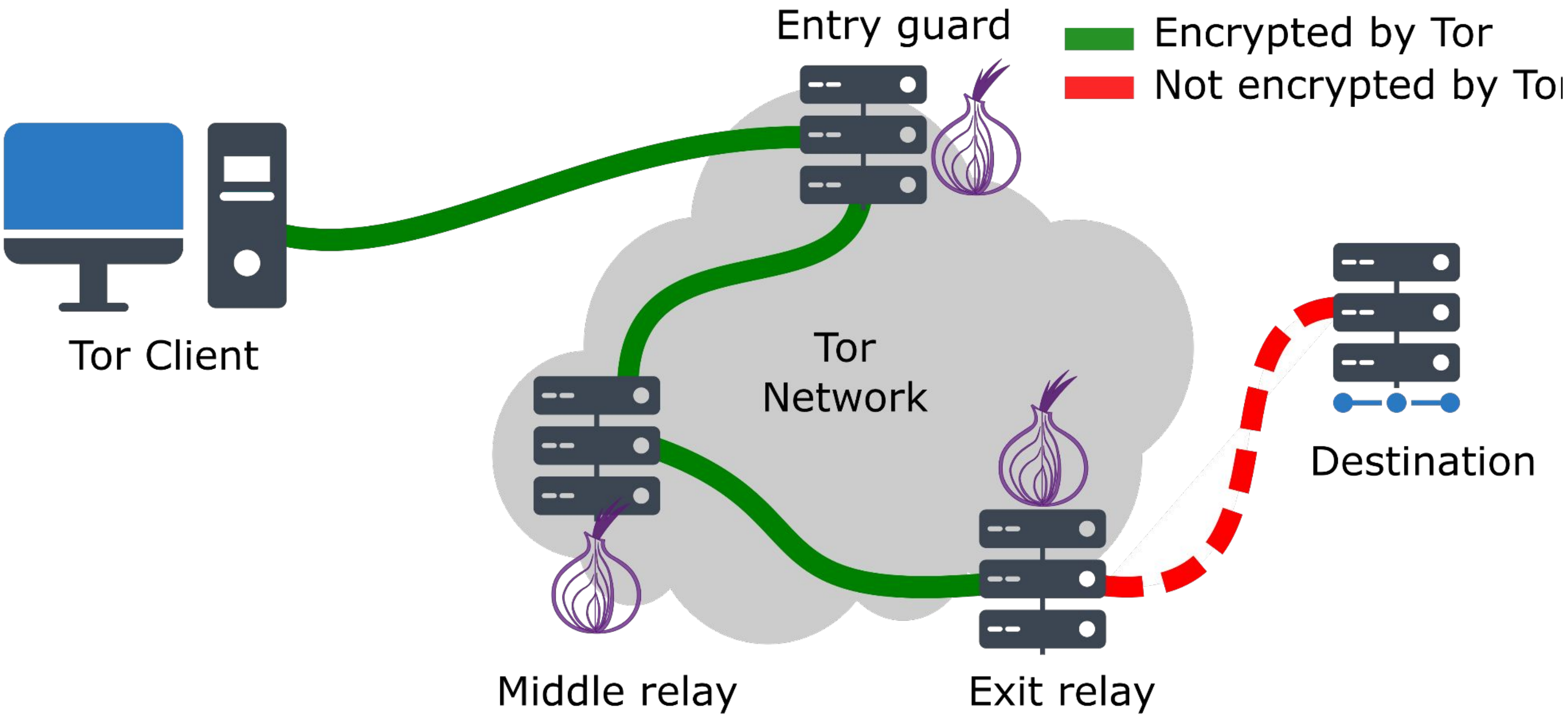


Tor: The Onion Router

- **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

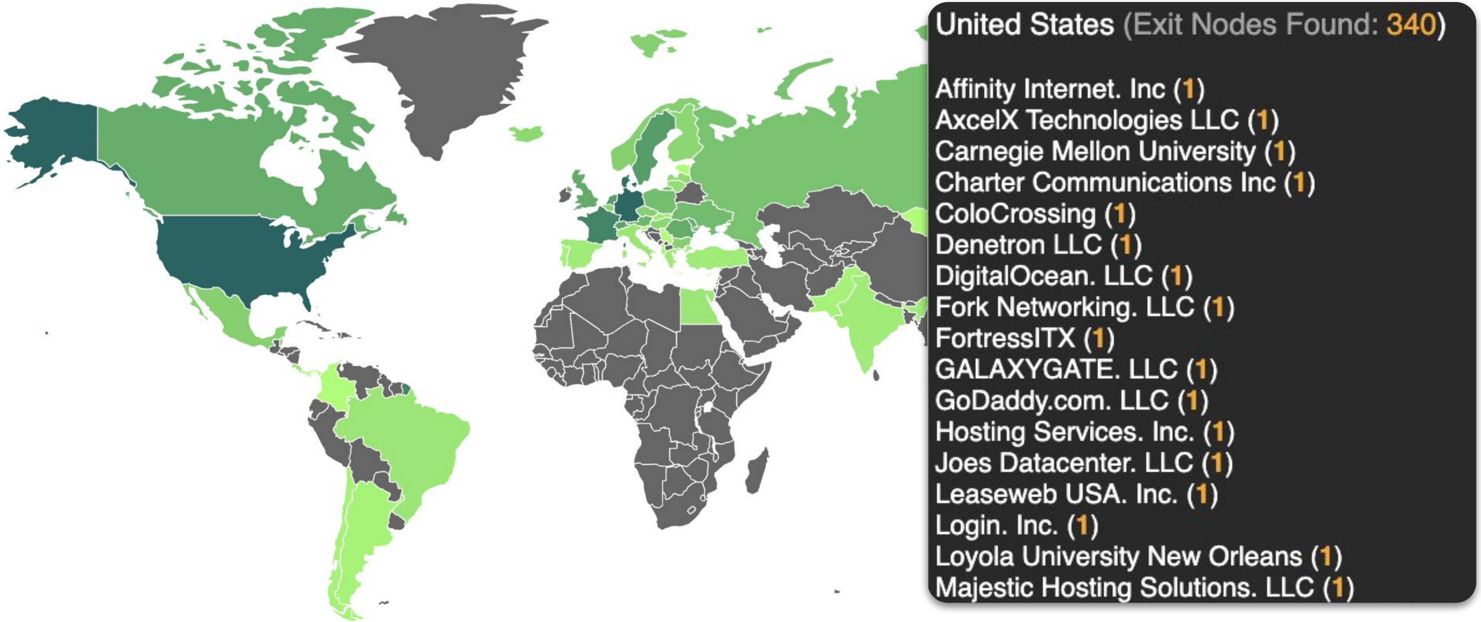


Trust in Tor

- **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

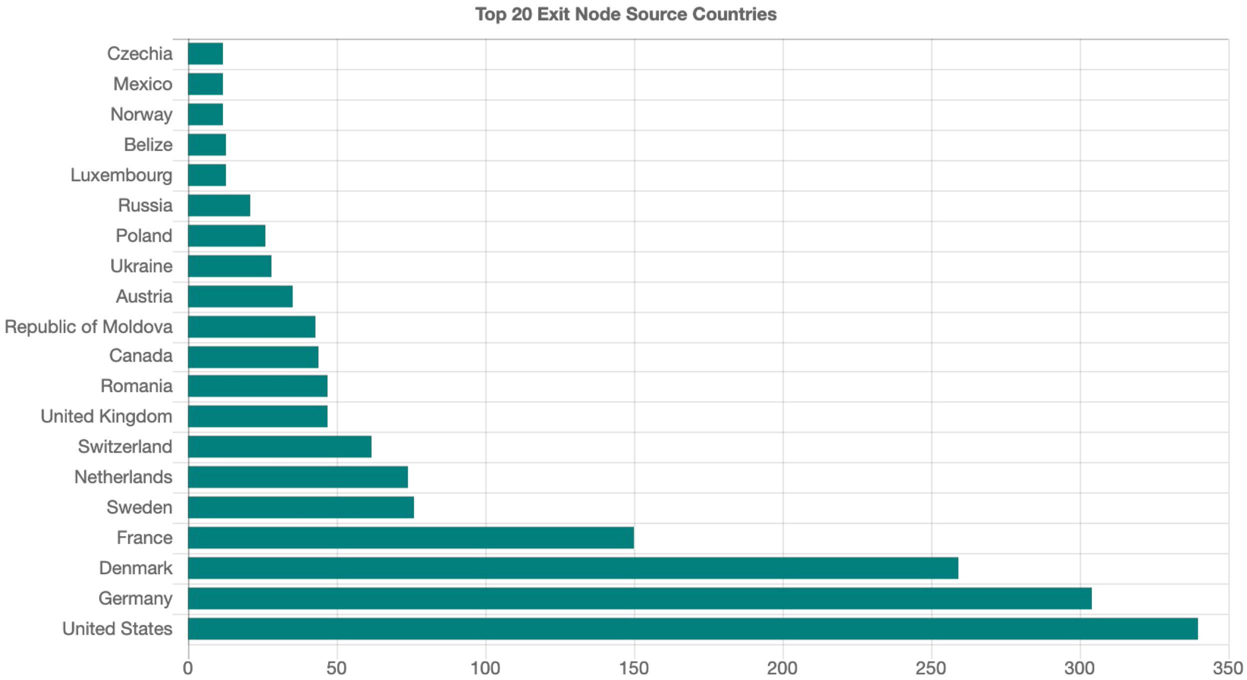


United States (Exit Nodes Found: 340)

Affinity Internet. Inc (1)
AxcelX Technologies LLC (1)
Carnegie Mellon University (1)
Charter Communications Inc (1)
ColoCrossing (1)
Denetron LLC (1)
DigitalOcean. LLC (1)
Fork Networking. LLC (1)
FortressITX (1)
GALAXYGATE. LLC (1)
GoDaddy.com. LLC (1)
Hosting Services. Inc. (1)
Joes Datacenter. LLC (1)
Leaseweb USA. Inc. (1)
Login. Inc. (1)
Loyola University New Orleans (1)
Majestic Hosting Solutions. LLC (1)

The Tor Network

- Lots of nodes spread out around the world

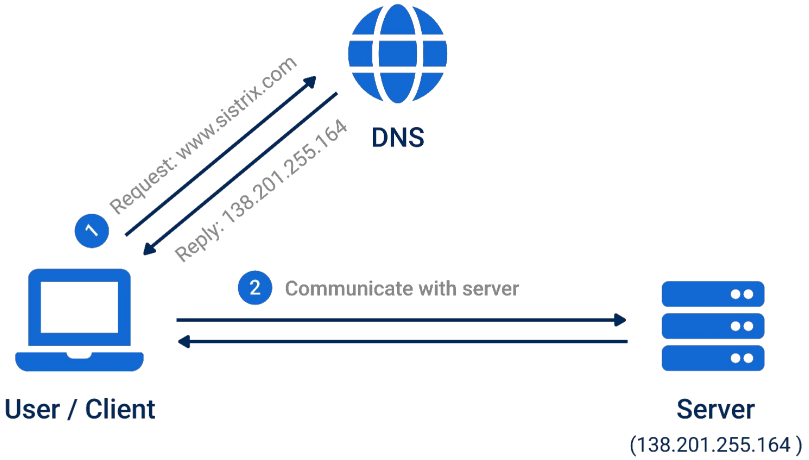


Questions?

Attacking Tor

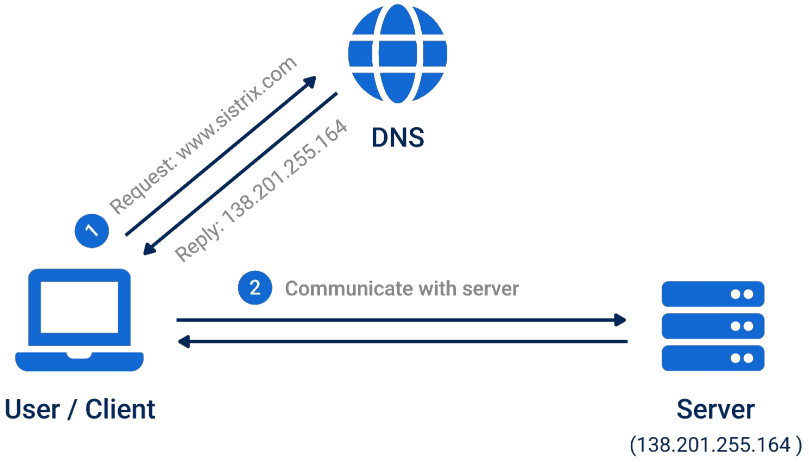
Recap: The Domain Name System

- ???



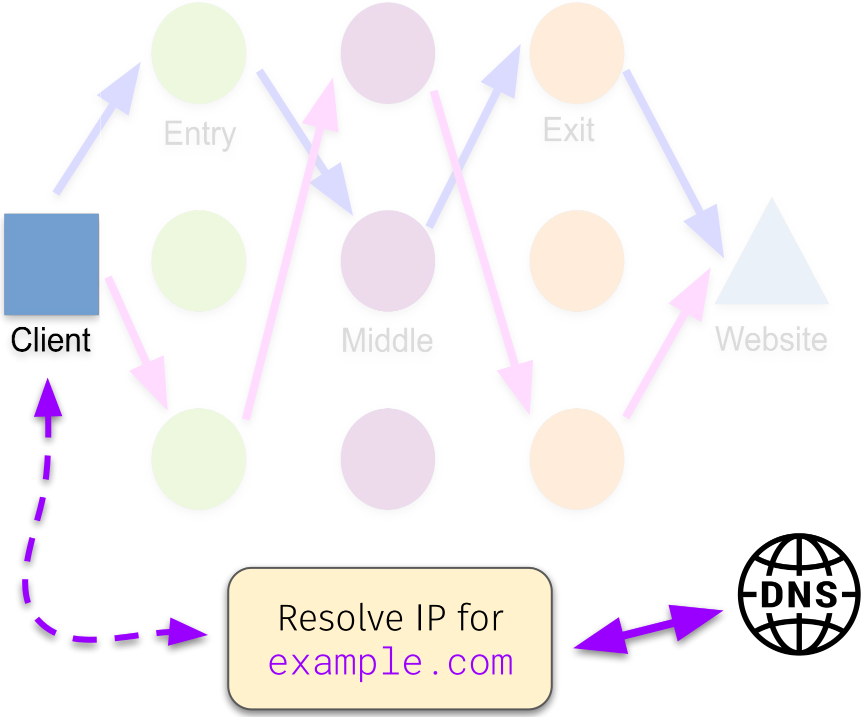
Recap: The Domain Name System

- **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

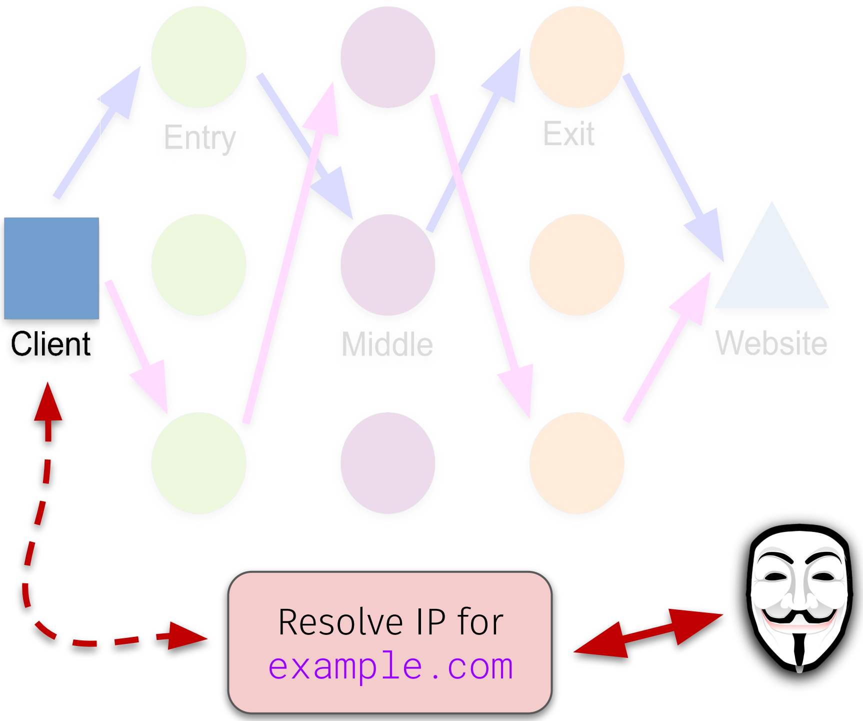
- **DNS requests** are **not** sent through Tor by default



Attack 1: DNS Leaks

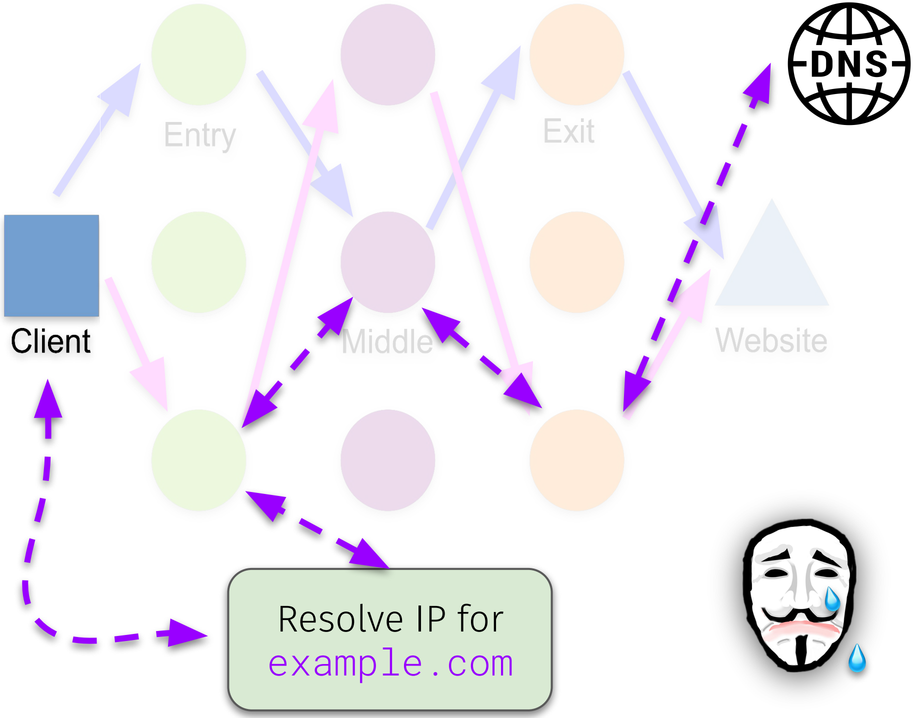
- **DNS requests** are **not** sent through Tor by default

- Attackers could see what **websites** are being visited

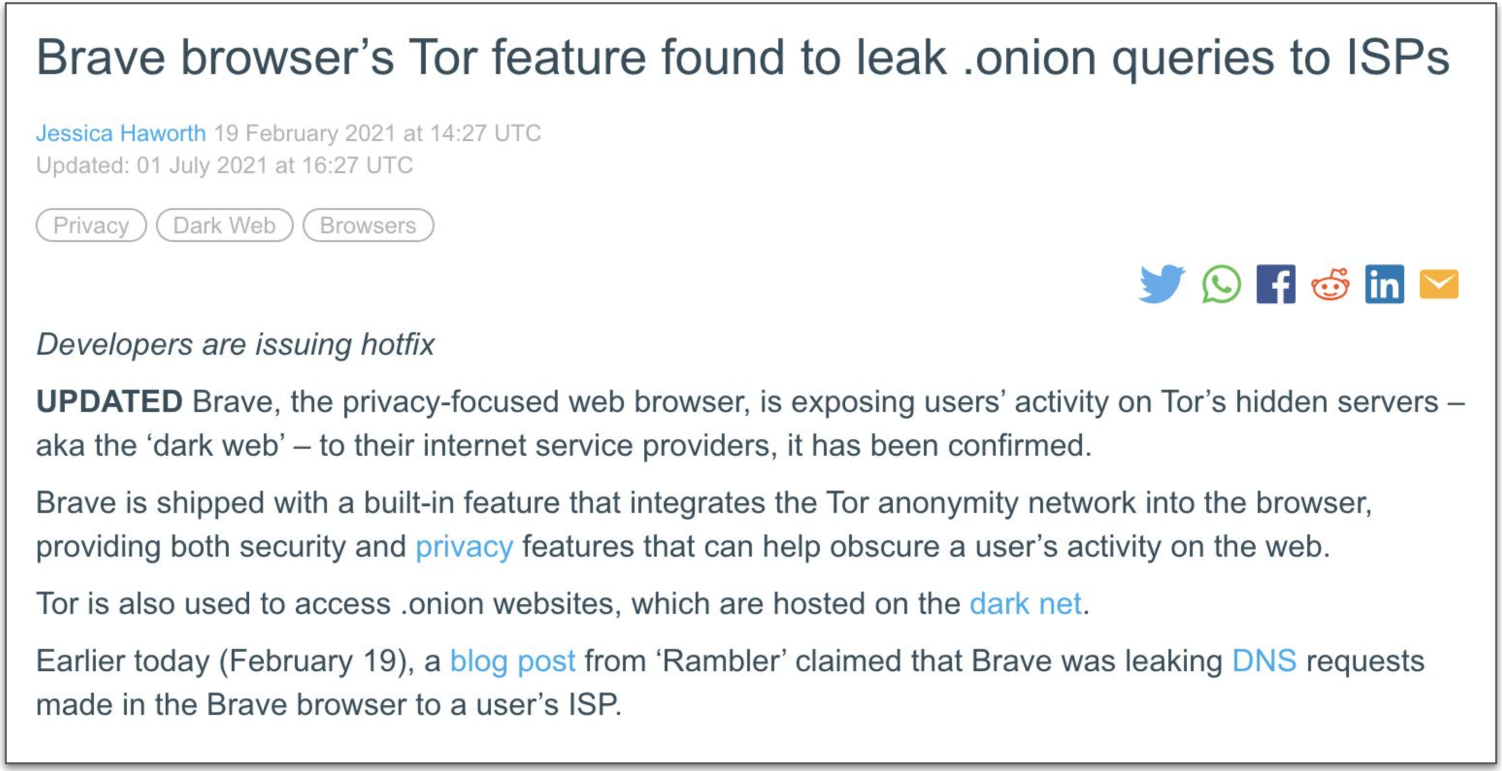


Attack 1: DNS Leaks

- **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

```
11:30:11 Server sent 5kb
```

```
11:30:12 Your node received 6kb
```

```
11:33:17 Server sent 14kb
```

```
11:33:18 Your node received 15kb
```

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

```
11:30:11 Server sent 5kb
```

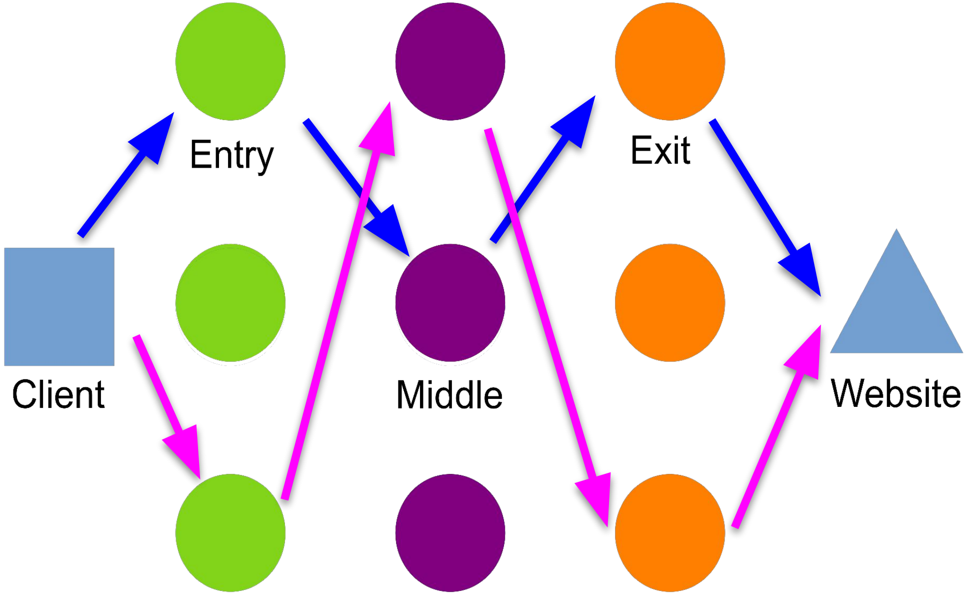
```
11:30:12 Your node received 6kb
```

```
11:33:17 Server sent 14kb
```

```
11:33:18 Your node received 15kb
```

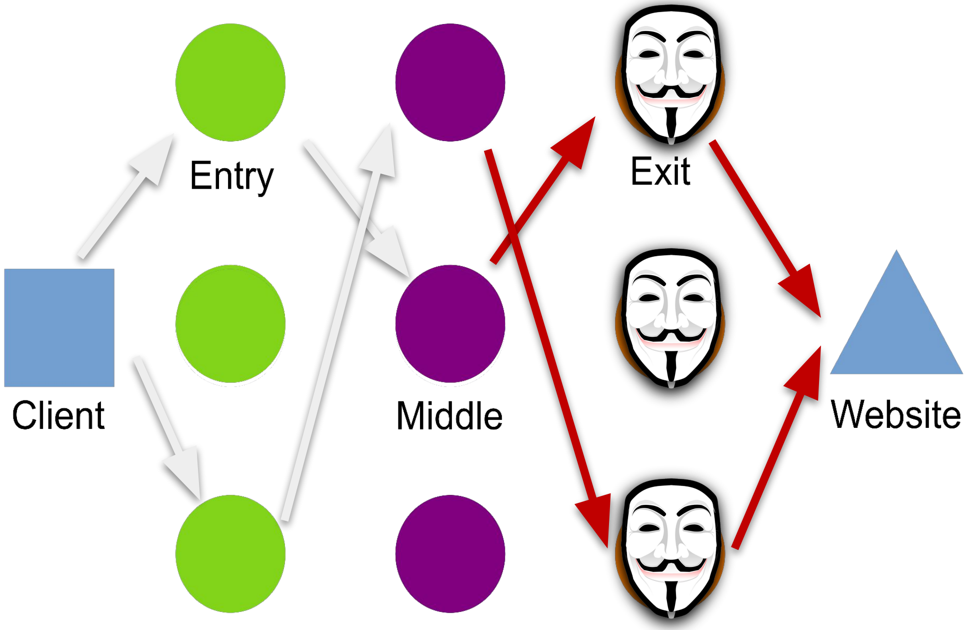
Attack 3: Malicious Nodes

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



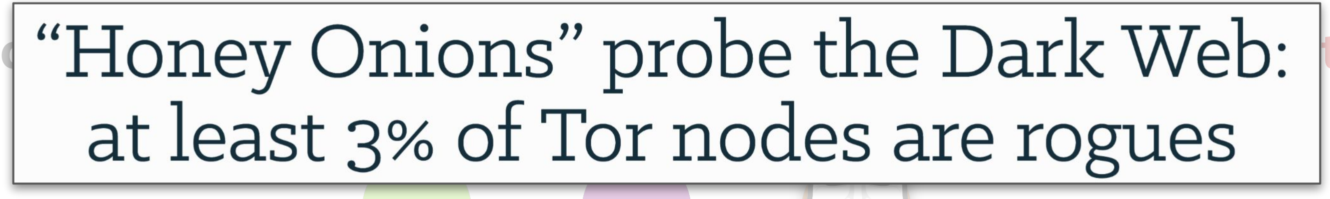
Attack 3: Malicious Nodes

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



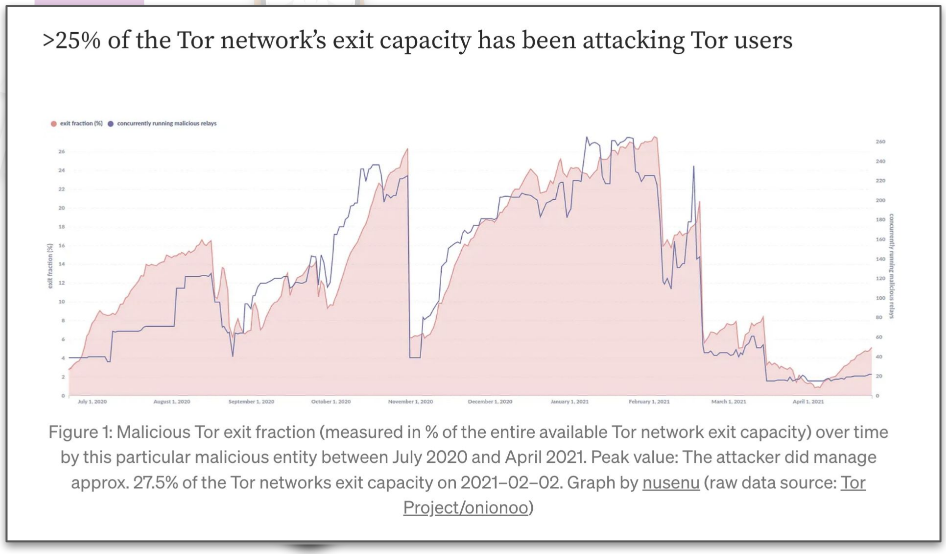
Attack 3: Malicious Nodes

- Traffi 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.”



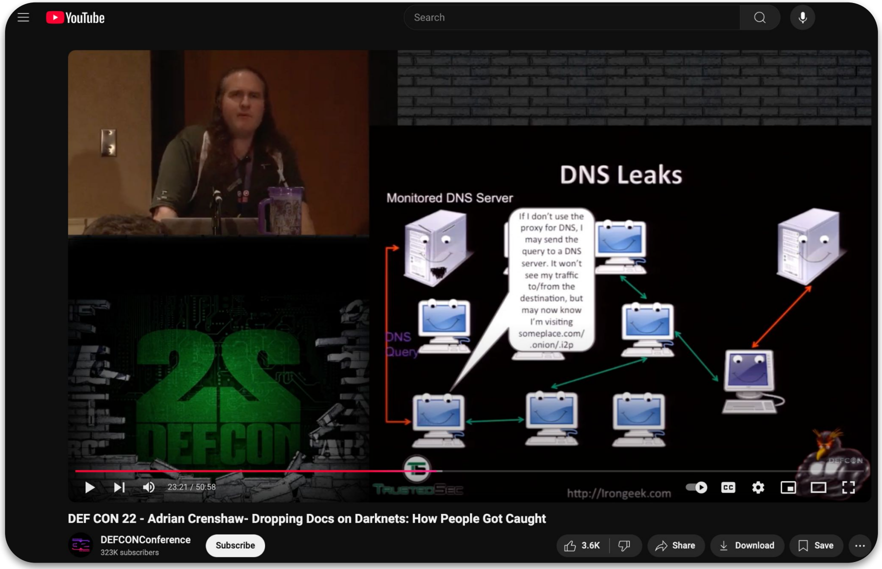
>25% of the Tor network’s exit capacity has been attacking Tor users

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)

Questions?

Supplemental: Dropping Docs on Darknets

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



If I don’t use the proxy for DNS, I may send the query to a DNS server. It won’t see my traffic to/from the destination, but may now know I’m visiting someplace.com/.onion/.i2p

DEF CON 22 - Adrian Crenshaw- Dropping Docs on Darknets: How People Got Caught

DEFCONConference
323K subscribers

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

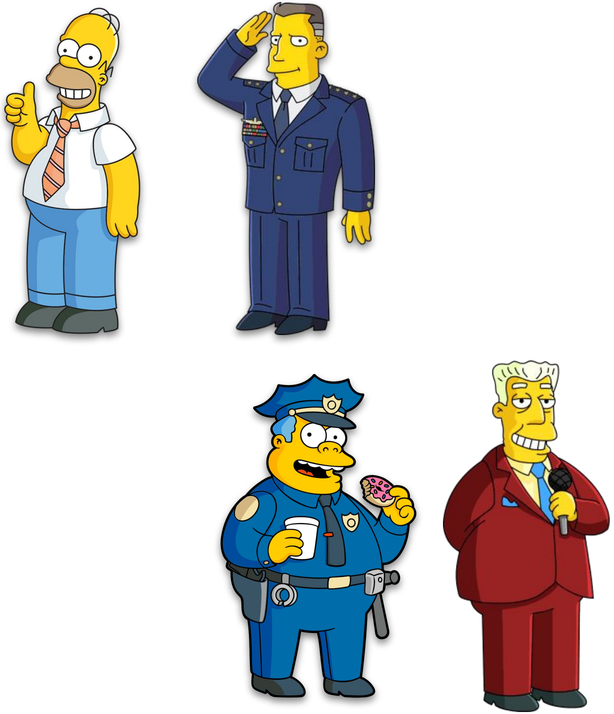
Tor Users and Websites

Who uses Tor?

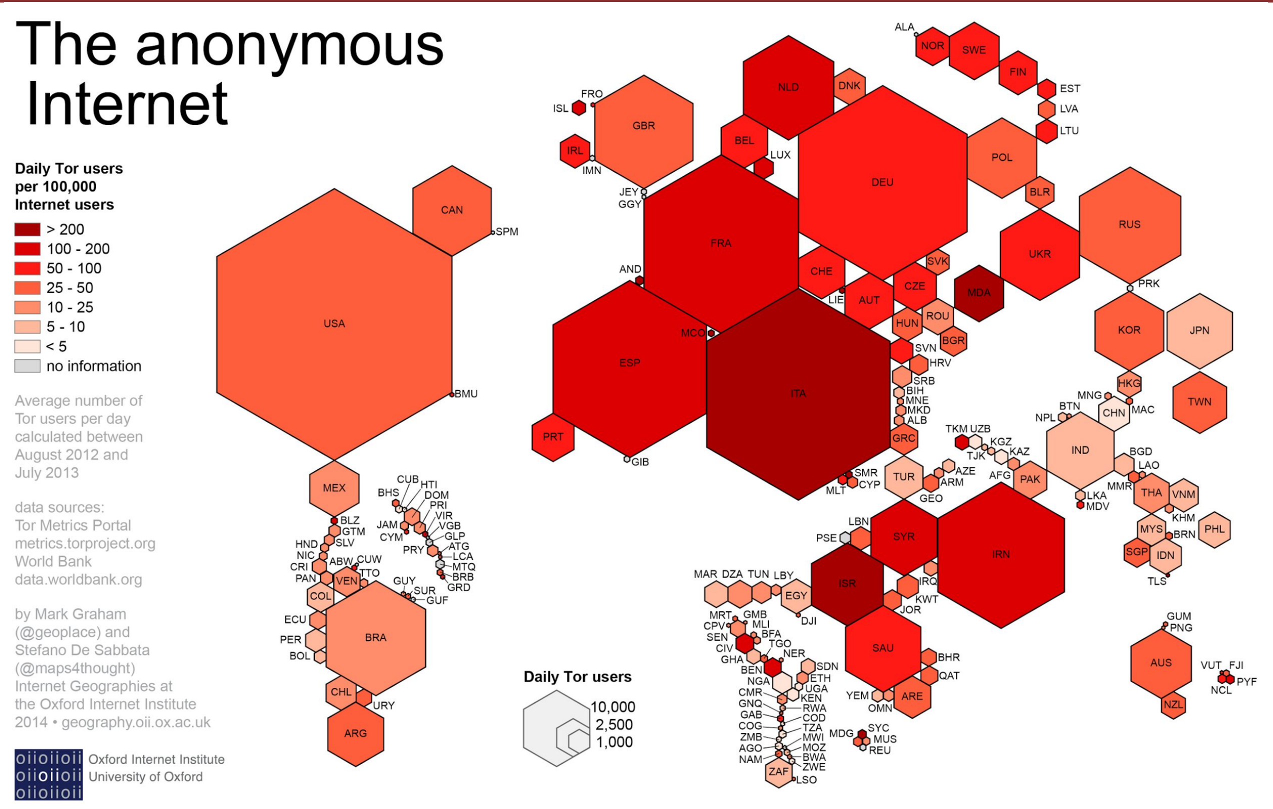
- **???**

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

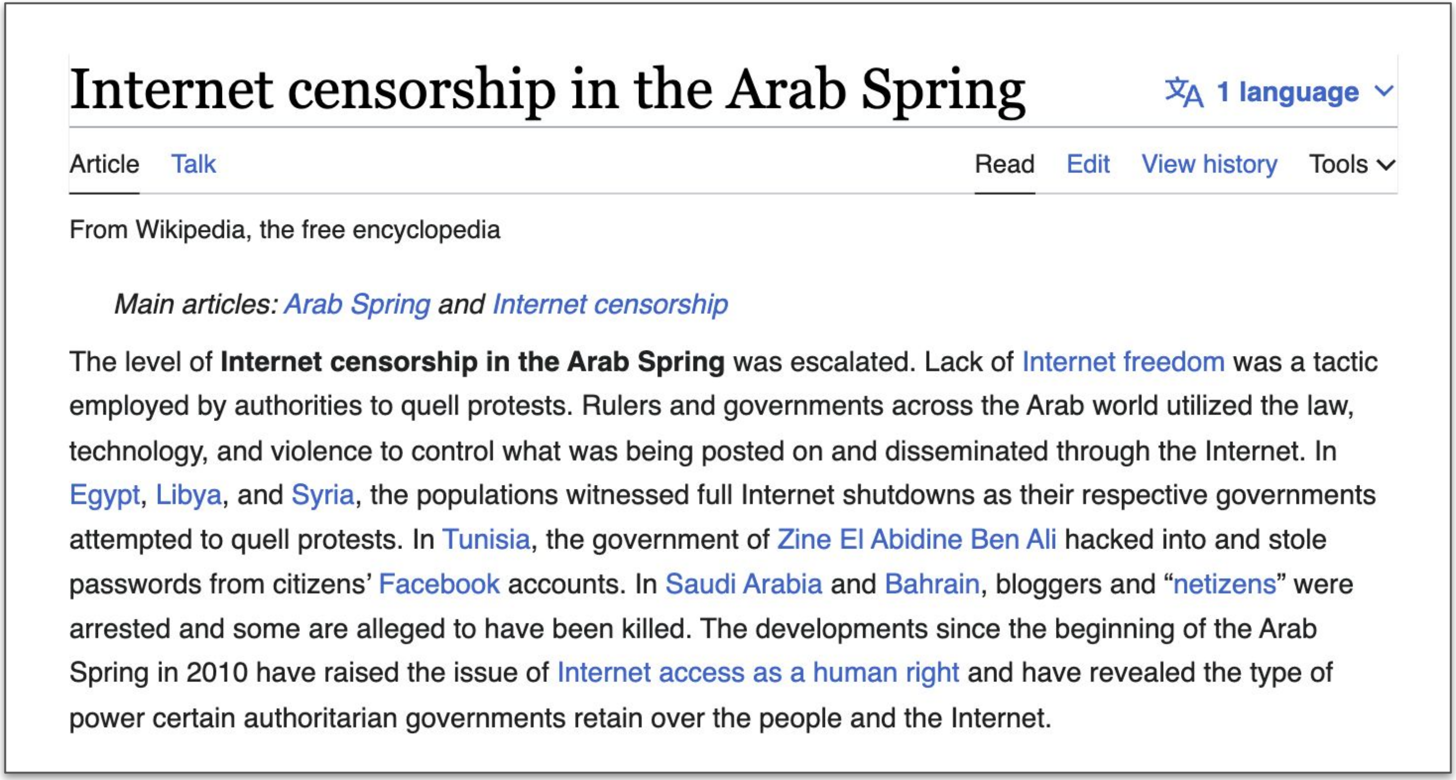
by Mark Graham (@geoplace) and Stefano De Sabbata (@maps4thought)
Internet Geographies at the Oxford Internet Institute
2014 • geography.oii.ox.ac.uk

Oxford Internet Institute
University of Oxford

Daily Tor users

10,000
2,500
1,000

Who uses Tor?



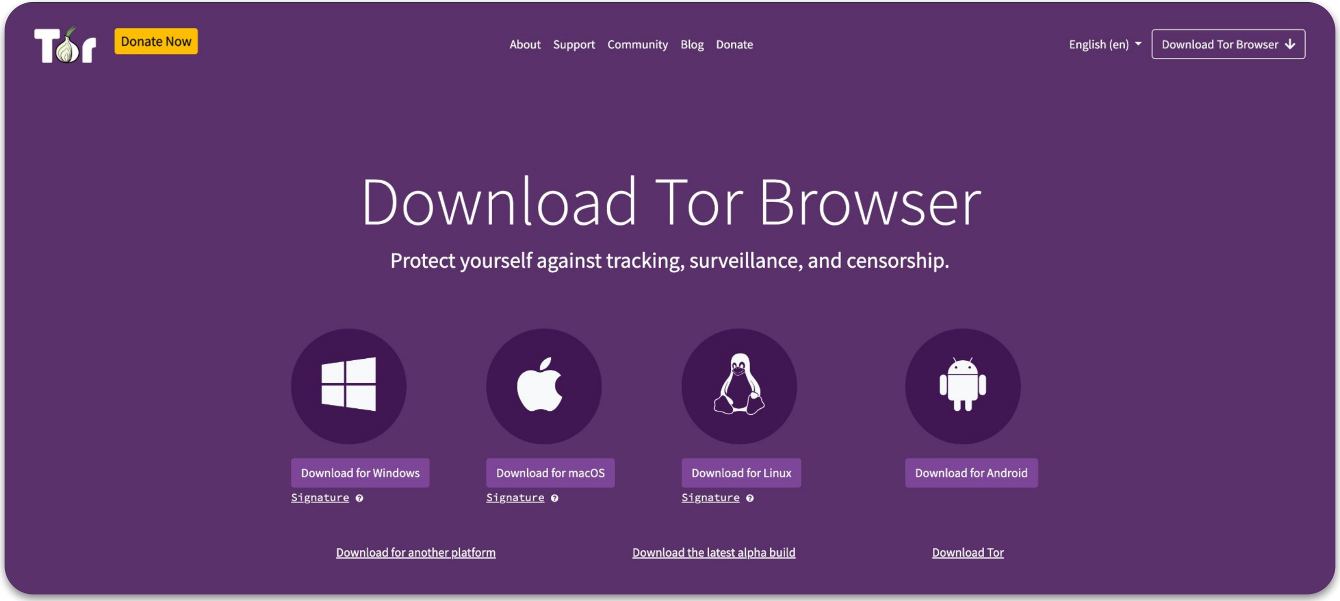
Internet censorship in the Arab Spring

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.

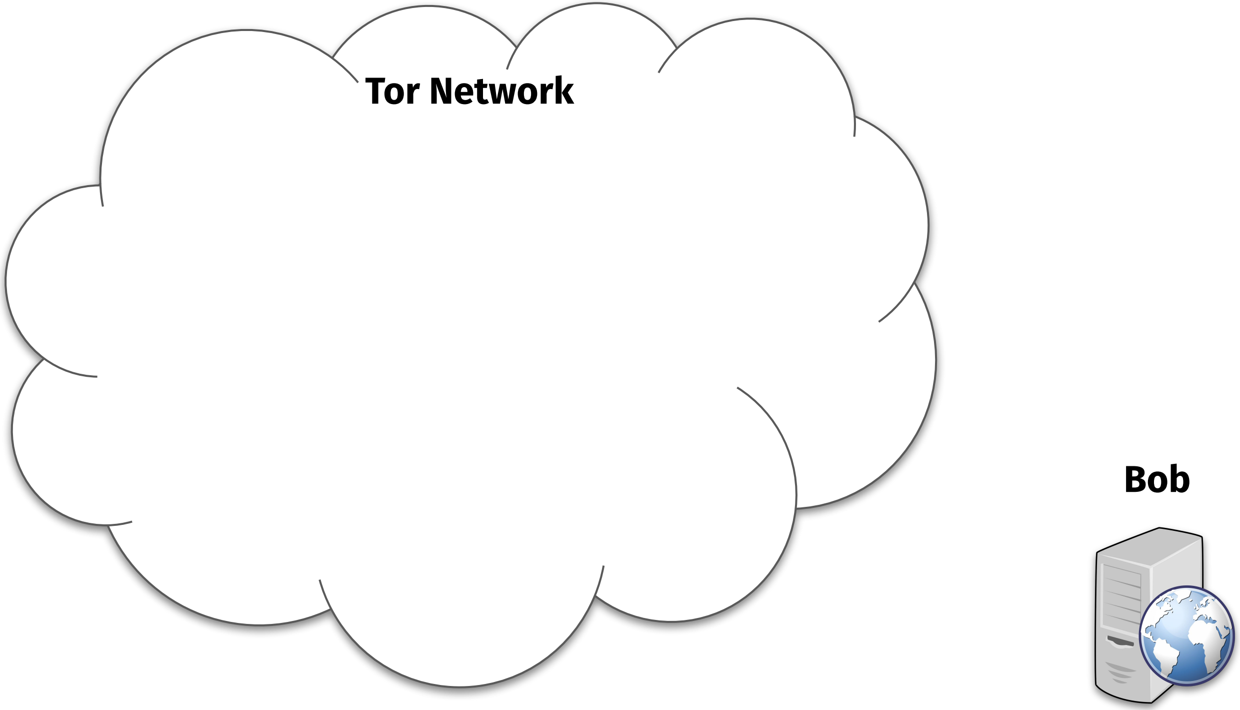
How can you use Tor?



Download Tor Browser

Protect yourself against tracking, surveillance, and censorship.

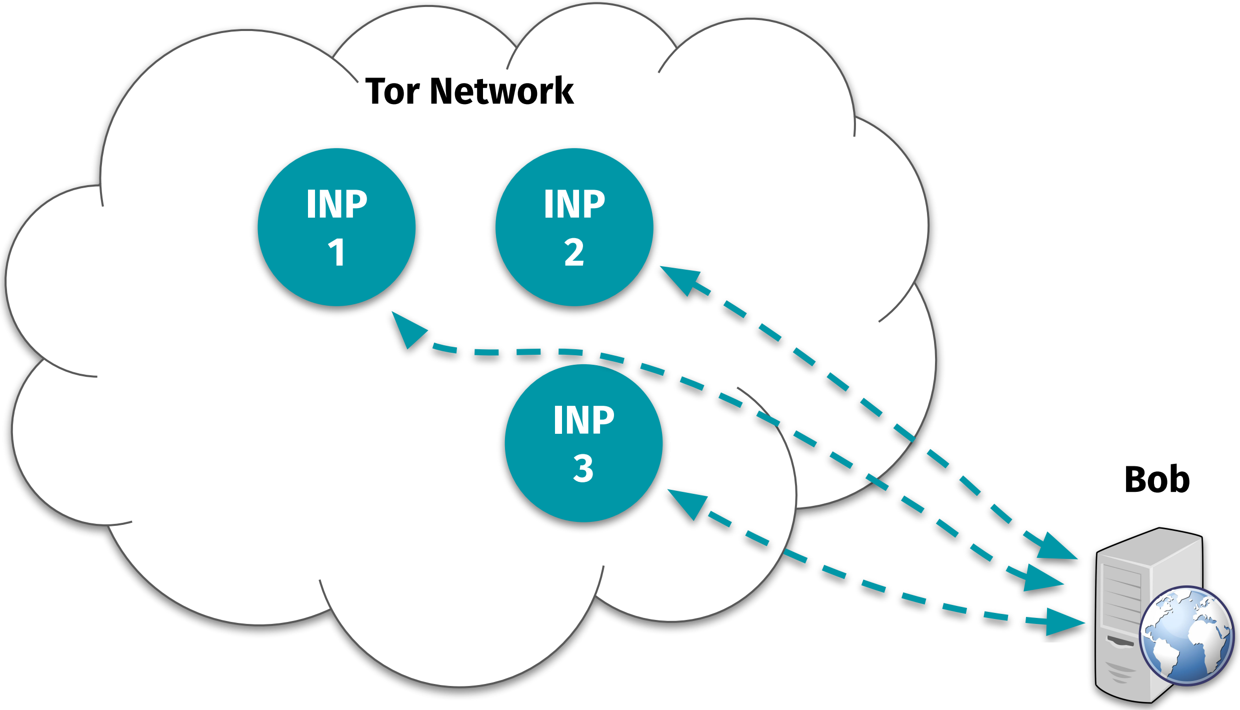
Hidden Services



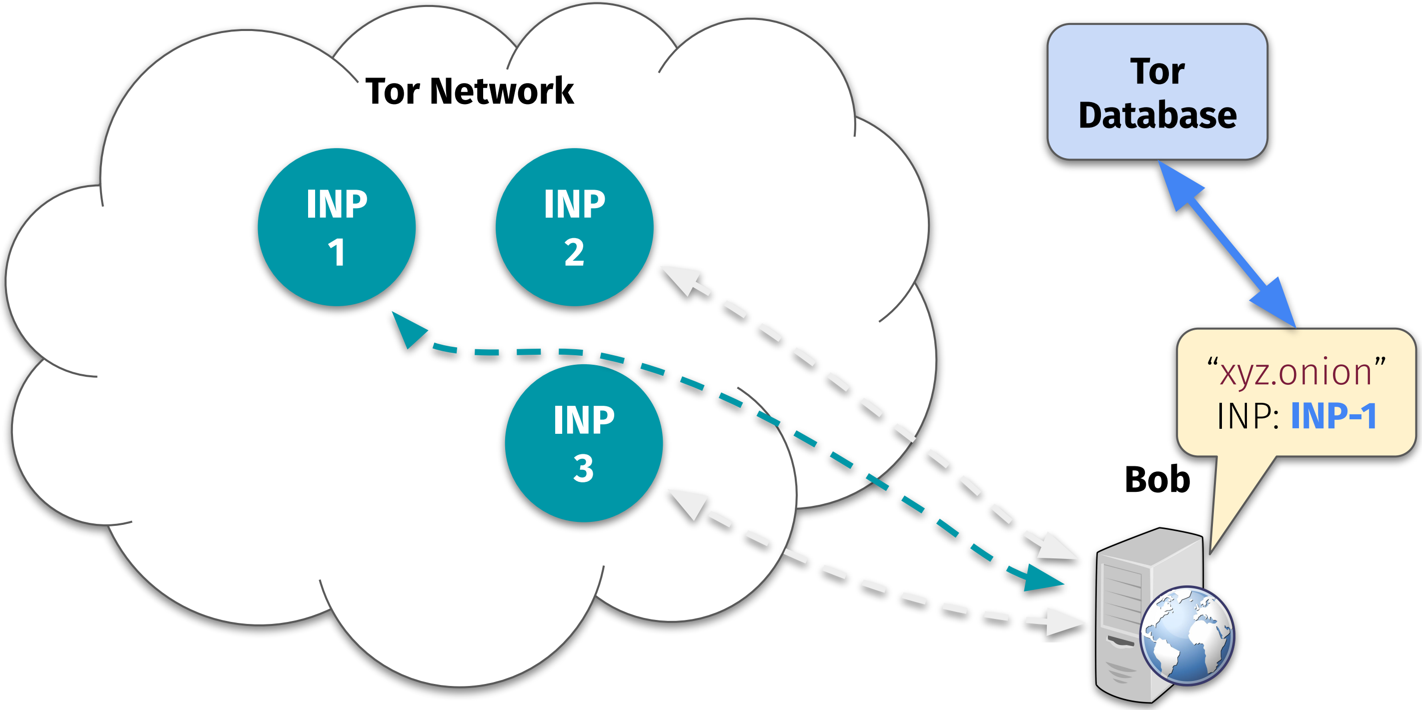
Tor Network

Bob

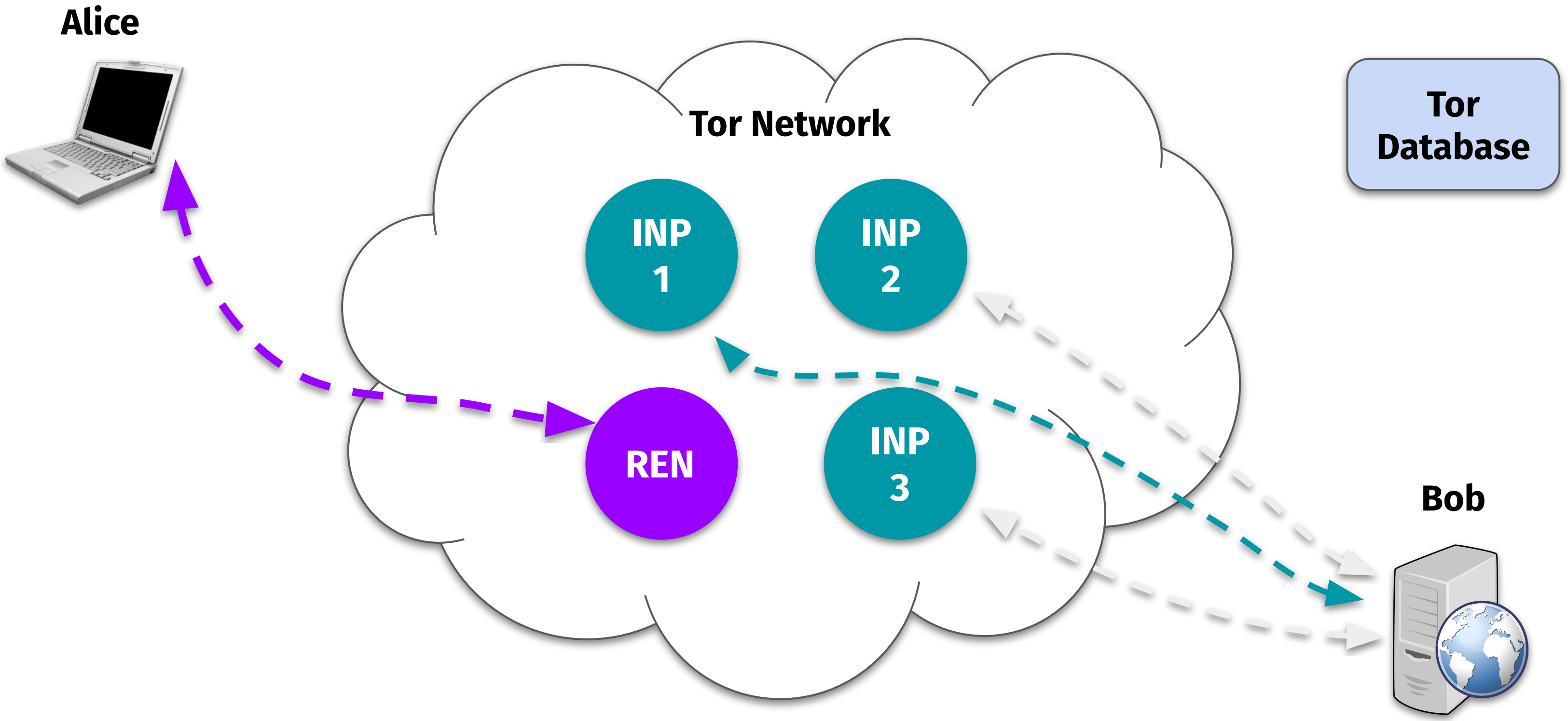
Hidden Services



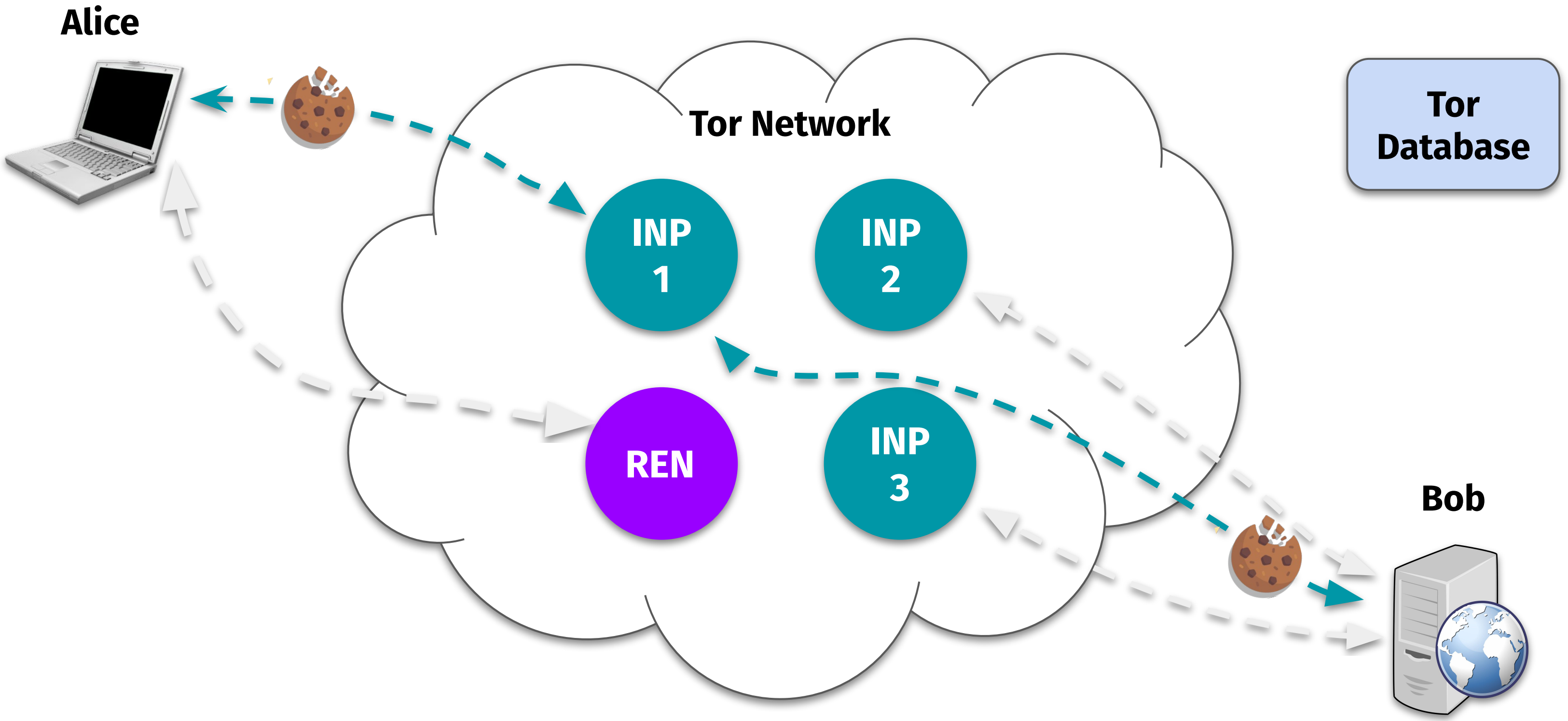
Hidden Services



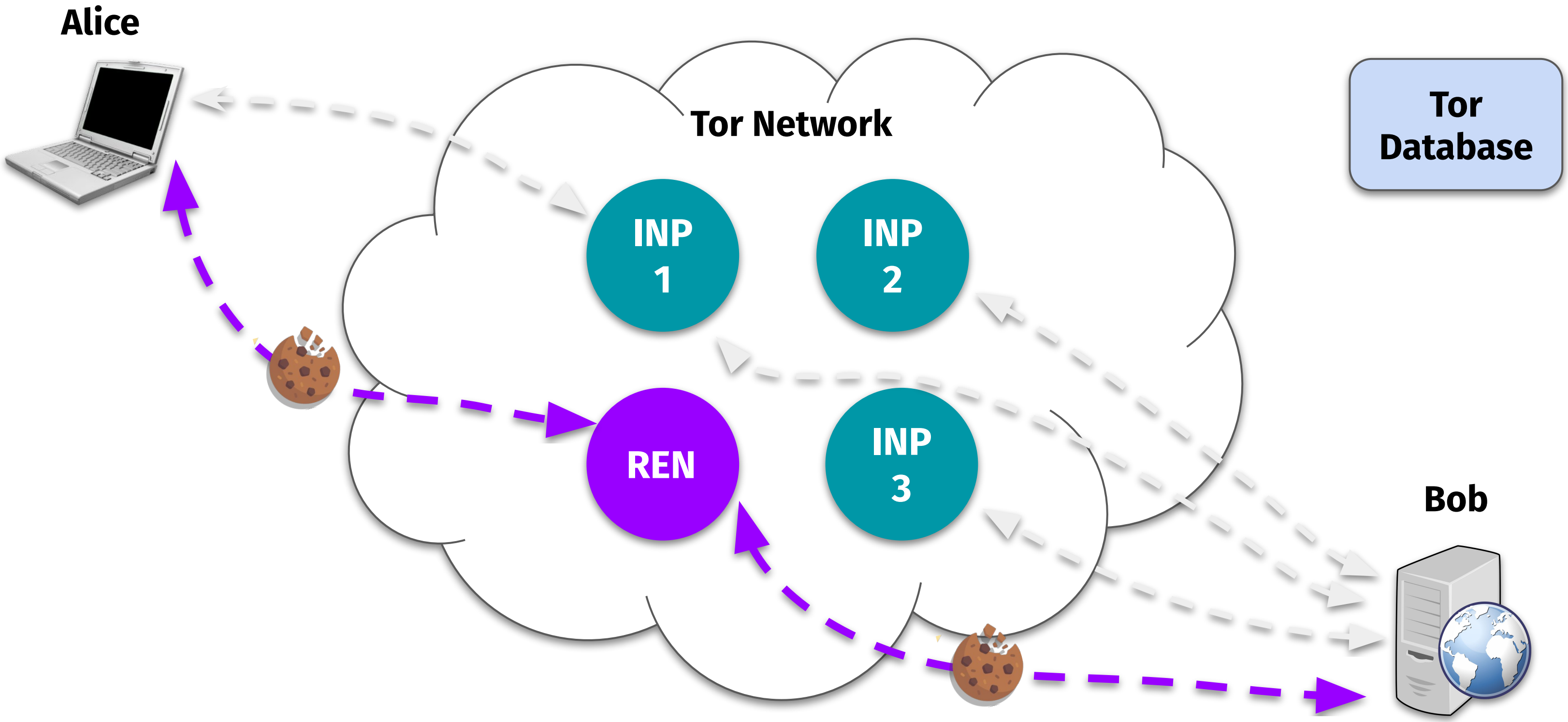
Hidden Services



Hidden Services

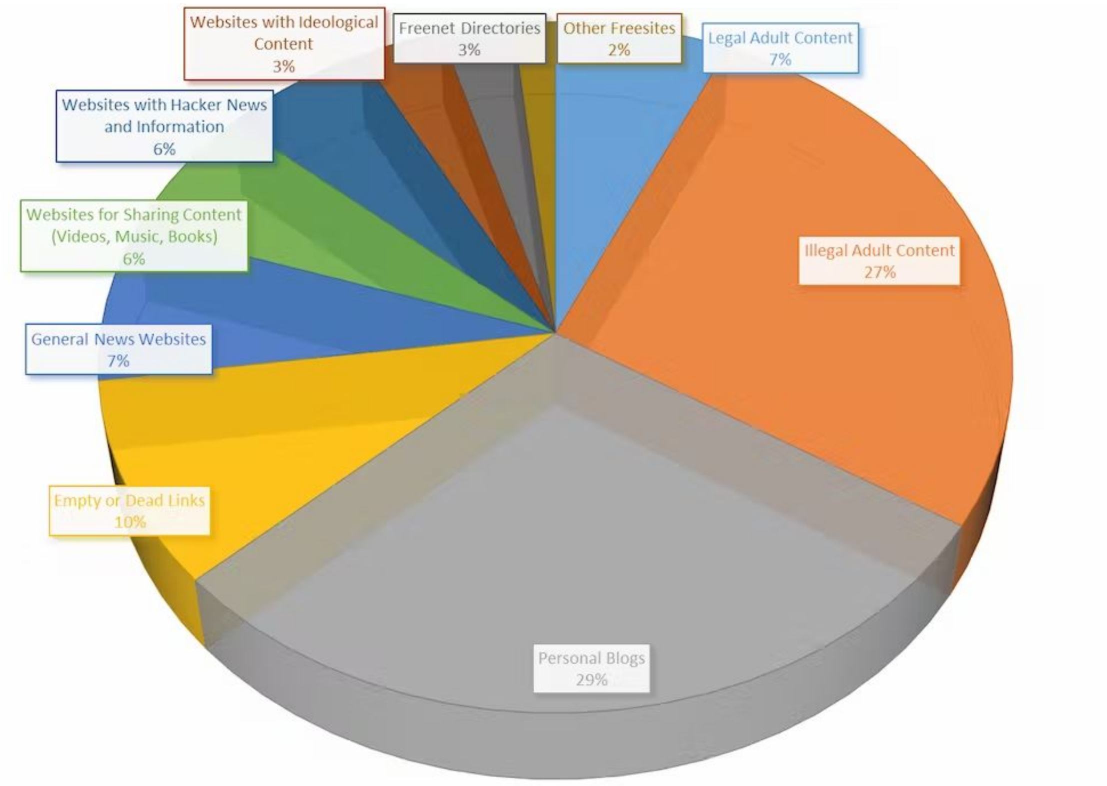


Hidden Services

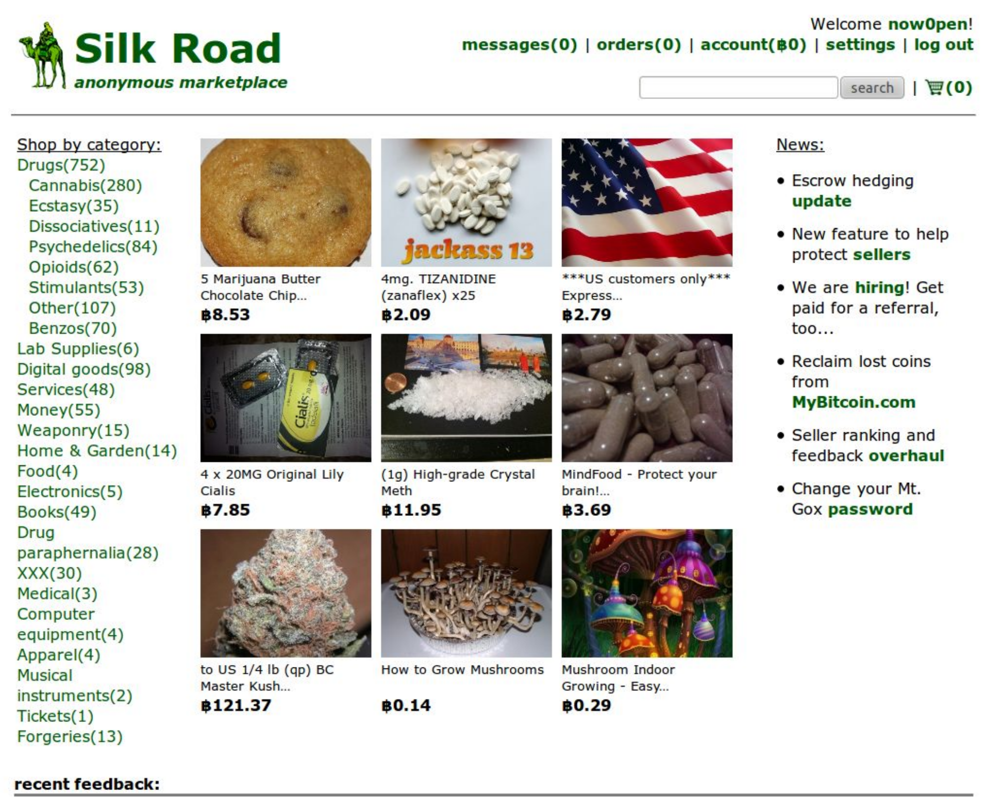


Hidden Services

What services get hidden?



What services get hidden?



Silk Road

anonymous marketplace

Welcome **now0pen**!

messages(0) | **orders(0)** | **account(฿0)** | **settings** | **log out**

search | (0)

Shop by category:

Drugs(752)
- Cannabis(280)
- Ecstasy(35)
- Dissociatives(11)
- Psychedelics(84)
- Opioids(62)
- Stimulants(53)
- Other(107)
- Benzos(70)

Lab Supplies(6)
Digital goods(98)
Services(48)
Money(55)
Weaponry(15)
Home & Garden(14)
Food(4)
Electronics(5)
Books(49)
Drug paraphernalia(28)
XXX(30)
Medical(3)
Computer equipment(4)
Apparel(4)
Musical instruments(2)
Tickets(1)
Forgeries(13)

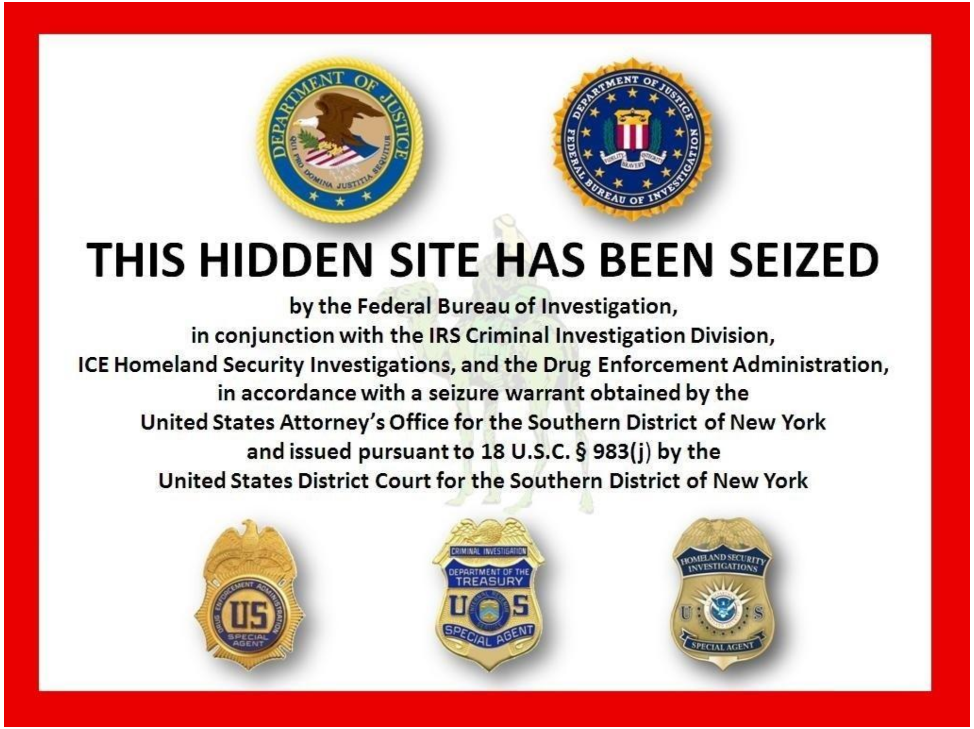
5 Marijuana Butter Chocolate Chip... **฿8.53**	4mg. TIZANIDINE (zanaflex) x25 **฿2.09**	***US customers only*** Express... **฿2.79**
4 x 20MG Original Lily Cialis **฿7.85**	(1g) High-grade Crystal Meth **฿11.95**	MindFood - Protect your brain!... **฿3.69**
to US 1/4 lb (qp) BC Master Kush... **฿121.37**	How to Grow Mushrooms **฿0.14**	Mushroom Indoor Growing - Easy... **฿0.29**

News:

- Escrow hedging **update**
- New feature to help protect **sellers**
- We are **hiring**! Get paid for a referral, too...
- Reclaim lost coins from **MyBitcoin.com**
- Seller ranking and feedback **overhaul**
- Change your Mt. Gox **password**

recent feedback:

What services get hidden?

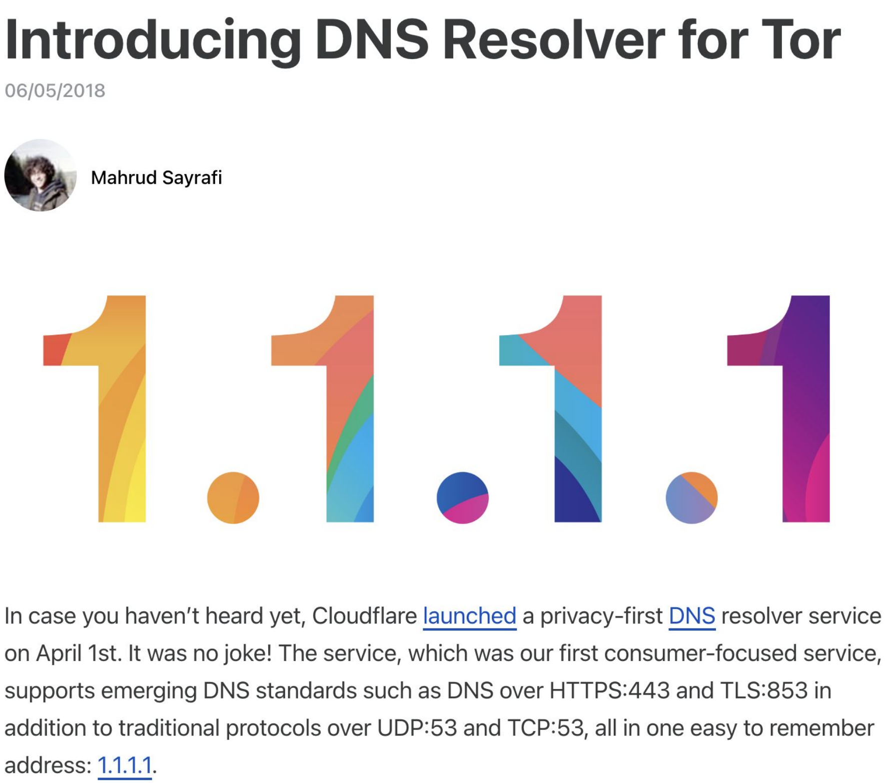


THIS HIDDEN SITE HAS BEEN SEIZED

by the Federal Bureau of Investigation,
in conjunction with the IRS Criminal Investigation Division,
ICE Homeland Security Investigations, and the Drug Enforcement Administration,
in accordance with a seizure warrant obtained by the
United States Attorney's Office for the Southern District of New York
and issued pursuant to 18 U.S.C. § 983(j) by the
United States District Court for the Southern District of New York

DEPARTMENT OF JUSTICE
QUI PRO DOMINA JUSTITIA SEQUITUR
DEPARTMENT OF JUSTICE
FEDERAL BUREAU OF INVESTIGATION
FIDELITY BRAVERY INTEGRITY
DRUG ENFORCEMENT ADMINISTRATION
US
SPECIAL AGENT
CRIMINAL INVESTIGATION
DEPARTMENT OF THE TREASURY
U S
SPECIAL AGENT
HOMELAND SECURITY INVESTIGATIONS
U S
SPECIAL AGENT

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

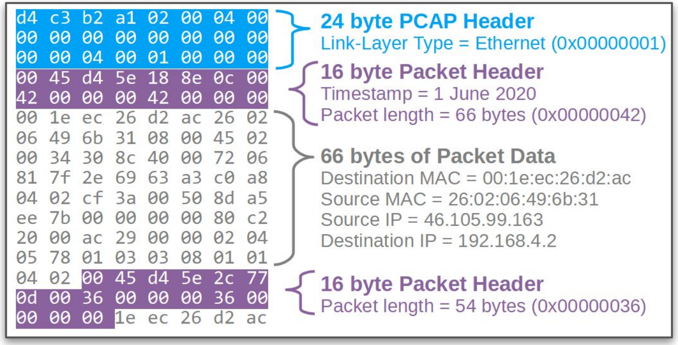
Project 4 Overview

- 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!



```
d4 c3 b2 a1 02 00 04 00
00 00 00 00 00 00 00 00
00 00 04 00 01 00 00 00
00 45 d4 5e 18 8e 0c 00
42 00 00 00 42 00 00 00
00 1e ec 26 d2 ac 26 02
06 49 6b 31 08 00 45 02
00 34 30 8c 40 00 72 06
81 7f 2e 69 63 a3 c0 a8
04 02 cf 3a 00 50 8d a5
ee 7b 00 00 00 00 80 c2
20 00 ac 29 00 00 02 04
05 78 01 03 03 08 01 01
04 02 00 45 d4 5e 2c 77
0d 00 36 00 00 00 36 00
00 00 00 1e ec 26 d2 ac
```

24 byte PCAP Header
Link-Layer Type = Ethernet (0x00000001)

16 byte Packet Header
Timestamp = 1 June 2020
Packet length = 66 bytes (0x00000042)

66 bytes of Packet Data
Destination MAC = 00:1e:ec:26:d2:ac
Source MAC = 26:02:06:49:6b:31
Source IP = 46.105.99.163
Destination IP = 192.168.4.2

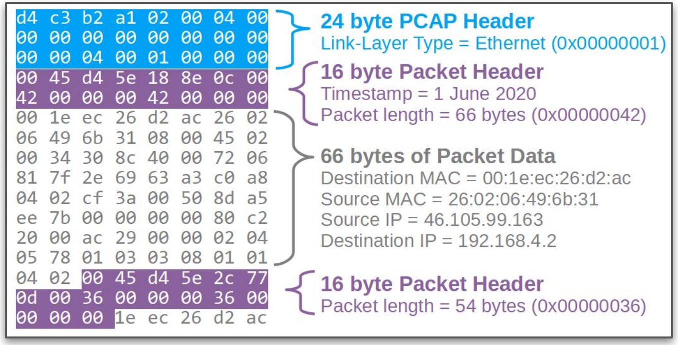
16 byte Packet Header
Packet length = 54 bytes (0x00000036)

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



```
d4 c3 b2 a1 02 00 04 00
00 00 00 00 00 00 00 00
00 00 04 00 01 00 00 00
00 45 d4 5e 18 8e 0c 00
42 00 00 00 42 00 00 00
00 1e ec 26 d2 ac 26 02
06 49 6b 31 08 00 45 02
00 34 30 8c 40 00 72 06
81 7f 2e 69 63 a3 c0 a8
04 02 cf 3a 00 50 8d a5
ee 7b 00 00 00 00 80 c2
20 00 ac 29 00 00 02 04
05 78 01 03 03 08 01 01
04 02 00 45 d4 5e 2c 77
0d 00 36 00 00 00 36 00
00 00 00 1e ec 26 d2 ac
```

24 byte PCAP Header
Link-Layer Type = Ethernet (0x00000001)

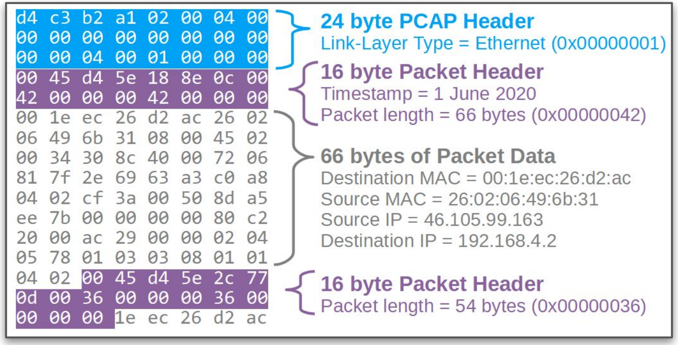
16 byte Packet Header
Timestamp = 1 June 2020
Packet length = 66 bytes (0x00000042)

66 bytes of Packet Data
Destination MAC = 00:1e:ec:26:d2:ac
Source MAC = 26:02:06:49:6b:31
Source IP = 46.105.99.163
Destination IP = 192.168.4.2

16 byte Packet Header
Packet length = 54 bytes (0x00000036)

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



```
d4 c3 b2 a1 02 00 04 00
00 00 00 00 00 00 00 00
00 00 04 00 01 00 00 00
00 45 d4 5e 18 8e 0c 00
42 00 00 00 42 00 00 00
00 1e ec 26 d2 ac 26 02
06 49 6b 31 08 00 45 02
00 34 30 8c 40 00 72 06
81 7f 2e 69 63 a3 c0 a8
04 02 cf 3a 00 50 8d a5
ee 7b 00 00 00 00 80 c2
20 00 ac 29 00 00 02 04
05 78 01 03 03 08 01 01
04 02 00 45 d4 5e 2c 77
0d 00 36 00 00 00 36 00
00 00 00 1e ec 26 d2 ac
```

24 byte PCAP Header
Link-Layer Type = Ethernet (0x00000001)

16 byte Packet Header
Timestamp = 1 June 2020
Packet length = 66 bytes (0x00000042)

66 bytes of Packet Data
Destination MAC = 00:1e:ec:26:d2:ac
Source MAC = 26:02:06:49:6b:31
Source IP = 46.105.99.163
Destination IP = 192.168.4.2

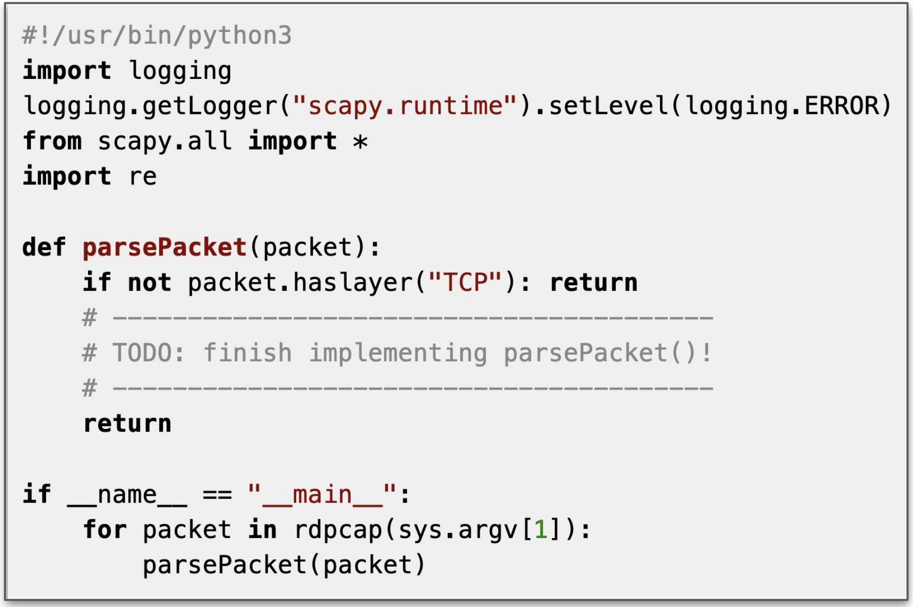
16 byte Packet Header
Packet length = 54 bytes (0x00000036)

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

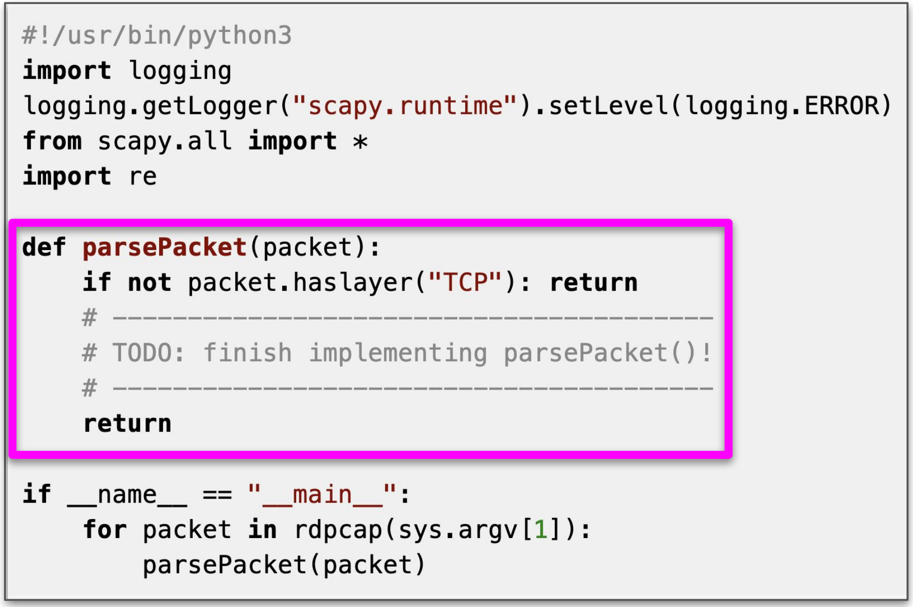


```python
#!/usr/bin/python3
import logging
logging.getLogger("scapy.runtime").setLevel(logging.ERROR)
from scapy.all import *
import re

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”**

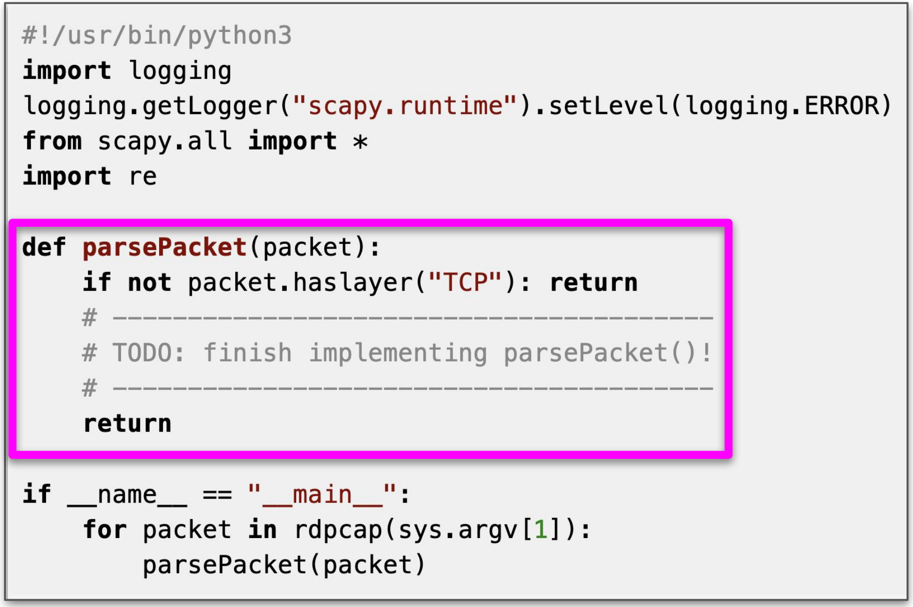
- We provide **skeleton code** template
  - Sets-up the packet parsing workflow
  - **Your job:** finish implementing the function **`parsePacket()`**



```
#!/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

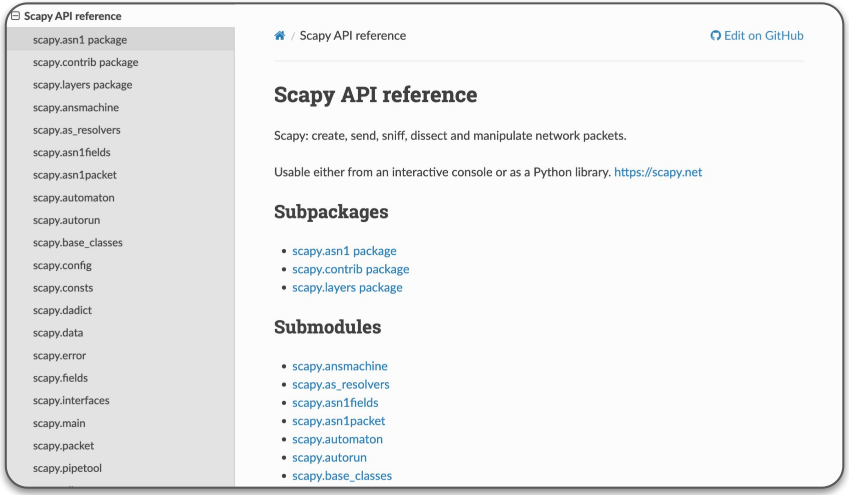
- Python API for programmatic packet capture and analysis
  - Think of it as **“Wireshark in API form”**
- We provide **skeleton code** template
  - Sets-up the packet parsing workflow
  - **Your job:** finish implementing the function **`parsePacket()`**
- You may also add **additional code**
  - E.g.,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



```python
#!/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 API reference

scapy.asn1 package
scapy.contrib package
scapy.layers package
scapy.ansmachine
scapy.as_resolvers
scapy.asn1fields
scapy.asn1packet
scapy.automaton
scapy.autorun
scapy.base_classes
scapy.config
scapy.consts
scapy.dadict
scapy.data
scapy.error
scapy.fields
scapy.interfaces
scapy.main
scapy.packet
scapy.pipetool

Scapy API reference

Scapy: create, send, sniff, dissect and manipulate network packets.

Usable either from an interactive console or as a Python library. https://scapy.net

Subpackages

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package

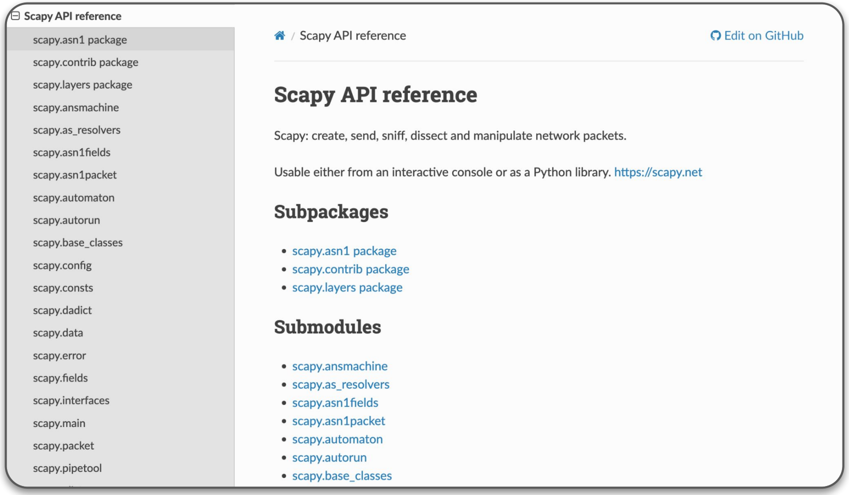
Submodules

- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes

Scapy Fundamentals

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

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



Scapy API reference

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package
- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes
- scapy.config
- scapy.consts
- scapy.dadict
- scapy.data
- scapy.error
- scapy.fields
- scapy.interfaces
- scapy.main
- scapy.packet
- scapy.pipetool

Scapy API reference

Scapy: create, send, sniff, dissect and manipulate network packets.

Usable either from an interactive console or as a Python library. https://scapy.net

Subpackages

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package

Submodules

- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes

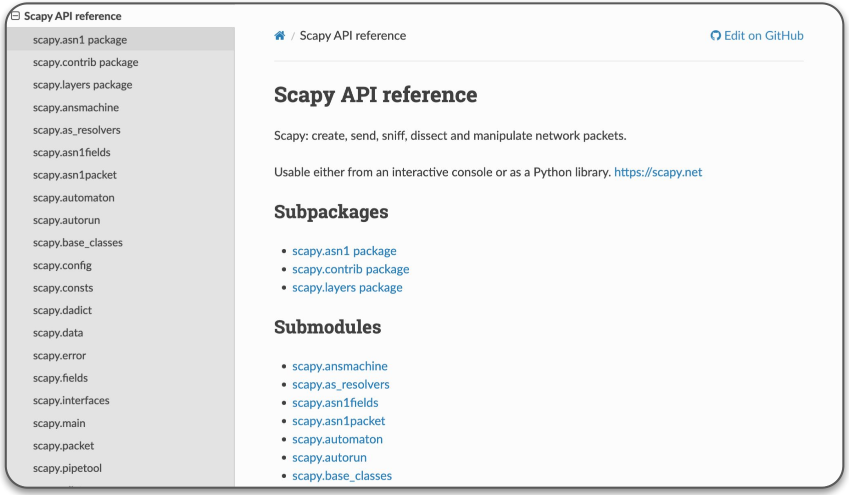
Scapy Fundamentals

- 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 API reference

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package
- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes
- scapy.config
- scapy.consts
- scapy.dadict
- scapy.data
- scapy.error
- scapy.fields
- scapy.interfaces
- scapy.main
- scapy.packet
- scapy.pipetool

/ Scapy API reference

Edit on GitHub

Scapy API reference

Scapy: create, send, sniff, dissect and manipulate network packets.

Usable either from an interactive console or as a Python library. https://scapy.net

Subpackages

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package

Submodules

- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes

Scapy Fundamentals

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

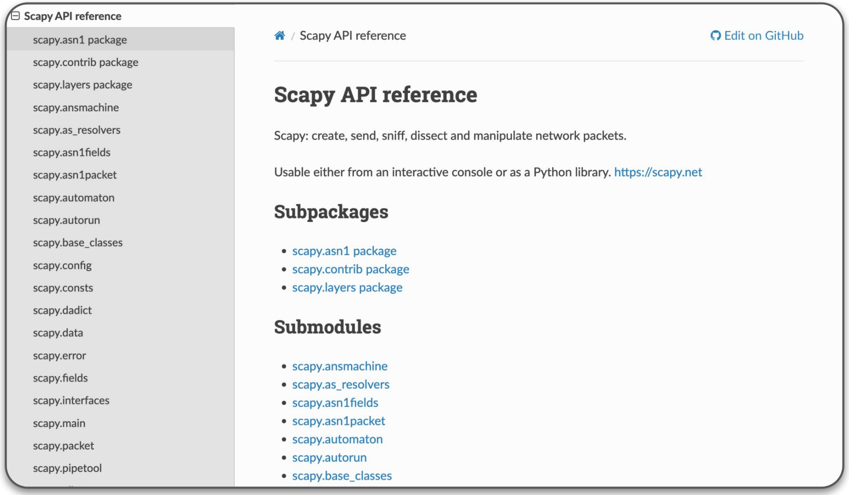
    ```
    packet[“TCP”].flags
    ```

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

    ```
    packet[“TCP”].dport
    ```

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

    ```
    packet[“IP”].src
    ```



Scapy API reference

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package
- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes
- scapy.config
- scapy.consts
- scapy.dadict
- scapy.data
- scapy.error
- scapy.fields
- scapy.interfaces
- scapy.main
- scapy.packet
- scapy.pipetool

/ Scapy API reference

Edit on GitHub

Scapy API reference

Scapy: create, send, sniff, dissect and manipulate network packets.

Usable either from an interactive console or as a Python library. https://scapy.net

Subpackages

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package

Submodules

- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes

Scapy Fundamentals

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

```
packet[“TCP”].flags
```

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

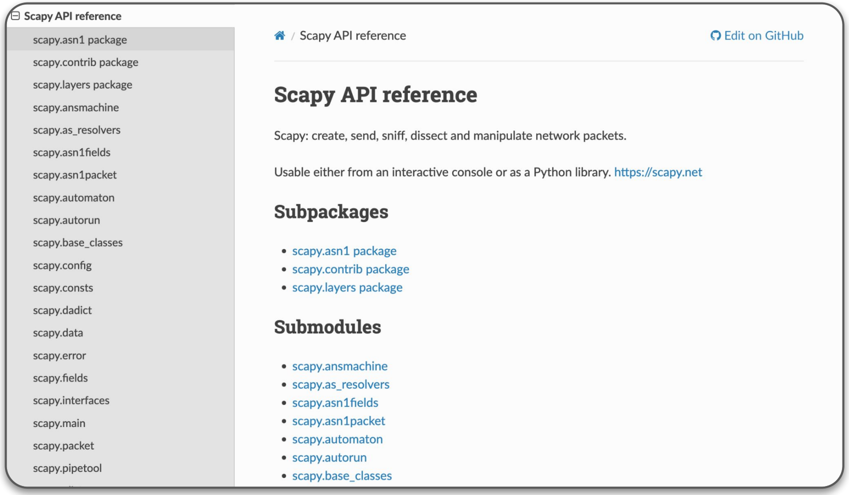
```
packet[“TCP”].dport
```

  - 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 API reference

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package
- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes
- scapy.config
- scapy.consts
- scapy.dadict
- scapy.data
- scapy.error
- scapy.fields
- scapy.interfaces
- scapy.main
- scapy.packet
- scapy.pipetool

Scapy API reference

Scapy: create, send, sniff, dissect and manipulate network packets.

Usable either from an interactive console or as a Python library. https://scapy.net

Subpackages

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package

Submodules

- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes

Scapy Fundamentals

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

    ```
    packet[“TCP”].flags
    ```

  - Get a pa

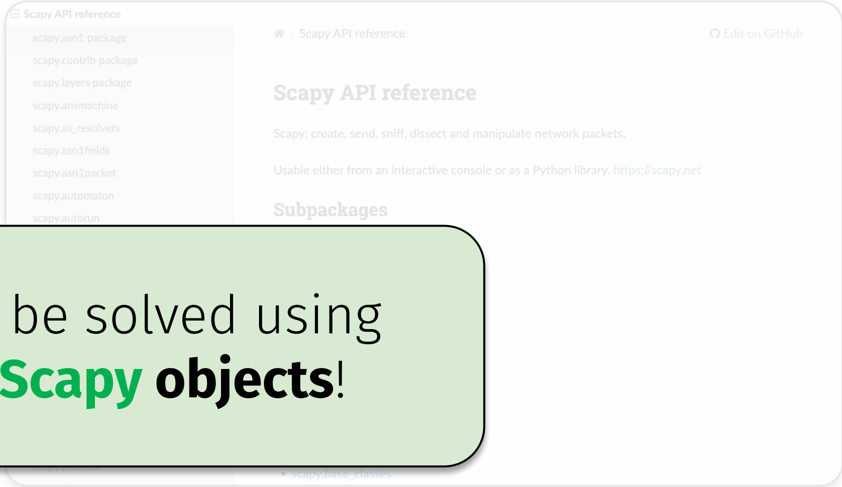
    ```
    pack
    ```

  - Get a pa

    ```
    packet[“IP”].src
    ```

  - Get a packet’s TCP **payload**:

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



Scapy API reference

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package
- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun

Scapy API reference

Scapy: create, send, sniff, dissect and manipulate network packets.

Usable either from an interactive console or as a Python library. https://scapy.net

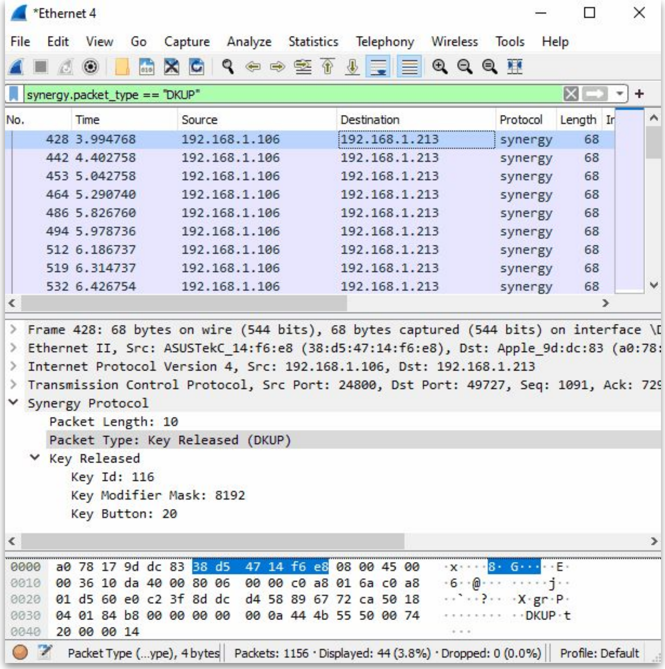
Subpackages

- scapy.base_classes

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!



synergy.packet_type == "DKUP"

```
No.      Time        Source           Destination      Protocol  Length  Ir
    428 3.994768    192.168.1.106    192.168.1.213    synergy      68
    442 4.402758    192.168.1.106    192.168.1.213    synergy      68
    453 5.042758    192.168.1.106    192.168.1.213    synergy      68
    464 5.290740    192.168.1.106    192.168.1.213    synergy      68
    486 5.826760    192.168.1.106    192.168.1.213    synergy      68
    494 5.978736    192.168.1.106    192.168.1.213    synergy      68
    512 6.186737    192.168.1.106    192.168.1.213    synergy      68
    519 6.314737    192.168.1.106    192.168.1.213    synergy      68
    532 6.426754    192.168.1.106    192.168.1.213    synergy      68
```

```
> Frame 428: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface \D
> Ethernet II, Src: ASUSTekC_14:f6:e8 (38:d5:47:14:f6:e8), Dst: Apple_9d:dc:83 (a0:78:
> Internet Protocol Version 4, Src: 192.168.1.106, Dst: 192.168.1.213
> Transmission Control Protocol, Src Port: 24800, Dst Port: 49727, Seq: 1091, Ack: 729
v Synergy Protocol
    Packet Length: 10
    Packet Type: Key Released (DKUP)
  v Key Released
      Key Id: 116
      Key Modifier Mask: 8192
      Key Button: 20
```

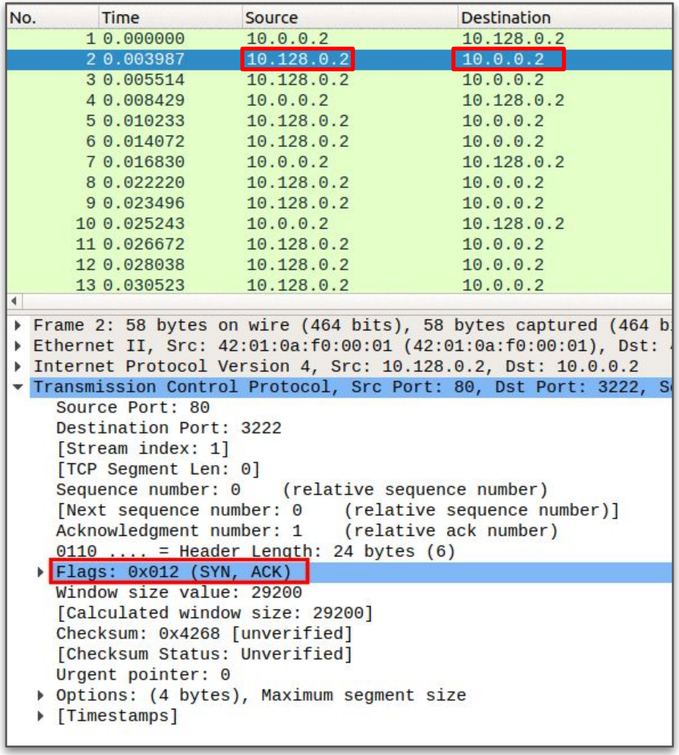
```
0000  a0 78 17 9d dc 83 38 d5  47 14 f6 e8 08 00 45 00   ·x····8· G·····E·
0010  00 36 10 da 40 00 80 06  00 00 c0 a8 01 6a c0 a8   ·6··@··· ·····j··
0020  01 d5 60 e0 c2 3f 8d dc  d4 58 89 67 72 ca 50 18   ··`··?·· ·X·gr·P·
0030  04 01 84 b8 00 00 00 00  00 0a 44 4b 55 50 00 74   ········ ··DKUP·t
0040  20 00 00 14                                         ···
```

Packet Type (...ype), 4 bytes | Packets: 1156 · Displayed: 44 (3.8%) · Dropped: 0 (0.0%) | Profile: Default

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?



```
No.   Time        Source        Destination
  1 0.000000    10.0.0.2      10.128.0.2
  2 0.003987    10.128.0.2    10.0.0.2
  3 0.005514    10.128.0.2    10.0.0.2
  4 0.008429    10.0.0.2      10.128.0.2
  5 0.010233    10.128.0.2    10.0.0.2
  6 0.014072    10.128.0.2    10.0.0.2
  7 0.016830    10.0.0.2      10.128.0.2
  8 0.022220    10.128.0.2    10.0.0.2
  9 0.023496    10.128.0.2    10.0.0.2
 10 0.025243    10.0.0.2      10.128.0.2
 11 0.026672    10.128.0.2    10.0.0.2
 12 0.028038    10.128.0.2    10.0.0.2
 13 0.030523    10.128.0.2    10.0.0.2
```

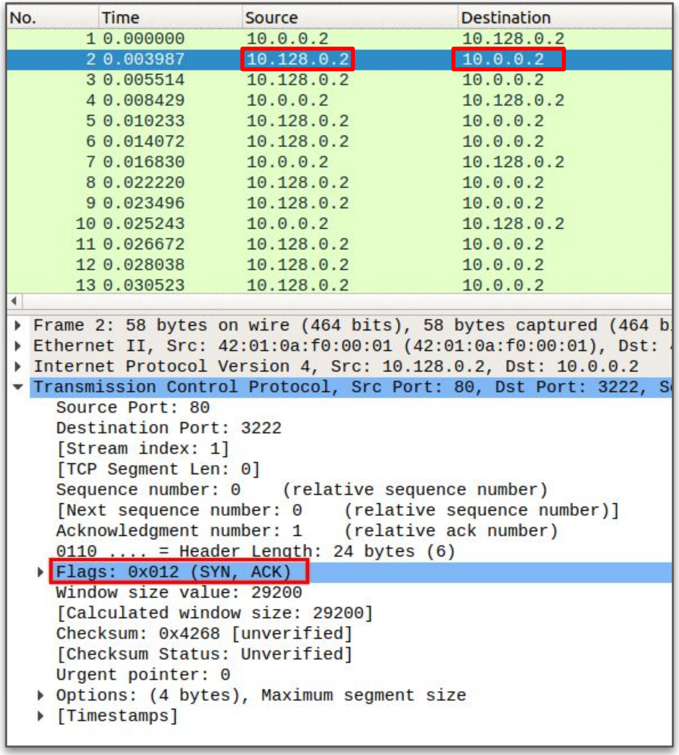
```
▸ Frame 2: 58 bytes on wire (464 bits), 58 bytes captured (464 b
▸ Ethernet II, Src: 42:01:0a:f0:00:01 (42:01:0a:f0:00:01), Dst:
▸ Internet Protocol Version 4, Src: 10.128.0.2, Dst: 10.0.0.2
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 3222, S
    Source Port: 80
    Destination Port: 3222
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0110 .... = Header Length: 24 bytes (6)
  ▸ Flags: 0x012 (SYN, ACK)
    Window size value: 29200
    [Calculated window size: 29200]
    Checksum: 0x4268 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (4 bytes), Maximum segment size
  ▸ [Timestamps]
```

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!



```
No.     Time        Source          Destination
  1 0.000000        10.0.0.2        10.128.0.2
  2 0.003987        10.128.0.2      10.0.0.2
  3 0.005514        10.128.0.2      10.0.0.2
  4 0.008429        10.0.0.2        10.128.0.2
  5 0.010233        10.128.0.2      10.0.0.2
  6 0.014072        10.128.0.2      10.0.0.2
  7 0.016830        10.0.0.2        10.128.0.2
  8 0.022220        10.128.0.2      10.0.0.2
  9 0.023496        10.128.0.2      10.0.0.2
 10 0.025243        10.0.0.2        10.128.0.2
 11 0.026672        10.128.0.2      10.0.0.2
 12 0.028038        10.128.0.2      10.0.0.2
 13 0.030523        10.128.0.2      10.0.0.2
```

```
Frame 2: 58 bytes on wire (464 bits), 58 bytes captured (464 b
Ethernet II, Src: 42:01:0a:f0:00:01 (42:01:0a:f0:00:01), Dst: 
Internet Protocol Version 4, Src: 10.128.0.2, Dst: 10.0.0.2
Transmission Control Protocol, Src Port: 80, Dst Port: 3222, S
  Source Port: 80
  Destination Port: 3222
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 0    (relative sequence number)
  [Next sequence number: 0    (relative sequence number)]
  Acknowledgment number: 1    (relative ack number)
  0110 .... = Header Length: 24 bytes (6)
  Flags: 0x012 (SYN, ACK)
  Window size value: 29200
  [Calculated window size: 29200]
  Checksum: 0x4268 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  Options: (4 bytes), Maximum segment size
  [Timestamps]
```

Questions?

Next time on CS 4440…

Software Reverse Engineering