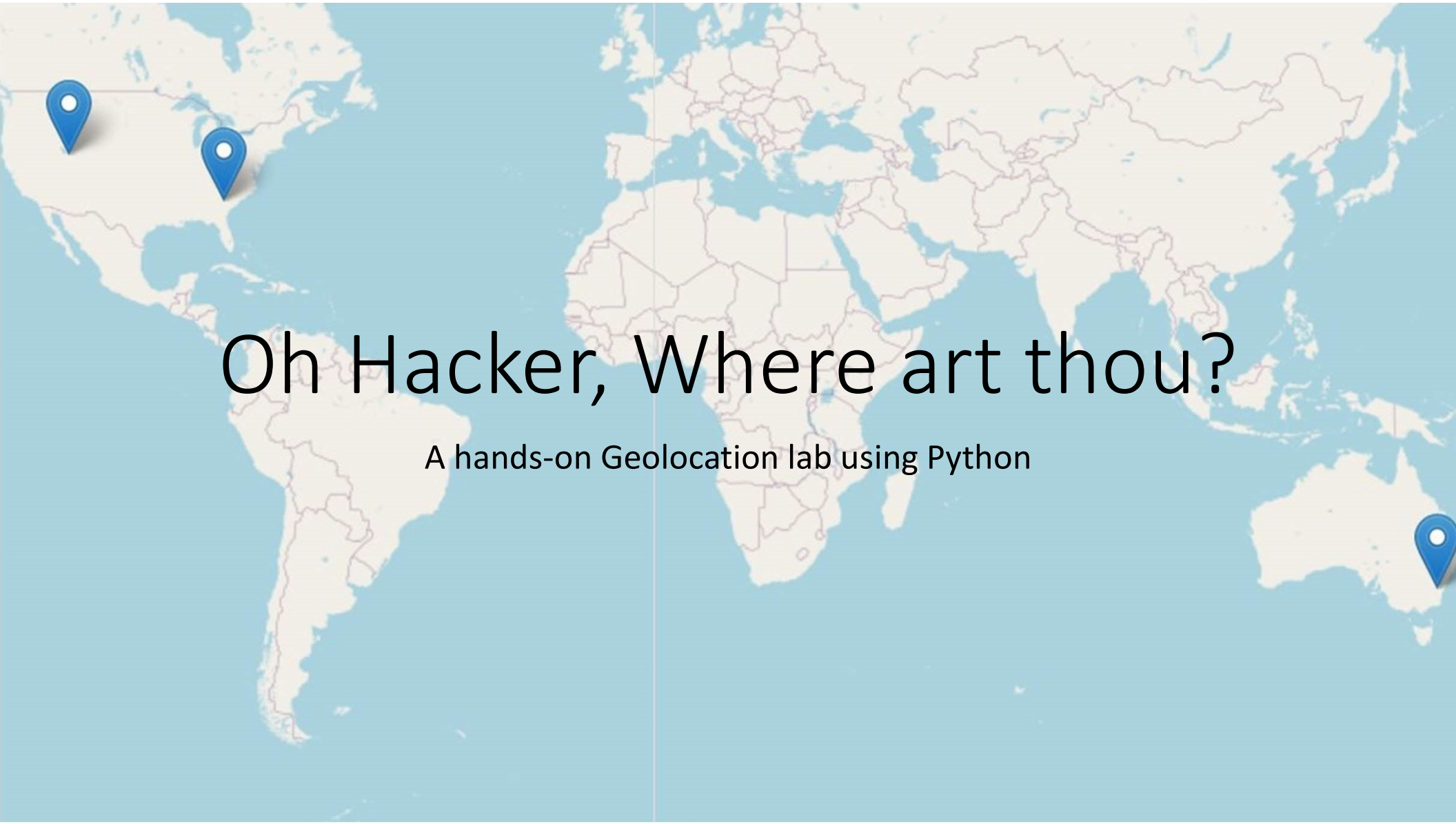


Oh Hacker, Where art thou?

A hands-on Geolocation lab using Python

SANS free resources are aimed to provide the latest in research and technology available to help support awareness and growth across a wide range of IT and OT security considerations.

Curriculum Free Resources

SANS CLOUD SECURITY
SANS CYBER DEFENSE
SANS DFIR
SANS ICS
SANS MANAGEMENT
SANS PENTEST

sans.org/free

- girls go cyberstart
- CYBER START
- Test Drive Courses
- Tech Tuesday Workshops
- Free Training
- CYBER ACES
- Cyber Discovery
- SANS CYBER IMMERSION ACADEMY
- Scholarships & Community
- Cyber Workforce Academy
- Diversity Cyber Academy
- Vet Success
- Women's Immersion Academy
- Cyber FastTrack
- Cyber Talent
- Cheat Sheets
- Posters
- WHITE PAPERS
- NEWSLETTERS
- NewsBites
- @Risk
- OUCH!
- News & Updates
- INTERNET STORM CENTER
- WEBCASTS
- BLOG
- Podcasts
- Podcast
- TRUST ME, I'm Certified GIAC CERTIFICATIONS
- BLUEPRINT
- Daily Stormcast https://isc.sans.edu
- Holiday Hack Challenge
- LEVEL UP
- Challenges & Ranges
- NETWARS Mini Ranges
- Jupiter Rockets
- Free Tools (Over 150)
- slingshot linux distribut
- SIFT WORKSTATION
- SOF-ELK
- REMnux

SANS

How to setup your machine

- Register and log into the SLACK Channel
  - https://sansurl.com/balance-17may
- Download the Class Virtual Machine and import it into VMWare
  - https://tinyurl.com/locatehacker
  - Download password is "whereRUhacker?"
- Login as "student" with password of "student"
- Double-Click "lab_setup.sh" on the Desktop inside the VM

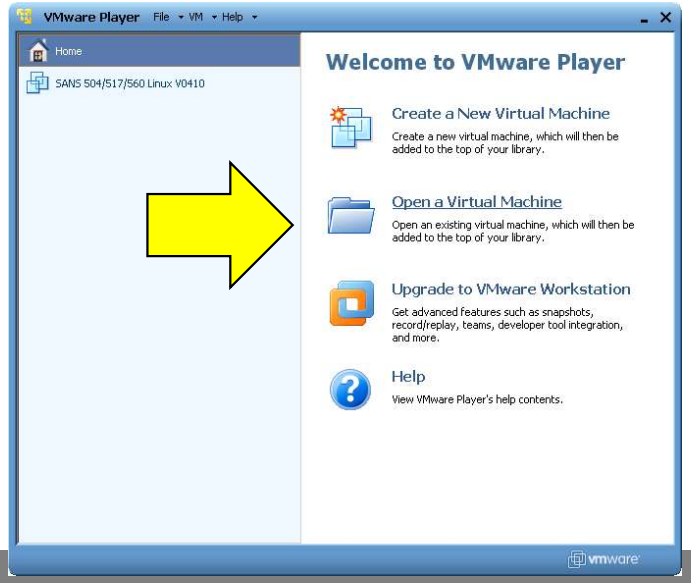
The Backup Plan

If you can not get the Virtual Machine working

http://github.com/markbaggett/GeoLocationNotebook

Start Configuring your Workshop VM Now

- You will need a virtualization product for this workshop
- VMWare Workstation offers a free 30 day trial:
  - https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html
- Import the VM Into VMWare or Virtualization Software:
  - select File > Import in VMware
  - Select "Open a Virtual Machine" and select the OVF file
- Run VMware, open the VM, and boot it
- Make sure your VM has access to the internet
- Once the Desktop appears you can click the lab_setup script to download the material.



```
Get-ADUser -Filter "Mark Baggett"| fl -Properties *
```

- Mark Baggett
- Penetration Testing and Incident Response Consulting
- Senior SANS Instructor
- Author of SANS SEC573 Automating InfoSec with Python
- Masters in Information Security Engineering
- GSE #15
- DoD Advisor, Former CISO 18 years commercial

```
student@573:/opt/metasploit-framework$ grep -Ri "mark baggett" | wc -l
7
```

Todays Topic

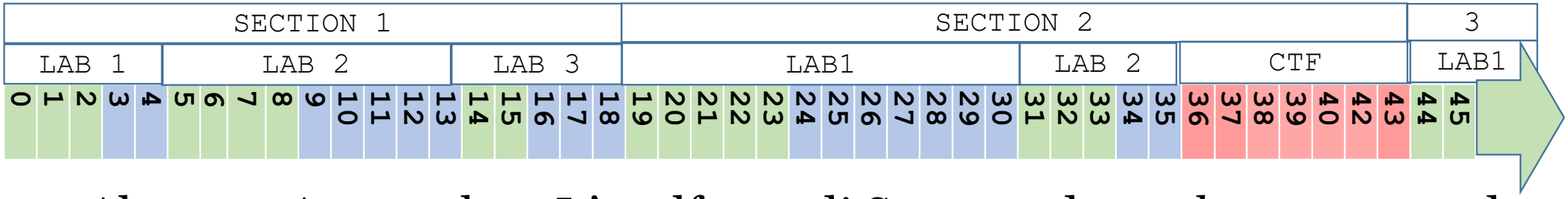
- Todays topic assumes you know some Python
- SANS Sec573 Automating Information Security with Python does not assume prior knowledge
- OnDemand is now available!

Course Overview

- Days 1 and 2: Essentials Workshop
  - Build the skills required to rapidly develop information security tools on your own
- Days 3–5: Continue Learning Coding Concepts
  - Day 3 - Defensive focused projects
  - Day 4 - Forensics focused projects
  - Day 5 - Offensive focused projects
- Day 6: Capstone "CTF" Event
- Days 1–5: pyWars Challenge and CTF
  - Master the nuances of Python programming

Recommended New Coders' GPYC Self-Paced Study Plan

- **Listen to lecture** and expect to **finish 33%–50% of labs** (shown in green) during time allotted in class. GREEN = Completed In Class
- **Finish** all the **non-pyWars labs** during the time allotted **in class**
- **Complete the rest** of labs (shown in blue) on your local pyWars server before the exam; **evenings** after class **or** when you get **back home**. BLUE = At Home
- CTF Challenges (shown in red) are scattered throughout pyWars. They are for the veterans' CTF. RED = Ignore Them



- Alternate Approaches: It's self-paced! Some students choose to spend all week on Sections 1 and 2, but then you're on your own for 3–5.

Common Veteran Coders' Self-Paced Study Plan

- Usually, veteran coders will ignore lectures and race through the first two days on Day 1 and then focus on Days 3–5
- Use the "Roadmap Slide" to determine when you need to pay closer attention
- After completing Days 3–5, there are 20+ CTF challenges that go far beyond GPYC and the course material
- The FIRST person to complete ALL the challenges gets a CTF coin
- Instructors will be vague when assisting students in CTF until someone finishes
- Finished? Most "Finishers" have skipped all non-pyWars labs, including Day 5!

Day 1		Day 2	Day 3	Day4	DAY 5	At Home		
Day 1 1-18	**Day 2 19-35**	**Day 3 36-61**	**Day 4 51-66**	**Advanced pyWars CTF 67 - 90+**		**Non-pyWars Labs**	**Day 5**	**Hall of Fame Challenges**

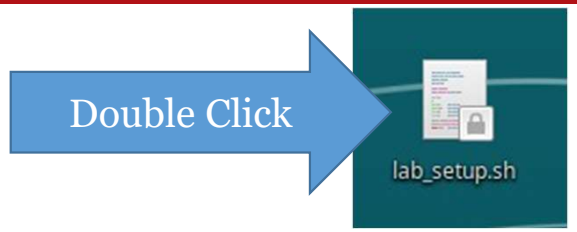
- The top 1% will complete all pyWars and only some of the non-pyWars labs

Majority of Students' Self-Paced Study Plan

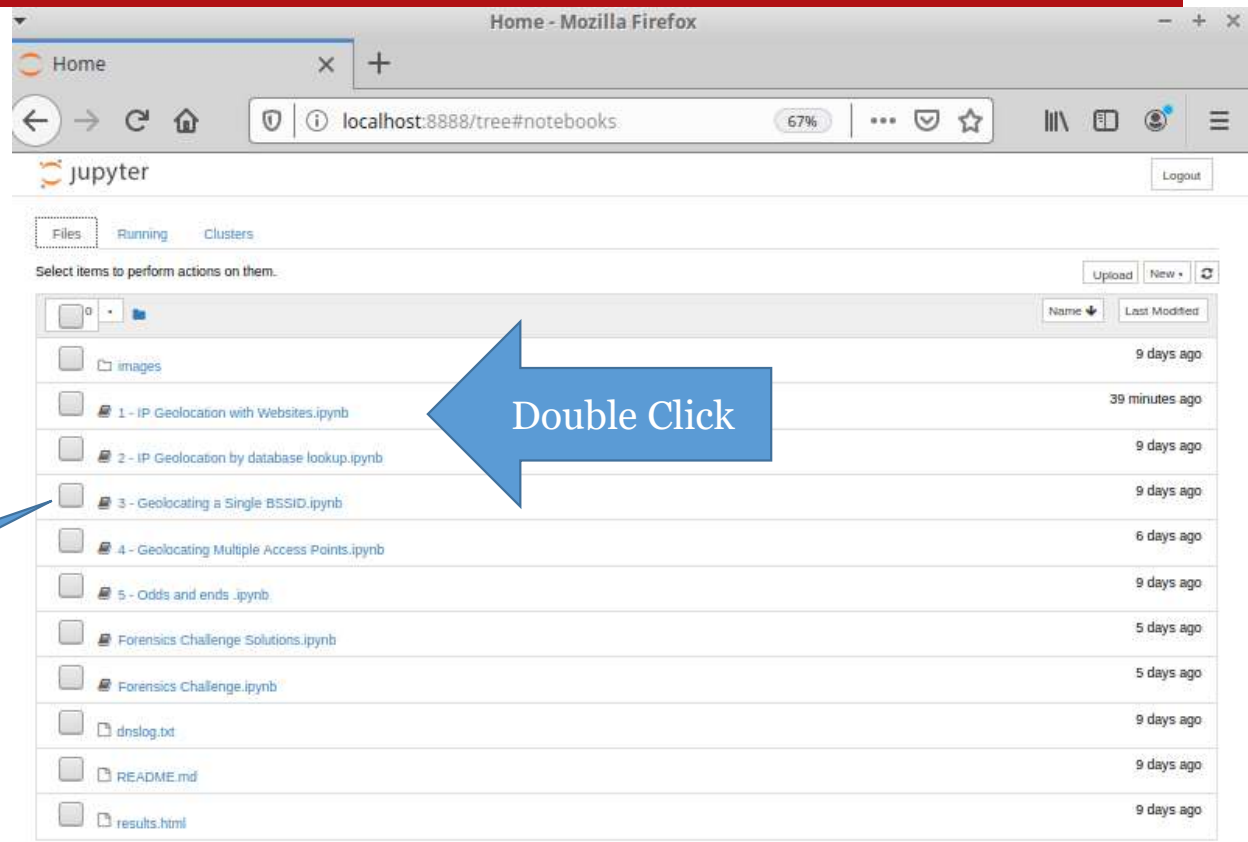
- The majority of you will be somewhere in between those two approaches
- Alternate back and forth between listening to lectures and self-paced labs
- You also have more work than you can finish! Choose where to focus your time
- You may have an incomplete lab or two that you can finish later, particularly on Day 3, when we provide more than double the labs than you can normally complete
- Keep in mind that the ONLY thing you don't have access to after class is the Veterans CTF. For this reason, some people choose to leave some non-pyWars labs or a few pyWars challenges unsolved and take a shot at those advanced CTF questions.
- You choose the learning model that is best for you. In other words, it is self-paced.

DAY 1	DAY 2	DAY 3	DAY 4	DAY 5		At Home
1-18 and Non-pyWars Labs	19-35 and Non-pyWars Labs	36-61 and Non-pyWars Labs	62-65 and Non-pyWars Labs	Day 5 is all Non-pyWars Labs	Advanced PyWars CTF 65 - 90+	Hall of Fame Challenges

Double Click lab_setup.sh on the Desktop to Start

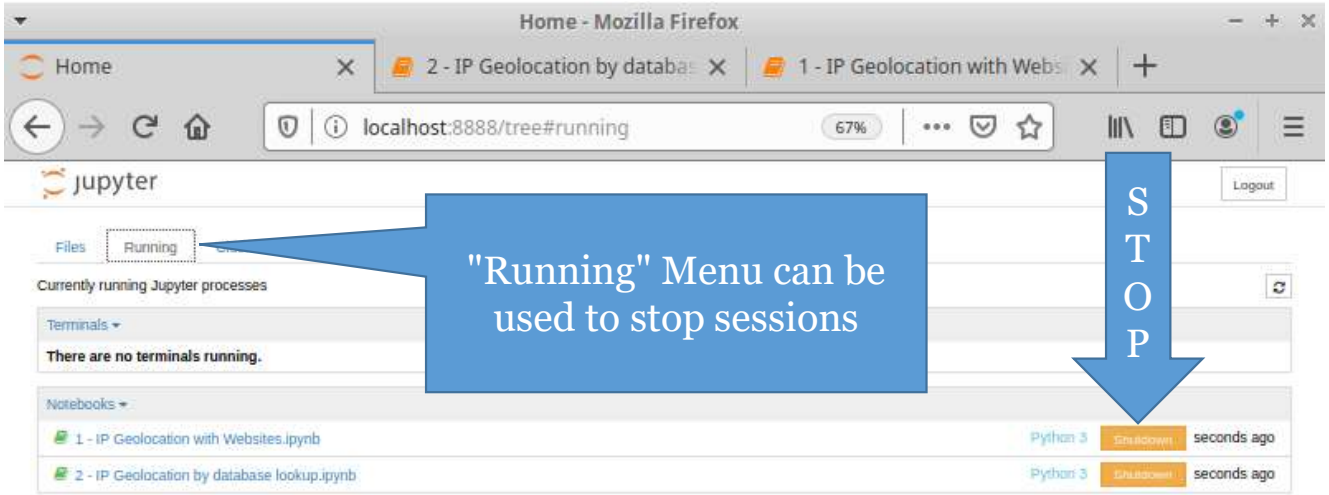


After running lab_setup Jupyter notebooks will open!



Double Click a page to **launch a new session** and interact with that page

Shutting Down a Session



You Can Change These Pages!

- On each page there are "CELLS" that contain code
- Blue line is read mode

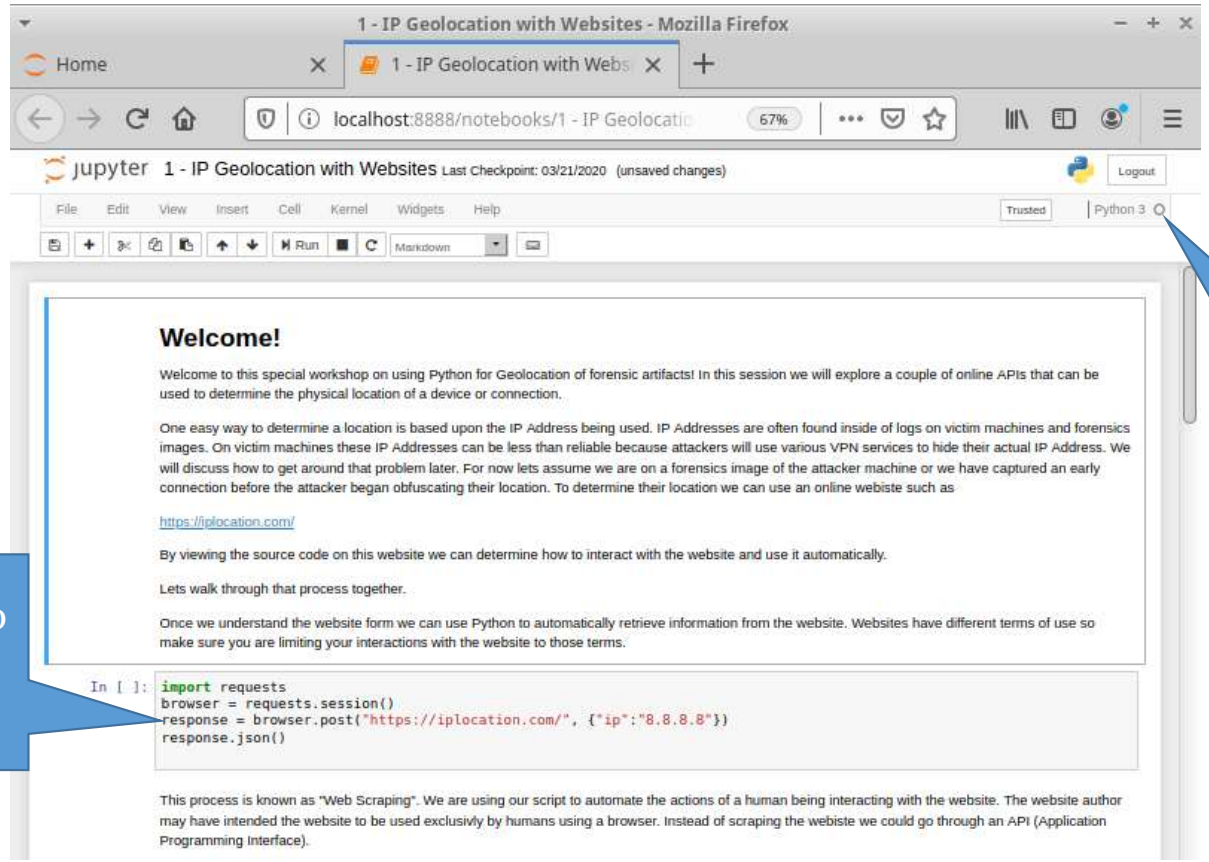
```
In [ ]: import requests
        browser = requests.session()
        response = browser.post("https://iplocation.com/", {"ip":"8.8.8.8"})
        response.json()
```

- Double click a cell to begin editing
- Green line is edit mode

```
In [ ]: you can edit a cell when the line is GREEN
        import requests
        browser = requests.session()
        response = browser.post("https://iplocation.com/", {"ip":"8.8.8.8"})
        response.json()
```

- You can add new cells

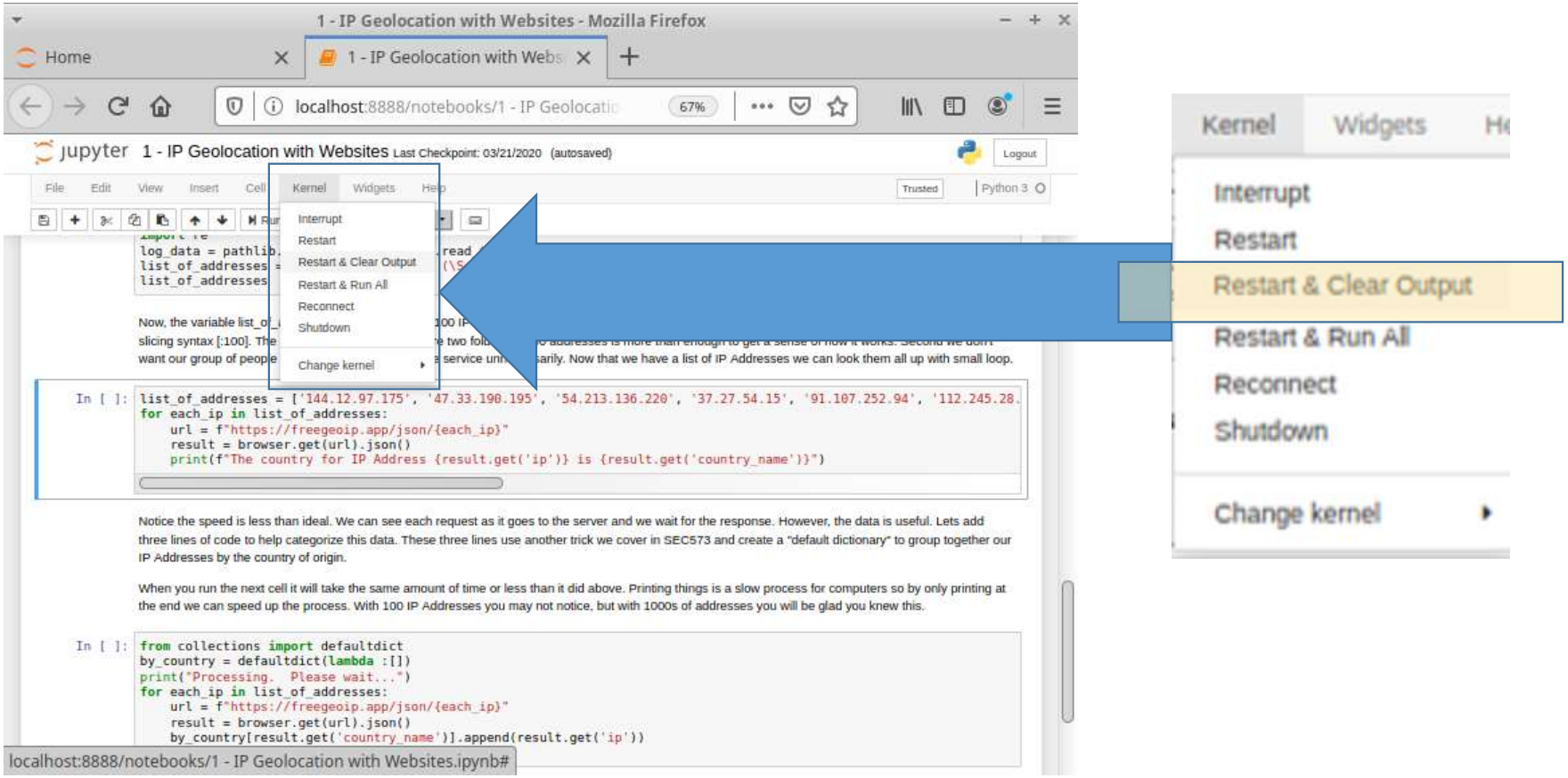
Run a single cell by pressing CONTROL+ENTER



Turns black while code is running

CONTROL + ENTER to execute a "cell" in Python session

Reset your Python Session with Kernel Menu



Cell Menu allows you to Run Previous Cells

1 - IP Geolocation with Websites - Mozilla Firefox

Home | 1 - IP Geolocation with Webs

localhost:8888/notebooks/1 - IP Geolocati 67%

jupyter 1 - IP Geolocation with Websites Last Checkpoint: 03/21/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Cells
Run Cells and Select Below
Run Cells and Insert Below
Run All
Run All Above
Run All Below
Cell Type
Current Outputs
All Output

Now, the variable ... slicing syntax [:1... want our group o...

In [ ]:
```
list_of_addr...
for each_ip ...
    url = f"...
    result = browser.get(url).json()
    print(f"The country for IP Address {result.get('ip')} is {result.get('country_name')}")
```

Notice the speed is less than ideal. We can see each request as it goes to the server and we wait for the response. However, the data is useful. Lets add three lines of code to help categorize this data. These three lines use another trick we cover in SEC573 and create a "default dictionary" to group together our IP Addresses by the country of origin.

When you run the next cell it will take the same amount of time or less than it did above. Printing things is a slow process for computers so by only printing at the end we can speed up the process. With 100 IP Addresses you may not notice, but with 1000s of addresses you will be glad you knew this.

In [ ]:
```
from collections import defaultdict
by_country = defaultdict(lambda :[])
print("Processing.  Please wait...")
for each_ip in list_of_addresses:
    url = f"https://freegeoip.app/json/{each_ip}"
    result = browser.get(url).json()
    by_country[result.get('country_name')].append(result.get('ip'))
```

localhost:8888/notebooks/1 - IP Geolocation with Websites.ipynb#

Cell Kernel Widgets

Run Cells
Run Cells and Select Below
Run Cells and Insert Below
Run All
Run All Above
Run All Below
Cell Type
Current Outputs
All Output

Challenge 1

Forensics Challenge 1 - The Outlook Web Access Attack

The victim lost 1.2 million dollars in an Outlook Web Access Phishing attack. You have pulled all of the SUCCESSFUL LOGINS from the Outlook Administrator console. But there are THOUSANDS of IP addresses to look through. The original attack came from Iran. You decide to focus your attention on those requests that are from the same country of origin.

```
['144.12.97.175',
 '47.33.190.195',
 '54.213.136.220',
 '37.27.54.15',
 '91.107.252.94',
 '112.245.28.187',
 '77.198.208.109',
 '79.195.246.215',
 '109.219.196.231',
 '111.14.43.156',
 '102.111.174.6',
 '176.65.253.46',
 '102.155.114.82']
```

Which IP Addresses originated in Iran?

Locating an Attacker by IP Address

- We find IP Addresses in Logs and Forensics Artifacts everywhere we look!
- Why turn those into locations?
  - Put bad guys in jail!
  - Determine relationships between IP Addresses in logs for more detailed investigations
- What are the Limitations?
  - UDP protocols connections can be spoofed
  - TCP Protocol can be spoofed by ISPs and any other "upstream" entities
  - Some TCP Protocol artifacts can be spoofed such as SMTP Headers
  - Criminals love to hide behind VPN connections that hide their real IP

Using websites to lookup IP Addresses

- Python can use website to determine where the originator of a connection is physically location based on their IP Address
- For Python to interact with website we will import a module called "Requests"

Using Requests Session()

- Think of this as creating a browser that remembers settings and headers like User-Agent and maintains state via cookies
- Call the browser objects get(), post(), and so on to use the settings

```
>>> import requests
>>> browser = requests.session()
>>> browser.headers
{'User-Agent': 'python-requests/2.21.0', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*',
'Connection': 'keep-alive'}
>>> browser.headers['User-Agent']
'python-requests/2.21.0'
>>> browser.headers['User-Agent']='Mozilla FutureBrowser 145.9'
>>> browser.headers
{'User-Agent': 'Mozilla FutureBrowser 145.9', 'Accept-Encoding': 'gzip, deflate', 'Accept':
'*/*', 'Connection': 'keep-alive'}
>>> x = browser.get("http://isc.sans.edu")
>>> type(x)
<class 'requests.models.Response'>
```

Create a browser object that remembers cookies, settings, and headers

Outbound headers are stored in a dictionary that you can modify to meet your needs

Response Objects

- All those methods return a response object with access to full details about the webpage's response

```
>>> resp = browser.get("http://isc.sans.edu")
>>> type(resp)
<class 'requests.models.Response'>
>>> dir(resp)
[ <dunders deleted>, '_content', '_content_consumed', 'apparent_encoding', 'close',
'connection', 'content', 'cookies', 'elapsed', 'encoding', 'headers', 'history',
'is_permanent_redirect', 'is_redirect', 'iter_content', 'iter_lines', 'json', 'links', 'ok',
'raise_for_status', 'raw', 'reason', 'request', 'status_code', 'text', 'url']
>>> resp.status_code, resp.reason
(200, 'OK')
>>> resp.headers
{'Content-Length': '29055', 'Content-Encoding': 'gzip', 'Set-Cookie':
'SRCHD=AF=NOFORM; domain=.bing.com; expires=Wed, 11-Apr-2019 12:48:57 GMT;
'Content-Type': 'text/html; charset=utf-8}
>>> resp.content[:70]
b'<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://'
```

The response from the server

All the headers in the response

The content of the webpage!

Browser GET/POST Requests

Create your browser object

```
>>> import requests
>>> browser = requests.session()
```

Make GET requests to retrieve content

```
>>> resp = browser.get("http://www.bing.com")
>>> resp.content[:60]
b'<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//E'
```

Make POST requests to submit data to forms

```
>>> postdata = {'username':'markb', 'password':'sec573'}
>>> resp = browser.post("http://web.page/login",postdata)
>>> print(resp.content)
b'Login Failed'
```

Cookies and other settings automatically persist across all actions that use the browser object

Instead of a website we could use a database!

- MaxMind maintains and distributes several IP information databases
- The free location database product is referred to as "GeoLite2"
- Module is easily installed with pip

```
$ pip install geoip2
```

- Databases can be directly downloaded from their website
  - https://dev.maxmind.com/geoip/geoip2/geolite2/
- Or a utility called geoipupdate can automatically downloads the monthly updates
  - First, add the MaxMind repository to your apt repository list
  - Then install their free geoipupdate utility and update your local database
  - Register for a free account to obtain an API key

```
$ sudo add-apt-repository ppa:maxmind/ppa
$ sudo apt update
$ sudo apt install geoipupdate
$ sudo geoipupdate
$ sudo python -m pip install geoip2
```

WALKTHROUGH WORKBOOK 1 and 2 TOGETHER

Challenge 2

Forensics Challenge 2 - At the scene of the crime

After obtaining the suspects laptop you need to place him at the scene of the crime. After pulling their wireless history from the registry,srum and event lots you come up with the following list of Wireless BSSIDs. What locations can you place the suspects laptop at from the following Wireless BSSID addresses?

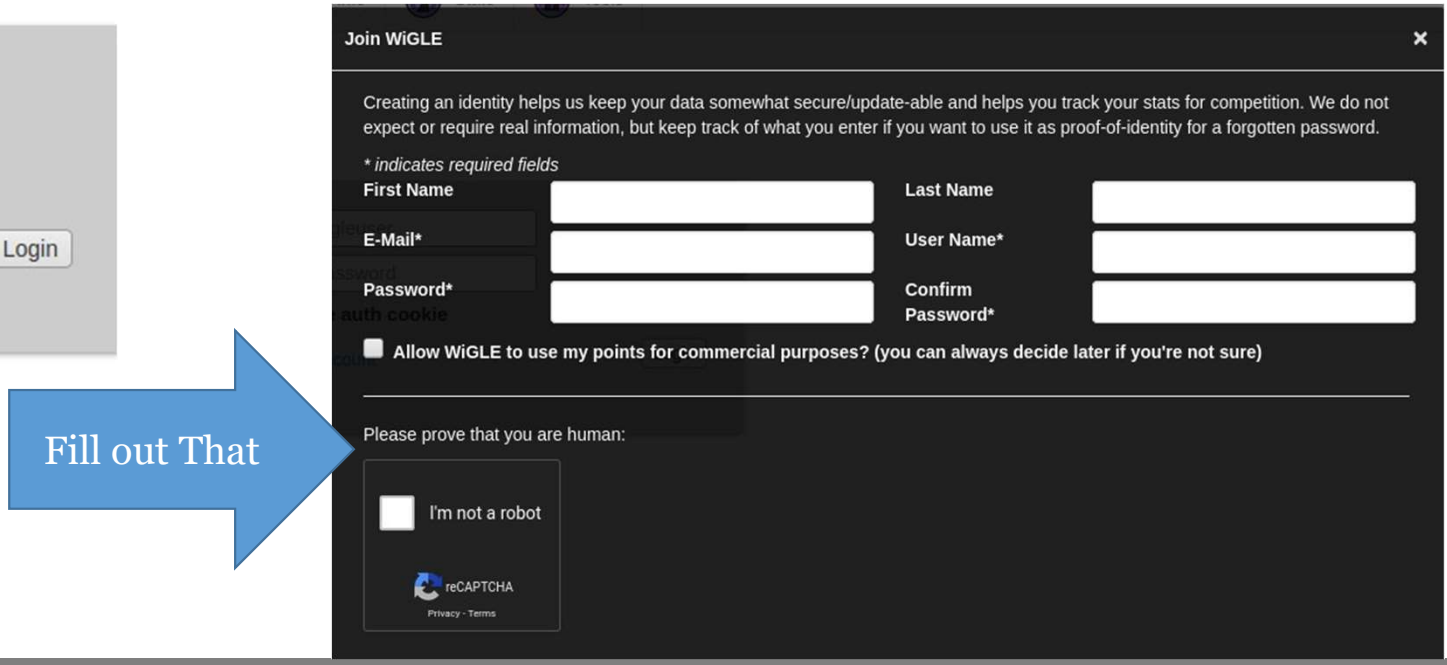
```
['70-0F-6A-32-2C-0F',
'0C-F5-A4-96-CD-C7',
'74-3E-2B-36-57-E8',
'08-4F-A9-3C-F5-A0',
'70-70-8B-29-12-81',
'0C-F5-A4-96-CD-CB']
```

Workbook 3 - Looking up a single BSSID

- There are limited resources available for looking up the location of 1 wireless network.
- Wigle.net - Free community based website
  - Fed by people collecting these from their phones and other devices and uploading that to wiggle
  - Free API is available for anyone to query the data
  - Administrators keep tight reigns on abuse
- A similar site to wigle focused on Europe
  - Much smaller database
  - "http://api.mylnikov.org/geolocation/wifi

How to get a Wigle.net API key

- http://wigle.net and click TOOLS -> ACCOUNT or
- https://wigle.net/login?destination=/account



Go back to your account page after logging in

- Login to wigle and go back to your ACCOUNT page

Workbook 4 - Geolocating BSSIDs

- Google and several others provide WIFI triangulation services
  - You submit BSSID and the signal strength and it calculates where you are relative to those Wireless Access Points
  - Event 6100s are pretty scarce in forensics information
  - With a little creativity we can still use these APIs to find artifacts that are plentiful in our forensics investigations

Google Geolocation API Pricing

- $5.00 for 1000 API requests
- The first $200.00 (40,000 request) a month is free

Obtaining a Google Maps API key

- Google "Get Google Maps API key"
- https://cloud.google.com/console/google/maps-apis/overview

This is a three step process

- 1) Enable the GEOLOCATION API in your account
- 2) Provide a Credit Card under your billing information
- 3) Create an API key that can use that API

Maps APIs and Servi... – Google Maps – My Project – Google Cloud Platform - Mozilla Firefox

Get an API Key | Maps JavaS | Maps APIs and Servi... – Goog | Google Cloud Platform

https://console.cloud.google.com/google/maps-apis/new?project=my-project-1494086980873&folder=&organizationId=

Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more

Free Trail

ACTIVATE

Google Cloud Platform | My Project | Search products and resources

Google Maps

Overview
APIs
Quotas
Credentials
Metrics
Support

Maps APIs and Services

Browse the Marketplace to find and use Maps APIs and services.

Geolocation API
Google
Location data from cell
WiFi nodes.

Click That

for Android
ur native Android app.

Roads API

Places API

Maps JavaScript API
website

Maps Embed API
Google
Make places easily discoverable with interactive Google Maps.

Time Zone API
Google
Time zon
world.

ion API
for any point in the

Then ENABLE the API

Google Cloud Platform | My Project

API Library

Geolocation API
Google
Location data from cell towers and WiFi nodes.

ENABLE

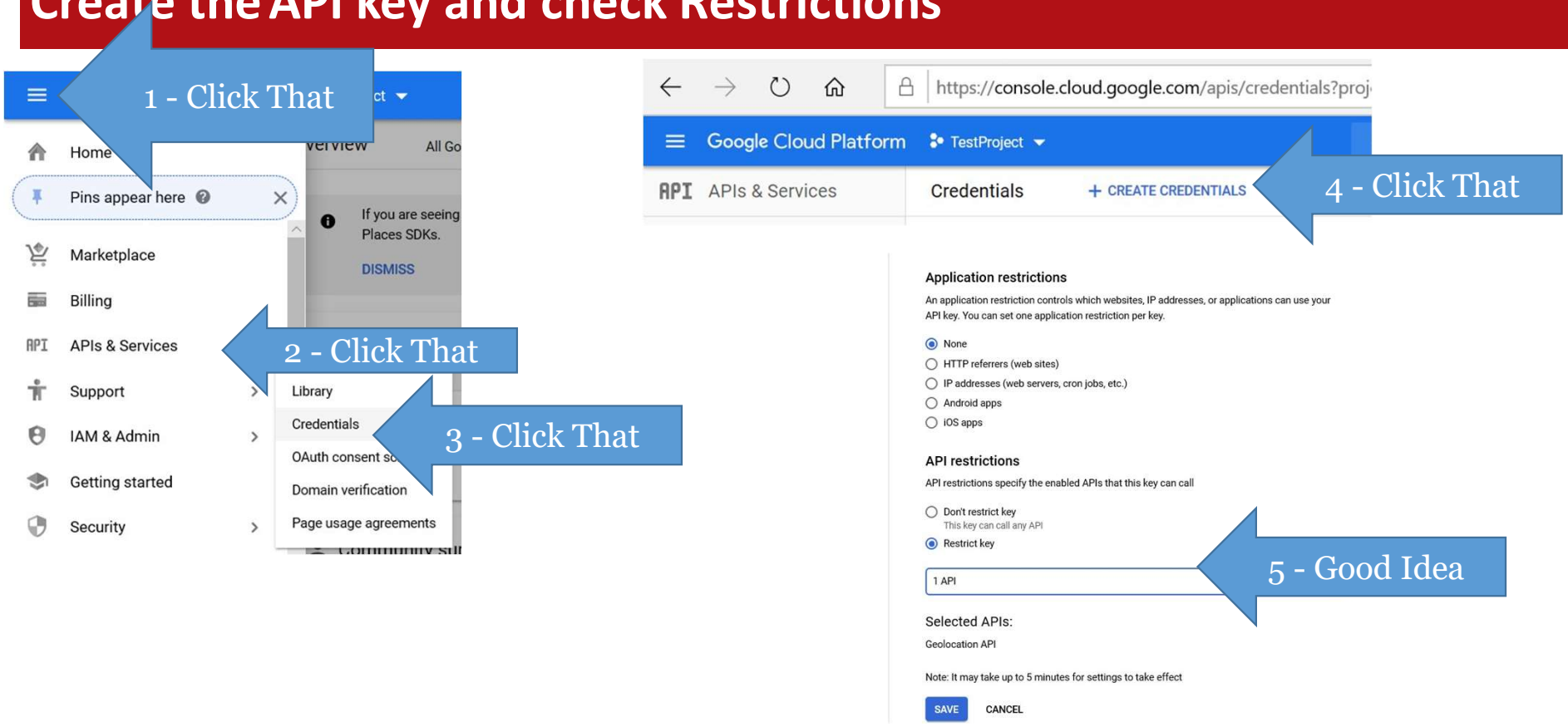
Directions API
Google
Directions between multiple locations.

Maps Static API
Google
Simple, embeddable map image with minimal code.

Geocoding API
Google
Convert between addresses and geographic coordinates.

Street View Static API
Google
Real-world imagery and panoramas.

Create the API key and check Restrictions



You have an API key!

Google Cloud Platform | TestProject | Search products and resources

Google Maps

- Overview
- APIs
- Quotas
- Credentials
- Metrics
- Support

Credentials | Geolocation API

Credentials compatible with this API

To view all credentials or create new credentials visit Credentials in APIs & Services

Remember to configure the OAuth consent screen with information about your application. CONFIGURE CONSENT SCREEN

API Keys

Name	Creation date	Restrictions	Key	Usage with this service (last 30 days)	Usage with all services (last 30 days)
API key 1	Feb 20, 2020	Geolocation API	[redacted]	47	47

WALKTHROUGH WORKBOOK 3 and 4 TOGETHER

A Tool that does all this for you?

- Werejugo does all of this for you.
  - BSSID in registry, event logs and other used to determine locations
    - Single BSSIDs via Wigle and BSSID pairs via google
  - Names of those wireless network in other artifacts used for date and ties
- It is Free!
- Download it here: https://github.com/MarkBaggett/werejugo
- You provide your own API keys
- "results.html" in your workbook is sample output

NOW IT IS YOUR TURN

- Can you solve the two forensics challenges?
- Use code from Workbooks 1 and 2 to solve Challenge #1
- Use code from Workbooks 3 and 4 to solve Challenge #2
- You can copy and paste code from the workbooks into your solutions book.
- You can continue to use my personal API key as you work on Forensics Challenge 2 but please be considerate.

LAB TIME!

- If you have questions or need help you can:
  - Post a question directly into SLACK
  - Reach out to one of our instructor via a Direct Message
  - Ask any of the following people for help!

Mark Baggett
Mike Pilkington
Mark Hallman
Don Williams
Bryan Scarbrough
Charlie Goldner