

CODING BOOTCAMP

FOR CYBER PROFESSIONALS

 thecyberinstructor.com



About Karan (Instructor)

- Hi I'm Karan Dwivedi!
- Sec Eng @ Google, Yahoo, 10+ yrs experience
- Founder @ The Cyber Instructor
- Wrote a book to get started in cybersecurity - kickstartseceng.com
- Received EB1A Extraordinary ability green card in US



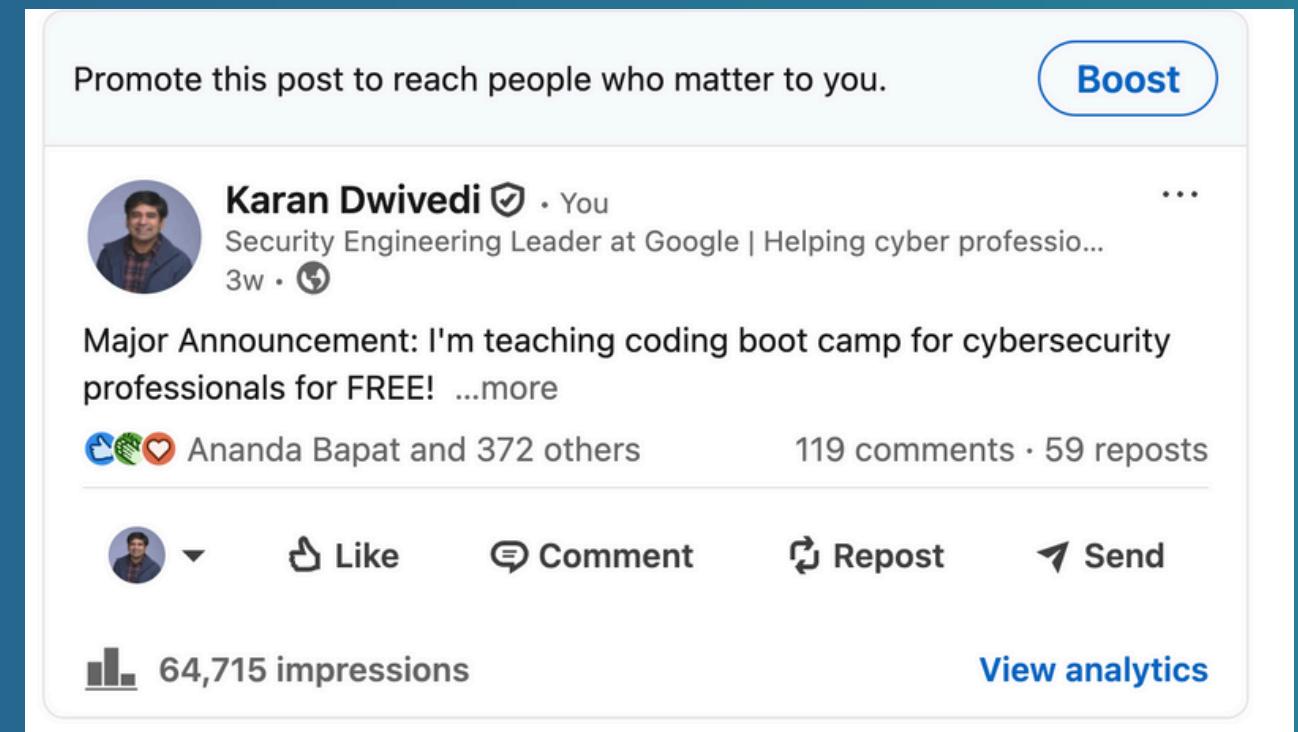
About Apurv (Course Staff)

- Hi I'm Apurv Singh Gautam !
- CTI Researcher @ Cyble, 6+ yrs experience
- Member @ Curated Intel
- Loves to contribute and help others
- 8k+ followers on LinkedIn



How to get the most out of this course!

- Follow lectures
- Do the exercises and capstone
- Ask questions!
- Network!
- Post your progress on LinkedIn/Social Media
- Tag Karan and Apurv for broader reach



Course Outline

- **Class 1: Python refresher (Exercise 1: Whois lookup tool)**
- Class 2: Common Algorithms (Exercise 2: Secrets Manager)
- Class 3: 1D data structures (Exercise 3: Mini Firewall)
- Class 4: Trees and Graphs (Exercise 4: Mini File System)
- Class 5: Capstone Demos and wrap up



The Cyber Instructor

- Mission: Provide highest quality cybersecurity education to the planet!
- Vision: Anyone can grow in cybersecurity using our offerings
- Impact: At scale!
- Principles:
 - Lead with Value
 - Quality >> Quantity
 - Strive for excellence
- <https://www.linkedin.com/company/103318086/>



Course Schedule

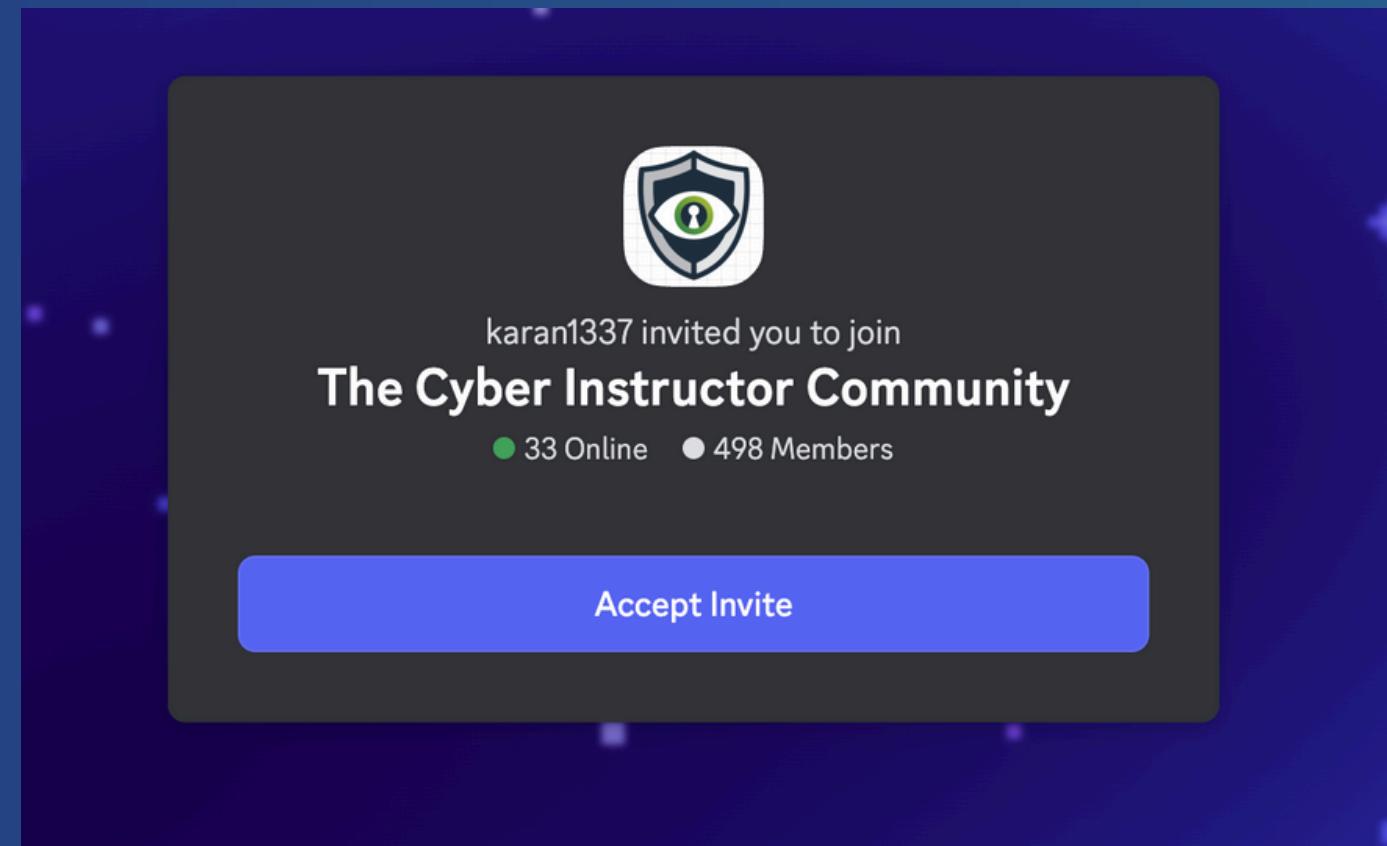
- Live lectures, exercises and capstone project
- We will remind you of each deadline
- #annoucements, email etc.

CODING BOOTCAMP FOR CYBERSECURITY PROFESSIONALS

Sept 12th	Class 1 (Basics of python)
Sept 18th	Exercise 1 Deadline (Whois lookup)
Sept 19th	Class 2 (Common Algorithms)
Sept 21st	Decide your capstone project (recommended deadline)
Sept 25th	Exercise 2 Deadline
Sept 26th	Class 3 (1D Data Structures)
Sept 27th	Certificate Request Deadline
Oct 2nd	Exercise 3 Deadline (Mini Firewall)
Oct 3rd	Capstone Project Deadline
Oct 10th	Class 5 (Capstone Demos and Course wrap up)



Course logistics – Discord



Join using:
<https://link.thecyberinstructor.com/cb-discord-invite>

verification

Text Channels ▾ +

welcome

Voice Channels ▾ +

General

Coding Bootcamp ▾ +

start-here

announcements

general-discussions

exercises

capstone-project

ask-course-staff

socials

testimonials

Welcome to # start-here!

This is the start of the # start-here channel.

Karan | The Cyber Instructor 8/20/2025 8:44 PM Start Here – Welcome to "The Cyber Instructor" Community Server.

Hey there, and welcome to the community! 🌟

This server is built for cyber professionals who are ready to grow their skillset. I'm Karan Dwivedi (a.k.a. Founder of The Cyber Instructor) and your mentor. My mission is to up level cybersecurity professionals like you so you can secure the lives of human beings at scale!

Before jumping in, please take a moment to review how this server works so you get the most value and stay aligned with the group's direction.

Community Guidelines

1. Stay Focused & Respectful

No harassment, hate speech, or discrimination of any kind. Keep it friendly and inclusive. Keep discussions educational and constructive. No hype or fear-based posting.

2. Use the Right Channels

Each channel has a purpose—let's keep things organized for everyone's benefit. Please see the message below for the purpose of each channel.

3. No spam or self-promotion

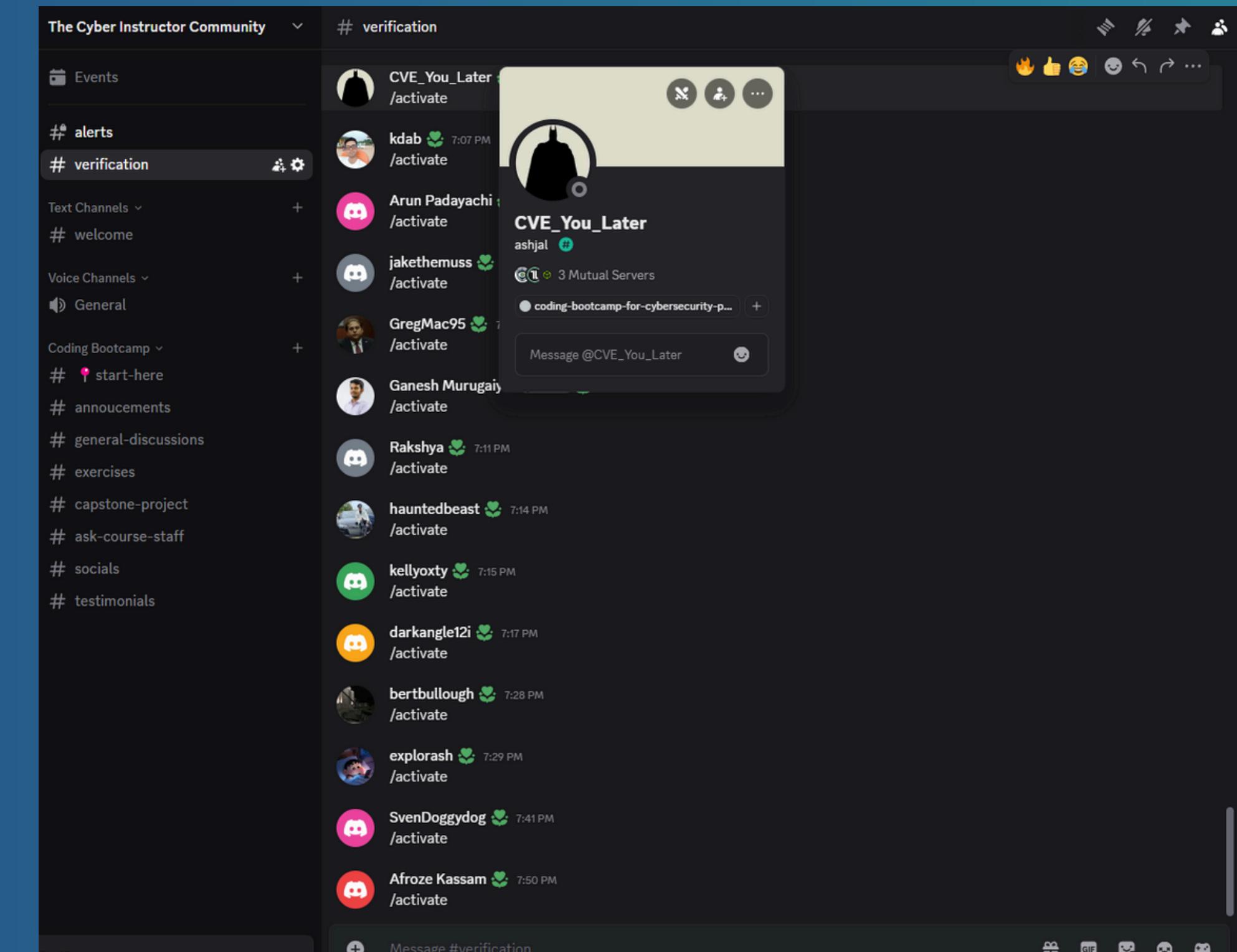
Don't flood channels, drop random links, or promote your social/content without permission. DM ads are not allowed.



Course logistics – Discord

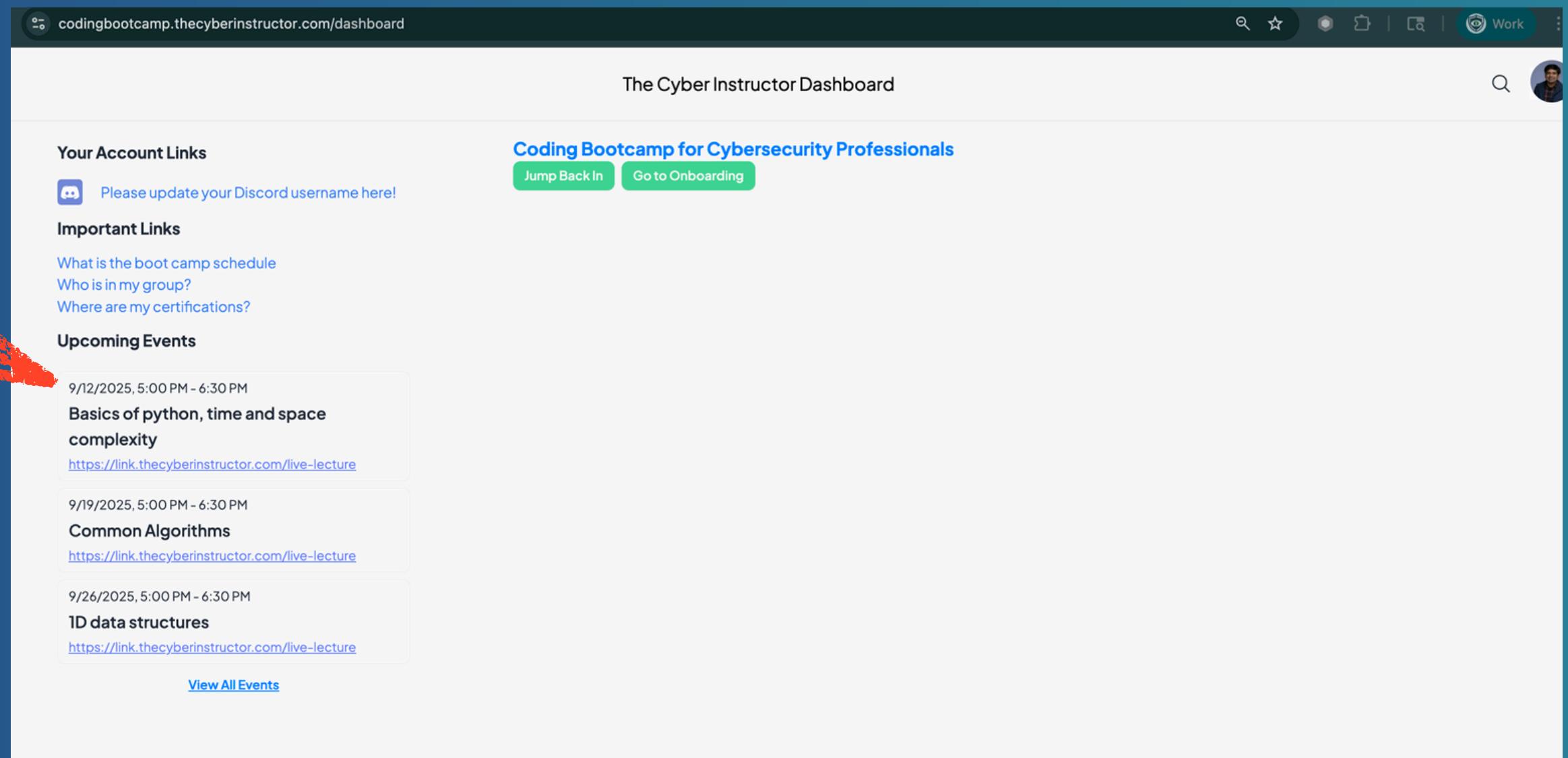
Use /activate in the #verification channel to be added to the right role for the course!

Role: @coding-bootcamp-for-cybersecurity-professionals



Course logistics – Lectures

- Lecture links will be available in the events page on your dashboard.
- The most recent link is always
<https://link.thecyberinstructor.com/live-lecture>
- We will also share in #annoucements



The screenshot shows the 'The Cyber Instructor Dashboard' with the URL 'codingbootcamp.thecyberinstructor.com/dashboard'. The dashboard features sections for 'Your Account Links' (with a note to update Discord username), 'Important Links' (including boot camp schedule, group info, and certifications), and 'Upcoming Events'. A red arrow points from the text 'The most recent link is always' to the first event listed under 'Upcoming Events': '9/12/2025, 5:00 PM - 6:30 PM Basics of python, time and space complexity <https://link.thecyberinstructor.com/live-lecture>'. Other events listed are '9/19/2025, 5:00 PM - 6:30 PM Common Algorithms' and '9/26/2025, 5:00 PM - 6:30 PM 1D data structures'. There are also 'Jump Back In' and 'Go to Onboarding' buttons.



Course logistics – Exercises

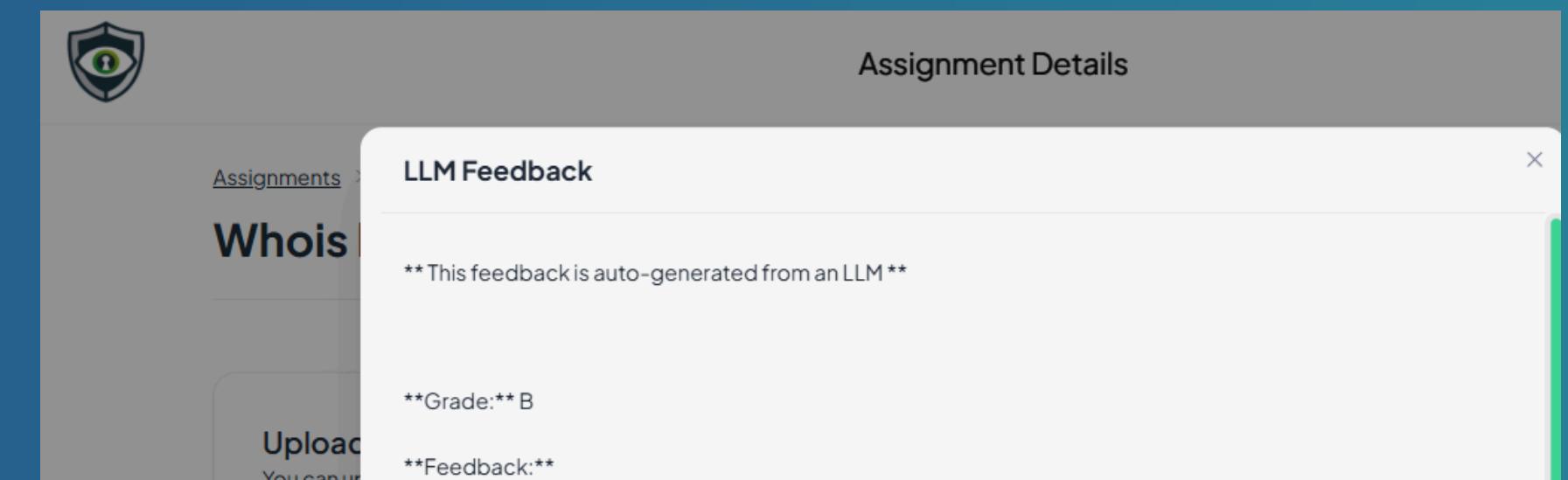
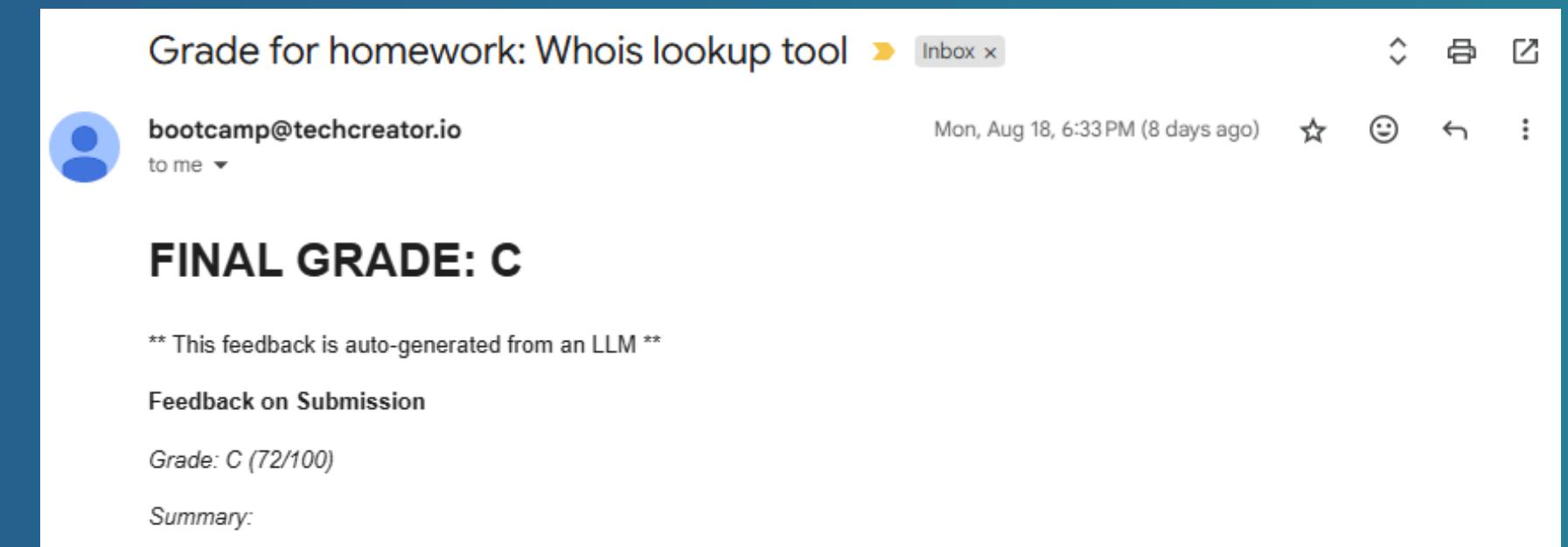
- Exercises can be completed locally in python.
- To submit, please use upload assignment to trigger autograder
- We will maintain a scoreboard for exercises for fun and reward!
- Link for assignments
- Honor system (please avoid AI)

The screenshot shows a web page titled "Whois lookup tool" under the "Assignment Details" section. At the top right is a "Download Template" button. Below the title, there's a "Upload Submission" area with a placeholder text "You can upload your submission here". A file input field shows "No file chosen" and a "CHOOSE FILE" button. A note below says "allowed file types: .zip". On the far right, there's a "Submit Assignment" button. The URL in the address bar is "Assignments > Whois lookup tool".



Course logistics – Exercises

- After submission you will get an email from autograder
- Ping in #exercises channel for questions



Course logistics – Capstone

- Project of your choice!
- Decide capstone idea early, due Oct 3rd.
- Use **#capstone-project** channel for any questions
- Capstone is due Friday night (Oct 3rd)

CODING BOOTCAMP FOR CYBERSECURITY PROFESSIONALS

Sept 12th	Class 1 (Basics of python)
Sept 18th	Exercise 1 Deadline (Whois lookup)
Sept 19th	Class 2 (Common Algorithms)
Sept 21st	Decide your capstone project (recommended deadline)
Sept 25th	Exercise 2 Deadline
Sept 26th	Class 3 (1D Data Structures)
Sept 27th	Certificate Request Deadline
Oct 2nd	Exercise 3 Deadline (Mini Firewall)
Oct 3rd	Capstone Project Deadline
Oct 10th	Class 5 (Capstone Demos and Course wrap up)



Course logistics – Certification

- You can opt to get a certificate at a nominal price.
- Prove your coding skills to anyone!
- [https://link.thecyberinstructor.com/
request-bootcamp-cert](https://link.thecyberinstructor.com/request-bootcamp-cert)



Course logistics – Certification

- Certificate opt in deadline is 27th Sept (right after 3rd class)
- Criteria:
 - PASS all assignments
 - WORKING capstone project
- <https://link.thecyberinstructor.com/request-bootcamp-cert>



Course logistics – How to get a free certificate

Awarded to:

- Top 3 students – Fastest, earliest and highest grade exercise submissions
- Top 2 capstone projects (as judged by course staff)



Python Intro

- Strings
- Lists, Tuples and Dictionaries
- Conditionals
- Loops
- Exception Handling
- Files
- Imports
- Functions
- Classes

Ref: <https://python101.pythonlibrary.org/intro.html>



Installing Python

- Install python 3.0
- <https://www.python.org/downloads/>
- Get the latest version for your OS
- Great for running and testing code locally



```
Karans-MacBook-Pro:~ Karan$ python3 --version
Python 3.9.6
Karans-MacBook-Pro:~ Karan$ █
```



Python Intro – Strings

Strings are immutable. Strings in python 3 are unicode by default

```
my_string = "this is a string"  
print(my_string)
```

```
# String concat  
second_string = "this is another string"
```

```
#String slicing  
print(my_string[0:5])
```

```
#String substitution  
my_string = "I like %s and %s" % ("Security", "Coding")
```

```
#Substitution using templates (dictionary)  
print("%(lang)s is fun" % {"lang":"Python"})
```

```
#Substitution using format function  
print("Python is a {0}, {1}".format("simple", "language"))
```



Python Intro – List, Tuples, Dicts

Lists can be created in two ways. List are mutable

```
my_list = []
my_list2 = list()
```

```
alpha_list = [34, 23, 67, 100, 88, 2]
print(alpha_list)
```

```
# Returns None as the alpha_list will be sorted in place instead of being assigned to sorted_list. The sort function does not return any value
```

```
sorted_list = alpha_list.sort()
print(sorted_list)
print(alpha_list)
```



Python Intro – List, Tuples, Dicts

```
# Tuples - Immutable list  
my_tuple = (1, 2, 3, 4, 5)  
print(my_tuple[0:3])
```

```
abc_list = list(my_tuple)  
print(abc_list)
```

```
# Dict - hash table with key:value pairs  
my_dict = {}  
another_dict = dict()  
other_dict = {"one":1, "two":2, "three":3}  
print(other_dict)
```

```
# Print keys in dict  
print(other_dict.keys())
```

```
# Check if a key exists in the dict  
print("two" in other_dict)
```

```
# Print value of a key  
print(other_dict["two"])
```



Python Intro - Empty Checks on Lists, Tuple and Strings

```
#Empty and None check on strings
```

```
empty_list = []
```

```
empty_tuple = ()
```

```
empty_string = ""
```

```
nothing = None
```

```
if empty_list == []:
```

```
    print("It's an empty list!")
```

```
if empty_tuple:
```

```
    print("It's not an empty tuple!")
```

```
if not empty_string:
```

```
    print("This is an empty string!")
```

```
if not nothing:
```

```
    print("Then it's nothing!")
```



Python Intro – Conditionals

Python cares about spaces, don't mix tabs and spaces, especially in if conditions

```
var1 = 1  
var2 = 2  
if var1 < var2:  
    print("var 1 is less than var 2")
```

```
val3 = int(input("input var3:"))  
if val3 < var2:  
    print("var 3 is less than var 2")
```

```
# Conditional operators are and, or, not  
if var1 == 1 and var2 == 2:  
    print("var1 is 1 and var2 is 2")
```

```
if var1 > 2:  
    print("var1 > 2")  
elif var1 < 2:  
    print("var1 < 2")  
else:  
    print("var1 == var2")
```



Python Intro – Loops

for and while loops

```
# range function creates a list that is n in length  
print(range(0,5))
```

```
for number in range(5):  
    print(number)
```

```
if __name__ == "__main__":  
    print("within main")  
    # write main code here
```

```
for key in other_dict.keys():  
    print(key)
```

```
while val3 > 0:  
    print(key)  
    val3 = val3 - 1
```



Python Intro - Exception Handling

```
try:  
    1/0  
except ZeroDivisionError:  
    print("cannot divide by zero")  
  
my_dict = {"a":1, "b":2, "c":3}  
try:  
    value = my_dict["a"]  
except IndexError:  
    print("This index does not exist!")  
except KeyError:  
    print("This key is not in the dictionary!")  
except:  
    print("Some other error occurred!")  
else:  
    print("No error occurred")  
finally:  
    print("The finally statement has executed")
```



Python Intro - Files

```
#File Handling  
# Default mode for file open is read only  
handle = open("test.txt", "r")  
# read() reads the entire file  
data = handle.read()  
print(data)  
handle.close()
```

readline() will read line by line and readlines() will read all lines.

Use with keyword to open files. Creates a context manager to automatically close the file when processing is done.

```
try:  
    with open("test.txt") as file_handler:  
        for line in file_handler:  
            print(line)  
    except IOError:  
        print("An IOError occurred")
```



Python Intro – Imports

Python comes with lots of pre-made code baked in.

These pieces of code are known as modules and packages.

A module is a single importable Python file whereas a package is made up of two or more modules.

A package can be imported the same way a module is.

```
>>> import math  
>>> math.sqrt(4)  
2.0
```

```
>>> from math import sqrt  
>>> sqrt(16)  
4.0
```



Python Intro - Functions

```
def func():
    print("Inside func")
```

```
def add(a,b):
    return a+b
```

```
print(add(1,2))
```

```
# For any number of args or keywords, use *args and **kwargs
```

```
def many(*args, **kwargs):
    # args turns into a tuple
    print(args)
    # kwargs turns into a dict
    print(kwargs)
```

```
many(1, 2, 3, name="Mike", job="programmer")
```



Python Intro – Classes

```
# Class definition
class Vehicle(object):
    def __init__(self, color, doors):
        self.color = color
        self.doors = doors

    def brake(self):
        return "Braking"

    def drive(self):
        return "Driving"
```



Python Intro - (Sub) Classes

```
# Subclass definition from Vehicle
```

```
class Vehiclesub(Vehicle):
```

```
    def brake(self):  
        return "this message is from a subclass"
```

```
if __name__ == "__main__":  
    print("within main")  
    # write main code here  
    car = Vehiclesub("blue color", "black door")  
    print(car.brake())
```



Exercise: Whois Look up

Design a script that calls the WHOIS server to look up the Registrar of a list of domains.

- Input: List of domains separated by a space in the command line
- Output: List of Registrar names corresponding to the domains

Extra Credit (optional)

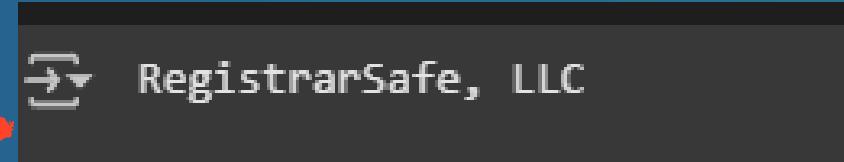
1. Take input from a file instead of the command line (e.g. input.txt)
2. Parallel lookups where more than one domain can be looked up at a time (read about threading in python!)
3. Rate-limiting the requests so you don't get blocked by WHOIS servers



Exercise: Whois Look up

Good Submission

- Should output only the Registrar and nothing else



```
{  
  "domain_name": "FACEBOOK.COM",  
  "registrar": "RegistrarSafe, LLC",  
  "registrar_url": [  
    "http://www.registrarsafe.com",  
    "https://www.registrarsafe.com"  
  ],  
  "reseller": null,  
  "whois_server": "whois.registrarsafe.com",  
  "referral_url": null,  
  "updated_date": "2025-04-23 19:08:37",  
  "creation_date": "1997-03-29 05:00:00",  
  "expiration_date": "2034-03-30 04:00:00",  
  "name_servers": [  
    "A.NS.FACEBOOK.COM",  
    "B.NS.FACEBOOK.COM",  
    "C.NS.FACEBOOK.COM",  
    "D.NS.FACEBOOK.COM"  
  ],  
  "status": [  
    "clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited",  
    "clientTransferProhibited https://icann.org/epp#clientTransferProhibited",  
    "clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited",  
    "serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited",  
    "serverTransferProhibited https://icann.org/epp#serverTransferProhibited",  
    "serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited",  
    "clientDeleteProhibited https://www.icann.org/epp#clientDeleteProhibited",  
    "clientTransferProhibited https://www.icann.org/epp#clientTransferProhibited",  
    "clientUpdateProhibited https://www.icann.org/epp#clientUpdateProhibited",  
    "serverDeleteProhibited https://www.icann.org/epp#serverDeleteProhibited",  
    "serverTransferProhibited https://www.icann.org/epp#serverTransferProhibited",  
    "serverUpdateProhibited https://www.icann.org/epp#serverUpdateProhibited"  
  ],  
  "emails": [  
    "abusecomplaints@registrarsafe.com",  
    "domain@fb.com"  
  ],  
  "dnssec": "unsigned",  
  "name": "Domain Admin",  
  "org": "Meta Platforms, Inc.",  
  "address": "1601 Willow Rd",  
  "city": "Menlo Park",  
}
```



Submit to Autograder

- Zip your script. You can name it anything
- Upload your zip under Whois lookup tool Assignment
- You will receive your grade and feedback within a few minutes

<https://codingbootcamp.thecyberinstructor.com/dashboard>

The screenshot shows the top navigation bar of the dashboard. It features a logo, the text "The Cyber Instructor Dashboard", and a section titled "Coding Bootcamp for Cybersecurity Professionals" with "Jump Back In" and "Go to Onboarding" buttons. A red circle highlights the "Go to Onboarding" button.

A red arrow points from the "Go to Onboarding" button on the dashboard to the "Contents" page. The "Contents" page lists modules: "Basics of python, time and space complexity" (1 Assignment), "Python Data Structures" (1 Assignment), "Assignments" (1 Assignment), and "Common Algorithms (Sorting, Searching, Hashing)" (1 Assignment). The "Whois lookup tool" assignment is circled in red.

A red arrow points from the "Whois lookup tool" assignment on the contents page to the "Assignment Details" page. This page shows an "Upload Submission" section with a "CHOOSE FILE" button and a placeholder "No file chosen". A red circle highlights the "Submit Assignment" button at the bottom right.



Things to remember

- Avoid using library functions that solve the problem completely (e.g., implement sorting manually instead of using `sort()`) and AI code generation. For the capstone project, you are free to use any library.
- Avoid DOS on whois servers
- Autograder is configured to check basic submission, but feel free to submit scripts with extra credit additions (you will get a higher grade for extra credits)
- Exercise is due Thursday night 11:59 PM PST (Sept 18th) (a day prior to next class)



Questions?

