A bit of research before class - top 11 common user authentication vulnerabilities to know about:

- Flawed brute force attack (e.g. dictionary attacks) - entering large numbers of usernames and passwords until they find one that works
    - If there is flawed brute force protection system, e.g. flaw in the authentication logic, firewall or SSH protocol attackers can hijack login credentials and processes
    - This compromises the security of user credentials
- Weak login credentials
    - If the user creates a predictable password - is easy to guess in a brute force attack
    - With no restrictions on weak passwords even sites protected against brute force attacks can be compromised
- Username enumeration - makes an attacker's life easier by lowering the cost of other attacks
    - Process of this is where the attacker confirms whether a username is valid or not - so if you have admin entered in and it says that you don't have the right password for this account - this validates that this is in fact a user
    - Problem with this is that attackers can tell what usernames are valid - can then try to hack into them using brute force techniques without wasting time on invalid usernames
- HTTP basic authentication
    - This is easy to implement but it has its risks - sends a username and password in the request
        - Note that if you had TLS encryption this wouldn't be as big of an issue

Some general things

What do you do when you have tried everything and don't know where to go next

- The realistic answer is to take a breath, take a break and then come back to it when you have the capacity to come back to it
- If you are stressed out - your mental health is more important
- Next thing there - google dork and google is your friend
    - Try to format the question as a google search to try and get more context to get what you are looking for
- When you have taken that break the next step is to ask someone nearby - ask on discord or slack
- Ask your peers first, then your tutor then ask Kristian

Robox.txt if you're curious
https://developers.google.com/search/docs/crawling-indexing/robots/intro

There are many DNS resolvers that are on the internet - sometimes they don't get updated

- Sometimes people put these up there to be malicious
- So if you have a bunch of DNS resolvers

- The more IPs that you have as your DNS resolvers then there is more possibilities that something is exploited or something can go wrong
- Why? Basically a scalability and numbers game

Scripting is a very useful tool - useful for us to

- If you're curious - missing semester course on MIT - this is up online is a self learning course
    - Has a lot of stuff from COMP2041 if you can't be bothered taking that course

Review

- We talked about authentication a lot last week
- We talked about cookies and how they interact with sessions
- We talked about passwords as well

TLS

Is a method of exchanging keys from a distance

1. Sends a cipher suite - the languages that we can talk in between the client and the ssl server
2. Client verifies the SSL certificate information and a pre-master key generated using the public key - if you don't do this this is how man in the middle attacks happen
3. The server then verifies the client certificate (the MTLS section)
4. Now we both have the same key - now we can generate a symmetric key using the really long private keys that both the client and the server has now generated

Building flask applications

- We're going to build a flask application together in this course right now to help those that haven't work with flask before

First thing to do is import flask

```python
# imports
from flask import *

# Create the application variable
app = Flask(__name__) # flask is the catch all for the web server

# endpoint that we want to interact with when we send things to the server
@app.route('/error')
def error():
  return "<p><strong>Enter correct password</strong></p>'

@app.route('/')
...
...
...

@app.route('/success'), methods = ['POST'])
def check_login():
  if request.method == "POST":
    email = requestform.
....
...
# Once you make the requests you just run in python and do things with the commands

# This is why we use flask, its basic, its slow but we can use interact with for the sake of this course
```

```python
if __name__ == "__main"
  app.run(Debug=True) # if this is the file run on command line run this
# runs as a synchronous so it waits for a response before it runs the next one
```

- If you want to know more details read the docs for more details
- Register_blueprint(my_app) - if you are building an application in and of itself this is a term that
- If you are given a task to create an application you can write it in whatever you want but flask is just nice to debug and understand

Be aware that people are doing things maliciously - sometimes people slightly misspell popular python package names and if you misspell it then you could accidentally be installing malicious code

Set_cookie in flask requires a key and a value - so if we send jwt we need to also send it a code

HS256 - is a hashing algorithm

Unless your cookie expires your site will just accept the cookie if you just stay on that page

So if you were to edit a cookie - if you didn't input the secret key that was used when the code initially generated a jwt then it won't work

- You would need to know the cookie, and you would need to ensure that you have the secret and add this to the jwt you have edited to make sure that the thing you are trying to access verifies your modified cookie

Note prod is a production environment - people can use it in production

- Other people can access it and is kind of bad if things go wrong
- You could store password in prod, but you have actual users so you would have to form a data breach notification because you have test cases that have had their passwords stolen

If you were to generate a new key each time such as .secrets (secrets import) this would be a good way to take away the secret from the source code

Reports

Due the Sunday following the Lecture in Wk 5 (19th of March) at 11:55pm

- There will be 5 minutes of leeway for network issues
- Will include all challenges up to and including those due at the end of week 5
- Sections of report
    - Executive summary - want to see you have put thought into it
        - Makes it easier for the report easier to read
        - Put in page numbers
        - Do the bare minimum to make the marker's life easier
    - Vulnerability tables - including risk, description, walkthrough and steps to remediate
        - Discuss what the vulnerabilities look like
        - We want the risk calculated with a risk matrix
        - Walkthrough of how you were able to exploit it
        - We want to know the steps to remediate the vulnerability
            - E.g. these are how you would remediate the risk altogether
            - E.g. this is the code to remediate it
        - Risk table - is a graph that plots the extent of the vulnerability on one axis and the likelihood of it getting exploited on the other axis
            - One axis is likelihood one axis is the risk
            - Don't care how you develop the matrix - but just make sure it makes sense
            - Don't rate things as high risk and high possibility because if everything is high risk nothing is high risk

        - E.g. all the stuff you find in a reconnaissance stage is still a vulnerability - e.g. would group this as information disclosure

Things to possibly add

- Table of contents might be a good idea
- Vulnerability table - we want to see both the consequences and the likelihood
- Give information as to where you can find the vulnerability - flag what the issue with each of the
    - The more specific you can be about a vulnerability the better we can remediate
    - **Get the content across because you are marked on substance**
- Add references if you found some references - you can include links - tell the client why it's important and then use the reference to back up the statements that you made
- In the impact - impact to a business sense rather than just a technical sense because this is a report
    - If you have personal information leakage by IDOR - IDOR is not a particularly bad attack but can lead to
    - There is a difference between business risk and technical risk
- Anything else that you have found during the term include this
- Vulnerability details - give some information to fill in what the vulnerability even is as if they don't know what it is

You get to define your own likelihood and consequences

- Vulnerability matrixes are 5x5 (Kristian likes this one)
    - Low, moderate, high, extreme, catastrophic
- You get to define it so what we are looking for is justification on why you defined it this way
    - Is a low actually a low
    - We also need consistency - we need to see that the risk matrix has been applied consistently in your report
- Remediation - tell the user how to fix it
- Expect more detail in the base course and expect more challenges to be completed for the advanced course
- If you can't complete a challenge and you can't get the flag still include it in your report - if you do something and you get a server and it could escalate to a denial of service attack
    - You should still write down everything regardless of whether you have gotten the flag or not
- Your risk matrix should reflect the level of access that you are able to gain when you do your challenges
- Report is done in groups - one person submits

PII - personally identifiable information - things that can be used to identify a unique individual

For the subdomain one - you are just opening up your attack
DNS subdomain - likelihood - is always will always be accessed

- Everyone is always doing it so this one is high likelihood

Given the nature of certain domains found - we know that there are other domains out there that should be protected by firewalls (or something to this effect)

- If you find an admin subdomain - then you classify this higher than just an www.
- For everything outside of just this course the consequence here
- Recon is the exception to all the rules - all the recon flags together, give it high likelihood and medium consequences/low

Midterm exam

- Final exam will be in 3 sections: part a, b, c all weighted equally
- Final mark will be: sum(max(final part a, midterm), final part b, final part c)
- You can sit this to understand how you are relative to the course and the content
    - This is entirely optional
    - This is an online one hour exam
- If you do the midterm and score 100% you can skip entirely the part a of the exam because you will get 100% because it is the max
    - NOTE DO THE REST OF THE EXAM THOUGH
- Midterm will be 1 hour long - will be like the ctfd challenges
    - Each flag will be one mark
    - The write ups are 4 marks - you submit the writeups on moodle
    - So for each flag there are 5 marks each
- Will be 6-7 pm
    - Will be open book
    - You can do it wherever you like
- Will include everything covered in weeks 1 to 4 - includes recon, session management, and sql injection
- What is a writeup - is similar to a vulnerability
    - Show what it is and how you fix it
    - Includes details of how you got the flag
    - Dot points
        - What you did , you freeshed page, etc - how you got the flag
    - Is steps on how to reproduce how you got the flag
    - If you did a good writeup you will get 4 marks
        - If your writeup is missing something you will get 3
    - If you don't get the flag and you got halfway there - do the writeup of what you've done so far
        - Is a method of demonstrating your knowledge without just having to submitting a flag

        - Can get marks even if you don't get the flag - just the instructions on getting halfway there could get you marks
    - Are brief summaries - similar to reports - want you to show you how to reproduce it and nothing else
        - You don't have to write how to fix it
- No brute forcing during the exam - will not have any requirement/permissible brute forcing during the exam
- There will be 5-6 flags to get during the exam
    - If it takes your more than 5 minutes then maybe move to the next one
- The course will be the same for the midterm - this is because this is a status check, is not supposed to be an advanced application of your techniques
- Ctfd will not be up during the exam
- No bruteforcing in the final exam either
- Writeups should be dot points - should take 8 seconds to write the dot points if it took 8 minutes to find the flag
    - Anything more than just the steps in dot points is just fluff
    - Dot Points - like seriously
- If you get 50% then you're going well in this course so far

Final exam will probably be a 4 hour exam with 3 hours of content

- Will be online
- Xml.rpc - wordpress always has this

Thinking

- If someone wants to transfer all the money in an account to yours after death - transferring all assets
    - Proof of death
    - Proof that the estate was in that person's name (ownership) - the person who died is the person who owns it
    - Proof of the person who has come in is who they say they are
    - Proof that the person in the bank is the executor of the will - the person that gets to choose what happens to the dead person's assets
        - The executor has the final say of what happens in the will
    - Proof that they are entitled to the inheritance
- Some of the things here is a mix of authentication and authorisation

Authentication - proving you are who you say you are
Authorisation - making sure you are allowed to do what you are trying to do

- Allowing the person you said you are, to access what they are allowed to access

Auth + Auth = A

- Authorisation - is who you say you are allowed to access that resource
- Authentication - are you who you say you are
- Authentication and authorisation form access - one of the CIA principles

Cookies here are a way that we can distinguish the things a user is authorized to access compared to others

Vertical access controls
- Where you are allowed or not allowed to access up and down the user profiles
    - E.g. regular user to admin
- E.g. student edit marks in a portal usually allowed for staff

Horizontal access controls
- Often forgotten
- Happens when a user can access the boundaries on the same level
    - Brings about IDOR - insecure direct object referencing
    - A student can read the reports and grades of other students

Privilege escalation - when you can elevate the access you have to certain resources or functions
This means both horizontally and vertical

Access control methods
- Role based - a user would have a role
- Attribute based - set an attribute for a user after verifying - assigning people attributes after you have verified them and their role
    - Are a more fine grained approach to role based approach
- Policy based
- Others but these are the main ones we look at - primarily role based access

Recon -> design attack ->get more info -> do more recon -> keep designing → is very cyclical

IDOR demo

Generally if you have a post id and you change it to 1 you are likely to get a test page
- Then if you do 2 usually is the admin page

Start thinking about how to do challenges under timed conditions

Develop a flask application and implement some role based applications