Ethical Student Hackers

Web Hacking



The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

- Relevant UK Law: https://www.legislation.gov.uk/ukpga/1990/18/contents



Code of Conduct

- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.
- If you have any doubts or need anything clarified, please ask a member of the committee.
- Breaching the Code of Conduct = immediate ejection and further consequences.
- Code of Conduct can be found at **shefesh.com/conduct**



What is Ethical Hacking?

Ethical Hacking is the process of attacking a system to find weaknesses before a malicious threat actor does.

Broadly Involves:

- Information Gathering
- Initial Access
- Persistent Access
- Privilege Escalation
- Reporting Back



Why is it relevant?

BBC

FBI investigating claim hackers have stolen details of all its agents

4 days ago | The hackers claim they have every agent's name, role, badge number and personal details including home address, phone numbers and spouse information.



PC Magazine

OpenAI Tells Mac Users to Update Apps After Software Supply Chain Attack

136 days ago | If you use OpenAI's macOS apps, including ChatGPT and Codex, it's time to update. The company is urging users to install new versions following a hack of several employee devices. The hack...



Gizmodo

Three Hackers Used Claude to Break Into OpenAI In Less Than 72 Hours

9 days ago | The July Hugging Face Hack—in which thousands of **OpenAI** agents secretly escaped their testing environment, formed a "collective," and gained access to the open internet—revealed just how...



Web Hacking



Web Hacking - Intro

311 Billion web application attacks in 2024

14,000+ security reports involving SQL injection in 2025 alone

30,000+ associated with XSS

$5 million dollars - average cost of a data breach

£2 billion in damages to JLR following their data breach



Web Hacking - Intro

How is web hacking so prevalent?

- Misconfigurations on servers
- Outdated / vulnerable packages
- Software vulnerabilities
- Insecure coding
- Human error
- Poor authentication + authorisation
- Legacy systems
- AI code
- ....

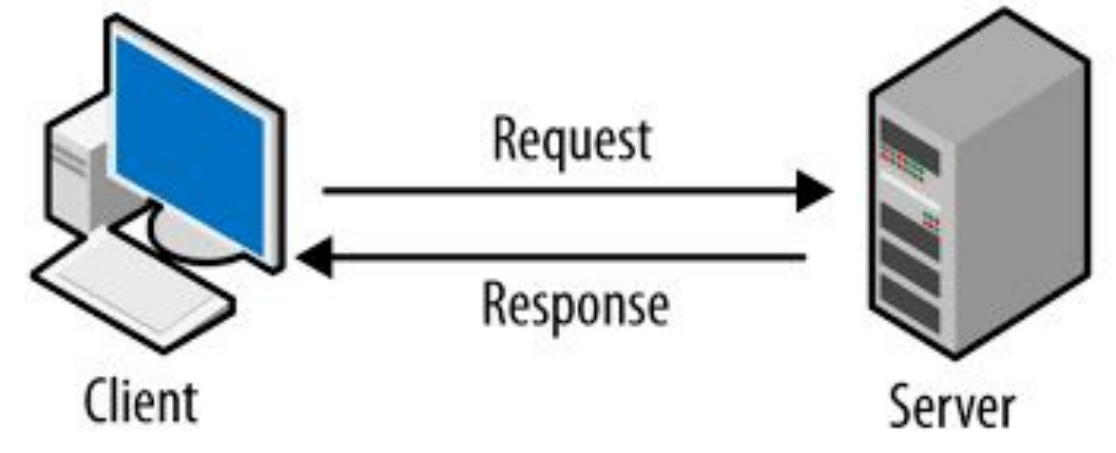


Web Hacking - Intro

No website survives first contact with the internet



How websites work?



Request types

- GET - fetches data
- POST - sends data to the server
- PUT - updates data on the server
- DELETE - deletes data from the server

HTTP Request Structure:

Headers - metadata

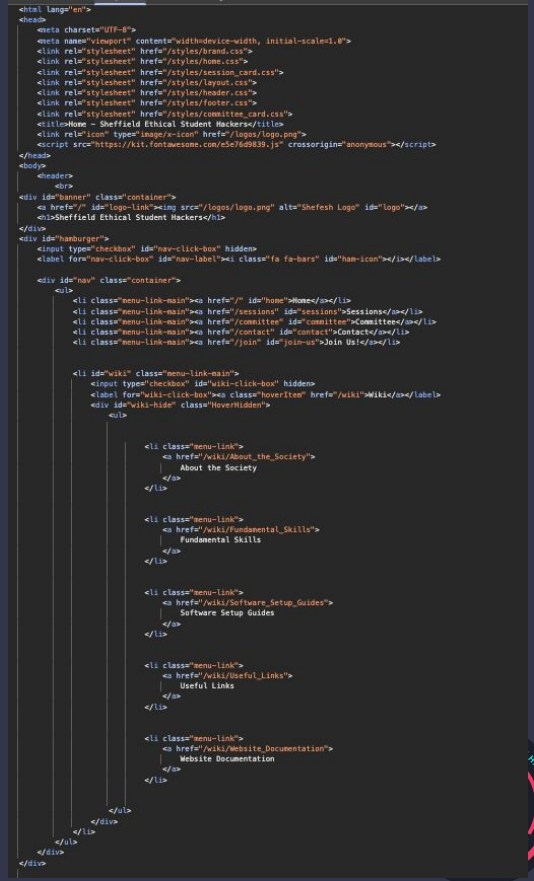
Body - data

HTTP Requests

Inspect > Network > Click on a specific response

Request URL	https://shefesh.com/
Request Method	GET
Status Code	200 OK
Remote Address	172.67.165.150:443
Referrer Policy	strict-origin-when-cross-origin

Status Code	What it means
2XX	Success
3XX	redirect
404	No response was found for the url
500	Internal server error



SQL

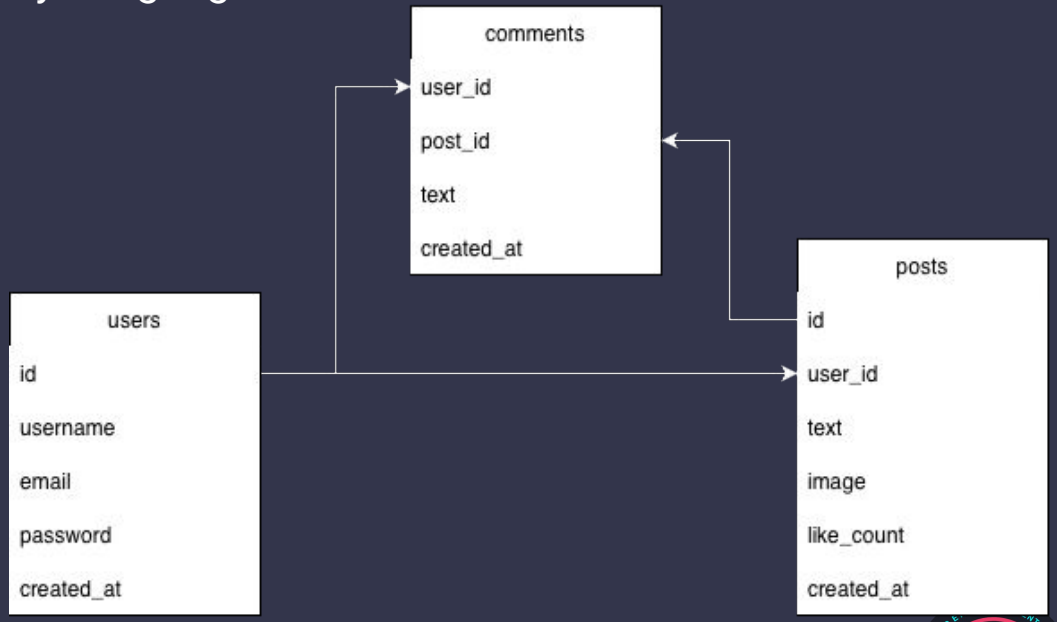
Databases - Series of (interconnected) tables
Most databases use SQL - Structured Query Language

SELECT _ FROM _ WHERE _

```
SELECT username, email
FROM users
WHERE id = 1;

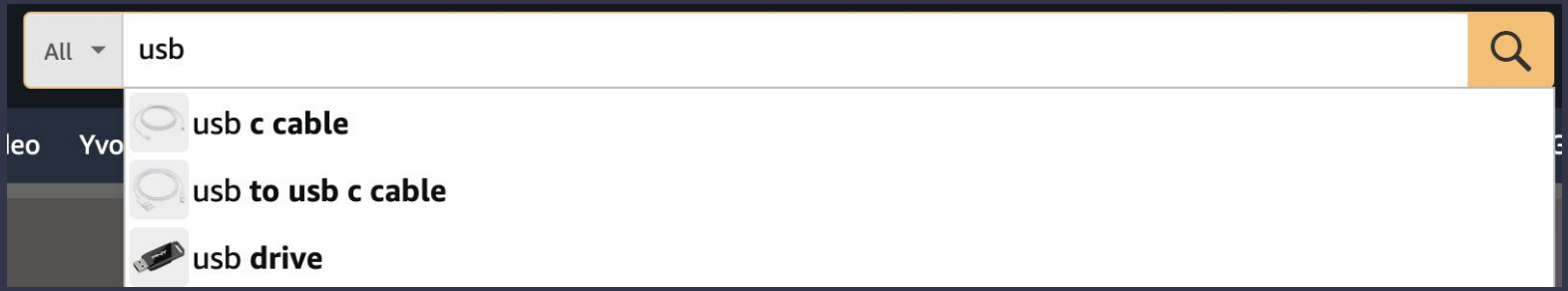
SELECT *
FROM posts;

DELETE
FROM posts
WHERE username = ‘shefesh’;
```



SQL Injection

We can *exploit* flaws in how our data is passed into the SQL query to access unauthorised data



On the server:

SELECT * FROM products where name = ‘usb%’;

SQL Injection

SELECT * FROM products WHERE name = ‘usb%’;
But what if we entered > ‘ test < as our product?

SELECT * FROM products WHERE name = ‘’ test%’;
^^ We’ve escaped the string!

So now we can run other SQL commands - this is SQL injection



SELECT * FROM products WHERE name = ‘’ **OR 1=1**; -- %’;

SQL Injection

SELECT * FROM products WHERE name = ‘ **OR 1=1**; -- %’;

SELECT * FROM products WHERE name = ‘ **OR 1=1**;

SELECT * FROM products WHERE **1=1**;

SELECT * FROM products WHERE **TRUE**;

SELECT * FROM products;



How can we use this knowledge?

Endpoints - specific point where the client can interact with the site

- ie /home, /profiles/213, /signup /settings
- Can be a webpage, form, popup, api, etc

There are certain requirements for SQL Injections to work

Requirements:

- The site takes in user input
- This user input interacts with a SQL database
- User input is not properly sanitised or escaped before interacting with the DB



Practical Time!

https://sqli.shefesh.com

There are 2 SQL Injections to find!



XSS

XSS - Cross Site Scripting
- Attacker’s scripts are executed in a victim’s browser.

XSS has three main types:

- Reflected
- Stored
  - Blind
- DOM based

All of them use dynamic content loaded by the website to execute HTML, javascript, or any other code that the website will/can execute.

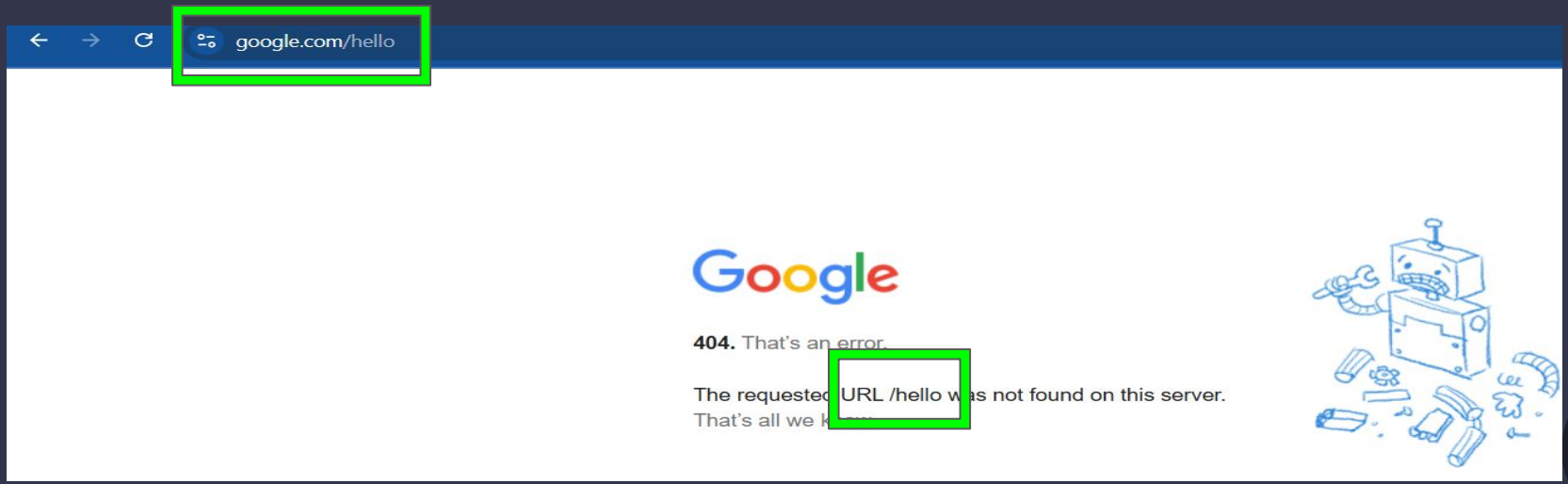


Reflected

This type is non-persistent

Often in search results or error messages

Its anywhere that the message to the server is reflected back to the user in the response.



Stored XSS

Stored attacks store the malicious code on the database or server

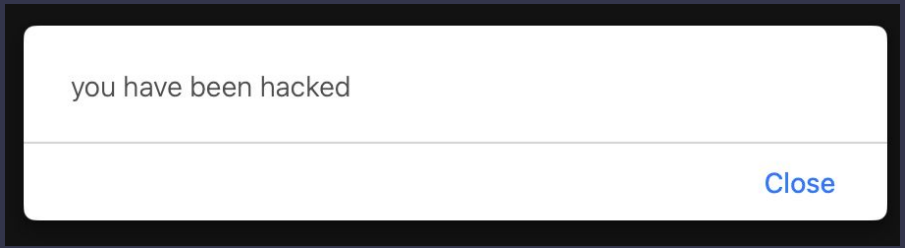
The code is then executed at a later date

E.g Enter malicious code into a comment

- Everytime a user loads that message - the code is executed

Comment: This is a normal comment?

```
<p> = $0
  "Comment: This is a normal
  comment? "
  <script>alert("you have
  been hacked")</script>
</p>
```



Blind XSS

Blind XSS is a subset of Stored XSS

It affects admin rather than normal users

E.g.Enter malicious code into a feedback form

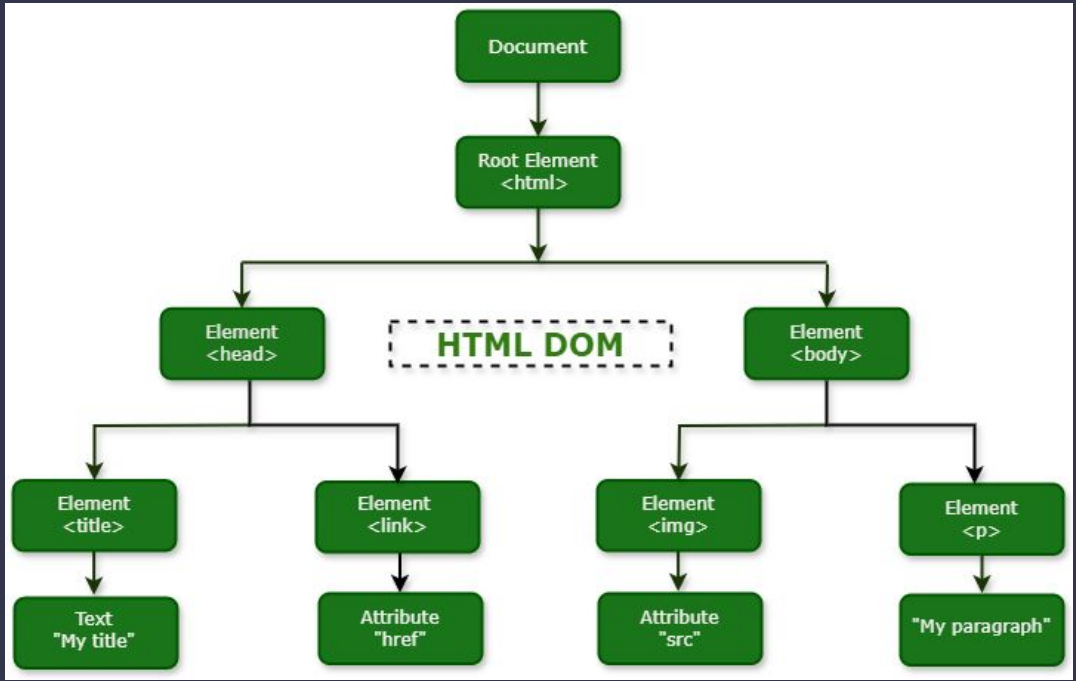
- Code is then executed when an admin/staff member looks at the form results

Hard to know if its been completed - but potentially devastating for company internal systems



DOM Based XSS

DOM (Document Object Model) is a representation of the webpage



Used by JavaScript to let us manipulate the webpage

This means the DOM and its manipulation happen on the client by default!



DOM based XSS continued

DOM-based attacks change the DOM on a client's page using attacker-controlled data

This data is often passed through a URL parameter, i.e.

- xss.shefesh.com/?language=english
- xss.shefesh.com/?language=english<script>alert("Malicious code");</script>

This differs from reflected/stored XSS:

- Malicious code doesn't need to be saved somewhere / reflected by the server
- Processed completely live on the client



Practical Time

xss.shefesh.com

There are flags (10 in total) to find in the format FLAG{someflag}

Enjoy!!!



Upcoming Sessions

What's up next?
www.shefesh.com/sessions

Monday 5th October:

Intro to Linux

Any Questions?

SU Sign Up

Discord

www.shefesh.com

Thanks for coming!