Ethical Student Hackers

Postgrad Session

SHEFFIELD ETHICAL STUDENT HACKERS

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?

Last week multiple European Airports, including Heathrow had their electronic check-in system shutdown. The software was hacked and service was disrupted massively. People queued for hours.

About a month and a half ago, The Co-op, Waitrose and other shops almost completely shutdown due to a ransomware attack. The same group then have gone onto attack Jaguar-Land-Rover.

Web Hacking - intro

Web Applications - fancy name for websites

https://www.google.com/search?q=google&rlz=1C1CHBF_en-GBGB913GB913&oq=googl&gs_lcrp=EgZjaHJvbWUqEAgAEAAYgwEY4wIYsQMYgAQyEAgAEAAYgwEY4wIYsQMYgAQyEwgBEC4YgwEYxwEYsQMY0QMYgAQyCggCEAAYsQMYgAQyDQgDEAAYgwEYsQMYgAQyBggEEEUYPDIGCAUQRRhBMgYIBhBFGDwyBggHEEUYPNIBBzc4OWowajeoAgCwAgA&sourceid=chrome&ie=UTF-8

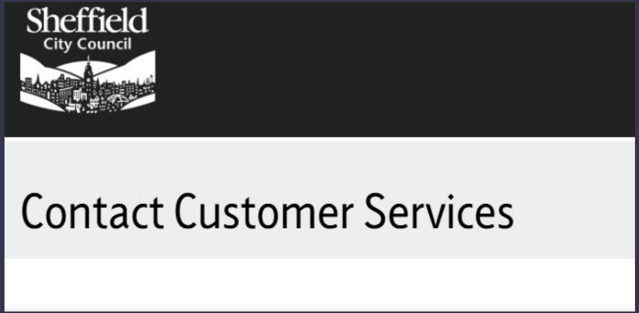
GET requests - fetching a web page and data

Parameters - bits of data passed inside the URL

Story Time!

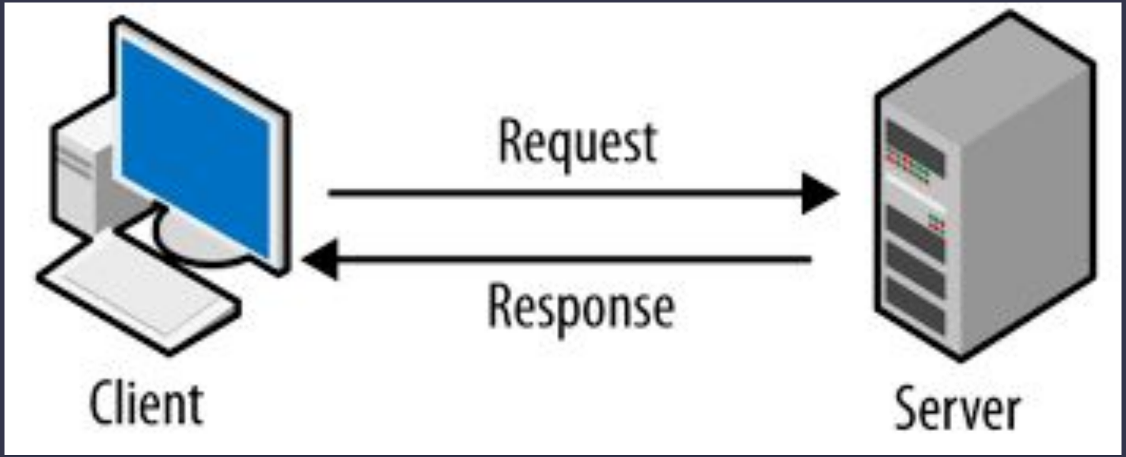
How can we use this data?

https://forms.sheffield.gov.uk/site/form/auto/make_enquiry_medium?team=Customer%20Services&email=customerservices@sheffield.gov.uk&eformURL=https://www.sheffield.gov.uk/your-city-council/contact-us



Contact Customer Services

How websites work?

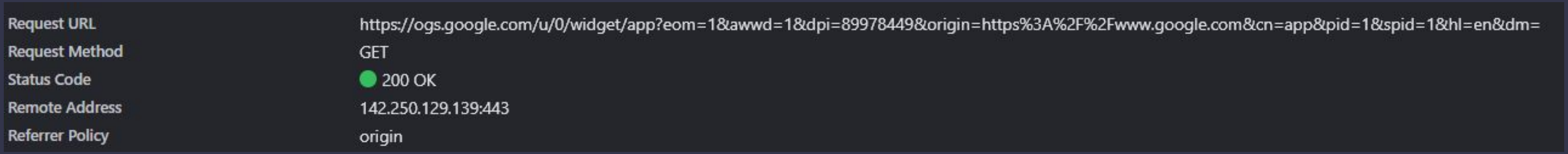


Request types

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

HTTP Request Structure:
Head - metadata
Body - data

HTTP Requests



Request URL	https://ogs.google.com/u/0/widget/app?eom=1&awwd=1&dpi=89978449&origin=https%3A%2F%2Fwww.google.com&cn=app&pid=1&spid=1&hl=en&dm=
Request Method	GET
Status Code	● 200 OK
Remote Address	142.250.129.139:443
Referrer Policy	origin

Inspect > Network > Click on a specific response

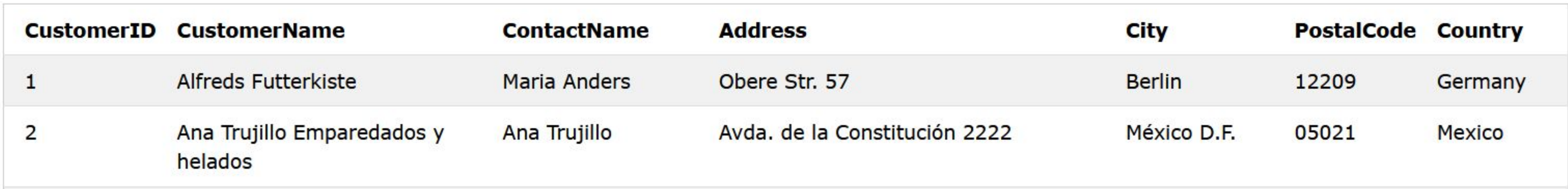
What this is saying:
Request URL - who sent the request
Request Method - what method the request is using
Status Code - 200 means it was a success

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

SQL Injection

Most (there are always exceptions) databases use SQL - Structured Query Language

SELECT * FROM user WHERE CustomerID = 1;



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

On its own, it doesn’t look like we can do much with it just yet…

SQL Injection Continued

So lets look at how its used

SELECT *values* FROM *table* WHERE *column* = *value;*

The values, table, column and value are set by the application, at some stage.

The issue comes when they are DIRECTLY set by the application.

How can we use this knowledge?

Endpoints - places on the website where data is exposed to users

We need to look for certain endpoints for SQL Injections to work

Requirements:

- User Input is taken by the website
- User input changes what is displayed
- The data displayed is stored in a database
- The database is vulnerable to a SQL injection attack

Surely no one is vulnerable??

SQL Injections have been around for a very long time (since 1998!). Surely it’s not relevant in 2025.

https://www.rapid7.com/blog/post/2025/02/13/cve-2025-1094-postgresql-psql-sql-injection-fixed/

You will use postgreSQL in at least second year comp sci. It is very popular as as a database option.

Why this occurs?

- Humans are fallible
- The internet was not built with security in mind

Practical Time!

http://13.40.138.221/

XSS

XSS has three main types:

- Reflected
- Stored
  - Blind
- DOM based

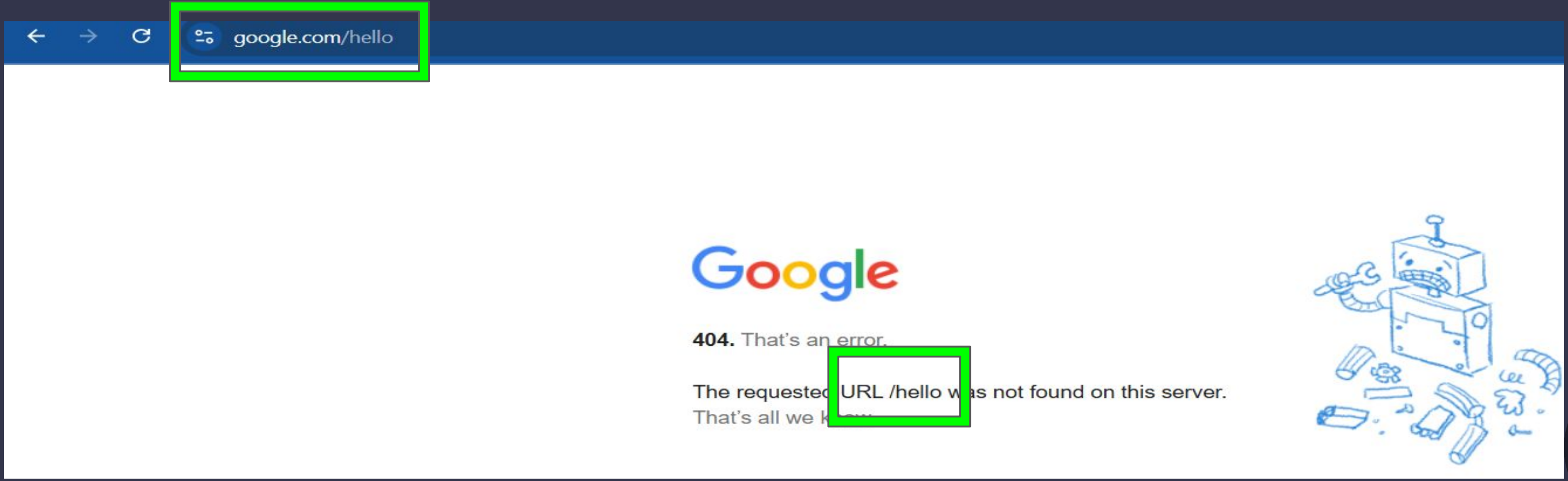
All of them use dynamic content loaded by the website to execute HTML, javascript, or any other code that the website will/can execute.

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.
This is an example for educational, demonstrative purposes.



google.com/hello

Google

404. That's an error.

The requested URL /hello was not found on this server. That's all we know.

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 message on a message board

Everytime a user loads that message - the code is executed

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

A bit of a different story … which needs a little history lesson first

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

Lets you change it with code - think document.write !

You actually use it all the time when you writing websites

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction for more info!

DOM based XSS continued

DOM based attacks change the DOM on a clients page

The javascript that is running on the page is manipulated to execute malicious code

How does this differ from reflected or stored?

Its (often)not going through the HTTP Response!

Give me an example already!

http://www.some.site/page.html?default=<script>alert(document.cookie)</script>

This URL has a parameter called default

The javascript on the page accesses this parameter and uses the value in a document.write piece of code.

The javascript passes the parameter to the DOM
This executes the HTML and subsequent javascript
Then renders it to the page

This actually goes through the server as its in the HTTP response header

The bypass? URI fragments (the bits after a #)

Practical Time

http://18.132.190.176/

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 6th October: Intro to Linux

Monday 13th October: OSINT

Any Questions?

SU Sign Up

SHEFFIELD ETHICAL STUDENT HACKERS

Discord

www.shefesh.com

Thanks for coming!