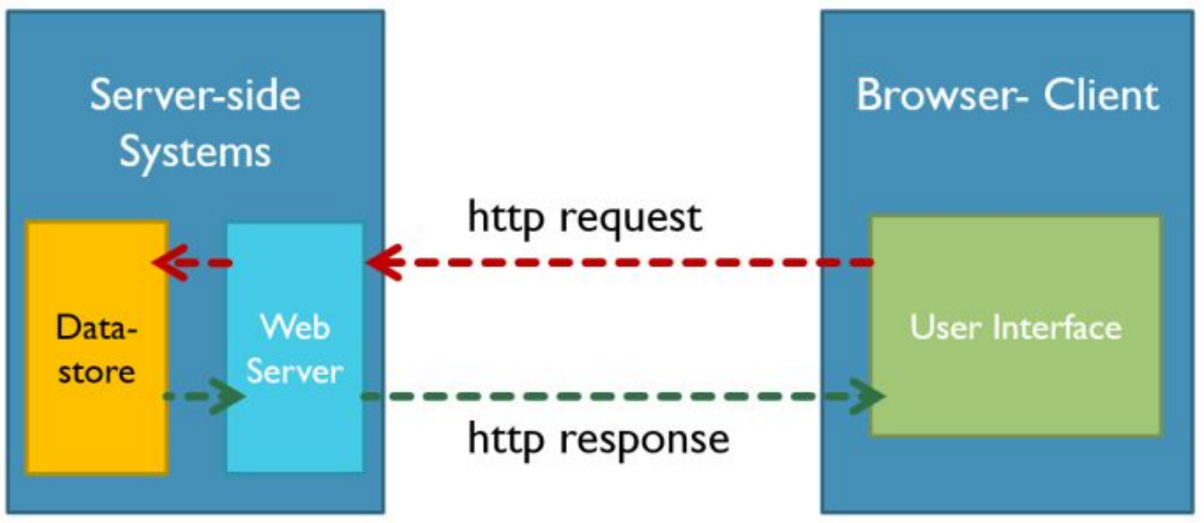
Web

2022-04-21
Axel Nilsson

Web fundamentals

- Server
- Client
- Request web pages



Problems

[Where are the robots](#)

[picobrowser](#)

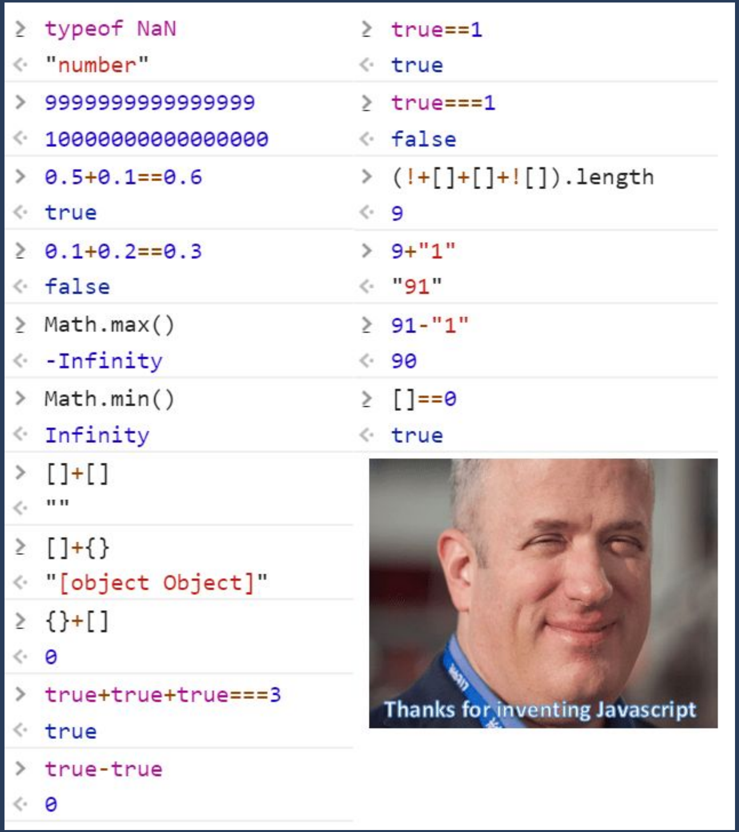
[roboto sans](#)

Server Side

- Javascript (Node.js)
- PHP
- Python

Client Side

- HTML
  - Structure
- CSS
  - Design and appearance
- Javascript
  - Dynamic actions



```
> typeof NaN
< "number"
> 9999999999999999
< 10000000000000000
> 0.5+0.1==0.6
< true
> 0.1+0.2==0.3
< false
> Math.max()
< -Infinity
> Math.min()
< Infinity
> []+[]
< ""
> []+{}
< "[object Object]"
> {}+[]
< 0
> true+true+true===3
< true
> true-true
< 0
```

```
> true==1
< true
> true===1
< false
> (!+[]+[]+![]).length
< 9
> 9+"1"
< "91"
> 91-"1"
< 90
> []==0
< true
```

Problems

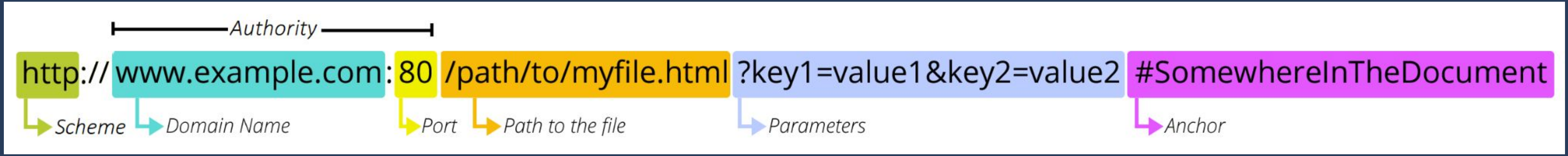
don’t-use-client-side

client-side-again

URL

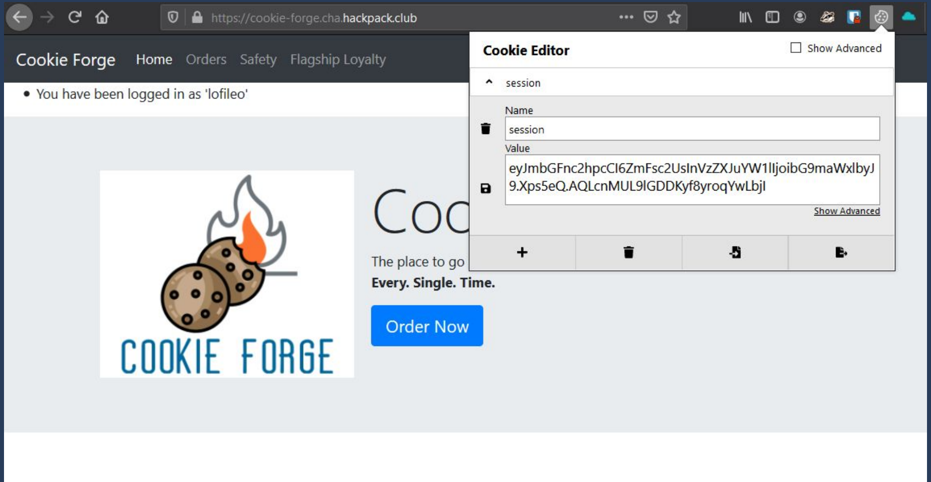
- Protocol
- Domain name / IP
- Port
- Path to file
- Parameters
- Anchor

When someone tries to "Rick Roll" you but you've memorized the URL:
You can't trick me anymore.
ifunny.co


Authority
http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#SomewhereInTheDocument
Scheme
Domain Name
Port
Path to the file
Parameters
Anchor

HTTP Cookies

- Stores information about a user
- Authentication cookies
- Tracking cookies



https://cookie-forge.cha.hackpack.club

Cookie Forge Home Orders Safety Flagship Loyalty

- You have been logged in as 'lofileo'

Coo

The place to go

Every. Single. Time.

Cookie Editor ☐ Show Advanced

session

Name: session

Value:

```
eyJmbGFnc2hpcCI6ZmFsc2UsInVzZXJuYW1lIjoibG9maWxlbyJ
9.Xps5eQ.AQLcnMUL9lGDDKyf8yroqYwLbjI
```

Problems

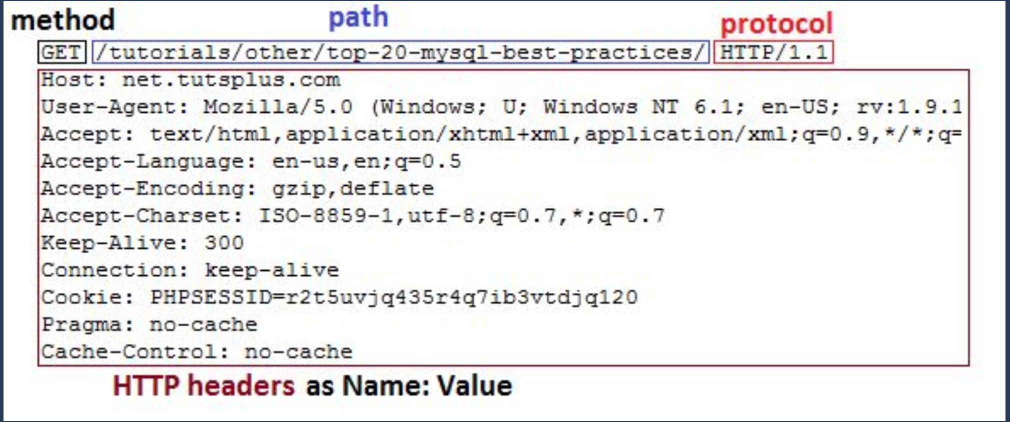
[Cookies](#)

[Logon](#)

[Power Cookie](#)

HTTP Methods

- GET
  - Request data from server
  - Send data via url parameters
- POST
  - Send data to the server
  - Send data in request body
- HEAD
  - Similar to GET but without response body
- OPTIONS
  - Ex. to find what method a server supports

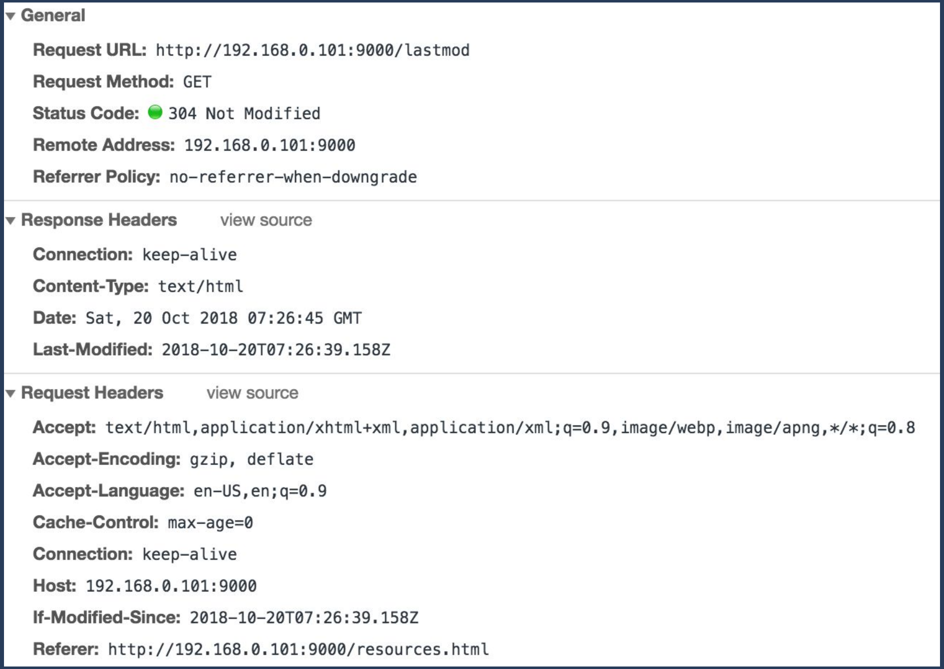


```
GET /tutorials/other/top-20-mysql-best-practices/ HTTP/1.1
Host: net.tutsplus.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120
Pragma: no-cache
Cache-Control: no-cache
```

Problems
get aHead

Headers

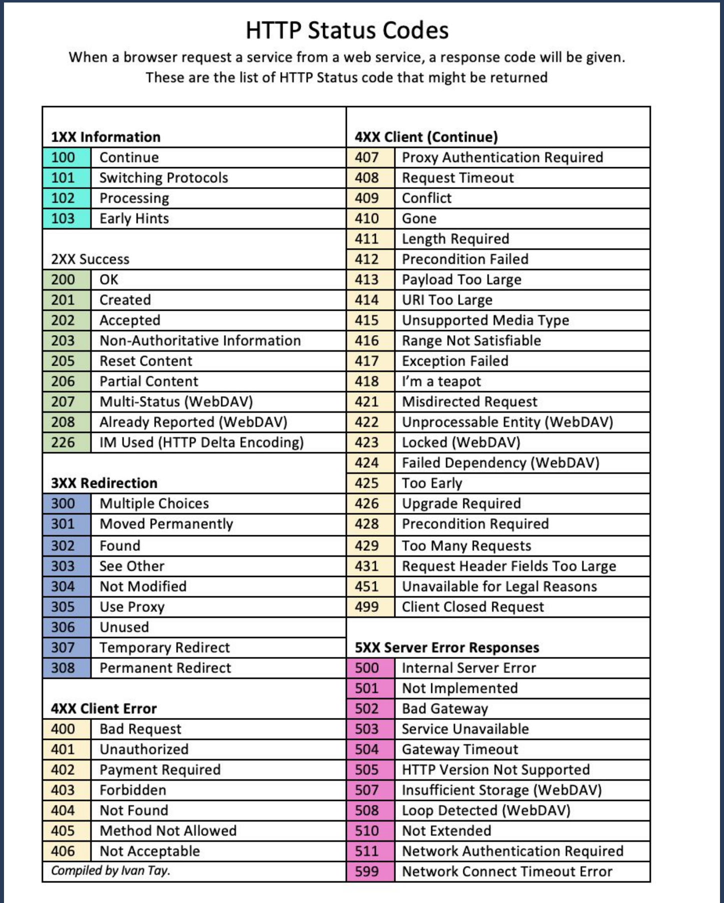
- Requests headers
  - Sent from the browser to the server
- Response headers
  - From the server back to the browser



```
▼ General
    Request URL: http://192.168.0.101:9000/lastmod
    Request Method: GET
    Status Code: ● 304 Not Modified
    Remote Address: 192.168.0.101:9000
    Referrer Policy: no-referrer-when-downgrade
▼ Response Headers      view source
    Connection: keep-alive
    Content-Type: text/html
    Date: Sat, 20 Oct 2018 07:26:45 GMT
    Last-Modified: 2018-10-20T07:26:39.158Z
▼ Request Headers      view source
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
    Accept-Encoding: gzip, deflate
    Accept-Language: en-US,en;q=0.9
    Cache-Control: max-age=0
    Connection: keep-alive
    Host: 192.168.0.101:9000
    If-Modified-Since: 2018-10-20T07:26:39.158Z
    Referer: http://192.168.0.101:9000/resources.html
```

Response Codes

- 1xx - Information
- 2xx - OK (success)
- 3xx - Redirection
- 4xx - Client side error
- 5xx - Server side error



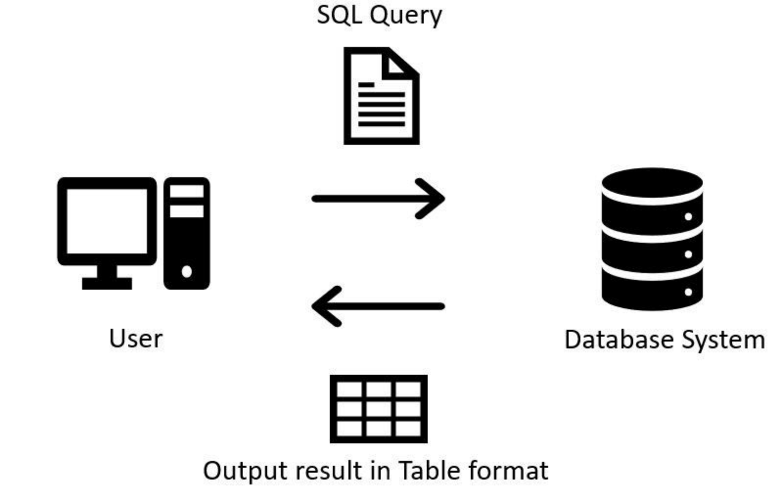
HTTP Status Codes

When a browser request a service from a web service, a response code will be given.
These are the list of HTTP Status code that might be returned

1XX Information		**4XX Client (Continue)**	
100	Continue	407	Proxy Authentication Required
101	Switching Protocols	408	Request Timeout
102	Processing	409	Conflict
103	Early Hints	410	Gone
		411	Length Required
2XX Success		412	Precondition Failed
200	OK	413	Payload Too Large
201	Created	414	URI Too Large
202	Accepted	415	Unsupported Media Type
203	Non-Authoritative Information	416	Range Not Satisfiable
205	Reset Content	417	Exception Failed
206	Partial Content	418	I'm a teapot
207	Multi-Status (WebDAV)	421	Misdirected Request
208	Already Reported (WebDAV)	422	Unprocessable Entity (WebDAV)
226	IM Used (HTTP Delta Encoding)	423	Locked (WebDAV)
		424	Failed Dependency (WebDAV)
3XX Redirection		425	Too Early
300	Multiple Choices	426	Upgrade Required
301	Moved Permanently	428	Precondition Required
302	Found	429	Too Many Requests
303	See Other	431	Request Header Fields Too Large
304	Not Modified	451	Unavailable for Legal Reasons
305	Use Proxy	499	Client Closed Request
306	Unused		
307	Temporary Redirect	**5XX Server Error Responses**	
308	Permanent Redirect	500	Internal Server Error
		501	Not Implemented
4XX Client Error		502	Bad Gateway
400	Bad Request	503	Service Unavailable
401	Unauthorized	504	Gateway Timeout
402	Payment Required	505	HTTP Version Not Supported
403	Forbidden	507	Insufficient Storage (WebDAV)
404	Not Found	508	Loop Detected (WebDAV)
405	Method Not Allowed	510	Not Extended
406	Not Acceptable	511	Network Authentication Required
Compiled by Ivan Tay.		599	Network Connect Timeout Error

SQL

- SQL - language used to access and manipulate databases
- DML - Data Manipulation statements
  - SELECT, INSERT, UPDATE, DELETE
  - Ex: SELECT * FROM MyTable
- DDL - Data Definition statements
  - CREATE, ALTER, DROP, TRUNCATE
  - Ex: CREATE TABLE Persons (ID int, Name varchar(255));



Problems

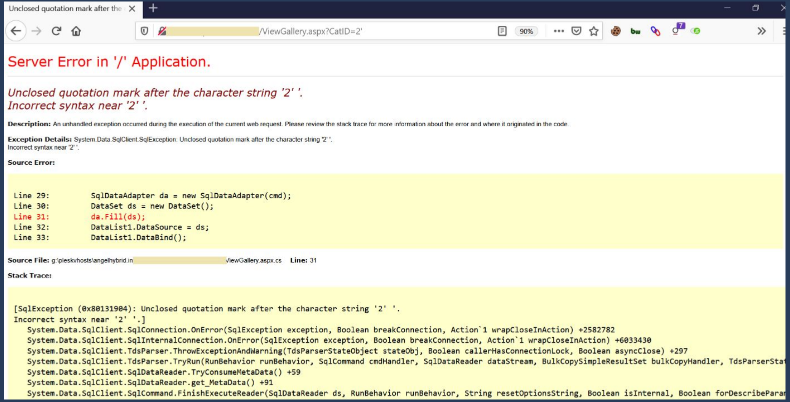
SQL Direct

SQL Injections (SQLi)

- Execute SQL scripts on the server that steals information from the server
- Examples
  - Authentication forms
  - Search engines
  - E-Commerce sites
  - Blog

Error based SQLi

- In-band injection
- Utilizes the error output
- Verification and exploitation
  - Break out of SQL query statements using for example single-quotes, double-quotes, backticks or semi-colons in the input field
  - Look for any error-messages or misbehavior in the application



[redacted]/ViewGallery.aspx?CatID=2'

Server Error in '/' Application.

Unclosed quotation mark after the character string '2' '.
Incorrect syntax near '2' '.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Unclosed quotation mark after the character string '2' '.
Incorrect syntax near '2' '.

Source Error:

```
Line 29:        SqlDataAdapter da = new SqlDataAdapter(cmd);
Line 30:        DataSet ds = new DataSet();
Line 31:        da.Fill(ds);
Line 32:        DataList1.DataSource = ds;
Line 33:        DataList1.DataBind();
```

Source File: g:\pleskvhosts\angelhybrid.in[redacted]ViewGallery.aspx.cs   **Line:** 31

Stack Trace:

```
[SqlException (0x80131904): Unclosed quotation mark after the character string '2' '.
Incorrect syntax near '2' '.]
   System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +2582782
   System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +6033430
   System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose) +297
   System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserSta
   System.Data.SqlClient.SqlDataReader.TryConsumeMetaData() +59
   System.Data.SqlClient.SqlDataReader.get_MetaData() +91
   System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString, Boolean isInternal, Boolean forDescribeParar
```

Error based SQLi

- Input field is a **String** and the query is **SELECT * FROM Table WHERE id = ‘1’**
  - **If ‘ gives false then ‘ ‘ must give true**
  - **If “ gives false then “ “ must give true**
  - **If \ gives false then \\ must give true**
- Input field is a **Numeric** and the query is **SELECT * FROM Table WHERE id = 1;**
  - AND 1 = True, AND 0 = False
  - AND true = True, AND false = False
  - 1*50 returns 50 if vulnerable and 1 if not vulnerable

Error base SQLi

- Input field is **LOGIN** and query is **SELECT * FROM Table WHERE username = ‘’;**
  - **‘ OR ‘1**
  - **‘ OR 1 --**
  - **“ OR “”” = “**
  - **‘LIKE’**
  - **‘=’**

Union based SQLi

- Extract information from the database by extending the results returned by the original query
- We can use the **Union** operator for this
- SELECT * FROM profiles WHERE id=$id UNION SELECT 1,2,3 -- LIMIT 0,1
- SELECT header, txt FROM news UNION ALL SELECT name, pass FROM members

Blind SQLi

- No error message is displayed by the application but its behaviour changes
- Boolean-based
  - Result varies depending on whether the sql-query is true or false
  - Ex. error message when login in disappear when we inject our payload
- Time-based
  - Sql-query that makes the database wait before returning the result. The difference in time can be used to leak data.

SQLi problems

- SQLiLite - PicoCTF
- Irish-Name-Repo 1 - PicoCTF
- Irish-Name-Repo 2 - PicoCTF
- Irish-Name-Repo 3 - PicoCTF
- https://sqlilabs.carelessfinch.me/ - bunch of SQLi challs
- https://www.wechall.net/challs/MySQL/by/chall_score/ASC/page-1

Cross Site Scripting (XSS)

- XSS is when a user of an application can send javascript that is executed by the browser of another user of the same application
- Javascript can
  - Modify the page (DOM)
  - Send HTTP requests
  - Access cookies
- This can be combined to form an xss attack. Ex. extract cookies, send to server of attacker

Reflected XSS

- XSS through URL parameters
- https://example.com?data=<script>alert(1)</script>
  - <html>
  - <body>
  - <script>alert(1)</script>
  - </body>
  - </html>

Stored XSS

- The XSS payload is provided from the website itself
- Ex. posting a XSS payload as a comment or blog post that is displayed to other users

DOM XSS

- The browser itself is injecting an XSS payload into the DOM
- The server might not be injectable but the client side javascript files are causing the issue
- Ex. dynamic javascript code such as eval() or innerHTML
- Exploit in URL which is accessed with the window.location object

http://www.example.com/userdashboard.html#context=<script>SomeFunction(somevariable)</script>.

XSS Problems

- https://xss.challenge.training.hacq.me/
- https://xss-game.appspot.com/
- https://xss.pwnfunction.com/

Tooling

- Python requests
- Burpsuite
- Sqlmap
- Ngrok
- Curl