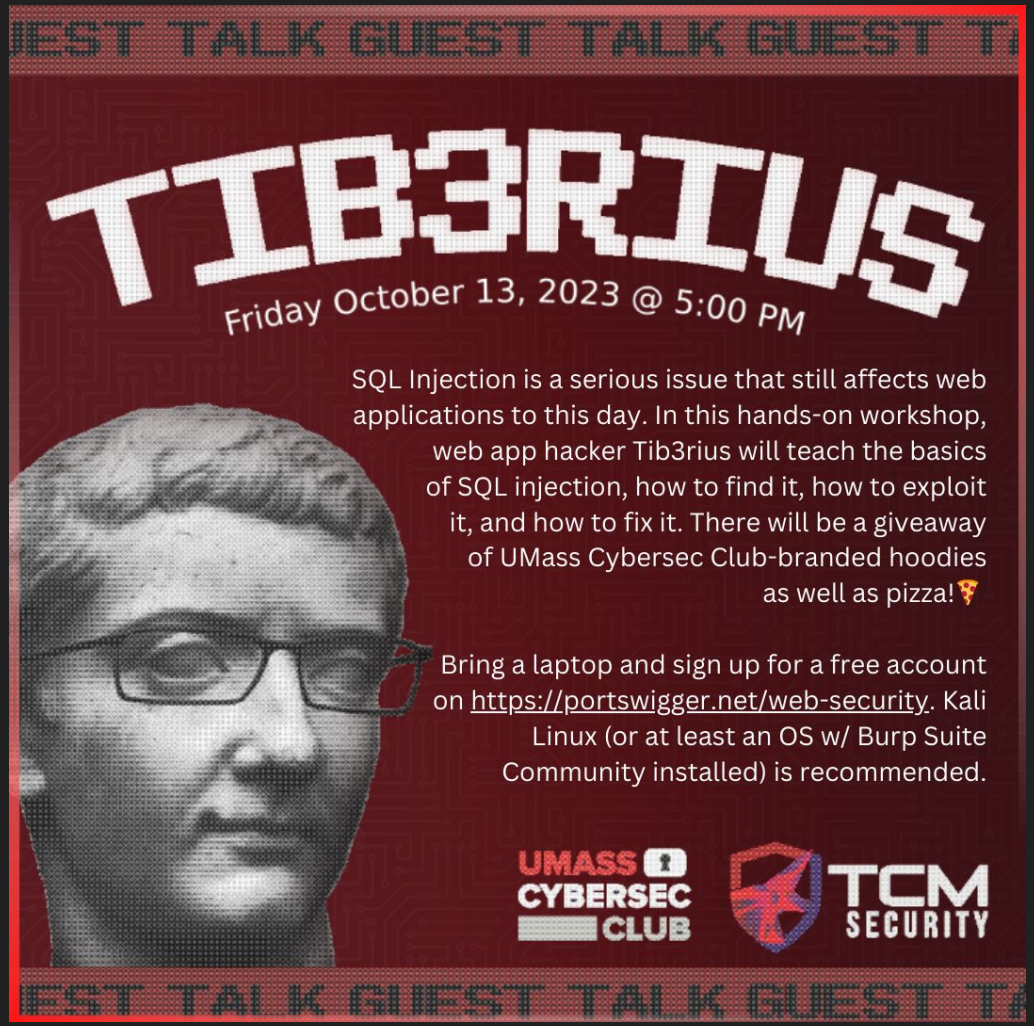
Web Exploitation

An Introduction

UMASS CYBERSEC CLUB



Interest Form

UMASS CYBERSEC CLUB

Disclaimer!

H4ck1ng is 1lleg4l 4nd b4d

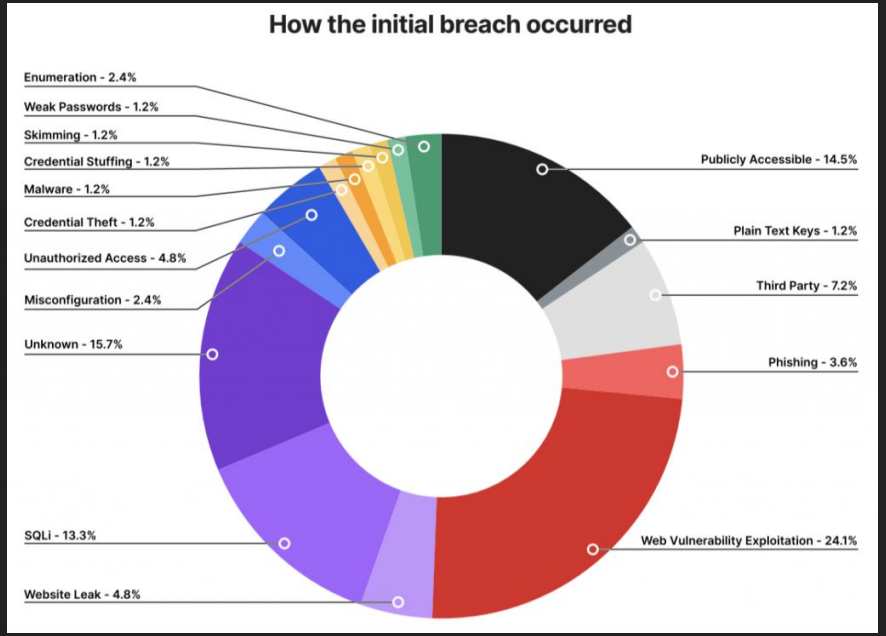
Don’t do this stuff without explicit permission. YOU WILL GET CAUGHT...

```
b'From: 104.28.234.24:31465 Data | sudo apt install ghidra\n'

b'From: 128.119.202.93:20507 Data | balls\n'

b'From: 128.119.202.93:63920 Data | hi\n'
b'From: 104.28.202.24:64838 Data | drugs\n'
b'From: 104.28.202.24:64838 Data | alcohol\n'
```

UMASS CYBERSEC CLUB

What is Web Exploitation

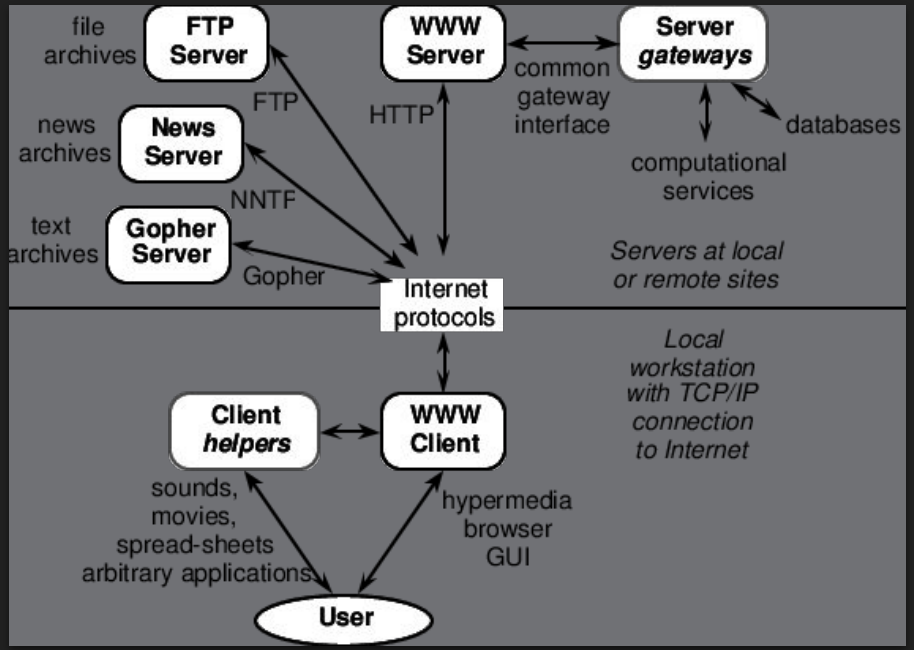
- Finding and exploiting vulnerabilities in web-based application
- Some common web vulnerabilities:
  - SQL Injection
  - Cross Site Scripting
  - Local File Inclusion
  - **Command Injection**
- Like the web itself, it can feel like a complicated mess
  - In both development and security, a lot of people don't know what to do. So if you're lost, don't worry, it's all part of the process.



UMASS CYBERSEC CLUB

What is the Web?

- **World Wide Web (WWW):** “an interconnected system of public webpages accessible through the Internet” - [mdn web docs]
- The web is NOT the internet, it is just an application built on top of the internet
  - If you want to learn more about this take CS 453!
- A browser is an interface to receive, display, and execute content retrieved from the web

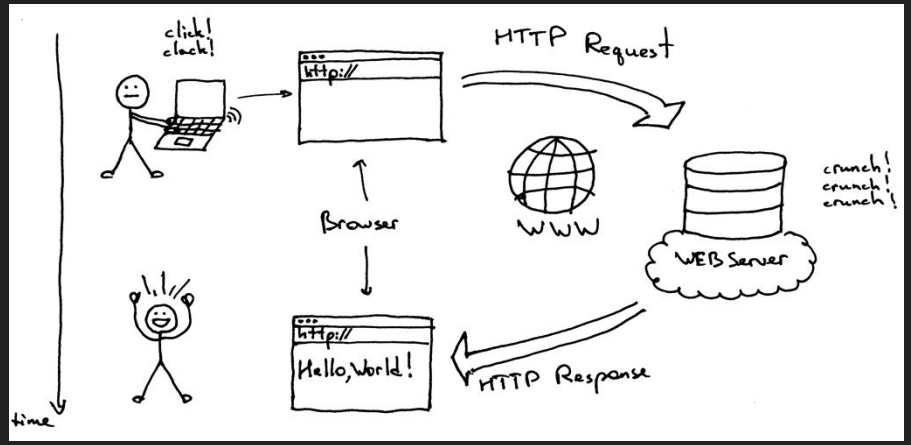


UMASS CYBERSEC CLUB

Client-Server Model

- **Client**: System/program that connects to a remote server to retrieve content
  - For most users: the browser
- **Server:** A local or remote system that provides data to a user
  - Can be local or remote
- You can set up a local file server of your own using:

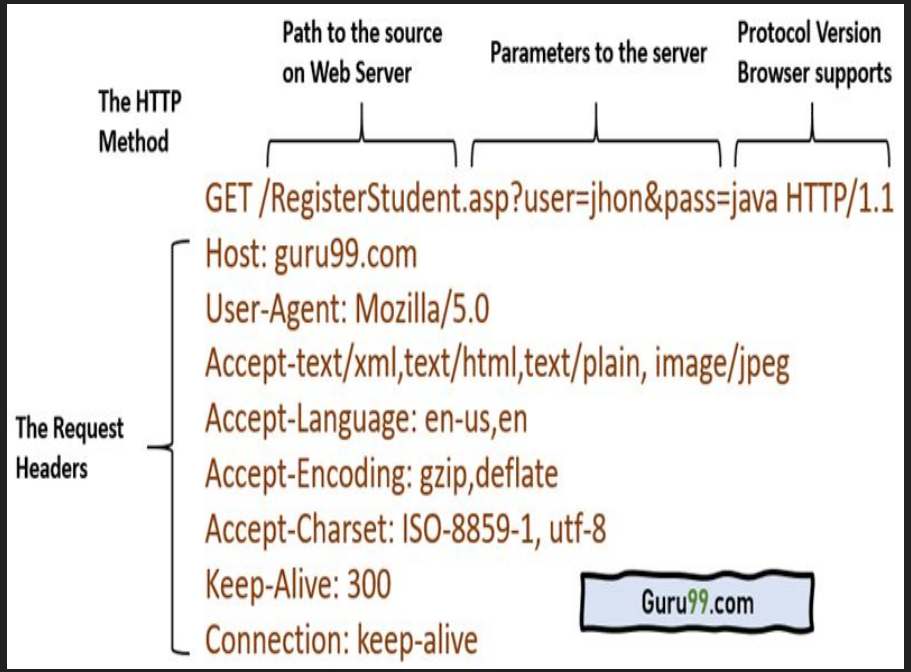
  python3 -m http.server 4444



UMASS CYBERSEC CLUB

Hypertext Transfer Protocol (HTTP)

- **HTTP:** A special protocol designed for communicating between web client and servers
  - Follows the client server model we mentioned earlier
- We send an HTTP request from our client and receive a response from the server
- An HTTP request consists of:
  - **HTTP version:** HTTP 1/2
  - **URL**: https://google.com
  - **HTTP Method:** GET, POST, etc..
  - **HTTP Request headers**
  - Optional: HTTP Body



UMASS CYBERSEC CLUB

Burp Suite and HTTP Proxies

- Today we're primarily going to use three features of Burp Suite:
  - **Target**: Let's us see the sitemap of a website we visit: this sitemap will grow as we visit more of the website
  - **Proxy:** Our proxy will intercept all HTTP requests it receives where we can the see and modify the contents while they are being sent
  - **Repeater:** We can send any HTTP request we intercept to repeater where we have a view of our request and response to debug an endpoint

UMASS CYBERSEC CLUB

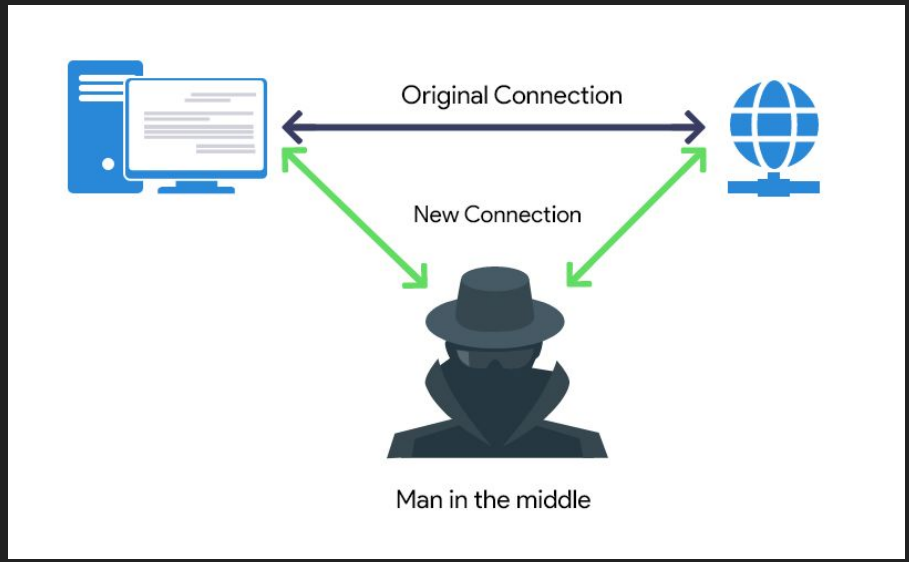
Burp Suite and HTTP Proxies

- Burp Suite is a pentesting tool to find, enumerate, and exploit vulnerable web applications
- Burpsuite has a proxy that receives all HTTP/s traffic on local port 8080 and forwards or intercepts based on settings on our settings

UMASS CYBERSEC CLUB

HTTPS - A Side Note

- You may have noticed most websites you connected to are using HTTPS
- This is a more secure version of HTTP encrypted using Transport Layer Security (TLS)
- This is use to prevent man in the middle attacks (MITM)
- You won't need to for the demo, but if you want Burp Suite to work over HTTPS you may need to follow this [tutorial]



UMASS CYBERSEC CLUB

Let’s intercept a request with Burp Suite.

HTTP methods - What are we requesting?

Common HTTP Request Method	Meaning
OPTIONS	Requests the server to tell us the available methods on an endpoint
GET	Requests the resource from a filename provided on the host, we receive the content in the body
HEAD	Same as GET, but body is not given
POST	Submits an entity to the specified resource, often causing a change in state or side effects on the server

- Can read more about different headers here

HTTP Status Codes - How is the Server Responding?

- Typically only seen in requests, for HTTP responses we get a status code:

Status Code	Meaning
200	OK
301	Moved permanently
302	Found
400	Bad/Invalid request

Status Code	Meaning
403	Unauthorized
404	Not found
418	I'm a teapot
500	Internal Server Error

- Sometimes we can find exploits because improper methods were allowed

UMASS CYBERSEC CLUB

HTTP Headers - How are we requesting our data.

- Headers are sent by both the server and client: to tell the client and the server information about each other
- Useful Request Headers:
  - **Cookie:** Cookies are used for tracking and/or authenticating users
    - We can edit and resend cookies to exploit logic bugs in developer code!
  - **Host:** What server are we requesting a resource on
  - **Content-Type:** Ensures our content is sent correctly to the server, examples are JSON, form-data, etc...
- A response header to look for:
  - **Server or x-powered-by**: Tells us what server is running in the background, useful for researching vulnerabilities
  - Read more about security with response headers here

UMASS CYBERSEC CLUB

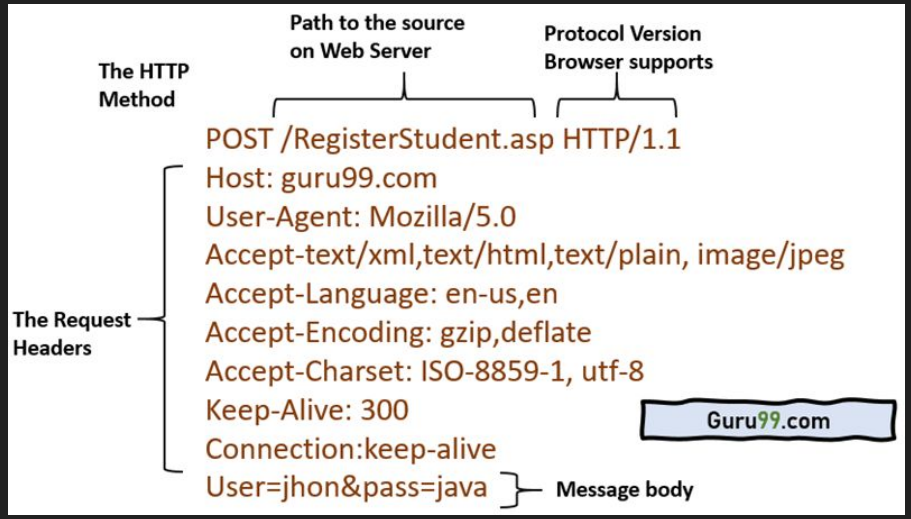
You can now try Challenges 1 and 2.

https://training.umasscybersec.org

UMASS CYBERSEC CLUB

HTTP Body - What is the data we are sending?

- With **POST** requests we can send data to the server via our HTTP body
  - Below all our headers of a post request we can add a new line and begin adding data
  - This data can be in multiple forms, we can specify what form in our request with the
- We can also send data to the server with **GET** requests using URL Parameters
  - https://www.google.com/search?q=hello_world



UMASS CYBERSEC CLUB

Command Injection

What's wrong with the following code? Discuss it with the people around you.

```
import os
def index(user_input):
    #Operating system executes ping -c 3 user_input
    return os.popen(f'ping -c 3 {user_input}').read()
```

UMASS CYBERSEC CLUB

How can we run multiple commands without a new line? Discuss with the people around you!

```
import os
def index(user_input):
    #Operating system executes ping -c 3 user_input
    return os.popen(f'ping -c 3 {user_input}').read()
```

UMASS CYBERSEC CLUB

How can we run multiple commands without a new line?

- echo "hi" **;** ls
- echo "hi" **|** ls
- echo "hi" **&&** ls
- We can find a cheat sheet here

UMASS CYBERSEC CLUB

You can now hack Challenge 3.

UMASS CYBERSEC CLUB

What now?

- We will have future talks, diving deep into web exploits
- In the meantime, you can:
  - Play PicoCTF
  - Try the PortSwigger labs
  - Play weekly CTFs with the club!

UMASS CYBERSEC CLUB