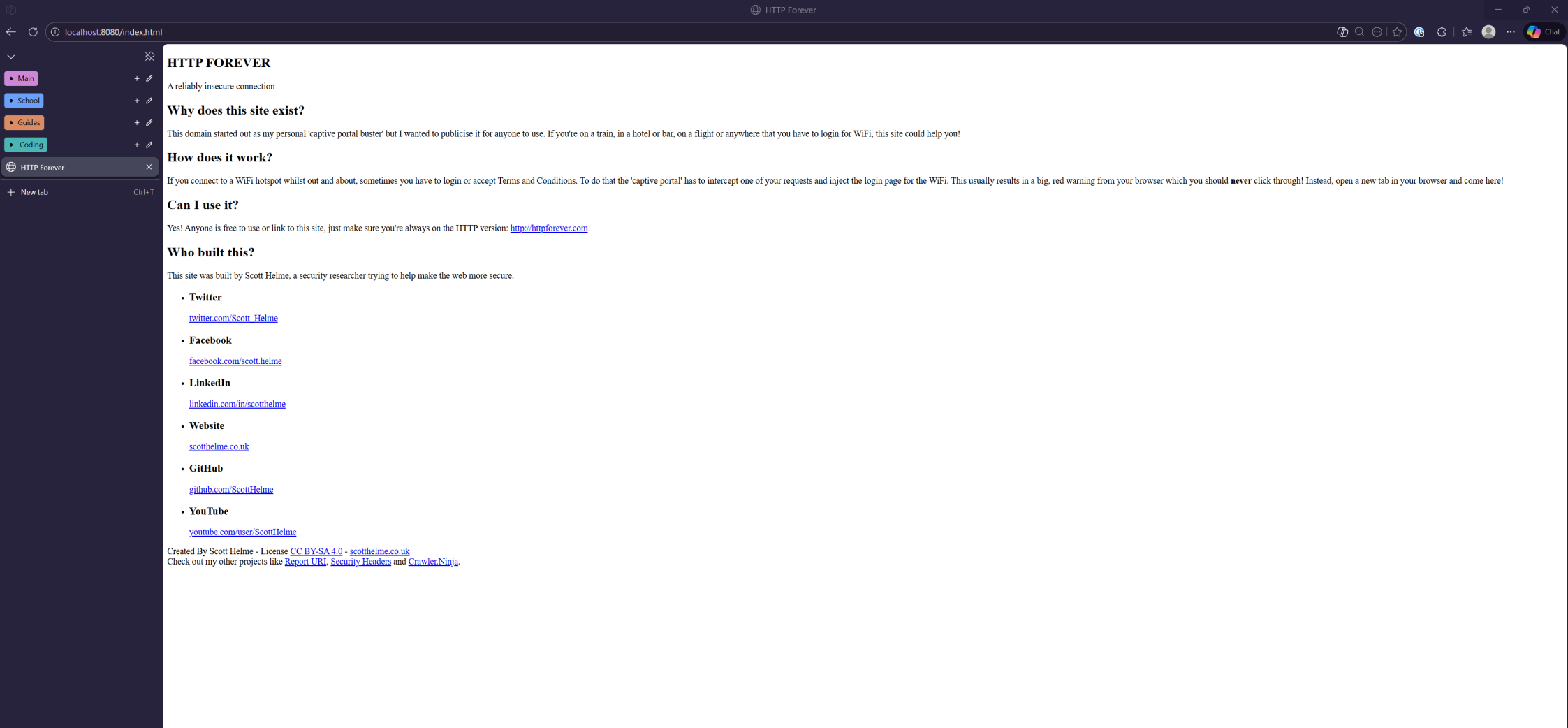

HTTP FOREVER

A reliably insecure connection

Why does this site exist?

This domain started out as my personal 'captive portal buster' but I wanted to publicise it for anyone to use. If you're on a train, in a hotel or bar, on a flight or anywhere that you have to login for WiFi, this site could help you!

How does it work?

If you connect to a WiFi hotspot whilst out and about, sometimes you have to login or accept Terms and Conditions. To do that the 'captive portal' has to intercept one of your requests and inject the login page for the WiFi. This usually results in a big, red warning from your browser which you should **never** click through! Instead, open a new tab in your browser and come here!

Can I use it?

Yes! Anyone is free to use or link to this site, just make sure you're always on the HTTP version: http://httpforever.com

Who built this?

This site was built by Scott Helme, a security researcher trying to help make the web more secure.

- **Twitter**

  twitter.com/Scott_Helme

- **Facebook**

  facebook.com/scott.helme

- **LinkedIn**

  linkedin.com/in/scotthelme

- **Website**

  scotthelme.co.uk

- **GitHub**

  github.com/ScottHelme

- **YouTube**

  youtube.com/user/ScottHelme

Check out my other projects like Report URI, Security Headers and Crawler.Ninja.

```
==========================================
TEST 2: Retriever -> Real Server
==========================================

Command:
  ./retriever http://httpforever.com/

Output:
Connecting to server: httpforever.com:80
Requesting file: /
Request sent, waiting for response...
Status: HTTP/1.1 200 OK
File saved as: index.html

Expected Result:
  - Status: HTTP/1.1 200 OK
  - File saved successfully

✓ TEST PASSED

```

```
==========================================
TEST 3: Retriever -> Your Server (200 OK)
==========================================

Command:
  ./retriever http://localhost:8080/index.html

Output:
Connecting to server: localhost:8080
Requesting file: /index.html
Request sent, waiting for response...
Status: HTTP/1.0 200 OK
File saved as: index.html

Expected Result:
  - Status: HTTP/1.0 200 OK
  - File saved successfully

✓ TEST PASSED
```

```
==========================================
TEST 4a: Unallowed Method - BREW
==========================================

Command:
  printf 'BREW /coffee.html HTTP/1.0\r\nHost: localhost\r\n\r\n' | nc localhost 8080

Output:
HTTP/1.0 418 I'm a teapot
Connection: close


Expected Result:
  - Status: HTTP/1.0 418 I'm a teapot

✓ TEST PASSED
```

```
==========================================
TEST 4b: Unallowed Method - POST
==========================================

Command:
  printf 'POST /index.html HTTP/1.0\r\nHost: localhost\r\n\r\n' | nc localhost 8080

Output:
HTTP/1.0 405 Method Not Allowed
Connection: close


Expected Result:
  - Status: HTTP/1.0 405 Method Not Allowed

✓ TEST PASSED
```

```
==========================================
TEST 5: Forbidden File (403 Forbidden)
==========================================

Command:
  ./retriever http://localhost:8080/MySecret.html

Output:
Connecting to server: localhost:8080
Requesting file: /MySecret.html
Request sent, waiting for response...
Status: HTTP/1.0 403 Forbidden
Server returned error code 403
Error page content:


Expected Result:
  - Status: HTTP/1.0 403 Forbidden
  - Error page displayed (not saved to file)

✓ TEST PASSED
```

```
========================================
TEST 6: Non-existent File (404 Not Found)
========================================

Command:
  ./retriever http://localhost:8080/doesnotexist.html

Output:
Connecting to server: localhost:8080
Requesting file: /doesnotexist.html
Request sent, waiting for response...
Status: HTTP/1.0 404 Not Found
Server returned error code 404
Error page content:
<html><body><h1>404 Not Found</h1><p>The requested file was not found on this server.</p></body></html>

Expected Result:
  - Status: HTTP/1.0 404 Not Found
  - Custom error page displayed

✓ TEST PASSED
```

```
==========================================
TEST 7: Malformed Request (400 Bad Request)
==========================================

Command:
  printf 'HELLO WORLD\r\n' | nc localhost 8080

Output:
HTTP/1.0 400 Bad Request
Connection: close


Expected Result:
  - Status: HTTP/1.0 400 Bad Request

✓ TEST PASSED
```