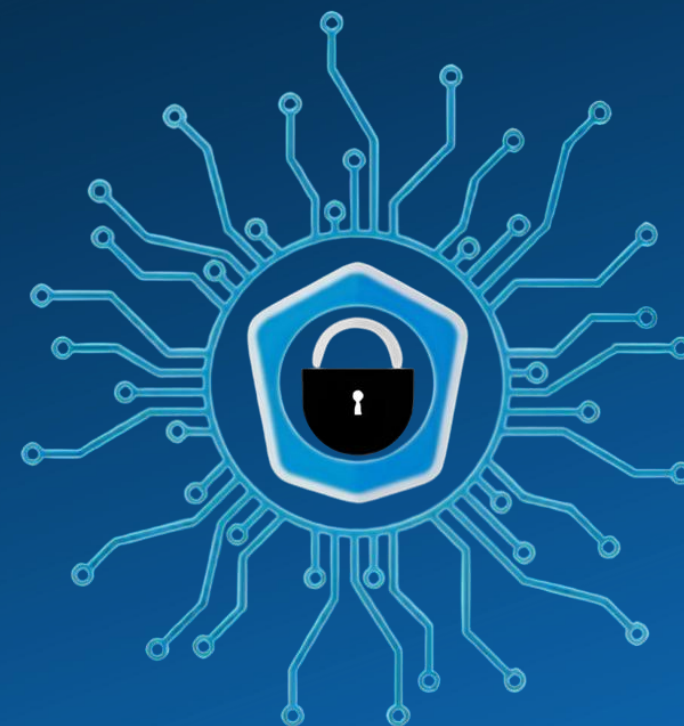


# **WIRELEAK CHALLENGE WALKTHROUGH**

**WRITTEN BY:  
TALEEN SKAFI**



**BZU - CSC**

Welcome to **WireLeak**, a challenge where a web page uses plain HTTP to transmit data with **no encryption**. Because of this, sensitive information travels in cleartext and can be captured by anyone watching the network. Use a packet sniffer to uncover the leaked credentials and capture the flag.



When you open the web page, you'll notice it has multiple sections. One of them is titled "Traffic Analysis", and it includes a message that says:

***"This site is currently undergoing testing to evaluate whether the data it sends and receives is transmitted securely."***

This observation hints that the site is using HTTP, not HTTPS , meaning the data is not encrypted. That could mean sensitive information might be leaking in plain text if you inspect the traffic closely.

**CSC-BZU**[Services](#) [Traffic Analysis](#) [Login](#)

## Traffic Security Analysis in Progress

Routine checks are performed to ensure our pages adhere to best practices and confidentiality standards.

### Our Services

Strategic, technical, and human-centric solutions to strengthen your security posture.

#### Penetration Testing

Simulate real-world attacks to identify vulnerabilities before adversaries do.

#### Security Audits

In-depth assessments of systems, networks, and applications to ensure compliance and resilience.

#### Training & Awareness

Equip your teams with the knowledge to recognize and mitigate evolving threats.

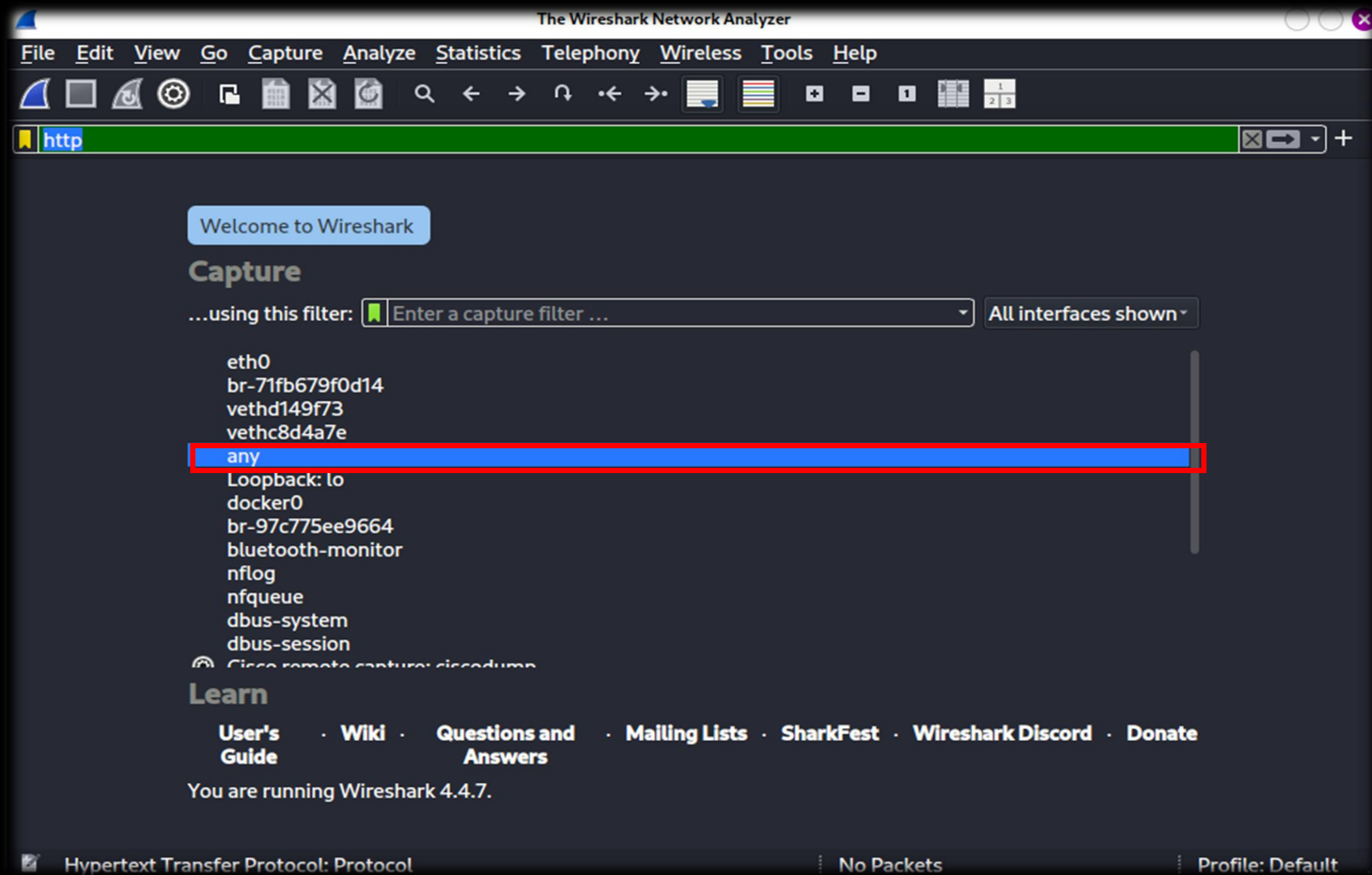


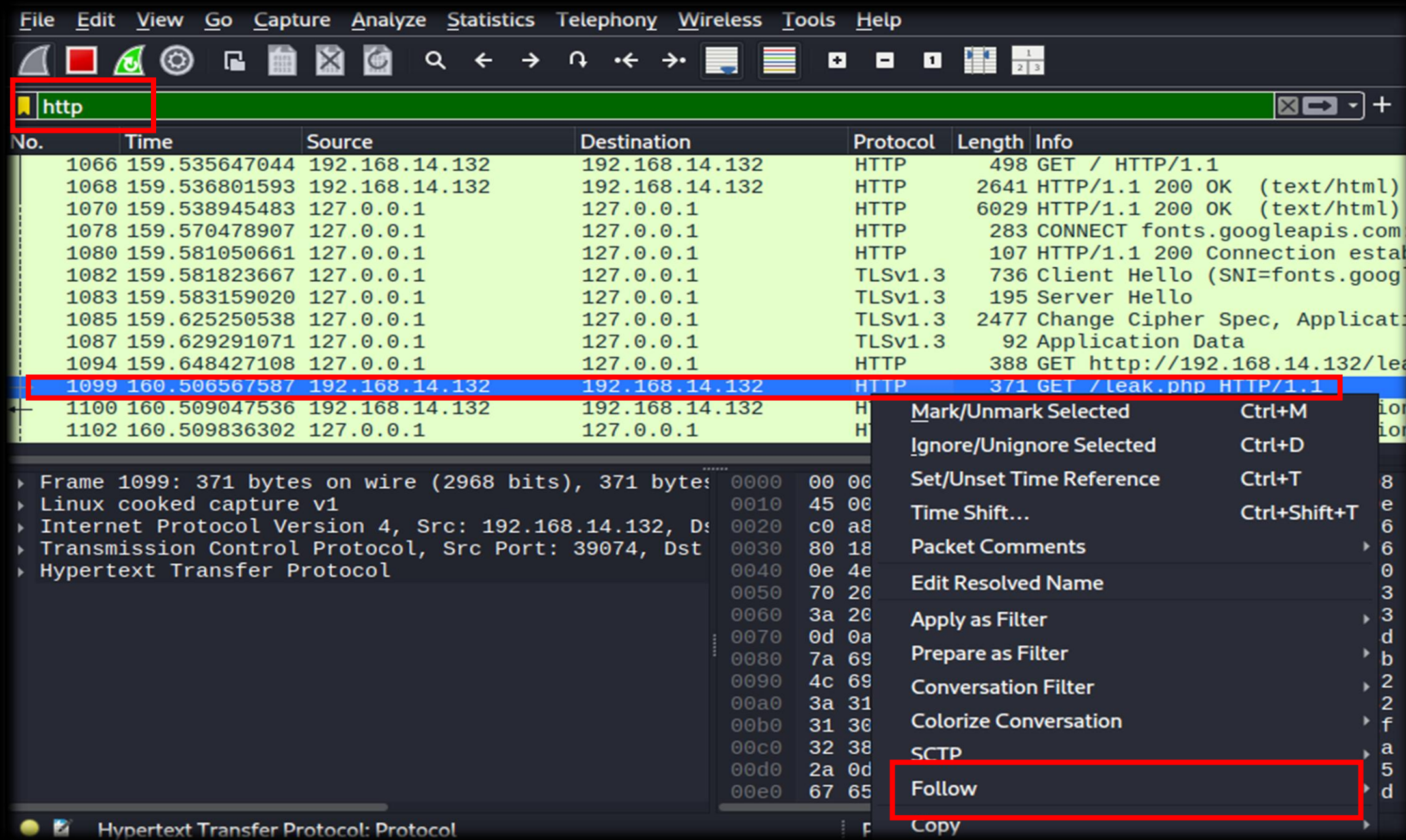
Since the website uses HTTP, the data it sends and receives is not encrypted. That means anyone on the same network can potentially intercept the traffic and read it in plain text.

In our case, we'll use Wireshark, a packet sniffer tool, to analyze the network traffic and see what kind of sensitive information might be leaking.

First, we'll open Wireshark and choose the network interface we want to monitor. In our case, we'll select "**any**", which is a special option that lets Wireshark capture traffic from all available network interfaces on the system. This is useful when you're not sure which specific interface the traffic is going through, or when multiple interfaces are active (like Ethernet, Wi-Fi, loopback, etc.).







The screenshot shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. The display filter bar at the top shows 'http' in a green box. The packet list pane displays a table of captured packets:

No.	Time	Source	Destination	Protocol	Length	Info
1066	159.535647044	192.168.14.132	192.168.14.132	HTTP	498	GET / HTTP/1.1
1068	159.536801593	192.168.14.132	192.168.14.132	HTTP	2641	HTTP/1.1 200 OK (text/html)
1070	159.538945483	127.0.0.1	127.0.0.1	HTTP	6029	HTTP/1.1 200 OK (text/html)
1078	159.570478907	127.0.0.1	127.0.0.1	HTTP	283	CONNECT fonts.googleapis.com
1080	159.581050661	127.0.0.1	127.0.0.1	HTTP	107	HTTP/1.1 200 Connection established
1082	159.581823667	127.0.0.1	127.0.0.1	TLSv1.3	736	Client Hello (SNI=fonts.google
1083	159.583159020	127.0.0.1	127.0.0.1	TLSv1.3	195	Server Hello
1085	159.625250538	127.0.0.1	127.0.0.1	TLSv1.3	2477	Change Cipher Spec, Applicat
1087	159.629291071	127.0.0.1	127.0.0.1	TLSv1.3	92	Application Data
1094	159.648427108	127.0.0.1	127.0.0.1	HTTP	388	GET http://192.168.14.132/lea
1099	160.506567587	192.168.14.132	192.168.14.132	HTTP	371	GET /leak.php HTTP/1.1
1100	160.509047536	192.168.14.132	192.168.14.132	H		
1102	160.509836302	127.0.0.1	127.0.0.1	H		

Packet 1099 is selected and highlighted in blue. A context menu is open over it, showing various actions. The 'Follow' option is highlighted in a red box. The packet details pane on the left shows the structure of the selected packet:

- Frame 1099: 371 bytes on wire (2968 bits), 371 bytes captured (2968 bits) on interface
- Linux cooked capture v1
- Internet Protocol Version 4, Src: 192.168.14.132, Dst: 192.168.14.132
- Transmission Control Protocol, Src Port: 39074, Dst Port: 80
- Hypertext Transfer Protocol

The packet bytes pane on the right shows the raw data in hexadecimal and ASCII. The status bar at the bottom indicates 'Hypertext Transfer Protocol: Protocol'.

Once we refresh the website, Wireshark will start capturing all the network traffic related to that action. To narrow it down and focus only on the web traffic, we can type **http** into the Wireshark display filter bar. This will filter out everything except HTTP packets, making it easier to spot any unencrypted data being exchanged between the browser and the server.



As we inspect the filtered traffic, we'll notice that a file called **leak.php** is being requested by the web page. This is suspicious. If we right-click on that packet and choose "Follow → TCP Stream", we'll see something interesting, possibly some plaintext sensitive data being leaked in the response.



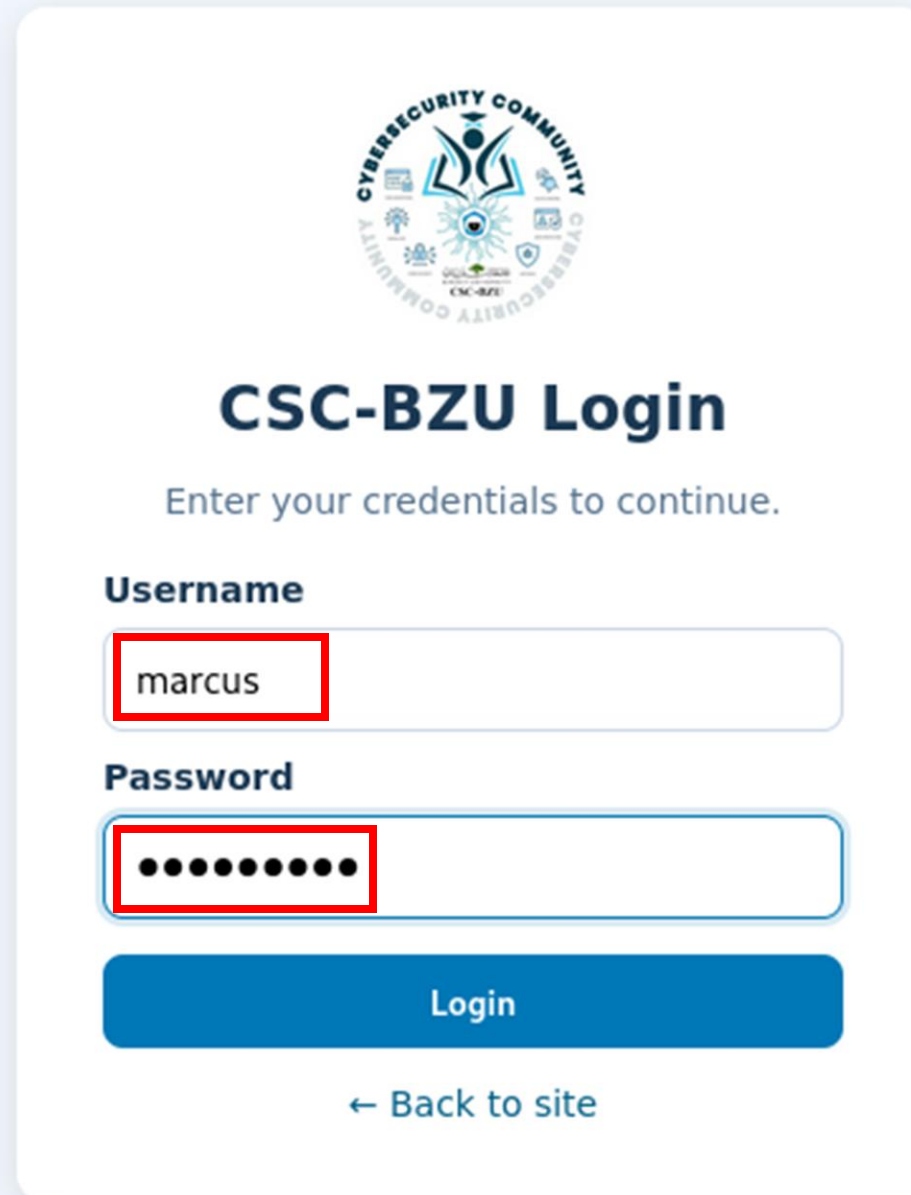
```
Wireshark · Follow TCP Stream (tcp.stream eq 13) · any

Host: 192.168.14.132
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://192.168.14.132/
X-Trigger: true
Connection: keep-alive
Priority: u=4

HTTP/1.1 200 OK
Date: Wed, 13 Aug 2025 20:34:22 GMT
Server: Apache/2.4.65 (Debian)
Content-Length: 59
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: application/javascript

// Internal auth log
// username=marcus password=m@rcu$123
```

Great! We've just uncovered some plaintext credentials for the user **Marcus**. Now, let's take these credentials and try logging into the website through the login page to check if they're actually valid.



The login form is centered on a light blue background. At the top is the CSC-BZU logo, a circular emblem with a book and a gear. Below the logo is the title "CSC-BZU Login" in bold dark blue, followed by the instruction "Enter your credentials to continue." in a smaller font. The form contains two input fields: "Username" with the value "marcus" and "Password" with ten black dots. Both fields are highlighted with red rectangles. Below the fields is a blue "Login" button, and at the bottom is a link "← Back to site".

**CSC-BZU Login**  
Enter your credentials to continue.

**Username**

**Password**

[Login](#)

[← Back to site](#)



Awesome, the credentials worked! We successfully logged in as **Marcus**, and as a result, we got our flag.



## CSC-BZU Login

Enter your credentials to continue.

**Welcome, marcus!**

CSC-  
BZU{Cl3@rText\_HTTP\_beacon}

[← Back to site](#)