

HTTP en detalle

Conozca cómo solicitar contenido de un servidor web utilizando el protocolo HTTP.

Tarea 1: ¿Qué es HTTP(S)?

- **¿Qué es HTTP? (Protocolo de transferencia de hipertexto)**
 - HTTP (creado en 1989-1991) es el protocolo que permite la comunicación con servidores web para transferir datos como HTML, imágenes o vídeos al visitar sitios.
 - **¿Qué es HTTPS? (Protocolo seguro de transferencia de hipertexto)**
 - HTTPS es la versión cifrada de HTTP, que protege la privacidad de los datos (evitando espionaje) y autentica el servidor (previniendo suplantación).
- ❖ **Pregunta:** ¿Qué significa HTTP?
- **Respuesta:** HyperText Transfer Protocol
- ❖ **Pregunta:** ¿Qué significa la S en HTTPS?
- **Respuesta:** secure
- ❖ **Pregunta:** On the mock webpage on the right there is an issue, once you've found it, click on it. What is the challenge flag?



An embedded page at static-labs.tryhackme.cloud says
You got the flag: THM{INVALID_HTTP_CERT}

- **Respuesta:** THM{INVALID_HTTP_CERT}

Tarea 2: Solicitudes y respuestas.

Las URL son las direcciones que el navegador usa para solicitar recursos (HTML, imágenes, etc.) a un servidor web, definiendo cómo (protocolo) y dónde (dominio/ruta) acceder a ellos.

- ❖ **Pregunta:** What HTTP protocol is being used in the above example?
- **Respuesta:** HTTP/1.1
- ❖ **Pregunta:** What response header tells the browser how much data to expect?
- **Respuesta:** Content-Length

Tarea 3: Métodos HTTP.

Los métodos HTTP definen la acción del cliente en una solicitud web. Los más comunes son GET (obtener datos), POST (enviar datos), PUT (modificar datos) y DELETE (eliminar datos).

- ❖ **Pregunta:** What method would be used to create a new user account?
 - **Respuesta:** POST
- ❖ **Pregunta:** What method would be used to update your email address?
 - **Respuesta:** PUT
- ❖ **Pregunta:** What method would be used to remove a picture you've uploaded to your account?
 - **Respuesta:** DELETE
- ❖ **Pregunta:** What method would be used to view a news article?
 - **Respuesta:** GET

Tarea 4: Código de estado HTTP.

- **HTTP Códigos de estado:**
 - Los códigos de estado HTTP indican el resultado de una solicitud al servidor.
- ❖ **Pregunta:** What response code might you receive if you've created a new user or blog post article?
 - **Respuesta:** 201
- ❖ **Pregunta:** What response code might you receive if you've tried to access a page that doesn't exist?
 - **Respuesta:** 404
- ❖ **Pregunta:** What response code might you receive if the web server cannot access its database and the application crashes?
 - **Respuesta:** 503
- ❖ **Pregunta:** What response code might you receive if you try to edit your profile without logging in first?
 - **Respuesta:** 401

Tarea 5: Encabezados.

Los encabezados HTTP son datos adicionales enviados entre el cliente y el servidor para controlar la comunicación.

- ❖ **Pregunta:** What header tells the web server what browser is being used?
 - **Respuesta:** User-Agent
- ❖ **Pregunta:** What header tells the browser what type of data is being returned?
 - **Respuesta:** Content-Type
- ❖ **Pregunta:** What header tells the web server which website is being requested?
 - **Respuesta:** Host

Tarea 6: Cookies.

- ¿Qué son las Cookies?
 - Son pequeños archivos de datos almacenados en el navegador para recordar información entre solicitudes HTTP (como sesiones, preferencias o tracking).
- ❖ **Pregunta:** Which header is used to save cookies to your computer?
 - **Respuesta:** Set-Cookie

Tarea 7: Realizar solicitudes.

Hacemos Click en el botón “View Site” para iniciar el emulador.

- ❖ **Make a GET request to /room page**

The screenshot shows a web browser interface with a dark theme. At the top, there is a search bar with the text "GET" and a dropdown arrow, followed by the URL "http://tryhackme.com/room". To the right of the URL bar are two buttons: a gear icon and a "Go" button. Below the URL bar, the browser displays the request details in a dark box:

```
GET /room HTTP/1.1
Host: tryhackme.com
User-Agent: Mozilla/5.0 Firefox/87.0
Content-Length: 0
```

Below the request details, there is a section labeled "Response" in a light gray header. The response body is displayed in a dark box:

```
HTTP/1.1 200 Ok
Server: nginx/1.15.8
Mon, 14 Jul 2025 22:54:7 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 252
Last-Modified: Mon, 14 Jul 2025 22:54:7 GMT

<html>
<head>
  <title>TryHackMe</title>
</head>
<body>
  Welcome to the Room page THM{YOU'RE_IN_THE_ROOM}
</body>
</html>
```

At the bottom of the screenshot, there is a window titled "THM Browser" with a standard macOS-style title bar (red, yellow, green buttons). The content of the window is:

```
Welcome to the Room page THM{YOU'RE_IN_THE_ROOM}
```


- **THM{YOU'RE_IN_THE_ROOM}**


- ❖ Make a GET request to /blog page and set the id parameter to 1
Note: Use the gear button on the right to manage URI parameters

Parameters

×

Here, you can add key-value pairs to customize your requests.

: 

GET ▼ 

GET /blog/?id=1 HTTP/1.1
Host: tryhackme.com
User-Agent: Mozilla/5.0 Firefox/87.0
Content-Length: 0

Response

```
HTTP/1.1 200 Ok
Server: nginx/1.15.8
Mon, 14 Jul 2025 22:57:25 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 250
Last-Modified: Mon, 14 Jul 2025 22:57:25 GMT

<html>
<head>
  <title>TryHackMe</title>
</head>
<body>
  Viewing Blog article 1 THM{YOU_FOUND_THE_BLOG}
</body>
</html>
```

THM Browser

Viewing Blog article 1 THM{YOU_FOUND_THE_BLOG}

➤ THM{YOU_FOUND_THE_BLOG}

❖ Make a DELETE request to /user/1 page

DELETE ▾

http://tryhackme.com/user/1

⚙

Go

DELETE /user/1 HTTP/1.1
Host: tryhackme.com
User-Agent: Mozilla/5.0 Firefox/87.0
Content-Length: 0

Response

HTTP/1.1 200 Ok
Server: nginx/1.15.8
Mon, 14 Jul 2025 23:0:14 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 250
Last-Modified: Mon, 14 Jul 2025 23:0:14 GMT

<html>
<head>
 <title>TryHackMe</title>
</head>
<body>
 The user has been deleted THM{USER_IS_DELETED}
</body>
</html>

THM Browser


The user has been deleted THM{USER_IS_DELETED}


➤ THM{USER_IS_DELETED}

- ❖ Make a PUT request to /user/2 page with the username parameter set to admin
Note: Use the gear button on the right to manage body parameters

Parameters ×

Here, you can add key-value pairs to customize your requests.

: 

PUT ▼ 

PUT /user/2 HTTP/1.1
Host: tryhackme.com
User-Agent: Mozilla/5.0 Firefox/87.0
Content-Length: 14
Content-Type: application/x-www-form-urlencoded

username=admin

Response

HTTP/1.1 200 Ok
Server: nginx/1.15.8
Mon, 14 Jul 2025 23:2:50 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 251
Last-Modified: Mon, 14 Jul 2025 23:2:50 GMT

```
<html>
<head>
  <title>TryHackMe</title>
</head>
<body>
  Username changed to admin THM{USER_HAS_UPDATED}
</body>
</html>
```

THM Browser

Username changed to admin THM{USER_HAS_UPDATED}

➤ THM{USER_HAS_UPDATED}

- ❖ Make a POST request to /login page with the username of thm and a password of letmein

Note: Use the gear button on the right to manage body parameters

Parameters

Here, you can add key-value pairs to customize your requests.

username	:	thm	
password	:	letmein	

[+ Add Param](#)

[Save](#)

POST

http://tryhackme.com/login

Go

POST /login HTTP/1.1
Host: tryhackme.com
User-Agent: Mozilla/5.0 Firefox/87.0
Content-Length: 29
Content-Type: application/x-www-form-urlencoded

username=thm&password=letmein

Response

HTTP/1.1 200 Ok
Server: nginx/1.15.8
Mon, 14 Jul 2025 23:6:26 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 256
Last-Modified: Mon, 14 Jul 2025 23:6:26 GMT

```
<html>
<head>
  <title>TryHackMe</title>
</head>
<body>
  You logged in! Welcome Back THM{HTTP_REQUEST_MASTER}
</body>
</html>
```

THM Browser

You logged in! Welcome Back THM{HTTP_REQUEST_MASTER}

➤ THM{HTTP_REQUEST_MASTER}