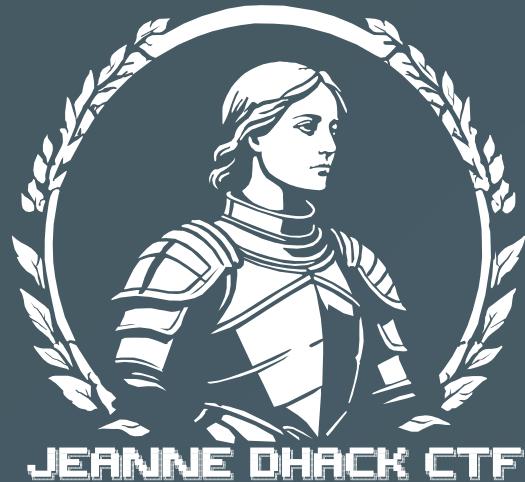


Introduction au pentest Web

Jeanne d'Hack CTF - 2025



loukabvn

Qu'est ce que le pentest ?

Le pentest, contraction de "penetration tests" en anglais, est une manière de tester la sécurité d'une application, d'une infrastructure ou d'un réseau en simulant une attaque.

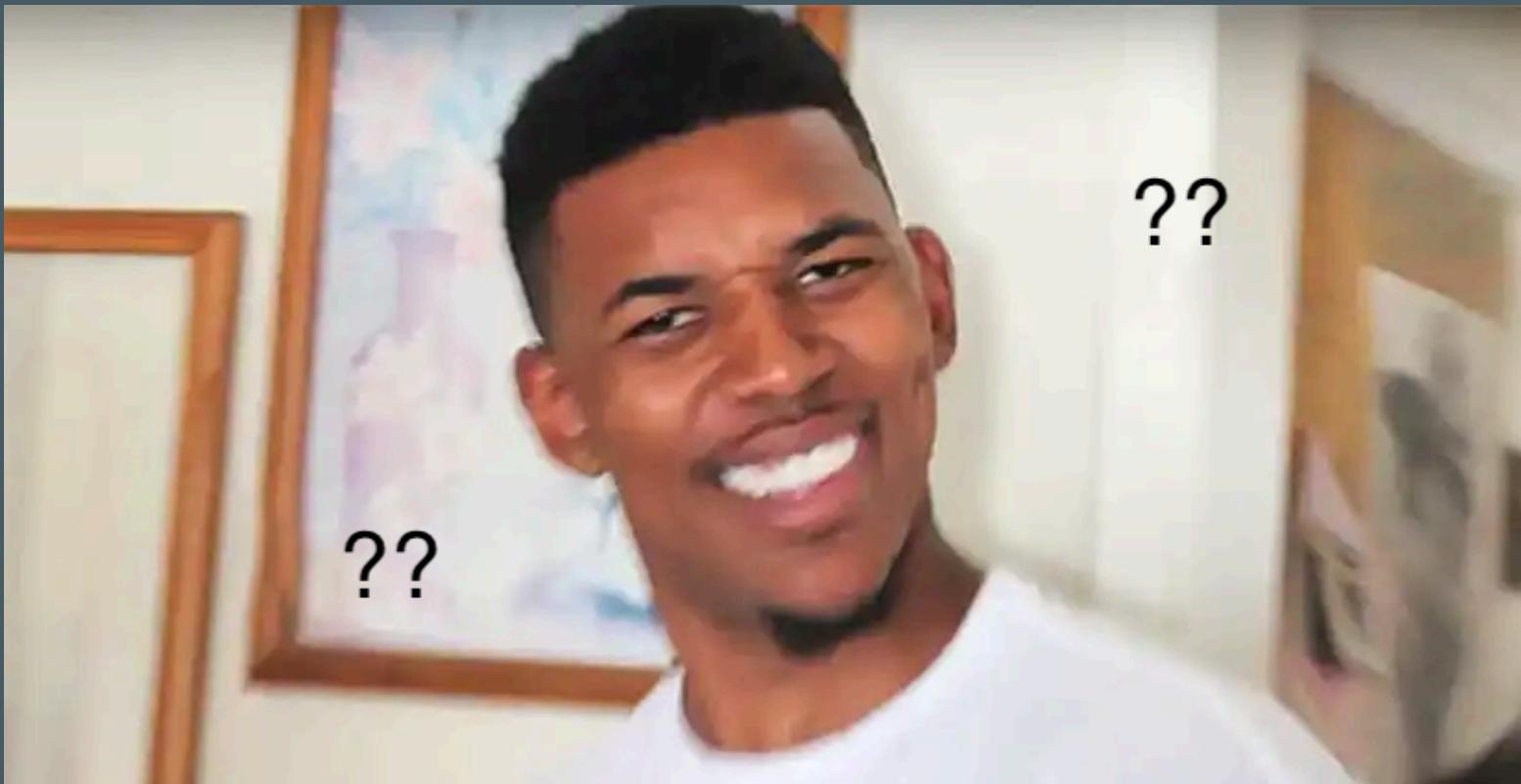
Introduction



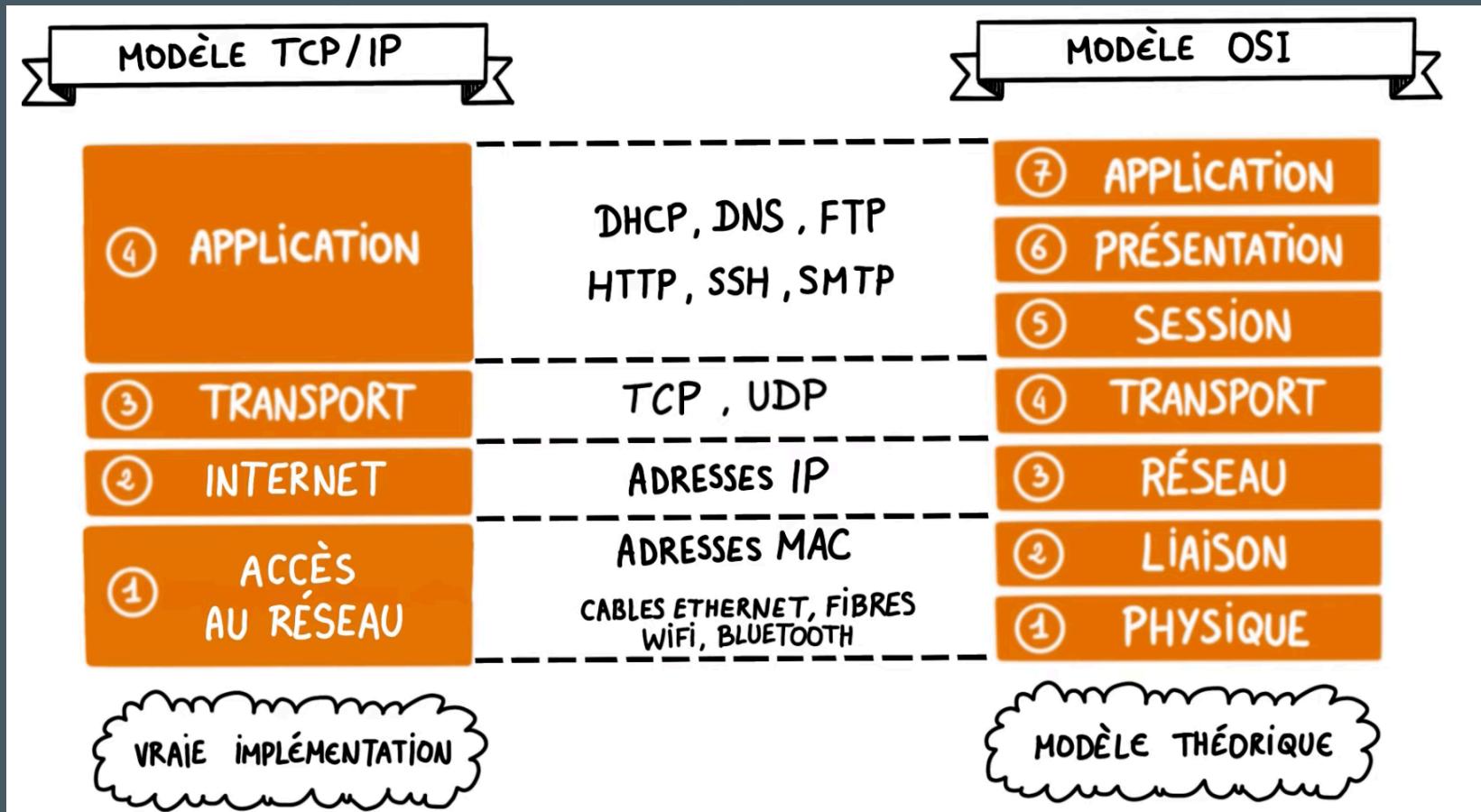
Introduction



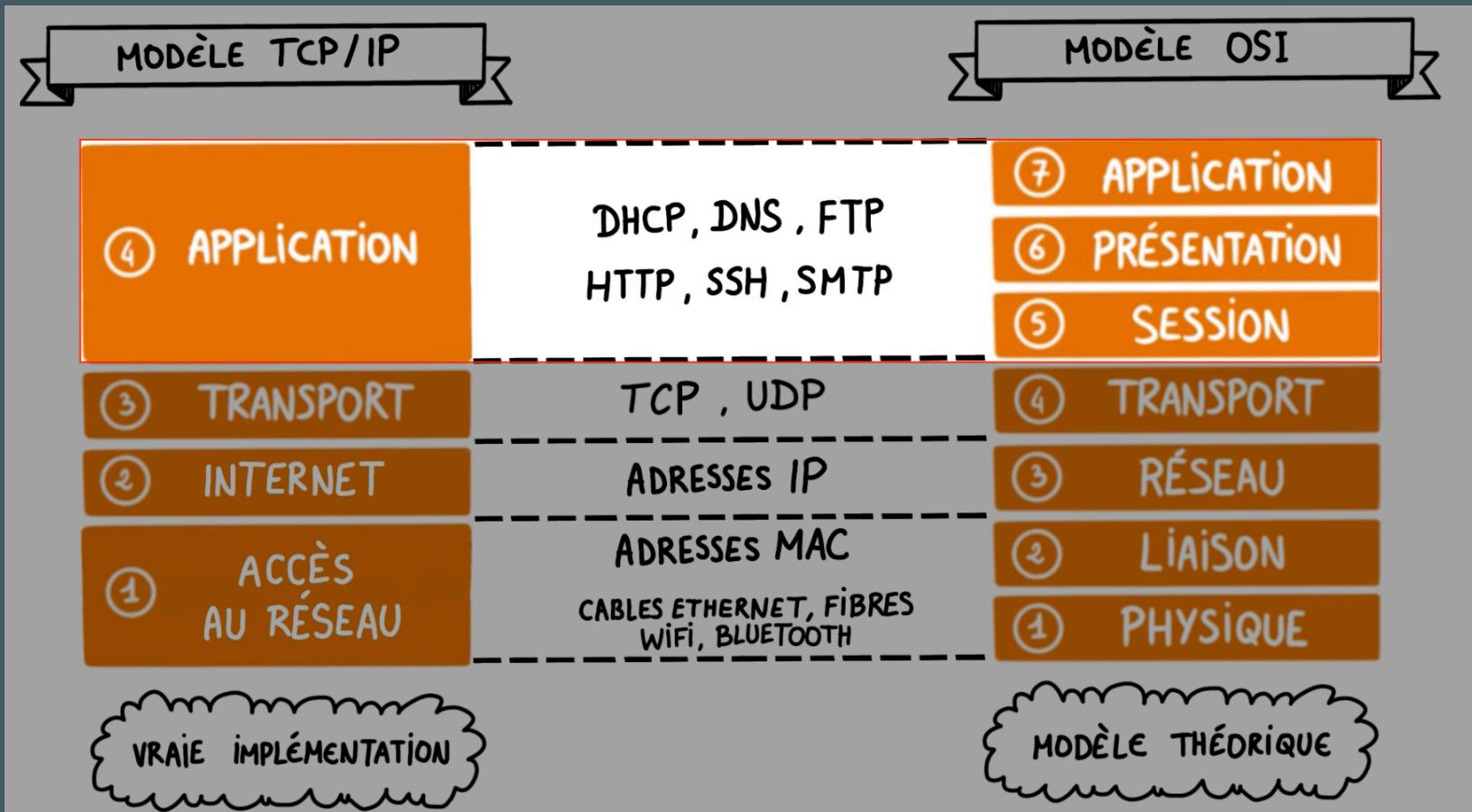
Comment ça marche le Web ?



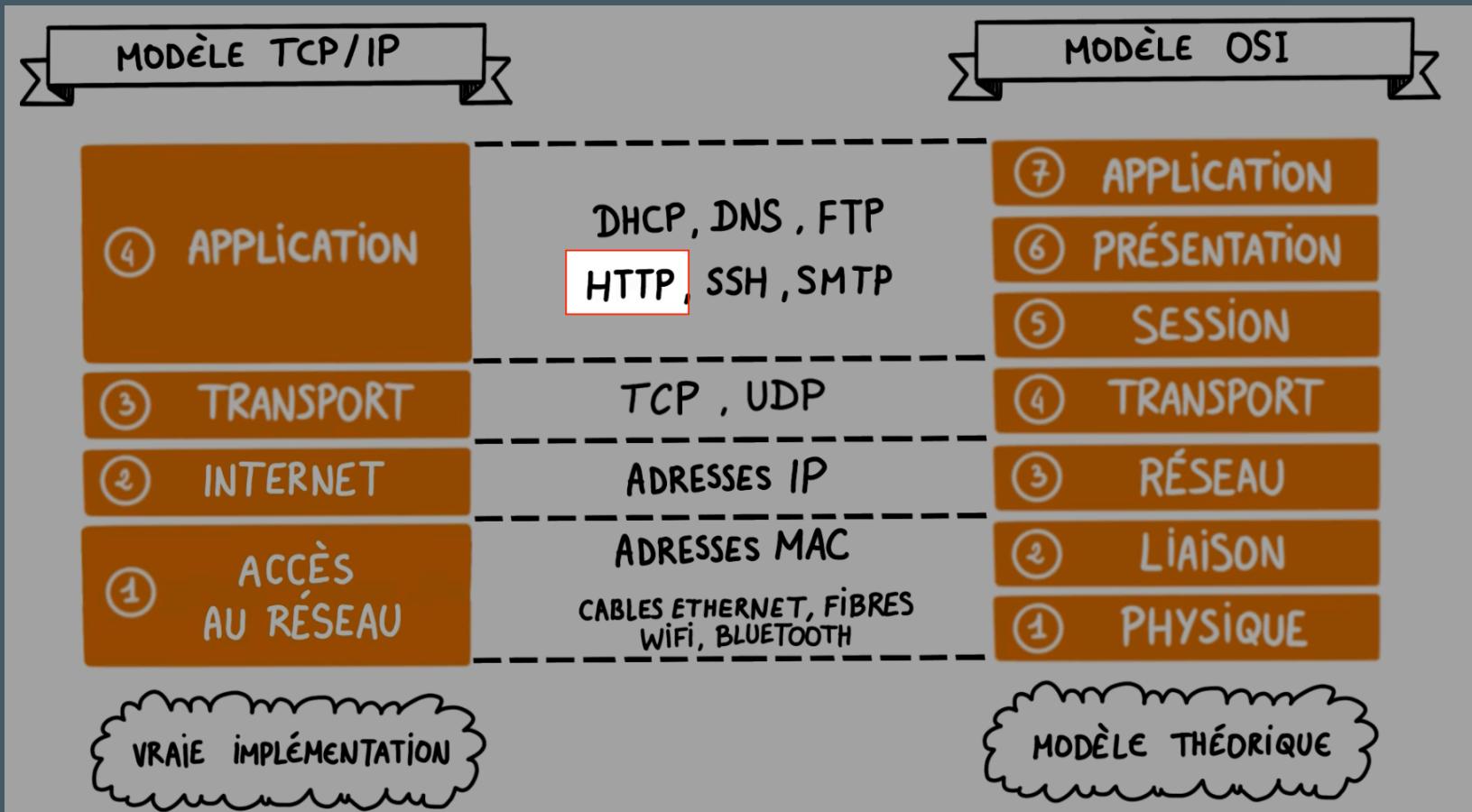
Comment ça marche le Web ?



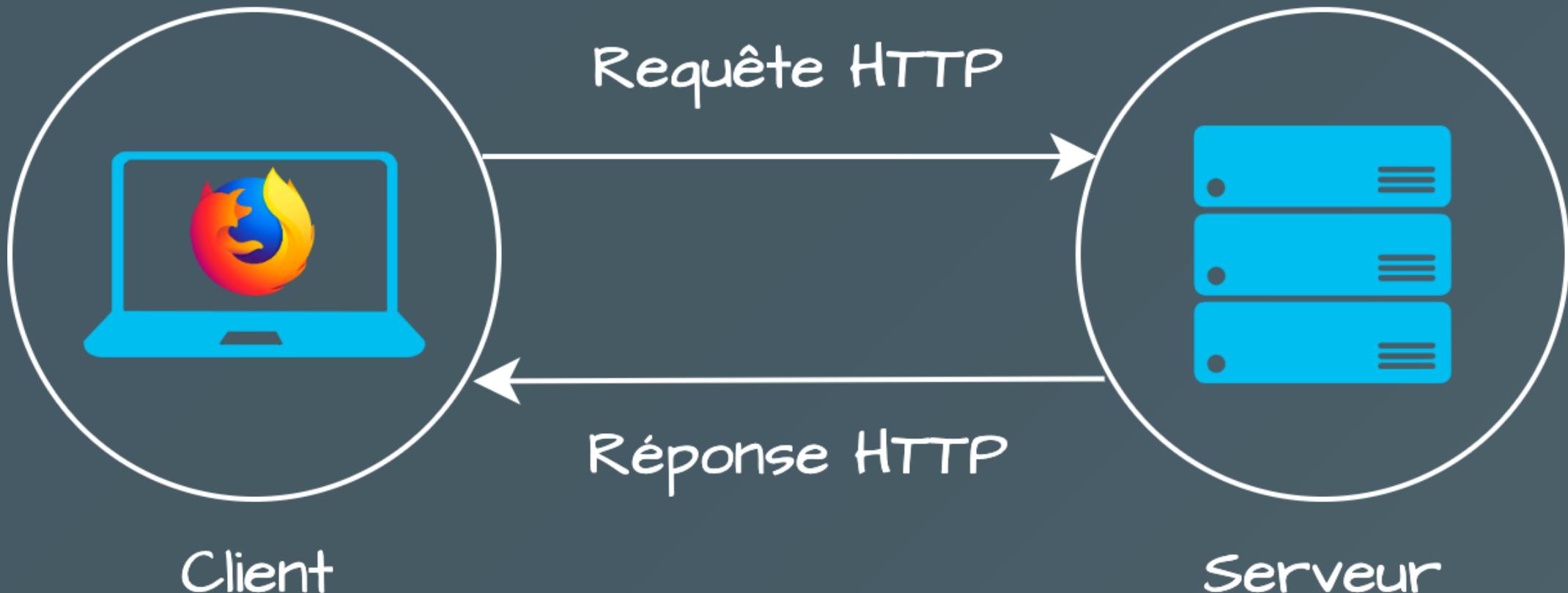
Comment ça marche le Web ?



Comment ça marche le Web ?



Comment ça marche le Web ?



Comment ça marche le Web ?

	Request	Méthode HTTP	Chemin	Version du protocole	
1	Pretty Raw Hex	GET	/wiki/World_Wide_Web	HTTP/2	Nom du serveur
2					
3					
4					
5					
6					En-têtes HTTP
7					
8					

```
1 GET /wiki/World_Wide_Web HTTP/2
2 Host: fr.wikipedia.org
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:134.0)
   Gecko/20100101 Firefox/134.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
```

Comment ça marche le Web ?

https://fr.wikipedia.org/wiki/World_Wide_Web

PwnFox-orange

WIKIPÉDIA L'encyclopédie libre

Rechercher sur Wikipédia Rechercher

Faire un don

World Wide Web

149 langues

Sommaire masquer

Début

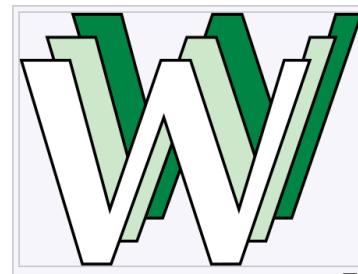
- > Terminologie
- > Histoire
- > Architecture
- > Types de ressource
- > Conception
- > Technologies
- Bibliographie
- > Voir aussi
- > Notes et références

Article Discussion Lire Modifier Modifier le code Voir l'historique Outils

Ne doit pas être confondu avec [Internet](#).
Pour le premier navigateur web, voir [WorldWideWeb \(navigateur\)](#). Pour le consortium, voir [World Wide Web Consortium](#).
Pour les articles homonymes, voir [Web \(homonymie\)](#).

Le **World Wide Web** (/wɜːld wɛd ˈweb/^a), abrégé en **WWW**, **W3**, le **Web**, la **toile mondiale** ou simplement la **toile**¹, est un système **hypertexte** public fonctionnant sur [Internet](#). Le Web permet de consulter, à l'aide d'un [navigateur](#), des [pages](#) regroupées en [sites](#). *Web* signifie littéralement « toile (d'araignée) », image représentant les [hyperliens](#) qui lient les pages web entre elles^b.

Le Web est une des applications d'Internet², qui est distincte d'autres applications comme le [courrier électronique](#), la [visioconférence](#) et le [partage de fichiers en pair à pair](#). Inventé en 1989-1990 au [CERN](#) par [Tim Berners-Lee](#) épaulé de [Robert Cailliau](#), le Web a popularisé Internet³. Depuis, le Web est fréquemment confondu avec Internet⁴ ; en particulier, le mot *toile* est souvent utilisé dans les textes non techniques sans que l'on sache toujours si l'auteur désigne



Logo historique du World Wide Web par Robert Cailliau.

Comment réaliser un pentest Web ?

En résumé



En résumé



Plusieurs approches possibles :



Toutes les informations disponibles
(ex. code source, documentation...)

Informations limitées sur le périmètre
(ex. un ou plusieurs compte fournis)

Aucune connaissance du périmètre
(ex. URL d'une application Web)

Les différents étapes

- Collecte d'informations passive
- Reconnaissance active
- Recherche/analyse de vulnérabilités
- Post-exploitation
- Rapport 😊

Collecte d'informations passive

- Recueillir un maximum d'informations sur le client, l'application, les technologies utilisés...
- Cf. cours d'OSINT (rendez-vous ici à 18h 😊)

Reconnaissance active

- Scan réseau de l'infrastructure (`nmap`, créé en 1997 mais toujours ton meilleur ami 😊)
- Identification des technologies et versions (`Wappalyzer`, en-têtes HTTP, extensions des fichiers)
- Fuzzing de l'application (`ffuf`, `wfuzz`, `gobuster` ...)
- ⚠️ Interdit (et inutile) d'utiliser des outils de fuzzing aujourd'hui !

Recherche/analyse de vulnérabilités (exemples)

La référence pour les vulnérabilités Web : OWASP Top 10
(<https://owasp.org/www-project-top-ten/>)

- Injections dans les champs (SQLi, XSS, SSTI...), ou manque de validation (LFI/RFI, Path traversal...)
- Contournement de l'authentification ou mauvaise gestion de la session
- Défaut de contrôle d'accès
- Dépôt de fichiers non contrôlé (File upload)

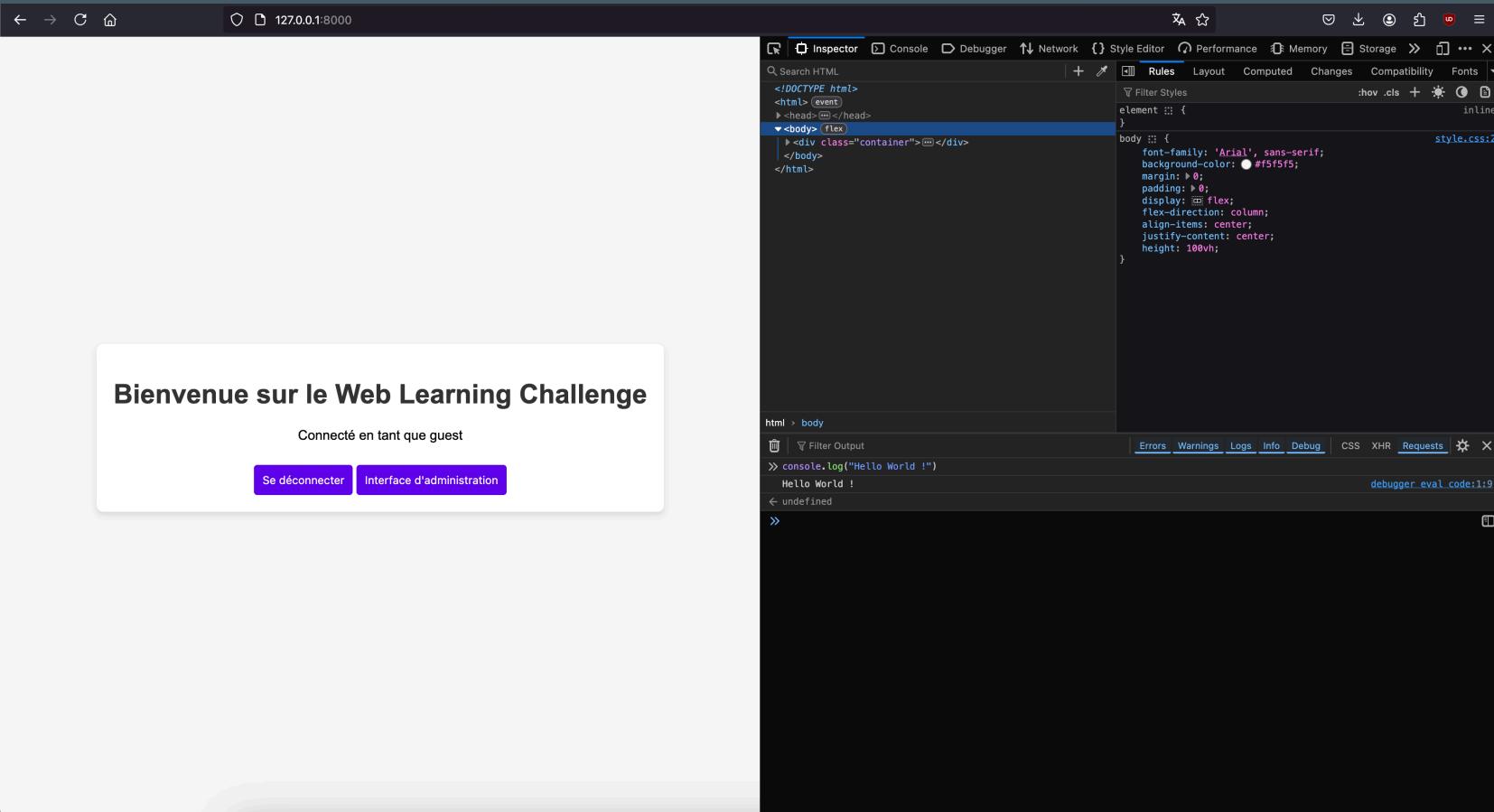
Post-exploitation

Après avoir identifié une vulnérabilité, on va essayer de l'exploiter afin d'évaluer le risque pour l'application

- Récupérer les comptes utilisateurs via une injection SQL
- Élever ses privilèges avec un défaut de contrôle d'accès
- Exécuter des commandes sur le serveur à partir d'un file upload arbitraire

Quels sont les outils ?

Outils



Burp Suite

The screenshot shows the Burp Suite interface with the following details:

- HTTP history tab:** Shows a list of captured requests and responses. The first request (POST /login) is highlighted.
- Request pane:** Displays the raw POST request sent to the server. The body contains the parameters: `username=user&password=password`.
- Response pane:** Displays the raw HTTP response received from the server. The response code is 200 OK, and the page content includes a form for login with fields for username and password.
- Inspector pane:** On the right, there are several tabs for inspecting the request and response: Request attributes, Request body parameters, Request cookies, Request headers, and Response headers.
- Bottom status bar:** Shows "Event log (1)" and "All issues (4)".
- System status:** Memory usage is shown as 224.2MB.

<https://portswigger.net/burp/communitydownload>



Quelques ressources :

- Hacktricks : <https://book.hacktricks.wiki/en/index.html>
- PayloadAllTheThings :
<https://swisskyrepo.github.io/PayloadsAllTheThings/>

Et de quoi s'entraîner :

- HackTheBox : <https://app.hackthebox.com/login>
- PortSwigger : <https://portswigger.net/web-security/learning-paths>
- RootMe : <https://www.root-me.org/>

Démo time ! 

Merci à tous !
Questions ?