

## Práctica 4: Fingerprinting de tecnologías con Wappalyzer y Nmap NSE

**Objetivo:** Identificar tecnologías, versiones y configuraciones activas en servicios web mediante fingerprinting pasivo (Wappalyzer) y activo (Nmap con NSE scripts). El objetivo es detectar frameworks, CMS, versiones de servidor, certificados SSL, etc.

### 1. Entorno virtualizado recomendado

#### Máquina atacante

- **SO:** *Kali Linux 2025.2*
  - **Herramientas:** Wappalyzer (extensión o CLI), Nmap con scripts NSE, WhatWeb (alternativa CLI), Firefox o Chromium

#### Máquina objetivo (local simulada)

- IP: 192.168.56.101
- Dominio simulado: `webapp.example-corp.com`
- Sistema: *Ubuntu Server*
- Servicios:
  - *Apache2 o Nginx*
  - *WordPress o Laravel*
  - *OpenSSL con certificado autofirmado*

### 2. Instalación y preparación

Instalación en máquina OBJETIVO: *Ubuntu Server 24.04*

#### Actualizar repositorios y paquetes

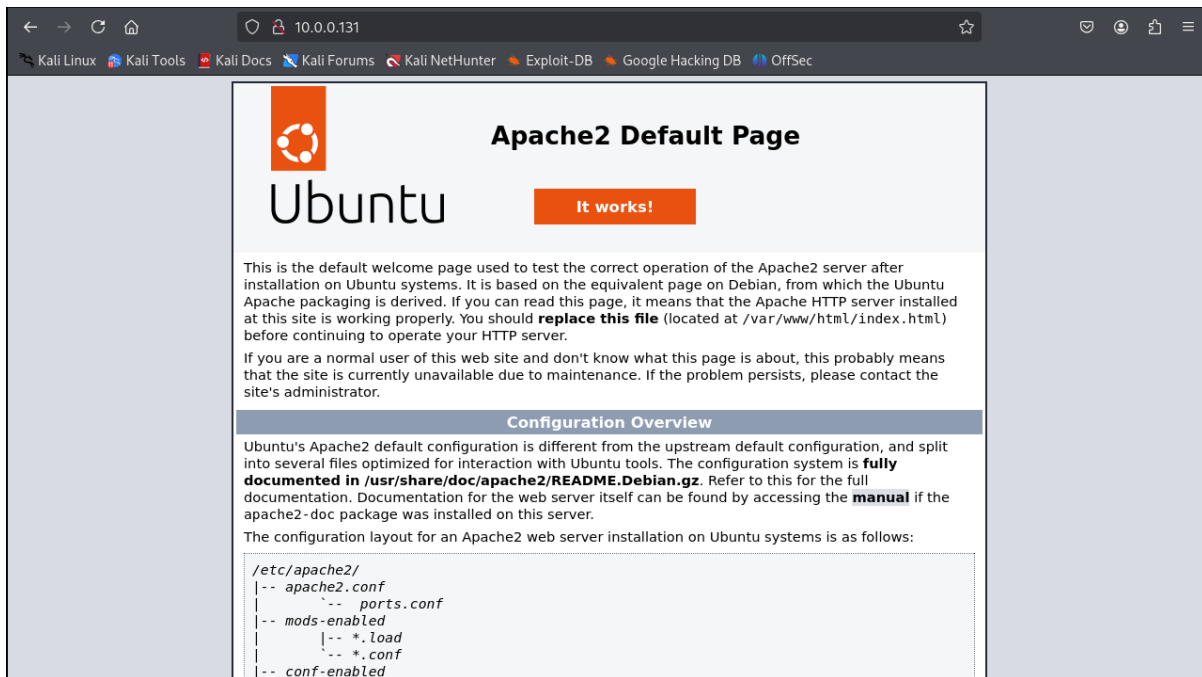
```
sudo apt update && sudo apt upgrade -y
```

#### Instalar servidor web

##### Opción A: Apache2

```
sudo apt install apache2 -y  
sudo systemctl enable apache2  
sudo systemctl start apache2
```

Acceder en navegador: `http://<IP_SERVIDOR>` Se debería ver la página por defecto de Apache.



## Opción B: Nginx (alternativa)

```
sudo apt install nginx -y
sudo systemctl enable nginx
sudo systemctl start nginx
```

## 3. Instalar PHP y módulos necesarios

```
sudo apt install php libapache2-mod-php php-mysql php-xml php-mbstring unzip wget curl -y
```

Verifica versión:

```
php -v
```

```
ubuntu@ubuntu-server:~$ php -v
PHP 8.3.6 (cli) (built: Jul 14 2025 18:30:55) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
ubuntu@ubuntu-server:~$
```

## 4. Instalar y configurar MySQL

```
sudo apt install mysql-server -y
sudo systemctl enable mysql
sudo systemctl start mysql
```

Configurar seguridad inicial:

```
sudo mysql_secure_installation
```

- Establece contraseña de root.
- Elimina usuarios anónimos.
- Desactiva acceso remoto root.
- Elimina DB de prueba.

Crear base de datos y usuario para WordPress:

```
sudo mysql -u root -p
```

Dentro del cliente:

```
CREATE DATABASE wordpress_db;
CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'StrongPassword123!';
GRANT ALL PRIVILEGES ON wordpress_db.* TO 'wp_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

## 5. Instalar WordPress

```
cd /tmp
wget https://wordpress.org/latest.zip
unzip latest.zip
sudo mv wordpress /var/www/html/webapp
```

Asignar permisos correctos:

```
sudo chown -R www-data:www-data /var/www/html/webapp
sudo chmod -R 755 /var/www/html/webapp
```

Configurar archivo de Apache:

```
sudo nano /etc/apache2/sites-available/webapp.conf
```

Ejemplo de configuración:

```
<VirtualHost *:80>
    ServerName webapp.example-corp.com
    DocumentRoot /var/www/html/webapp
```

```
<Directory /var/www/html/webapp>
    AllowOverride All
</Directory>

ErrorLog ${APACHE_LOG_DIR}/webapp_error.log
CustomLog ${APACHE_LOG_DIR}/webapp_access.log combined
</VirtualHost>
```

Habilitar sitio y módulos:

```
sudo a2ensite webapp.conf
sudo a2enmod rewrite
sudo systemctl reload apache2
```

## 6. Crear archivo de configuración de WordPress

```
cd /var/www/html/webapp
cp wp-config-sample.php wp-config.php
nano wp-config.php
```

Modifica estas líneas con tu DB:

```
define( 'DB_NAME', 'wordpress_db' );
define( 'DB_USER', 'wp_user' );
define( 'DB_PASSWORD', 'StrongPassword123!' );
define( 'DB_HOST', 'localhost' );
```

## 7. Configurar hosts en Kali Linux - máquina atacante

Editar el archivo:

```
sudo nano /etc/hosts
```

Añadir línea, modificando la IP según tu caso:

```
192.168.56.101    webapp.example-corp.com
```

## 8. Configurar SSL con certificado autofirmado

Generar certificado:

```
sudo mkdir /etc/ssl/webapp
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/ssl/webapp/webapp.key \
    -out /etc/ssl/webapp/webapp.crt
```

Configurar VirtualHost SSL:

```
sudo nano /etc/apache2/sites-available/webapp-ssl.conf
```

Ejemplo:

```
<VirtualHost *:443>
    ServerName webapp.example-corp.com
    DocumentRoot /var/www/html/webapp

    SSLEngine on
    SSLCertificateFile /etc/ssl/webapp/webapp.crt
    SSLCertificateKeyFile /etc/ssl/webapp/webapp.key

    <Directory /var/www/html/webapp>
        AllowOverride All
    </Directory>
</VirtualHost>
```

Habilitar SSL:

```
sudo a2enmod ssl
sudo a2ensite webapp-ssl.conf
sudo systemctl reload apache2
```

## 9. Verificación final

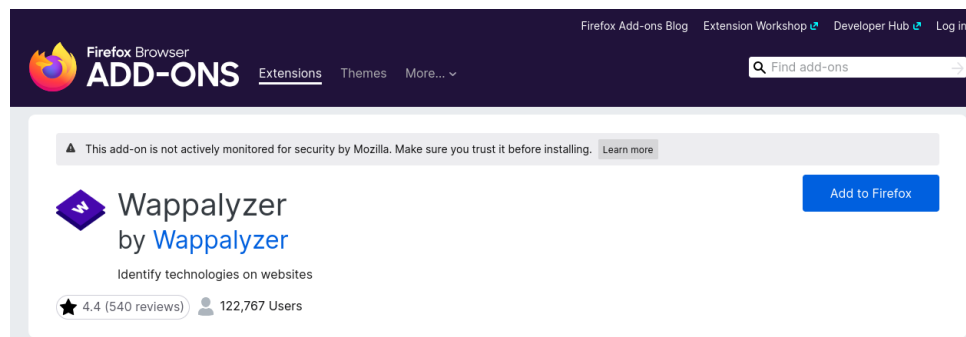
En navegador desde Kali:

- o <http://webapp.example-corp.com> se obtiene el instalador de WordPress.
- o <https://webapp.example-corp.com> muestra la advertencia de certificado autofirmado.

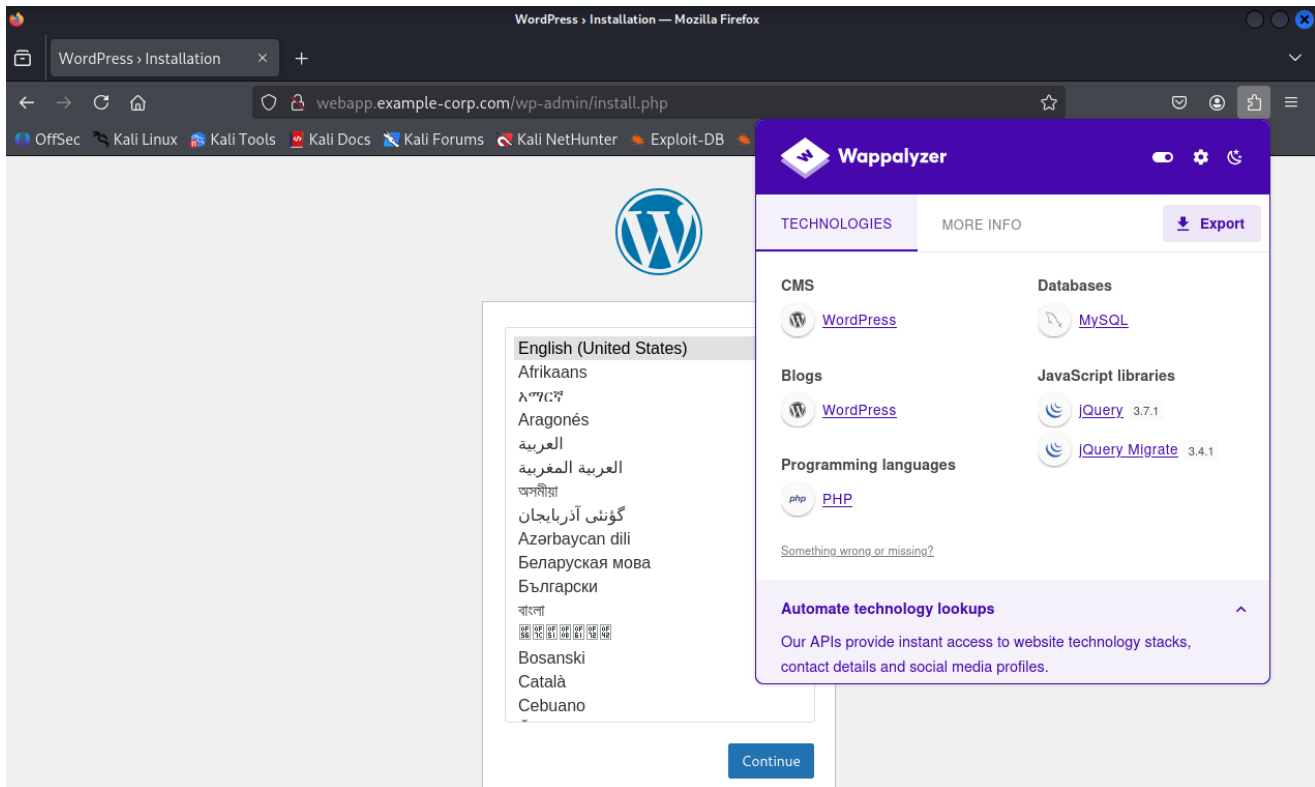
## 10. Herramientas de Fingerprinting

### A) Wappalyzer (navegador)

Instalar la extensión para Firefox o Chromium: <https://www.wappalyzer.com/>



Acceder a <http://webapp.example-corp.com>



Observar tecnologías detectadas: CMS, librerías JS, frameworks, etc.

## B) WhatWeb (alternativa CLI)

```
whatweb -a 3 http://webapp.example-corp.com
```

```
(kali㉿kali)-[~/webanalyze]
$ whatweb http://webapp.example-corp.com
http://webapp.example-corp.com/ [302 Found] Apache[2.4.58], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.58 (Ubuntu)], IP[10.0.0.131], RedirectLocation[http://webapp.example-corp.com/wp-admin/install.php], UncommonHeaders[x-redirect-by]
http://webapp.example-corp.com/wp-admin/install.php [200 OK] Apache[2.4.58], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.58 (Ubuntu)], IP[10.0.0.131], JQuery[3.7.1], PHP, Script[text/javascript], Title[WordPress &rsquo; Install
ation], WordPress

(kali㉿kali)-[~/webanalyze]
$
```

Ejecuta **WhatWeb** (*herramienta de fingerprinting web*) con el **nivel de agresividad 3** contra el dominio de tu laboratorio. ¿Qué significa **nivel de agresividad 3**?

- ❖ whatweb: herramienta que identifica **CMS, frameworks, librerías, servidores, versiones** y otras tecnologías web a partir de firmas en HTML, cabeceras HTTP, cookies, etc.
- ❖ -a 3: define el **nivel de agresividad**.

Los niveles van de 1 a 4:

Nivel	Significado	Comportamiento
-a 1	<b>Pasivo</b>	Solo analiza contenido HTML ya recibido, sin enviar peticiones extra.
-a 2	<b>Predeterminado</b>	Incluye análisis pasivo + algunas comprobaciones adicionales.
-a 3	<b>Agresivo</b>	Hace más pruebas activas: sigue redirecciones, solicita archivos comunes (robots.txt, favicon, etc.) y más cabeceras.
-a 4	<b>Muy agresivo</b>	Puede enviar aún más peticiones intrusivas (no recomendado fuera de entornos de prueba).

## D) Nmap NSE scripts

```
nmap -sV --script=http-enum,http-title,http-headers -p 80,443 webapp.example-corp.com
```

- ❖ **nmap** herramienta de escaneo de red.
- ❖ **-sV** activa la **detección de versiones** de servicios. Intenta identificar el software y su versión (p. ej. *Apache httpd 2.4.52*).
- ❖ **--script=http-enum,http-title,http-headers** ejecuta **scripts NSE (Nmap Scripting Engine)** relacionados con HTTP:
  - **http-enum**: intenta enumerar directorios y recursos comunes de aplicaciones web (por ejemplo */wp-admin/*, */phpmyadmin/*, etc.). Útil para descubrir la existencia de **WordPress, Joomla, etc.**
  - **http-title**: recupera el título HTML de la página principal (`<title>`), por ejemplo *"Just another WordPress site"*.
  - **http-headers**: muestra las cabeceras HTTP de respuesta (ejemplo: *Server: Apache/2.4.52 (Ubuntu), X-Powered-By: PHP/8.1*).
- ❖ **-p 80,443** escanea los puertos **80 (HTTP)** y **443 (HTTPS)**, los habituales de servidores web.
- ❖ **webapp.example-corp.com** el **objetivo**

```
(kali@kali)-[~/webanalyze]
$ nmap -sV --script=http-enum,http-title,http-headers -p 80,443 webapp.example-corp.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-11 10:00 EDT
Nmap scan report for webapp.example-corp.com (10.0.0.131)
Host is up (0.00078s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.58 ((Ubuntu))
| http-title: WordPress &rsquo; Installation
|_ Requested resource was http://webapp.example-corp.com/wp-admin/install.php
|_ http-server-header: Apache/2.4.58 (Ubuntu)
|_ http-headers:
|   Date: Thu, 11 Sep 2025 14:00:31 GMT
|   Server: Apache/2.4.58 (Ubuntu)
|   Expires: Wed, 11 Jan 1984 05:00:00 GMT
|   Cache-Control: no-cache, must-revalidate, max-age=0, no-store, private
|   Connection: close
|   Content-Type: text/html; charset=utf-8
```

```
nmap --script=ssl-cert,ssl-enum-ciphers -p 443 webapp.example-corp.com
```

- ❖ **nmap** herramienta de escaneo de red.
- ❖ **--script=ssl-cert,ssl-enum-ciphers** ejecuta **scripts NSE específicos de SSL/TLS**:
  - **ssl-cert** obtiene la información del certificado SSL del servidor:
    - CN (Common Name) y SAN (Subject Alternative Name).
    - Emisor (Issuer).
    - Fechas de validez (inicio y expiración).
    - Longitud de la clave.
    - Algoritmo de firma (ej. SHA256).
  - **ssl-enum-ciphers** enumera todos los protocolos y cifrados soportados por el servidor:
    - Protocolos: SSLv2, SSLv3, TLS 1.0, 1.1, 1.2, 1.3 (cuáles están habilitados).
    - Ciphers permitidos por protocolo.
    - Fuerza de cada cipher (fuerte, débil, inseguro).
- ❖ **-p 443** escanea el puerto **443 (HTTPS)**, el estándar para SSL/TLS.
- ❖ **webapp.example-corp.com** el **objetivo**

```
(kali㉿kali)-[~/webanalyze]
$ nmap --script=ssl-cert,ssl-enum-ciphers -p 443 webapp.example-corp.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-11 10:01 EDT
Nmap scan report for webapp.example-corp.com (10.0.0.131)
Host is up (0.00081s latency).

PORT      STATE SERVICE
443/tcp   open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM_8 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CCM (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CCM_8 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256 (dh 2048) - A
```

## 4. Actividades de evaluación

Actividad	Detalle
Informe con Wappalyzer	Screenshot o log con tecnologías identificadas
Resultado de WhatWeb o CLI	Detalle de versiones de CMS/frameworks
NSE: análisis de encabezados HTTP	http-headers, http-title, http-enum
NSE: análisis criptográfico SSL	ssl-cert, ssl-enum-ciphers