**Full Path Disclosure (FPD)**

**Descripción**

Full Path Disclosure (FPD) es una vulnerabilidad de seguridad en la cual una aplicación web revela la ruta completa (full path) del archivo del servidor donde se encuentra alojado el sitio web. Esta información puede ser utilizada por atacantes para realizar ataques más avanzados, como Directory Traversal o Local File Inclusion (LFI).

**Ejemplo**

Una aplicación web puede arrojar un mensaje de error que incluya la ruta completa del archivo que causó el error. Por ejemplo:

php

Copy code

Warning: include(file.php): failed to open stream: No such file or directory in /var/www/html/site/file.php on line 3

**Impacto**

Conocer la estructura de directorios del servidor puede ayudar a un atacante a:

* Identificar la ubicación de archivos sensibles.
* Planificar ataques de escalada de privilegios.
* Realizar ataques de Directory Traversal.

**Mitigación**

* Configurar el servidor para que no muestre errores detallados en las respuestas HTTP.
* Utilizar manejo de errores adecuado que no revele la estructura de directorios.
* Validar y sanitizar todas las entradas del usuario para evitar errores que puedan llevar a la revelación de rutas completas.

**Directory Traversal**

**Descripción**

Directory Traversal (también conocido como Path Traversal) es una vulnerabilidad que permite a los atacantes acceder a archivos y directorios que están fuera del directorio raíz de la aplicación web. Esto se logra manipulando variables que hacen referencia a archivos con secuencias como ../ para salir del directorio actual y acceder a otros directorios.

**Ejemplo**

Un atacante podría explotar una vulnerabilidad de Directory Traversal con una URL como la siguiente:

bash

Copy code

http://example.com/view.php?file=../../etc/passwd

Esta URL intentaría acceder al archivo /etc/passwd en un sistema Unix/Linux.

**Impacto**

* Acceso a archivos sensibles como /etc/passwd o archivos de configuración de la aplicación.
* Exposición de información sensible que puede ser utilizada para comprometer aún más el sistema.

**Mitigación**

* Validar y sanitizar todas las entradas del usuario.
* Utilizar funciones del sistema operativo que no permiten la navegación fuera del directorio raíz de la aplicación.
* Configurar el servidor para restringir el acceso a directorios específicos.

**Ejemplo de Mitigación en PHP**

php

Copy code

<?php

// Lista blanca de archivos permitidos

$whitelist = ['about.html', 'contact.html', 'home.html'];

// Obtener el archivo solicitado por el usuario

$file = $\_GET['file'];

// Validar si el archivo está en la lista blanca

if (in\_array($file, $whitelist)) {

include($file);

} else {

echo "Archivo no permitido.";

}

?>

En este ejemplo, solo se permiten archivos específicos y seguros para ser incluidos, evitando así que un atacante pueda manipular la ruta para acceder a archivos sensibles del sistema.

1. Full Path Disclosure: Es una vulnerabilidad que permite conocer la ruta completa de algún archivo en particular dentro de una aplicación web. Esto se provoca generando errores en la aplicación web. Estos errores se pueden provocar con distintos métodos como por ejemplo ingresar datos no esperados dentro de algún formulario o un parámetro en el URL. Tambien hay distintas vulnerabilidades que permiten generar errores para ver la ruta completa, entre ellas el LFI, RFI, SQL Injection y algunas otras. Full Path Disclosure traducido al español es : Divulgación de Ruta Completa.
2. LFI permite cargar archivos y los ejecuta, lo cual permite ejecutar comando y acciones en el sistema operativo del servidor.
3. Directory Traversal permite cargar archivos pero no los ejecuta, solo carga su contenido, aunque con otras técnicas es posible provocar que se ejecuten comandos.