**Informe Seg en App Web Vulnerabilidad Directory Trasversal**

La vulnerabilidad Directory Traversal (o File Path Traversal) permite a un atacante acceder a archivos sensibles en el servidor, como configuraciones del sistema operativo, credenciales y datos de la aplicación. En algunos casos, también es posible escribir en archivos del servidor, modificando su comportamiento y comprometiendo su seguridad.

**Diagrama

Descripción generada automáticamente**

**Descripción del Caso y Análisis**

1. **Identificación de la Vulnerabilidad:**
   * Un caso común ocurre cuando una aplicación permite cargar archivos mediante un parámetro como filename sin validar el input del usuario.
   * Ejemplo:

<img src="/loadImage?filename=218.png">

* + Un atacante podría manipular el parámetro para leer archivos sensibles:

https://insecure-website.com/loadImage?filename=../../../etc/passwd

1. **Pruebas Realizadas:**
   * Se ejecutó un ataque de prueba para leer el archivo /etc/passwd en un sistema Unix, demostrando la vulnerabilidad.
   * Se usaron herramientas como DotDotPwn para automatizar pruebas y evaluar posibles rutas de explotación.

**Metodología OSSEM y Mapeo MITRE ATT&CK**

1. **OSSEM:**

|  |  |
| --- | --- |
| **Componente** | **Detalle** |
| Fuentes de Datos | Logs del servidor web, rutas de archivos accedidas |
| Indicadores | Rutas de archivos inusuales, intentos de acceso a directorios superiores |
| Acciones | Explotación de secuencias de ruta como ../ para acceder a archivos no deseados |

1. **MITRE ATT&CK:**

|  |  |  |
| --- | --- | --- |
| **Táctica** | **Técnica** | **ID** |
| Initial Access | Exploit Public-Facing Application | T1190 |
| Discovery | File and Directory Discovery | T1083 |
| Impact | Data Manipulation | T1565 |

**Medidas de Mitigación**

* **Validación del Input:**
  + Bloquear caracteres como ../ o ..\ y sanitizar las entradas del usuario.
* **Canonicalización de Rutas:**
  + Usar funciones como getCanonicalPath() para validar las rutas antes de acceder a los archivos.
* **Restricción de Acceso:**
  + Limitar el acceso a directorios específicos y utilizar listas blancas para nombres de archivo.

**Medidas para Prevenirlo**:

Evitar pasar la entrada proporcionada por el usuario a las API del sistema de archivos de manera directa.

● Muchas funciones de la aplicación que hacen esto se pueden reescribir para ofrecer el mismo comportamiento de una manera más segura.

● Si se considera inevitable pasar la entrada proporcionada por el usuario a las API del sistema de archivos, entonces se deben usar dos capas de defensa juntas para prevenir ataques:

○ Validar la entrada del usuario antes de procesarla contra una lista blanca de valores permitidos o “sanitizarla”

○ Luego usar una API del sistema de archivos para canonicalizar la ruta y verificar que la ruta canonicalizada comience con el directorio base esperado