INTRODUCCIÓN

Resumen del objetivo y alcance del ejercicio:

El objetivo principal de este ejercicio fue demostrar la explotación de vulnerabilidades web comunes en un entorno controlado utilizando la plataforma "Damn Vulnerable Web App" (DVWA). El alcance se centró en dos tipos específicos de vulnerabilidades: Inyección de Comandos (Command Injection) y Cross-Site Scripting (XSS) Reflejado. El propósito fue entender cómo estas fallas permiten a un atacante ejecutar comandos del sistema operativo en el servidor o ejecutar código malicioso en el navegador de un usuario, respectivamente, y comprometer la seguridad de la aplicación.

METODOLOGÍA

Herramientas y técnicas utilizadas:

Entorno de Pruebas: DVWA (Damn Vulnerable Web App) configurada en nivel de seguridad "Low" (Bajo).

Herramientas: Navegador web estándar. No se requieren herramientas externas automatizadas para estas explotaciones.

Técnicas de Explotación:

Validación y explotación manual a través de campos de entrada de la aplicación web.

Encadenamiento de comandos de sistema operativo,usando el delimitador ; para Linux.

Inyección de código JavaScript usando etiquetas <script> para XSS.

RESULTADOS

Detalles de las vulnerabilidades explotadas:

A.Inyección de Comandos (Command Injection)

Ubicación: Módulo "Command Injection" de DVWA.

La aplicación presenta una funcionalidad de "ping" que toma una dirección IP ingresada por el usuario y la pasa directamente a una terminal del sistema operativo sin la debida validación o sanitización.

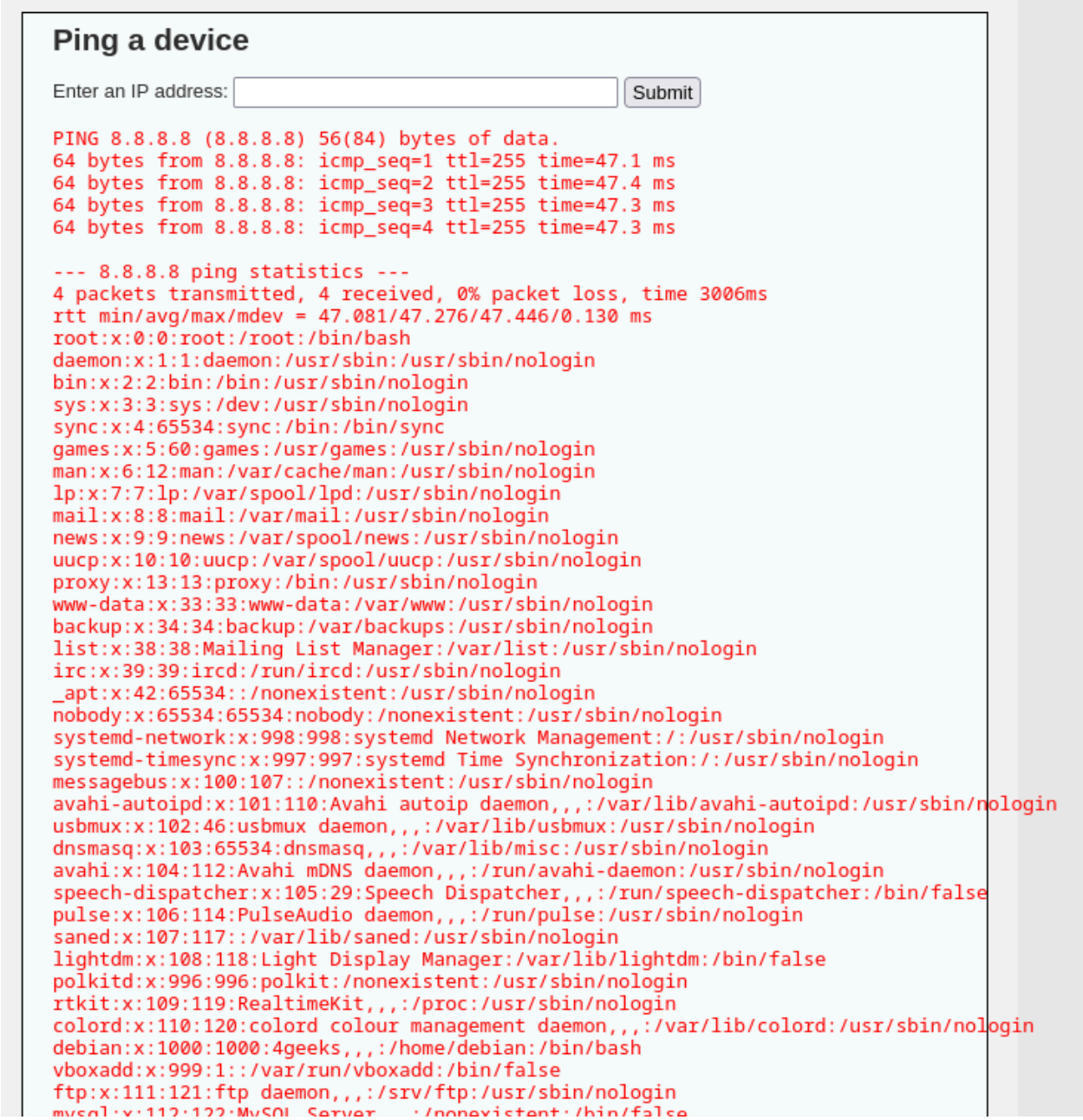
Comandos utilizados para la explotación:

8.8.8.8; whoami: Para confirmar la ejecución de comandos y descubrir el usuario actual del servidor web (resultado esperado: www-data).

8.8.8.8; ls -la: Para listar los archivos en el directorio actual del servidor.

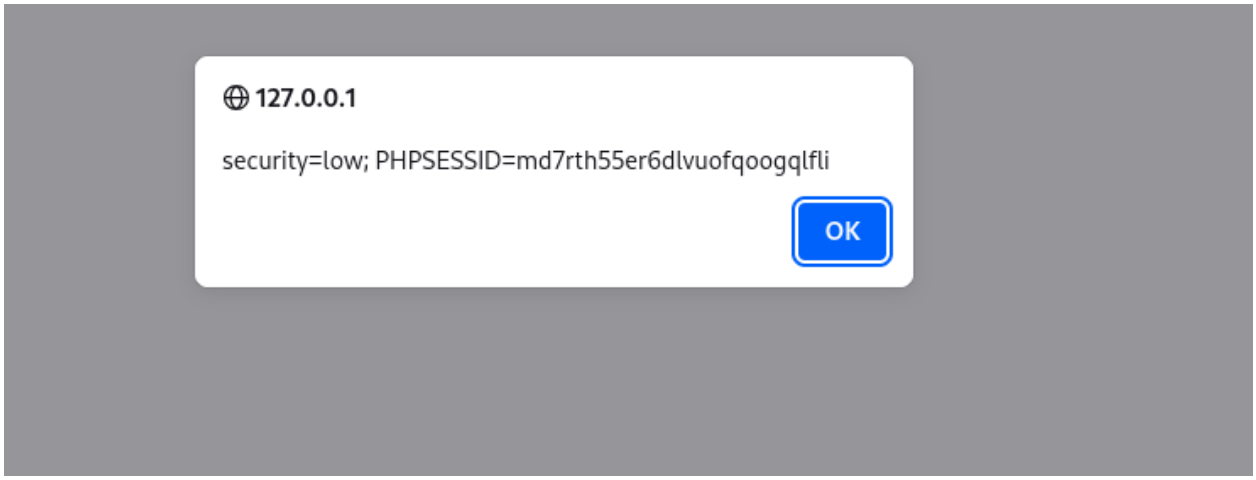
8.8.8.8; cat /etc/passwd: Para leer un archivo sensible del sistema que lista los usuarios.

Evidencias: Al inyectar los comandos después de la dirección IP separada por un punto y coma, la aplicación devolvió la salida estándar del comando ping seguida inmediatamente por la salida de los comandos inyectados (whoami, listado de directorios, contenido de /etc/passwd), confirmando la ejecución remota de comandos.



B. Cross-Site Scripting (XSS) Reflejado
Ubicación: Módulo "XSS (Reflected)" de DVWA.
Descripción: La aplicación solicita el nombre del usuario y lo refleja en la pantalla "Hello [nombre]" sin codificar la salida, permitiendo que el navegador interprete el texto ingresado como código HTML/JavaScript.

Codigos utilizados para la explotación:
<script>alert('¡Hackeado!');</script>: Prueba de concepto básica para verificar la ejecución de JavaScript (genera una ventana emergente).
<script>alert(document.cookie);</script>: Payload para demostrar el impacto real, mostrando la cookie de sesión del usuario en una ventana emergente.



Evidencias: Al enviar el payload, el navegador no mostró el texto literal, sino que ejecutó el script incrustado, resultando en la aparición de ventanas emergentes (pop-ups) con el mensaje personalizado o la información de la cookie de sesión (ej. PHPSESSID=...).

MITIGACIÓN

Propuestas para remediar las vulnerabilidades explotadas:

A. Para Inyección de Comandos:

Principio Fundamental: NUNCA confiar en la entrada del usuario.

Validación de Entrada: Implementar validaciones estrictas (ej. usar expresiones regulares) para asegurar que la entrada solo contenga una dirección IP válida.

Sanitización: Limpiar la entrada eliminando caracteres peligrosos utilizados para encadenar comandos (como ;, &, |).

Funciones Seguras: Utilizar funciones específicas del lenguaje que manejen argumentos de forma segura, como escapeshellarg() en PHP, para tratar la entrada del usuario como una cadena de texto y no como un comando ejecutable.

B. Para XSS Reflejado:

Codificación de Salida: Convertir los caracteres especiales de HTML en sus entidades correspondientes antes de reflejar la entrada del usuario en la página. Esto asegura que el navegador interprete los datos como texto plano y no como código ejecutable.

CONCLUSIÓN

Impacto de las vulnerabilidades y reflexión sobre el proceso:

El ejercicio demostró cómo la falta de validación y sanitización adecuadas de las entradas del usuario puede llevar a vulnerabilidades críticas.

La inyección de comandos permitió obtener acceso no autorizado al servidor subyacente, con la posibilidad de leer archivos sensibles y potencialmente tomar control total del sistema.

El XSS reflejado mostró cómo un atacante podría comprometer las sesiones de los usuarios, llevando al robo de identidad o a la realización de acciones en nombre de la víctima.

La lección clave reafirmada es que "nunca se debe confiar en la entrada del usuario", y que la seguridad debe implementarse tanto en la validación de entrada como en la codificación de salida para proteger eficazmente las aplicaciones web.