Proyecto de Explotación en Pentesting en un Sitio Web Vulnerable

Estudiante: Eric Grossermann
Fecha De Entrega: 11/10/25

Formato del Reporte de Pentesting v2

- Introducción
    - En este ejercicio se realizara una inyección de comandos en una aplicación vulnerable (DVWA). Vamos a aprovechar la caja de “ping” para mostrar como una entrada no validada puede permitir ejecutar comandos del sistema en el servidor.

      El objetivo es entender la vulnerabilidad, validar la explotacion y ver sus consecuencias (acceso a archivos, informacion del sistema).

      Objetivo: demostrar como una entrada insegura puede convertirse en ejecucion remota de comandos y aprender medidas bqsicas de mitigacion.

- Metodología

-Preparo el laboratorio: arranco Debian, levanto DVWA y pongo security en low.
-Pruebo la pagina: hago un ping a 8.8.8.8 para ver si responde.
-Intento la inyeccion: meto "8.8.8.8 ; whoami" y veo la salida.
-Exploro: con ls, cat y ifconfig reviso archivos y configuracion.
-Guardo todo: capturo pantalla y pego resultados en un documento.
-Validar la entrada, sanear caracteres y usar funciones seguras.

Herramientas y Tecnicas utilizadas

- Debian, navegador y DVW
- Terminal (bash) y comandos basicos (whoami, ls, cat, ifconfig, ping)
- Documentacion con capturas
- Tecnicas: reconocimiento, manipulacion de input, explotacion manual, recoleccion de evidencia y recomendaciones de mitigacion

---

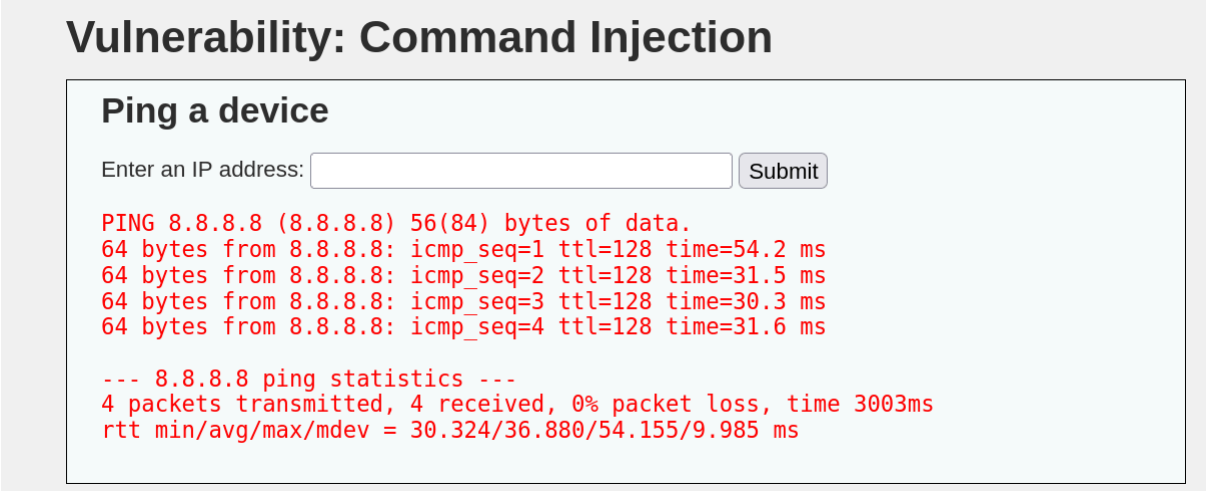
- Resultados

PARTE 1: **Reconocimiento y Preparación (DVWA)**

Reconocimiento y Preparación

1._Escribir:

8.8.8.8 (DNS de Google) a "Submit".



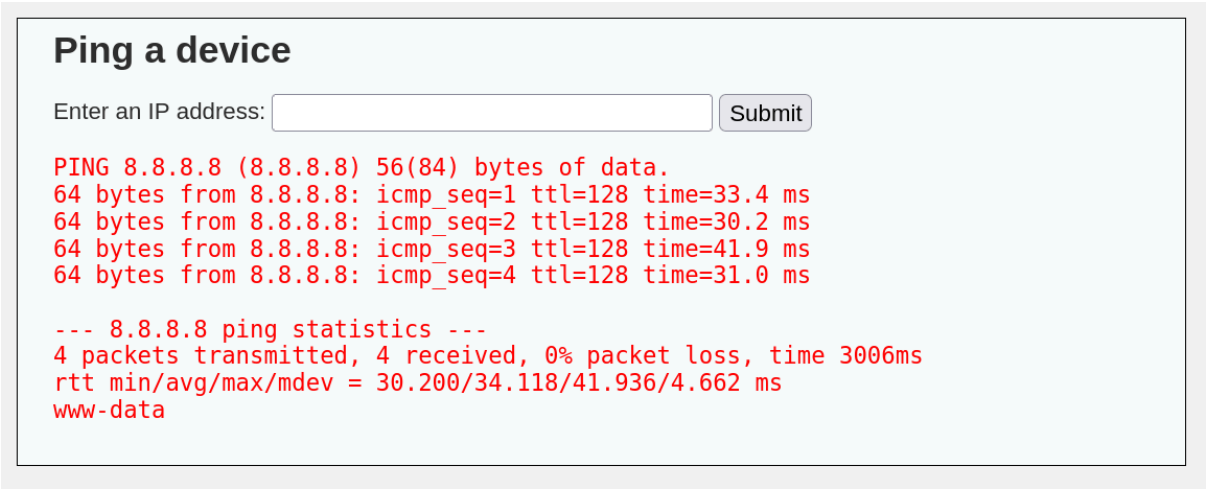
Resultado:

El servidor esta tomando lo que escribimos en la caja y se lo está pasando a la terminal del sistema operativo sin una posible “Validacion”

Validación y Explotación

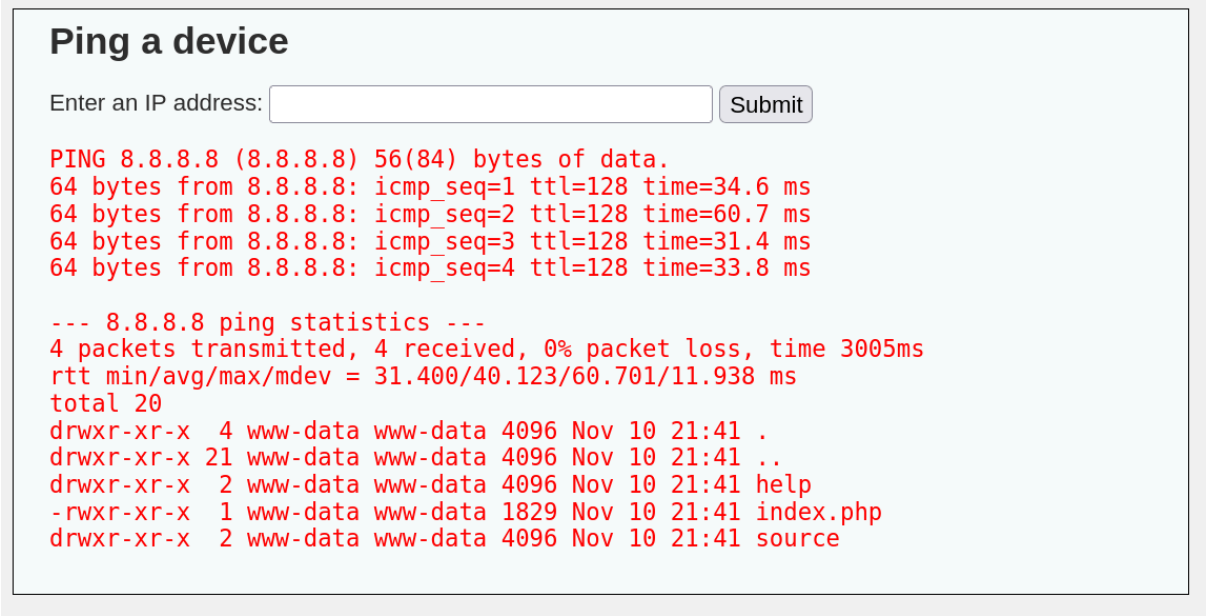
2._Comando:

8.8.8.8 ; whoami



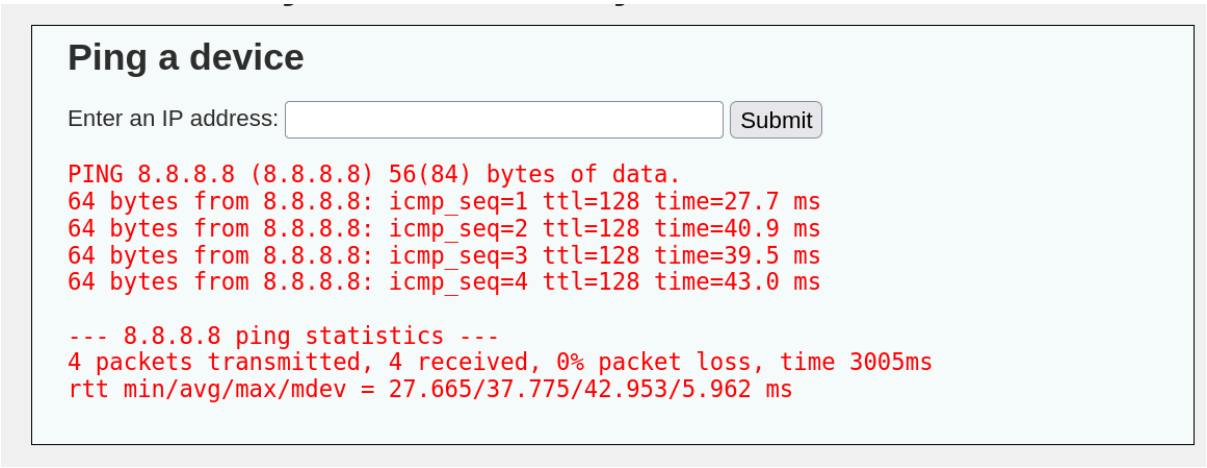
Resultado:
Nos refleja: www-data
Que es el nombre del usuario con el que corre el servidor web Apache.

Listar archivos
3._Comando:
8.8.8.8 ; ls -la



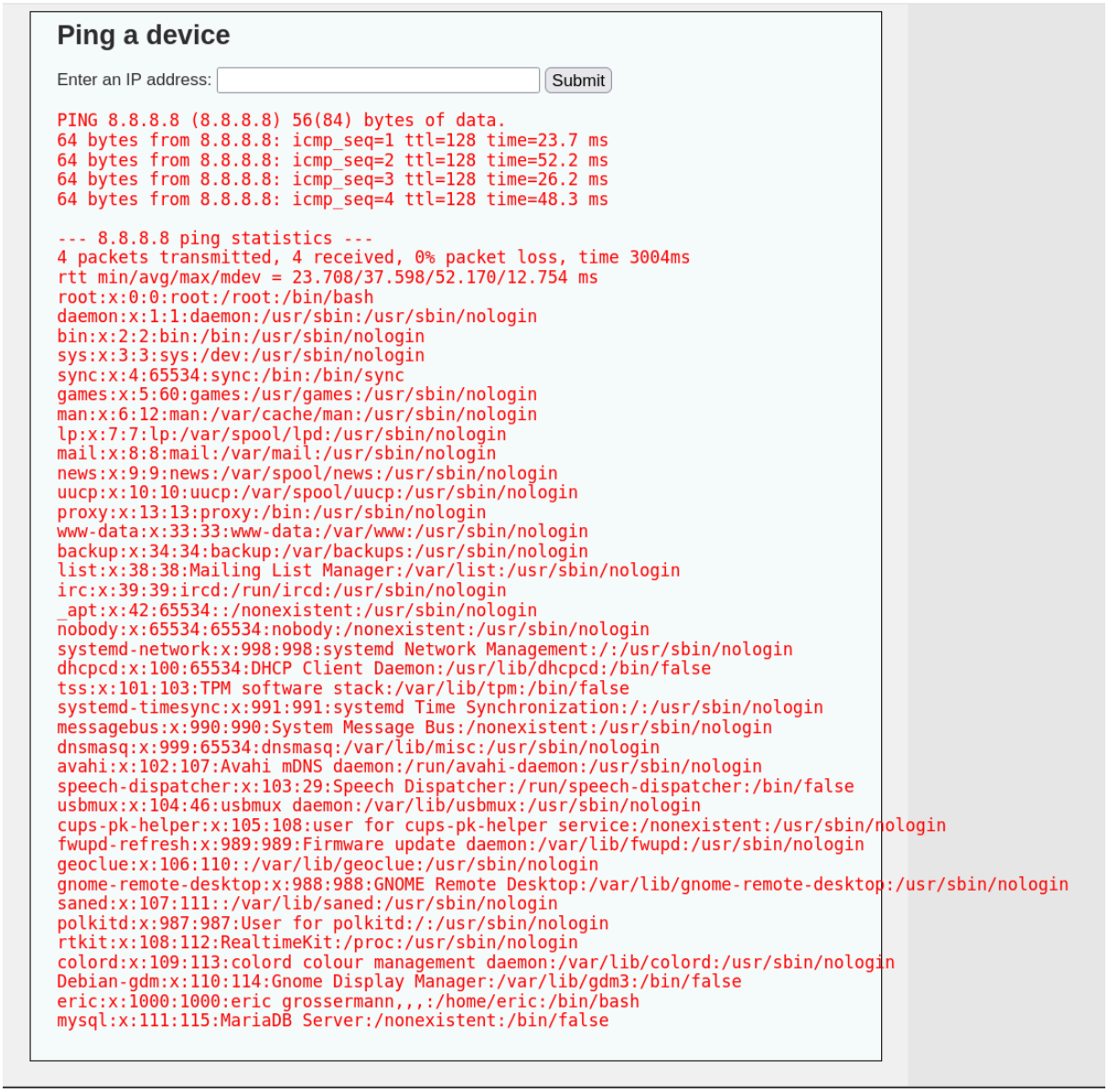
Resultado:
Refleja una lista de todos los archivos en el directorio actual del servidor web.

Ver la configuración de red
4._Comando:
8.8.8.8 ; ifconfig



Resultado: Refleja la salida de ifconfig del servidor.

Leer un archivo sensible:
5._Comando:
8.8.8.8 ; cat /etc/passwd

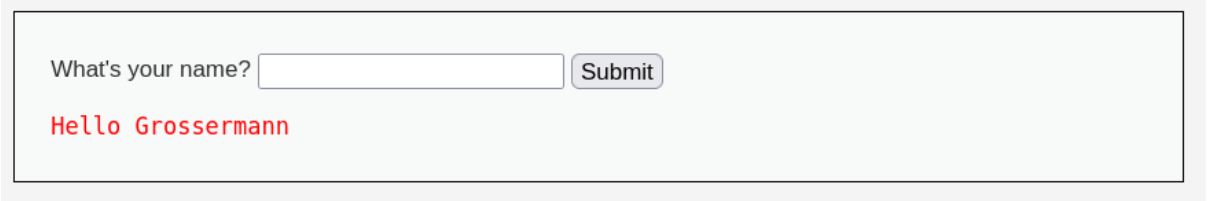


Resultado:
La página te mostrará el contenido del archivo /etc/passwd del servidor, que lista a todos los usuarios del sistema

El servidor tomo lo que escribi y lo metio directamente en un comando de sistema sin comprobarlo.
Por eso al enviar 8.8.8.8 ; whoami se ejecuto primero el ping y luego whoami , la entrada no fue validada ni sanitizada, lo que permite inyeccion de comandos.

PARTE 2: **XSS Reflejado en DVWA**

Localizamos la seccion de XSS (Reflected) para posterior mente llegar al:
Reconocimiento y Preparación
Donde el servidor recibe una peticion de nombre donde suministramos Grossermann



Dando una respuesta del servidor con “Hello Grossermann”
Que pasaria si aplicamos codigo?

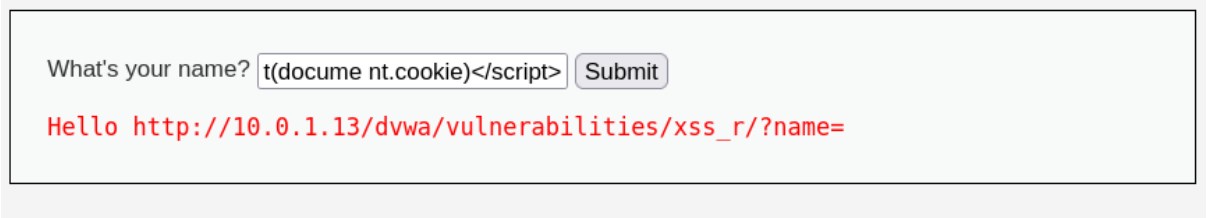
Explotación
1._Comando:
<script>alert('¡Hackeado!');</script>

Resultado:
No refleja "Hello...", sino que mostrará una ventana emergente (pop-up) con el texto "¡Hackeado!"

PARTE 3: **Explotación (El Ataque "Real": Robar Cookies)**
1._Comando:
<script>alert(document.cookie);</script>

Resultado:
El servidor hizo el ping y luego ejecuto el comando que agregamos.
No comprobo que lo que escribimos fuera solo una IP, por eso pudimos meter otro comando mostrando una ventana emergente : PHPSESSID=a1b2c3d4e5f6g7h8...; security=low

El Vector de Ataque
3._Comando:
http://10.0.1.13/dvwa/vulnerabilities/xss_r/?name=<script>alert(document.cookie)</script>



El navegador vio el codigo `<script>` y lo ejecuto porque penso que era parte normal de la pagina.

- Mitigación

Para evitar lo sucedido hay que hacer lo siguiente, facil y directo:

1. No aceptar cualquier cosa que ponga el usuario. Si piden una IP, comprobar que tenga formato de IP y punto.

2. Antes de mandar la entrada al sistema hay que Codificación de Salida (Output Encoding) Antes de "reflejar" la entrada del usuario, el servidor debe "neutralizar" los caracteres especiales de HTML.

3. Usar una función que ponga la entrada entre comillas (ej. escapeshellarg() en PHP) para que el sistema la trate como texto y no como comandos.

4. Pon al servicio con un usuario que no tenga permisos importantes. Si algo falla, que lo maximo que haga sea listar un directorio sin acceso a todo el server.

5. Logs y alertas: guarda todo lo que entre y pon alertas si alguien intenta meter caracteres raros o muchos comandos seguidos.

6. Mantener el servidor y el software al dia y limitar qué servicios estan expuestos a la red. Con eso, la probabilidad de que te inyecten comandos baja muchisimo.

- Conclusión

La vulnerabilidad permite que la aplicacion ejecute comandos del servidor, lo que puede llevar a leer archivos sensibles, obtener acceso al sistema y escalar a otros ataques.

En practicas como esta se ve claro que un fallo small en validar input puede convertirse en un control total sobre la maquina.

La leccion es simple: validar y sanitizar todo, reducir privilegios y monitorear, hacerlo en un lab como DVWA nos ayuda a entender el riesgo sin dañar a nadie.