Blandskron

Paso a paso: inyección SQL en DVWA

preparación del entorno

abrir un navegador moderno y dirigirse a la dirección http://localhost/dvwa esta dirección apunta al entorno local donde está alojada la aplicación vulnerable damn vulnerable web application dvwa esta aplicación simula fallos comunes en aplicaciones web para realizar pruebas de penetración éticas en un ambiente controlado

iniciar sesión utilizando las credenciales por defecto de dvwa usuario admin y contraseña password si estas credenciales han sido cambiadas previamente utilizar las actuales correspondientes en caso de error verificar que el servidor apache o nginx y mysql estén corriendo correctamente en el entorno local

una vez dentro del dashboard de dvwa buscar la opción dvwa security ubicada en el menú lateral izquierdo o superior según la versión de la interfaz seleccionar el nivel de seguridad low esto es esencial ya que los niveles medium high y impossible aplican técnicas de mitigación que podrían impedir la explotación manual de la vulnerabilidad

guardar la configuración seleccionada asegurándose de que el cambio de nivel de seguridad fue exitoso refrescar la página para confirmar que el entorno quedó configurado correctamente

identificación del módulo vulnerable

dirigirse al menú lateral izquierdo y seleccionar la opción sql injection esta sección ha sido diseñada para contener una implementación vulnerable a inyección de sentencias sql el objetivo es comprobar si el parámetro id usado en las consultas sql no está siendo sanitizado adecuadamente

observar la estructura del formulario presentado típicamente es un input que solicita un identificador numérico de usuario seguido de un botón submit o similar al enviar el formulario se genera una petición http que incluye el parámetro id

análisis de la solicitud http

abrir burp suite y configurar el navegador para que redirija su tráfico a través del proxy de burp generalmente esto implica configurar el navegador para usar el proxy local en localhost puerto 8080 una vez configurado capturar la solicitud generada al enviar el identificador de usuario por ejemplo id=1

ir a la pestaña http history de burp suite y analizar la solicitud capturada observar si el parámetro id está siendo enviado mediante el método get en la url o mediante post en el cuerpo de la solicitud

este análisis permite identificar el punto de inyección es decir el lugar exacto donde se inserta el input del usuario en la consulta sql interna ejecutada por la aplicación

ejecución de payloads simples

ingresar en el campo de texto un payload tautológico como el siguiente

```
1' OR '1'='1
```

esto busca forzar la condición booleana a verdadera independientemente del valor original del parámetro al enviar el formulario observar si el sistema retorna múltiples resultados o todos los registros de la tabla usuarios

esto indica que la consulta fue modificada exitosamente mediante la inyección y confirma la existencia de una vulnerabilidad de tipo inyección sql clásica por concatenación directa sin validación

ejecución de payloads avanzados

probar con una carga más sofisticada como

```
1' UNION SELECT null, database(), null --
```

este payload intenta forzar la ejecución de una sentencia union select que expone el nombre de la base de datos activa la función database es una función sql estándar que retorna el nombre del esquema activo

si la respuesta de la página incluye el nombre de la base de datos en lugar de datos de usuario esto confirma que se ha logrado manipular la lógica interna de la consulta con éxito y se ha accedido a información del sistema gestor de base de datos

repetición de pruebas con burp repeater

en burp suite seleccionar la solicitud correspondiente en la pestaña http history y enviarla al módulo repeater allí modificar el valor del parámetro id e insertar distintos payloads para observar de forma controlada las respuestas que devuelve el servidor

probar payloads como

```
1' AND 1=2 --
1' ORDER BY 3 --
1' UNION SELECT user, password FROM users --
```

verificar si el sistema muestra errores de sintaxis sql errores del motor de base de datos o filtrado incorrecto de datos esto ayuda a inferir la estructura de las tablas y columnas internas de la base de datos

documentación técnica del hallazgo

una vez identificada y validada la vulnerabilidad proceder a elaborar un informe técnico profesional estructurado con los siguientes apartados

título del hallazgo

por ejemplo inyección sql reflejada en parámetro id del módulo user info

descripción del problema
 explicación técnica de cómo el parámetro id no es sanitizado y permite manipular directamente la consulta sql ejecutada

evidencia técnica
 incluir los payloads utilizados capturas de pantalla de la respuesta del servidor descripción del comportamiento observado

evaluación de riesgo
 asignar una criticidad según criterios como confidencialidad integridad y disponibilidad aplicar métricas como cvss para justificar el nivel de riesgo

recomendación técnica
 proponer el uso de sentencias preparadas parametrizadas validación estricta del input del usuario separación de privilegios y auditoría continua

referencias
 incluir enlaces a documentación oficial de owasp base de datos mitre cwe y otras fuentes reconocidas para apoyar las recomendaciones

reflexión final

responder a las preguntas guía como por qué funcionó la inyección que medidas hubieran evitado este hallazgo y qué impacto real podría haber tenido en un entorno de producción esto permitirá contextualizar la importancia de la validación y codificación de entradas en el ciclo seguro de desarrollo de software

con este paso a paso se logra resolver completamente el ejercicio demostrando competencia técnica en la identificación explotación y documentación de vulnerabilidades de tipo inyección sql en aplicaciones web vulnerables como dvwa