SQL Injection Scanner Report

✔ http://vulnapp.example.com

Summary

Overall risk level:

High

Risk ratings:

High:	1
Medium:	0
Low:	0
Info:	1

Scan information:

Start time:	2018-12-03 16:47:15
Finish time:	2018-12-03 16:47:56
Scan duration:	41 sec
Tests performed:	2/2
Scan status:	Finished

Findings

SQL Injection

Vulnerable Page	Vulnerable Parameter	Method	Attack Vector	
/travel.php	id	GET	http://vulnapp.example.com/travel.php?id=5-2	
/bookings.php	cat	GET	http://vulnapp.example.com/bookings.php?cat=4+AND+1%3D1+--+	
/user_profile.php	uname	POST	http://vulnapp.example.com/user_profile.php **POST Data:** uname=ZAP' OR '1'='1' --	
/user_profile.php	pass	POST	http://vulnapp.example.com/user_profile.php **POST Data:** pass=ZAP' OR '1'='1' --	

Details

Risk description:
SQL injection may be possible.

Recommendation:
Do not trust client side input, even if there is client side validation in place.
In general, type check all data on the server side.
If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?'
If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries.
If database Stored Procedures can be used, use them.
Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality!
Do not create dynamic SQL queries using simple string concatenation.
Escape all data received from the client.
Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed characters in user input.
Apply the principle of least privilege by using the least privileged database user possible.
In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact.
Grant the minimum database access that is necessary for the application.

Light spider results: 11 dynamic URLs of total 25 URLs crawled

METHOD	URL	PARAMS

GET	/bookings.php	cat=2
GET	/bookings.php	cat=3
GET	/bookings.php	cat=4
POST	/user_profile.php	uname=ZAP&pass=ZAP
POST	/search.php?test=query	searchFor=ZAP&goButton=go
GET	/for_rent.php	file='%20+%20pict.item(0).firstChild.nodeValue%20+%20'
POST	/guestbook.php	name=anonymous+user&text=&submit=add+message
GET	/travel.php	artist=3
GET	/travel.php	artist=2
GET	/bookings.php	cat=1
GET	/travel.php	artist=1

Scan coverage information

List of tests performed (2/2)

- ✔ Spidering target
- ✔ Scanning for SQL Injection...

Scan parameters

Website URL:	http://vulnapp.example.com
Scan type:	Light
Authentication:	False