**Testphp.vulnhub.com**

**Cross-Site-Scripting(XSS)**

In this website first we seen that if we were searching anything on the search bar then it is returning it as it is as for ex- I searched html on the search for and that reflected me the same I typed. A screenshot of a computer

AI-generated content may be incorrect.

So as it is reflecting the same command we put inside the search box then We know that there can be posibility of Reflected XSS So know we will apply commands and check whether it is vulnerable or not.

As I tried the command it is reflecting that back on screenA screenshot of a computer

AI-generated content may be incorrect.

So it is vulnerbale to Reflected XSS

Now we will try XSS on every search option like there is a option for the guestbook also so we will see that whether this will work on it also

A screenshot of a computer

AI-generated content may be incorrect.

As it is visible, guestbook is also vulnerable to XSS.

Now I used <img src="https://imaget.io/wp-content/uploads/all-image-downloader.jpg"/>

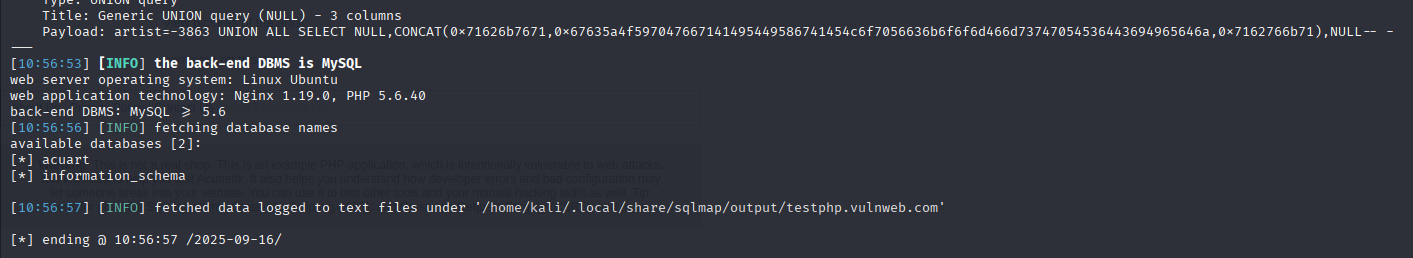
I picked the image from google and tried this tag and it is reflection back the image. A screenshot of a computer

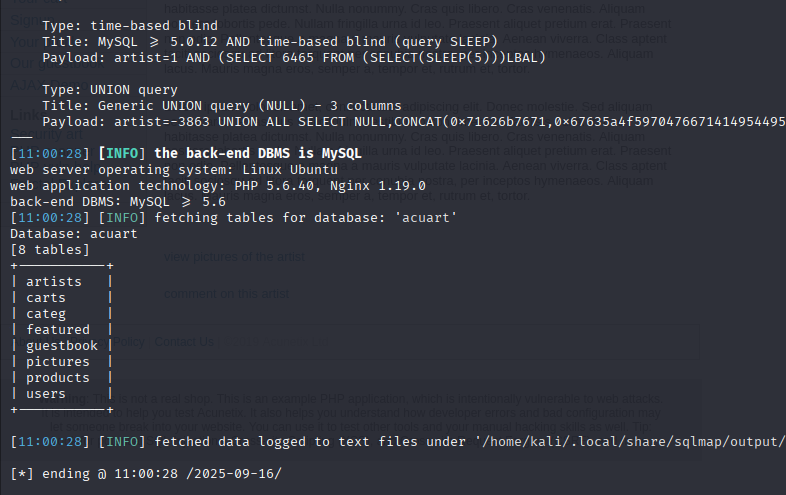
AI-generated content may be incorrect.

**SQL Injecton**

For this we have used sql map to find out the vulnerale parameter so we get to know that artist is vulnerable. We get the database out of it using sqlmap.

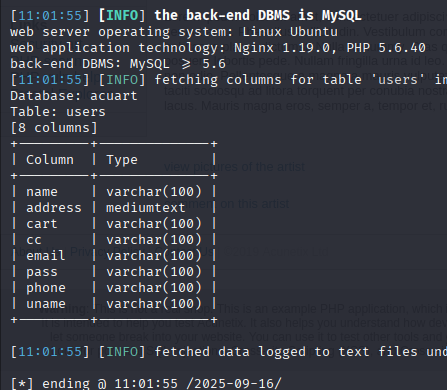
**sqlmap -url http://testphp.vulnweb.com/artists.php?artist=1 --dbs**

****

After that we will target acuart database and try to fetch Tables****

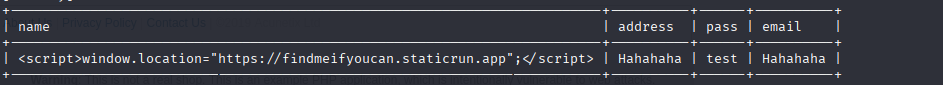
Now out of tables we will fetch columns using command

**sqlmap -url http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users –columns**

****

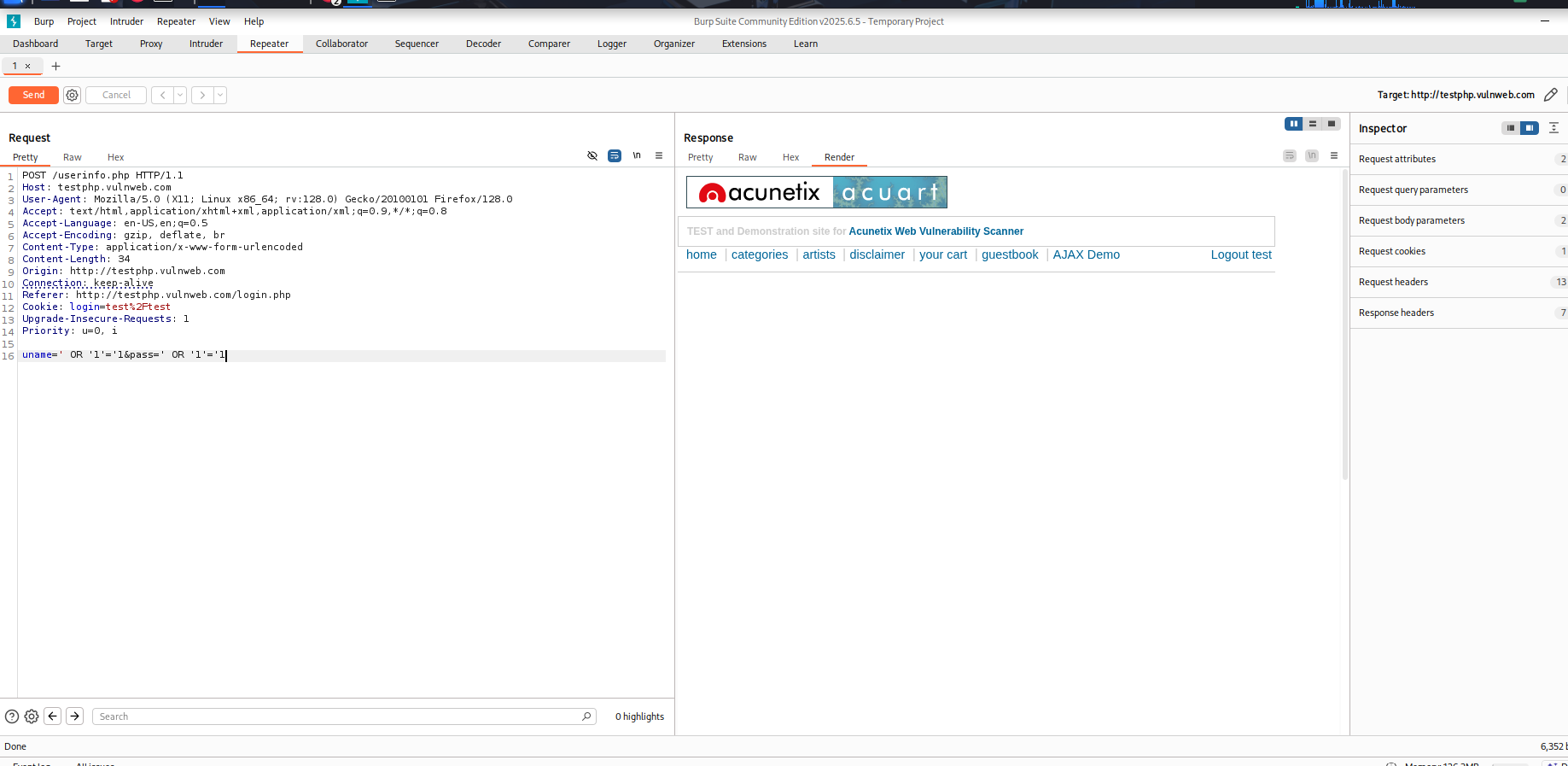
Now we will dump all the columns using command

**sqlmap -url http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users -C name,address,pass,email –dump**

**we get the output as**

**Bypass login through SQL injection using burp proxy**

* Firstly, try any username and password on the login page and pass the request on burp proxy.
* After that try to bypass login using SQL injection in the burp proxy.
* I used the payload as username=’ OR ‘1’=’1 and same criteria in the password also .
* With this we bypassed the login and logged in as test using this only.



**CSRF(cross-site request forgery)**

I tried CSRF on this website using html code the code which help me in implementing CSRF on DVWA, but I found out after applying that code on this website that this is not vulnerable to CSRF.

Code I tried is -A close-up of a person

AI-generated content may be incorrect.

I was trying to change username from josh to hacked but it didn’t implement. But it is not changing the name to hacked name is displaying as it is.

A screenshot of a computer

AI-generated content may be incorrect.

Rather it is displaying same output as we simply log in.