**LAB 3: Exploitation**

Demonstrate a non-destructive proof of concept (PoC)- show inputs /payload.

**Non-destructive Proof of Concept (PoC)** means showing that a vulnerability (like SQL Injection) exists without altering, deleting, or damaging data or systems. You simply demonstrate the flaw in a safe and controlled way — for educational or ethical hacking purposes.

Previous lab we already done with SQLinjection. Let us use the same example to do the demonstrate.

* Prove SQL injection exists on a system.
* Use SQLMap to demonstrate it.
* Avoid destructive actions like DROP TABLE, DELETE, UPDATE.
* Show your payload/input used, such as a URL or form data.

**What is SQL Injection?**  
SQL Injection (SQLi) is a technique where attackers insert malicious SQL queries into input fields, tricking the server into executing unintended database commands.

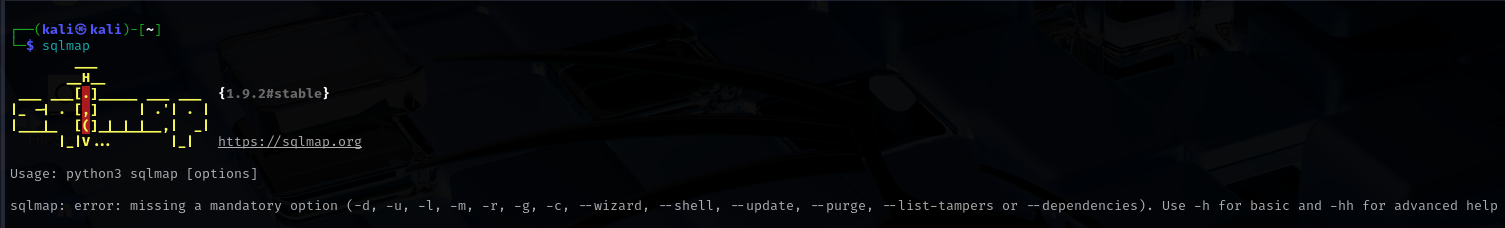
**Why is SQL Injection dangerous?**  
It can allow attackers to read, modify, or even delete data from a database if input validation is not properly handled.

**Before SQLMap Commands**

In this lab, we use **SQLMap**, an automated tool that helps identify SQL injection vulnerabilities in a given URL or form input.

**Demonstrating SQLinjection**

* 1. First identify website that you target to do SQLinjection.
  2. Explore the website till you are able to get any link as follow, example *id=4, cat=7, productid=3*

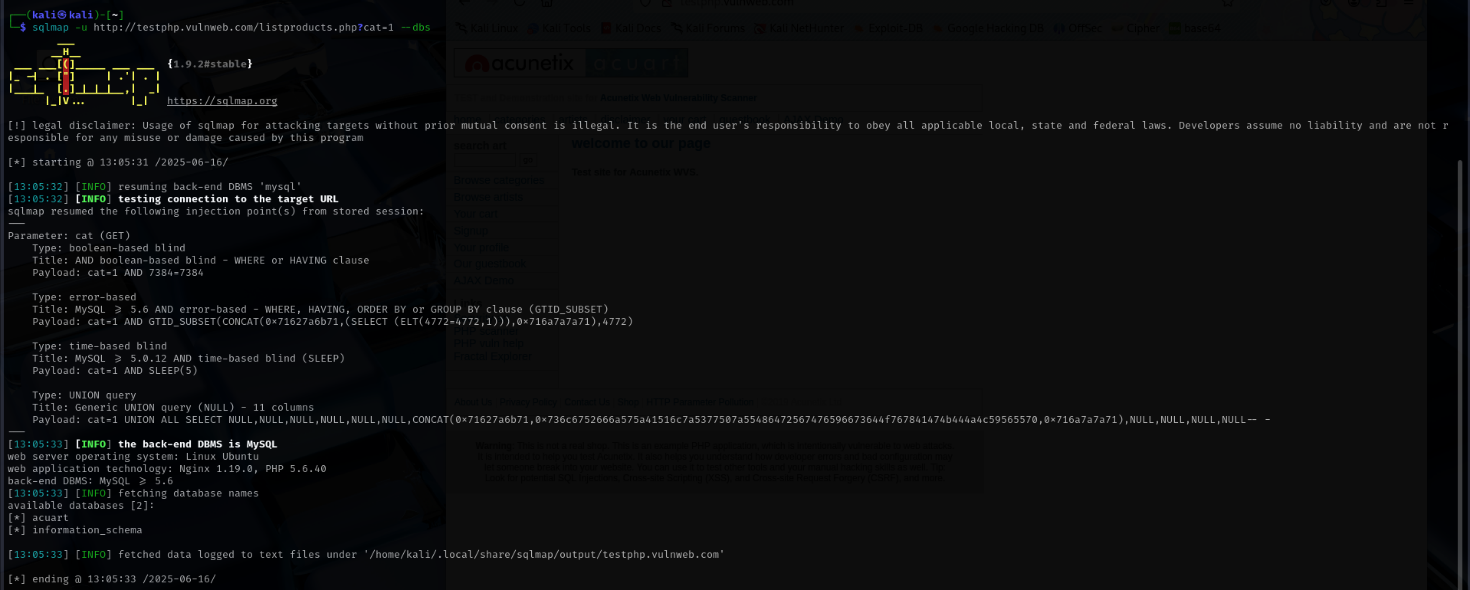


* For today’s Lab, we are going to attack **http://testphp.vulnweb.com**
  + - Explore the website till you manage to get the link this

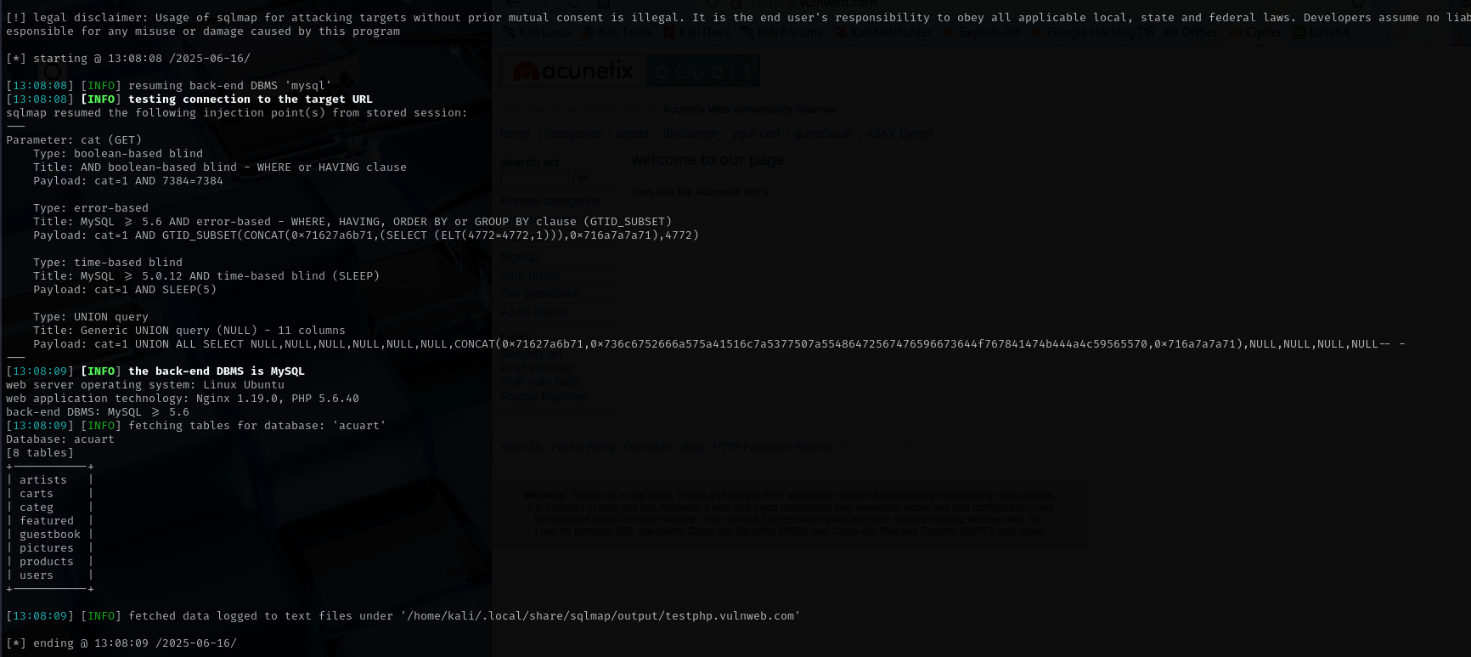
**This is your payload**  
[**http://testphp.vulnweb.com/listproducts.php?cat=1**](http://testphp.vulnweb.com/listproducts.php?cat=1)

* + - Then open VBox and Kali Linux. Open SQLmap
    - **Type on your terminal Kali Linux:**
    - sqlmap -u <http://testphp.vulnweb.com/listproducts.php?cat=1> - -dbs

Will display all the databases names.

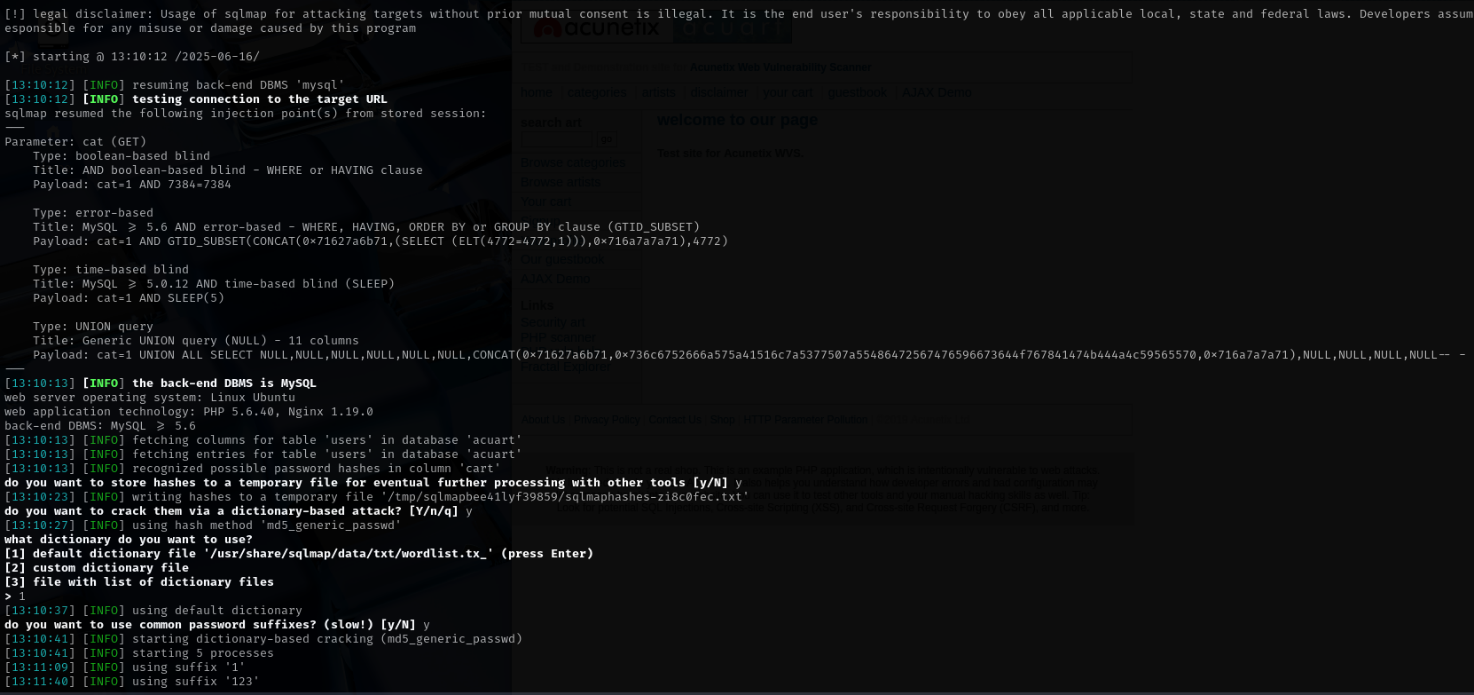


* sqlmap -u <http://testphp.vulnweb.com/listproducts.php?cat=1> -D **acuart** - -tables

Will display tables in that particular database

* sqlmap -u <http://testphp.vulnweb.com/listproducts.php?cat=1> -D acuart -T **users** - -dump

Will display data in those particular users table



**Demonstrating Reflected XSS**

**What is XSS?**

**Cross-Site Scripting (XSS) allows attackers to inject scripts into web pages viewed by others.  
Why is this bad? It can lead to stolen cookies, session hijacking, or defacement.  
What is Reflected XSS? It happens when the injected code is returned immediately in the response.**

**What is Reflected XSS?  
Reflected XSS occurs when user input is immediately echoed back in the page response without proper sanitization.**

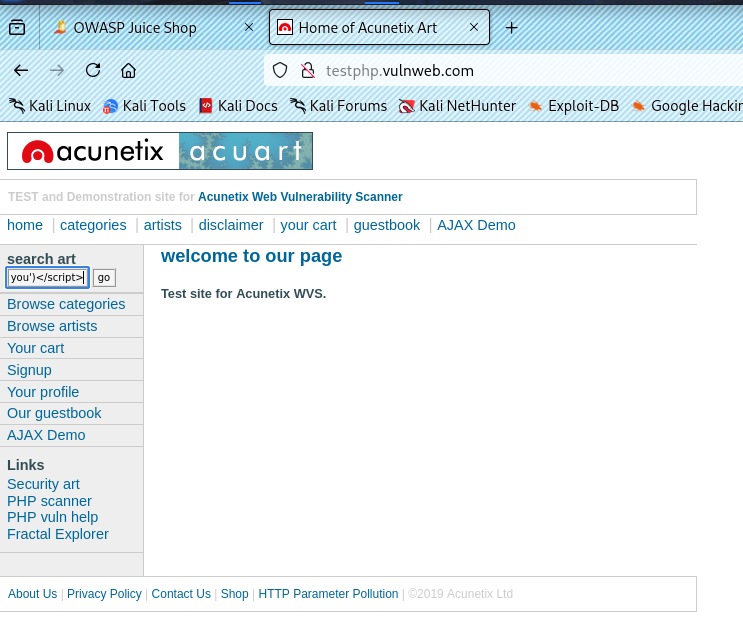
Step 1: Open the Target Page

1. Open your web browser.
2. Visit this link:  
   👉 http://testphp.vulnweb.com/search.php
3. You'll see a search box on the page with a button next to it that says **Search**.

This is the place where we’ll inject our XSS payload.

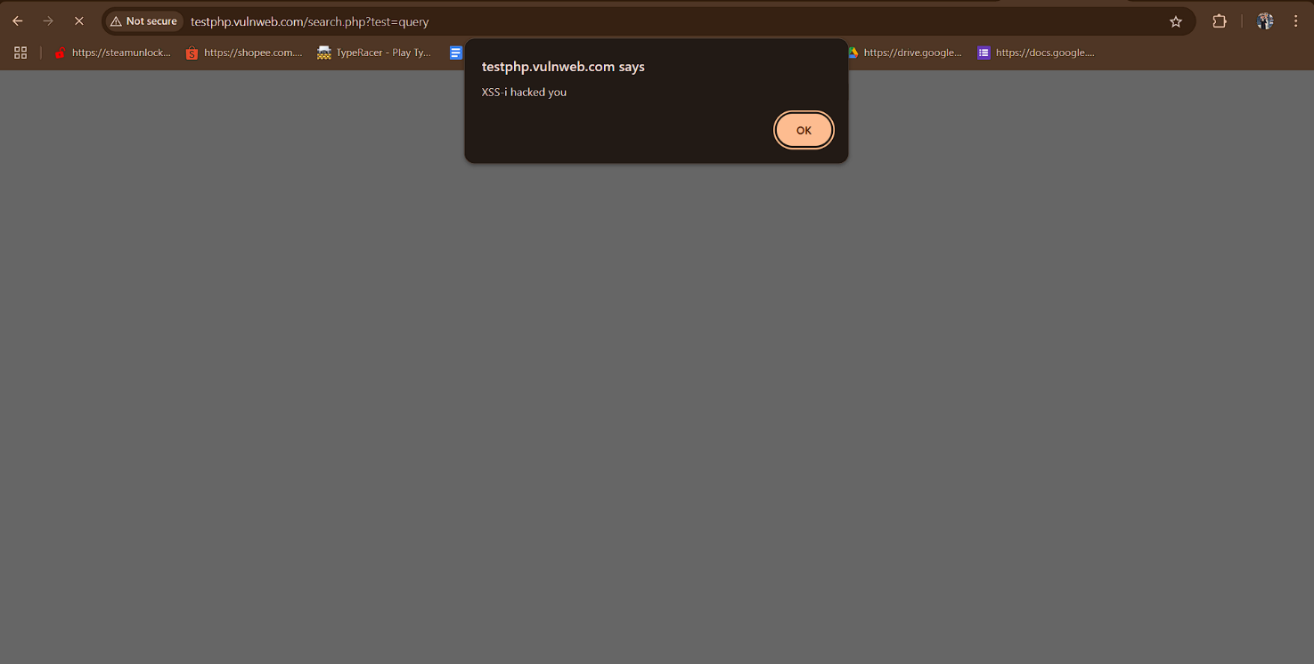
Step 2: Paste the XSS Payload into the Search Box

1. In the **Search box** (on the actual webpage, not in the source code), type or paste this code: **><script>alert('XSS-I hacked you')</script>**
2. Click the **Search** button.



Step 3: Observe What Happens

The browser will redirect you to a URL like this: If the site is vulnerable, you will see a **popup box**



* The payload you used is **harmless** — just shows an alert box.
* You didn’t hack, break, or modify any data.
* This is **non-destructive** and purely for ethical learning.

CONCLUSION

In this lab, we demonstrated how to identify SQL Injection, Reflected XSS, and Security Misconfiguration vulnerabilities using safe, non-destructive techniques. Understanding these attacks helps developers and testers build more secure web applications.