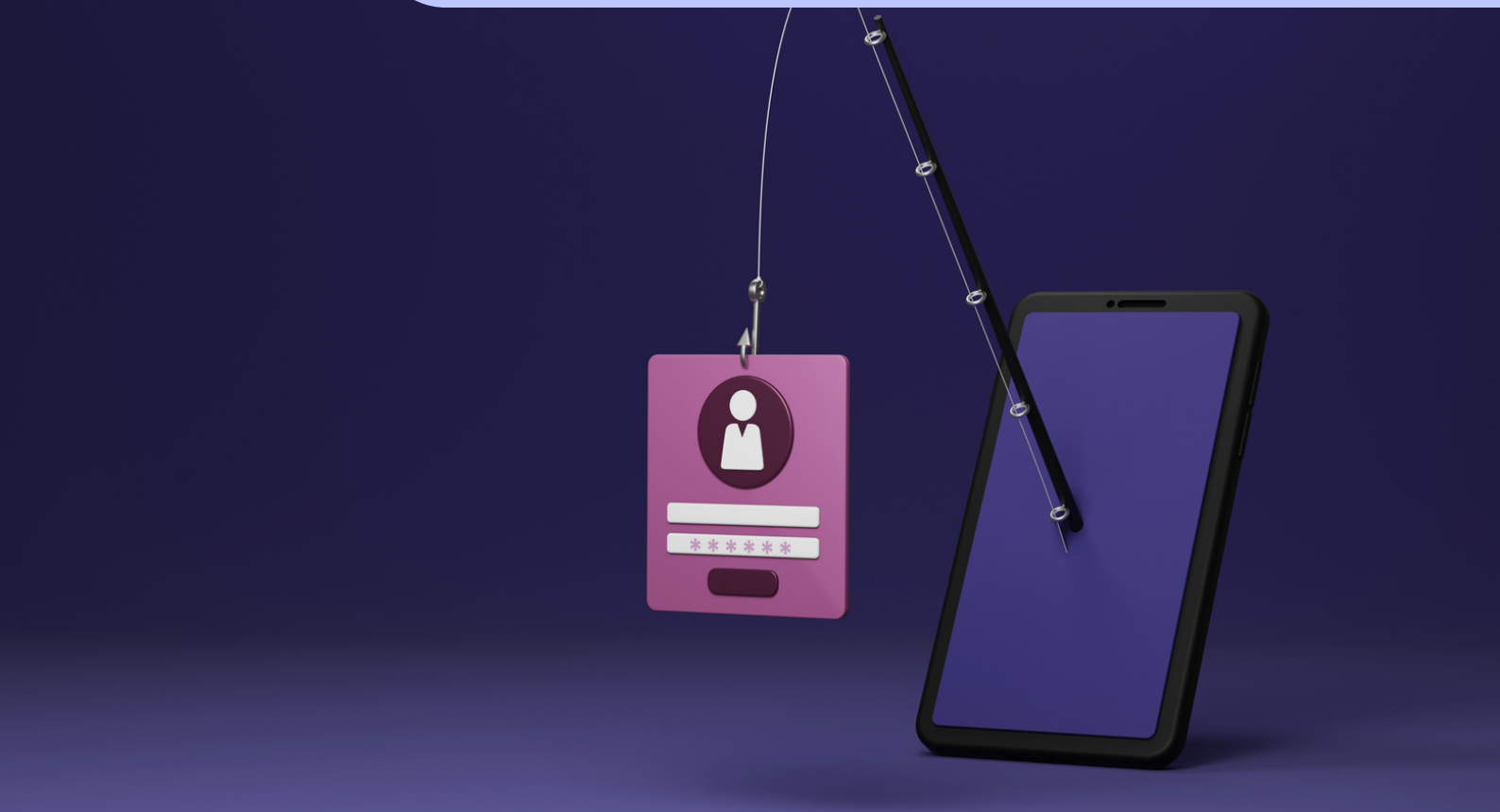
CHALLENGE

DATA BREACH *WRITE-UP*

trustline

TrustLine is your all-in-one solution for detecting, managing, and swiftly fixing security vulnerabilities.

BY: EXPLOITMINDER



AGENDA

NOTE

This writeup is exclusively intended for educational purposes. that reflect my individual perspective.

هذا المنشور هو لغرض نشر المعرفة و التعلم فقط، طريقتي ربما ليست الأصح لكنها تعكس وجهة نظري الشخصية.

CHALLENGE INTRODUCTION

DataBreach

By Trustline

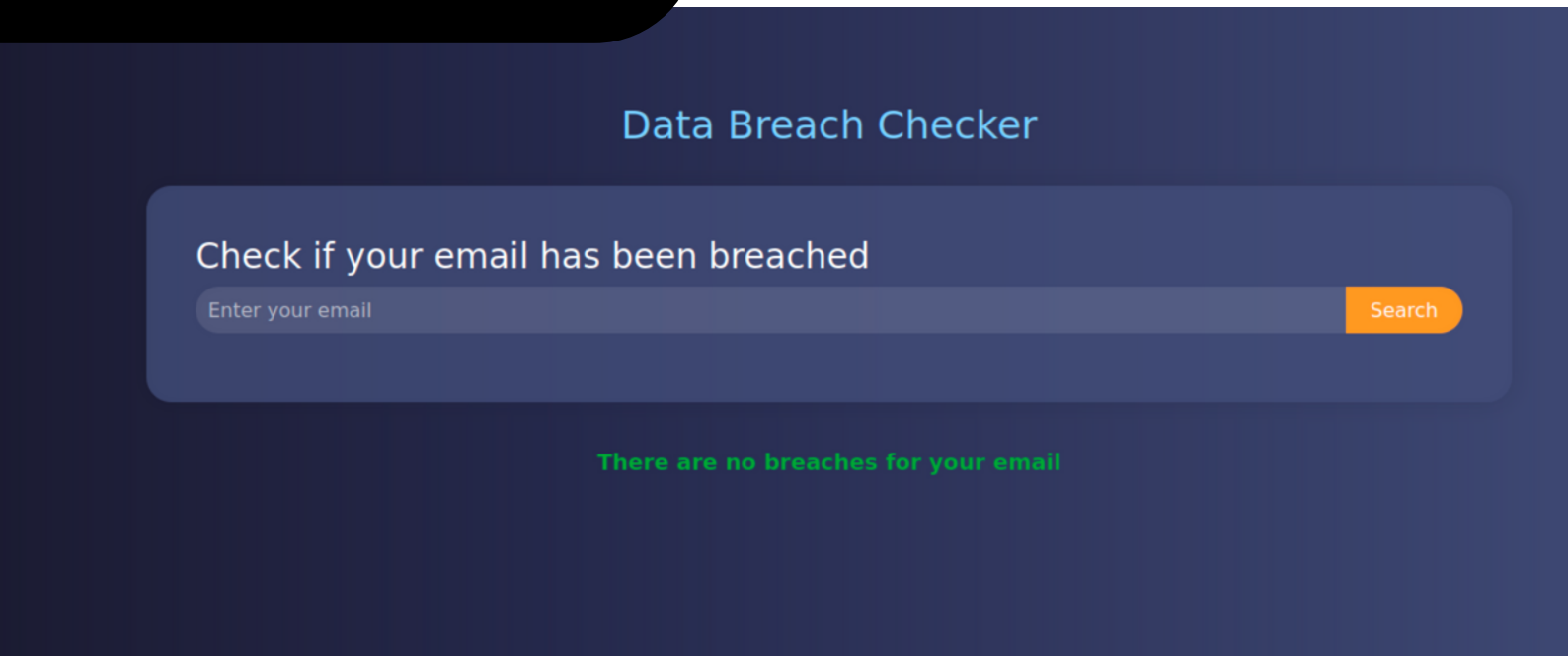
Challenge status	Challenge difficulty	Challenge category	Challenge points
Active	Medium	Web	20

ABOUT THE CHALLENGE

Data breaches can disrupt your life and privacy. It's time to take charge and find out if your personal accounts have been compromised.

SOLVING THE CHALLENGE

WEBSITE

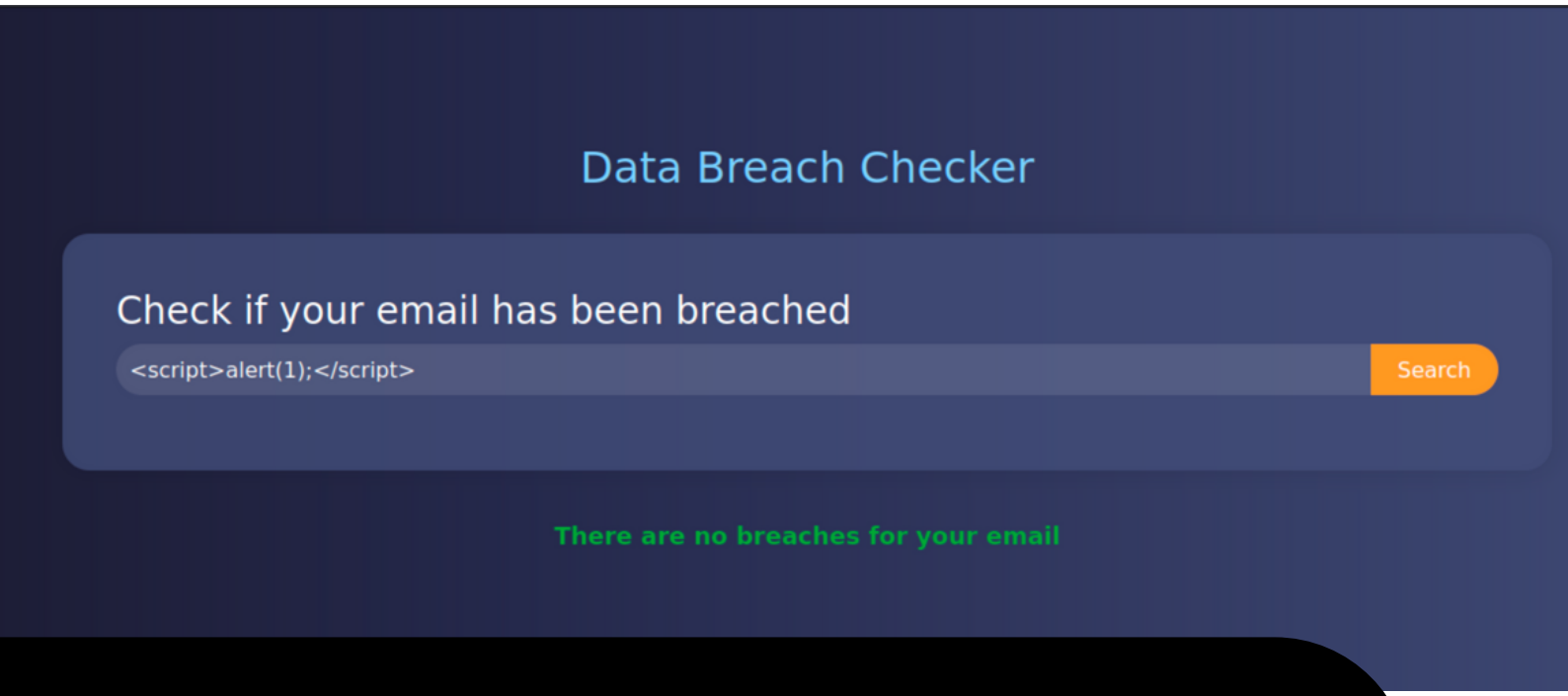


Upon reviewing the web page, it became apparent that the primary focus of the challenge lies in injection vulnerabilities, particularly emphasizing fundamental (OWASP 10) vulnerabilities such as XSS or SQL injection.

TRIAL 1: XSS

Attempting an XSS payload resulted in no observable outcome.

The command used:
<script>alert(1);</script>



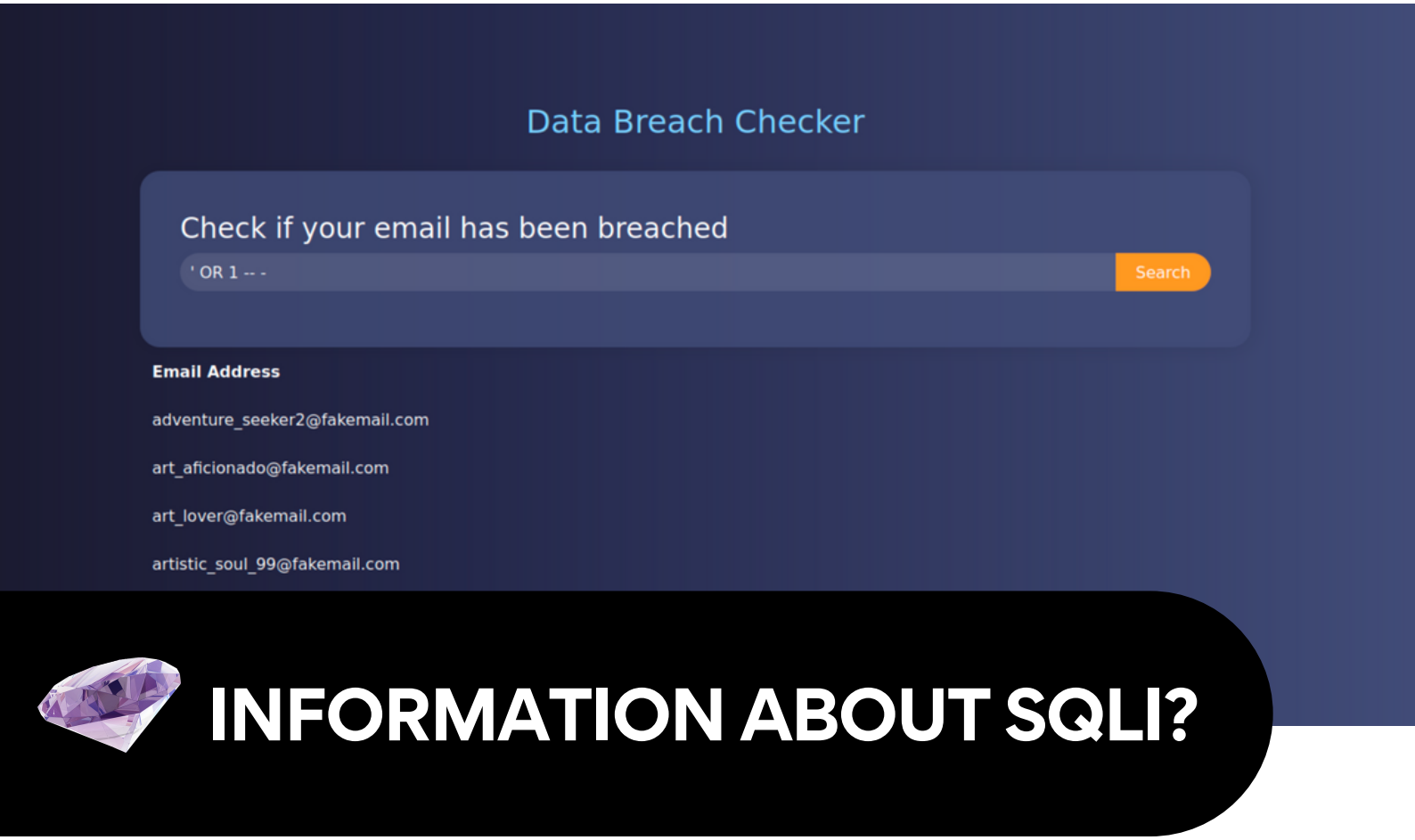
INFORMATION ABOUT XSS?

- XSS is a vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users.
- It occurs when a web application includes untrusted data in a webpage without proper validation or escaping, allowing an attacker to execute scripts in the victim's browser.
- There are three main types of XSS: stored XSS, reflected XSS, and DOM-based XSS.

TRIAL 2: SQL

The SQL injection produced a significant result.
retrieved a list of emails!

The command used:
' OR 1 -- -

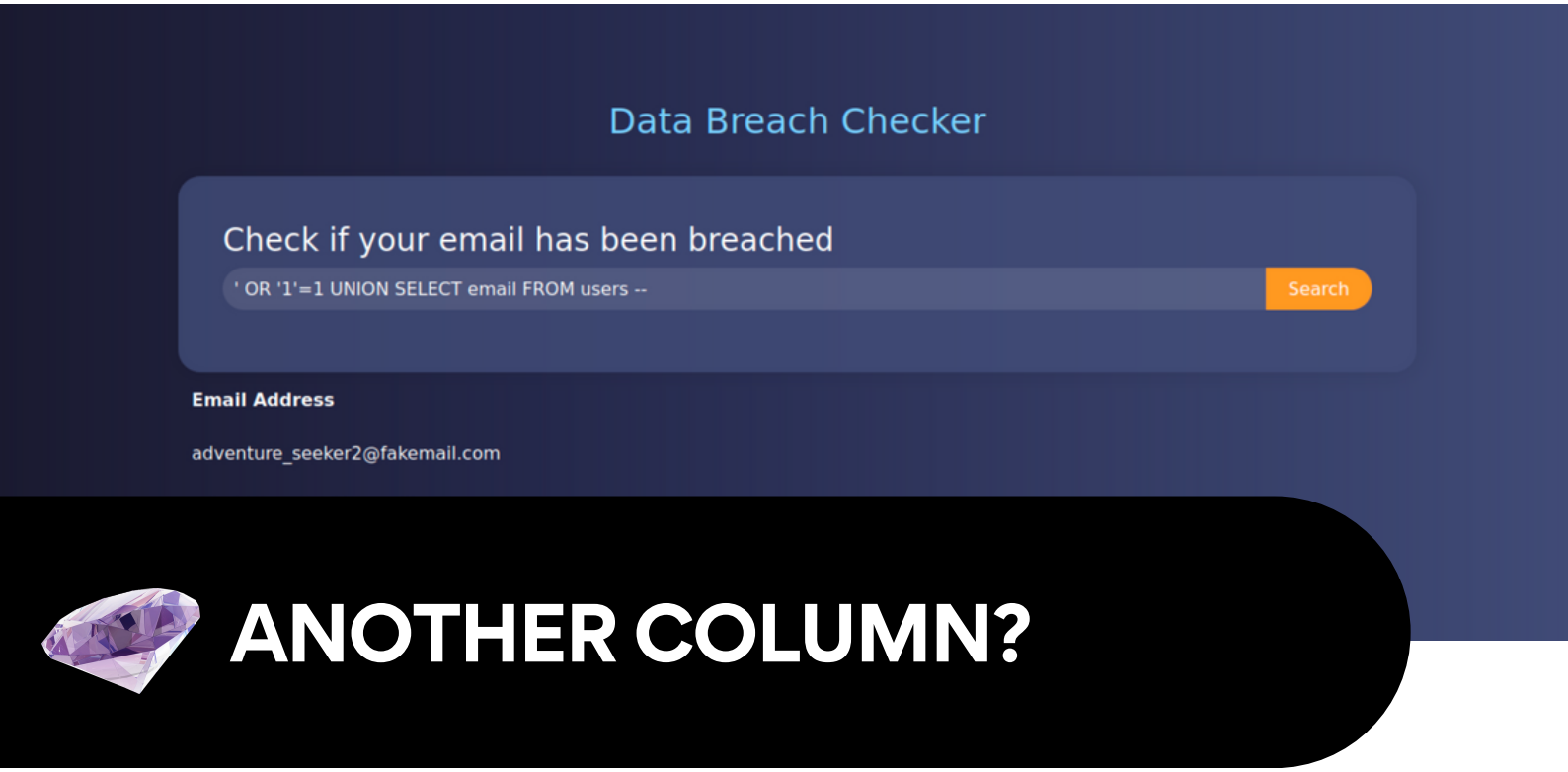


INFORMATION ABOUT SQLI?

- SQL Injection is a vulnerability that allows an attacker to interfere with the queries that an application makes to its database.
- It occurs when untrusted data is inserted into SQL commands without proper validation or sanitization.
- Attackers can use SQL Injection to bypass authentication, **retrieve sensitive data,** modify data, or execute administrative operations on the database.

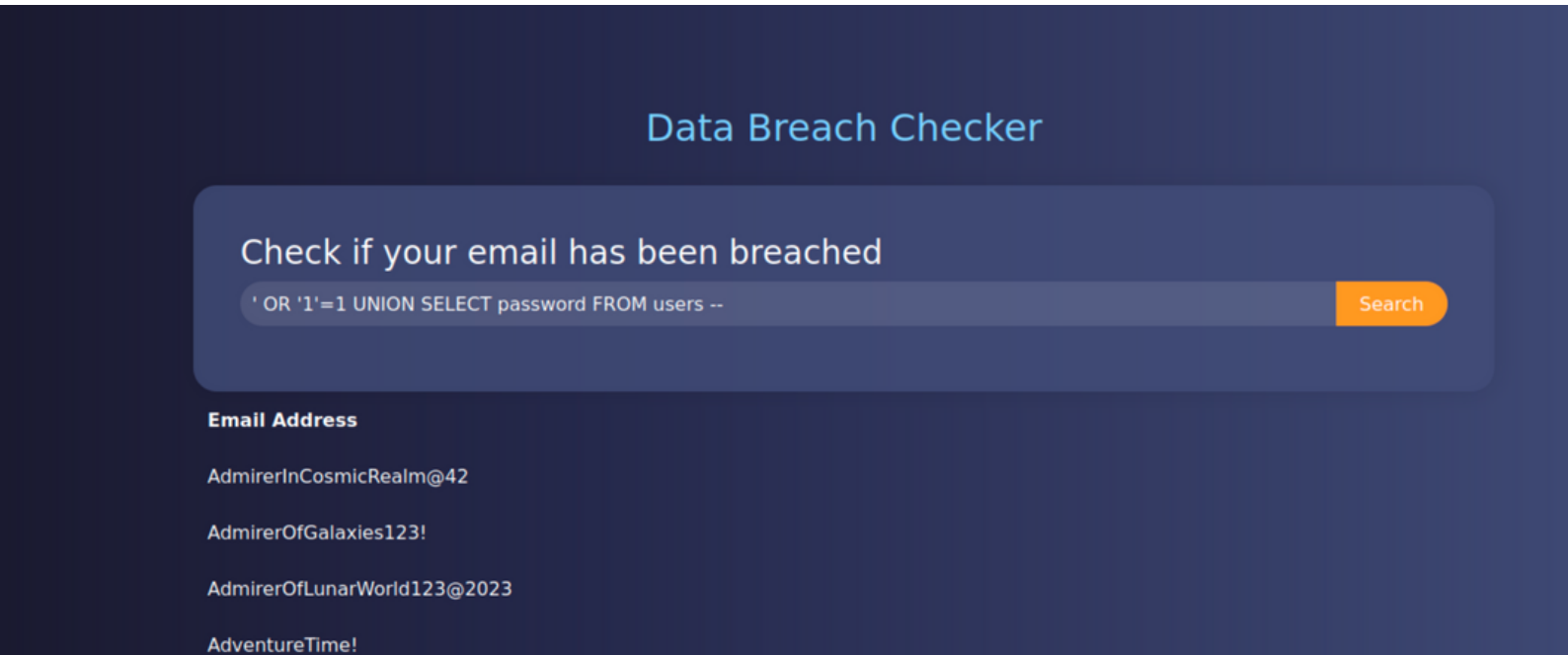
TRIAL 2: SQL

The payload takes advantage of the 'OR' condition to bypass authentication and the 'UNION SELECT' statement to retrieve data from another table.
Payload:
OR '1'=1 UNION SELECT email FROM users -
*retreived email from users to check it there is users table :)



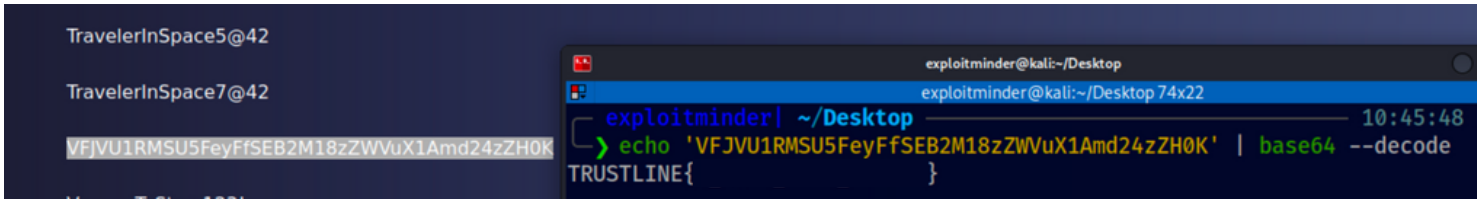
ANOTHER COLUMN?

Therefore, I tried to find if there is a password column, and it worked.



TRIAL 2: SQL

While inspecting the outputs, I stumbled upon something encoded. Upon decoding it, I uncovered the flag.



EXTRA INFORMATION?

WHY BASE64 ENCODING?

- Base64 encoding:
  - Converts binary data to text.
  - Ensures compatibility across systems.
- Not encryption:
  - No security benefits.
- Used for:
  - Transmission and storage.
  - Preserving data integrity.

CONCLUSION

PREVENTION METHODS FOR SQLi

KEEP THING ON TRACK

INPUT VALIDATION

Implement strict input validation techniques to ensure that user-supplied data is sanitized and adheres to expected formats. This involves validating input data types, length, and format to reject any input that could potentially be malicious.

PARAMETERIZED QUERIES

Utilize parameterized queries or prepared statements in database interactions. Parameterized queries separate SQL code from user input, preventing attackers from injecting malicious SQL commands. These queries use placeholders for user input, which are then bound to parameters, thus avoiding the concatenation of user input directly into SQL statements.

REFERENCE

OWASP SQL INJECTION

SQLI PAYLOADS

PORTSWIGGER SQLI

THANK YOU

MADE BY: EXPLOIT MINDER