

**Ghulam Ishaq Khan Institute of Engineering Sciences and Technology**

***Cybersecurity and Principles (CY-201)***

***Module 15 – SQL Injection******Group 09***

***Instructor:*** *Mr. Abdullah Bin Zarshaid****Date of Submission:*** *16th May 2025*

**Faculty:** CYS

**Group Members**

* 2023346 Muhammad Abdullah Khan Mahsud
* 2023907 Taamir Aatif

Task Distribution:

Lab 1 : Muhammad Abdullah Khan Mahsud 2023346

Lab 2 : Taamir Aatif 2023907

**Objective**

The objective of this module is to understand the mechanics, risks, and defenses related to SQL Injection vulnerabilities in web applications. Through this module, we aim to gain hands-on experience in identifying different types of SQLi such as classic, error-based, union-based, and blind SQL injection. We will simulate real-world attack scenarios in a controlled environment to observe how malicious SQL inputs can be used to access or manipulate backend databases. The goal is also to explore secure coding practices and mitigation strategies that developers can use to protect applications. This module helps strengthen our foundational knowledge of web security and database attack vectors.

**Environment Setup**

* Parrot OS
* Windows 11
* Windows SQL Server 2022
* Virtualization Tool Used: Oracle VirtualBox
* Network Settings: Host-only Adapter and bridged-adapter
* All the required tools were installed on all VMs
* Database imported into the MSSM and SQL Server

For this project, we used a combination of virtual machines and operating systems to create a secure and realistic lab environment. Our setup included **Windows 11**, **Parrot OS**, and **Windows Server 2022**, all running on **Oracle VirtualBox**. The network was configured using Host-only and Bridged adapters to allow smooth communication between the systems. On the Windows 11 machine, we installed **Microsoft SQL Server 2019** along with **SQL Server Management Studio (SSMS)** to host and manage the target database. Parrot OS was used as the attacker machine, equipped with tools like **sqlmap** to simulate real-world SQL injection attacks. Windows Server 2022 was available for administrative purposes and observation, where necessary. Once all machines were connected and the environment was configured, we proceeded to perform the SQL injection tests.

**Tools Used**

* **Microsoft SQL Server 2019** – Used as the backend database to simulate a real-world target for SQL injection.
* **SQL Server Management Studio (SSMS) 18.12** – Used to manage the SQL Server database, execute queries, and observe database behavior during attacks.
* **Parrot OS (Home Edition 5.3)** – Used as the attacker machine, equipped with pre-installed penetration testing tools.
* **sqlmap v1.7.3** – Automated tool used to detect and exploit SQL injection vulnerabilities in the database.
* **Oracle VirtualBox 7.0** – Virtualization software used to host and run multiple operating systems simultaneously.
* **Windows Server 2022** – Served as an optional administrative and monitoring system in our lab setup.

**Lab 1 Tasks and Execution Steps**

**Performed By: M. Abdullah Khan Mahsud 2023346**

**Title:** Perform SQL Injection Attacks

**Objectives:**

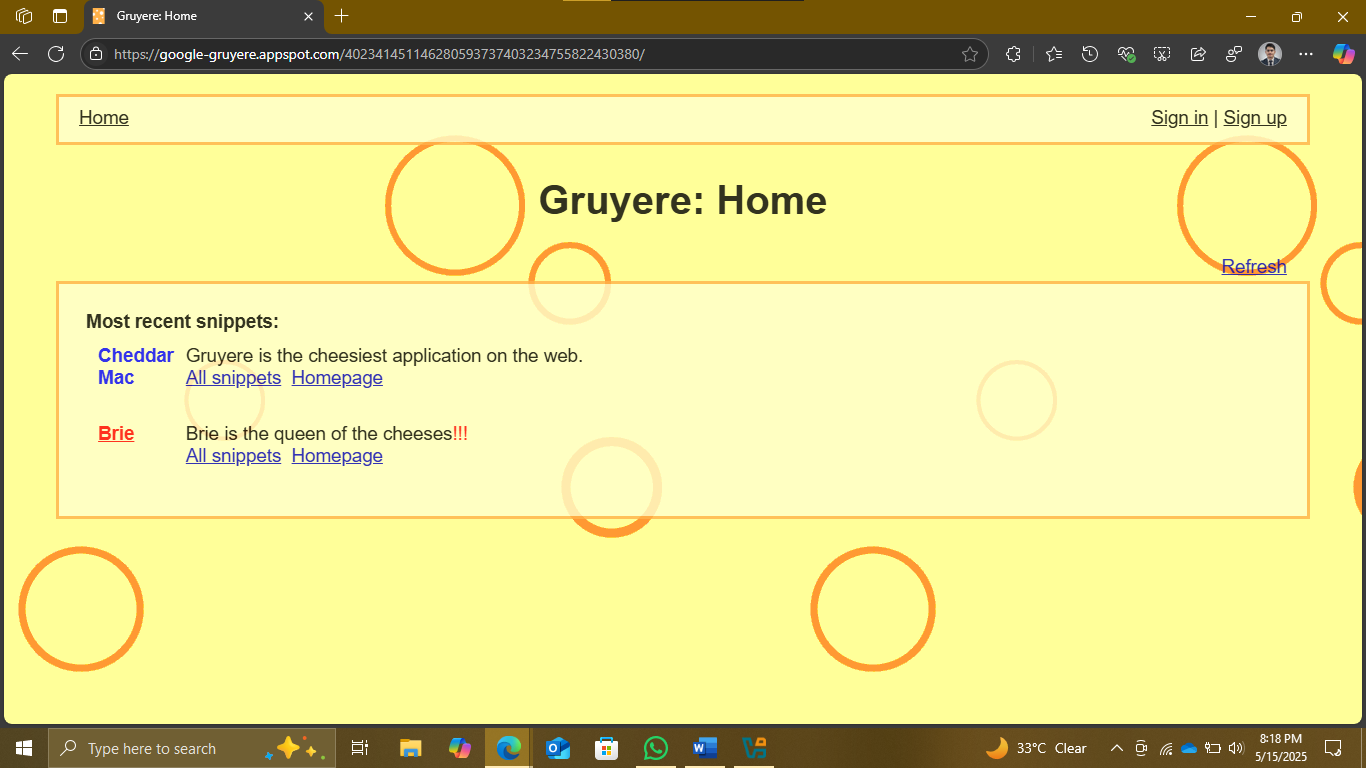
* Perform an SQL injection attack on an MSSQL database
* Perform an SQL injection attack against MSSQL to extract databases

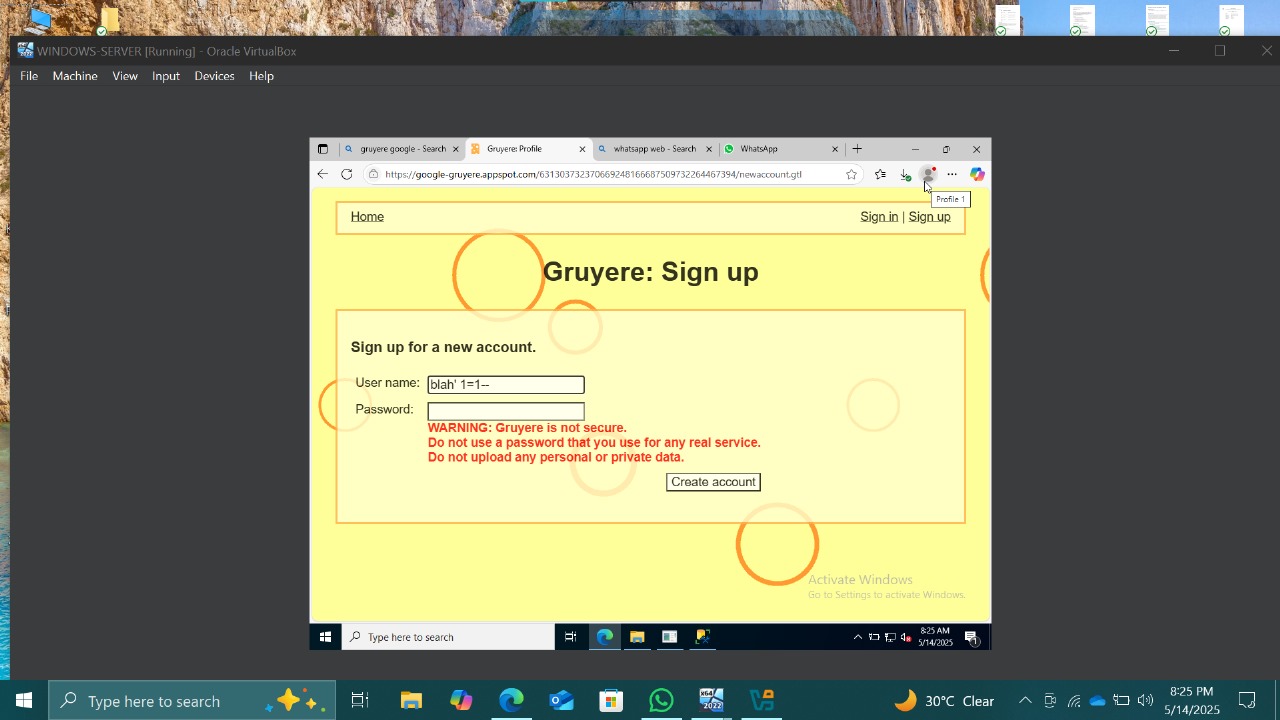
**Procedure:**

**Lab 1:**

In this task, the machine hosting the website (Windows Server 2019) is the victim machine and the Windows 11 machine will perform the attack.

1. Turn on the Windows 11 and Windows Server 2019 virtual machines. Switch to the Windows 11 virtual machine. Open any web browser, place the cursor in the address bar, type http://www.gruyere.com/, .
2. The Gruyere home page loads. Assume that you are new to this site and have never registered with it; click LOGIN on the menu bar.
3. In the Username field, type the query blah' or 1=1 -- as your login name, and leave the password field empty. Click the Log in button.



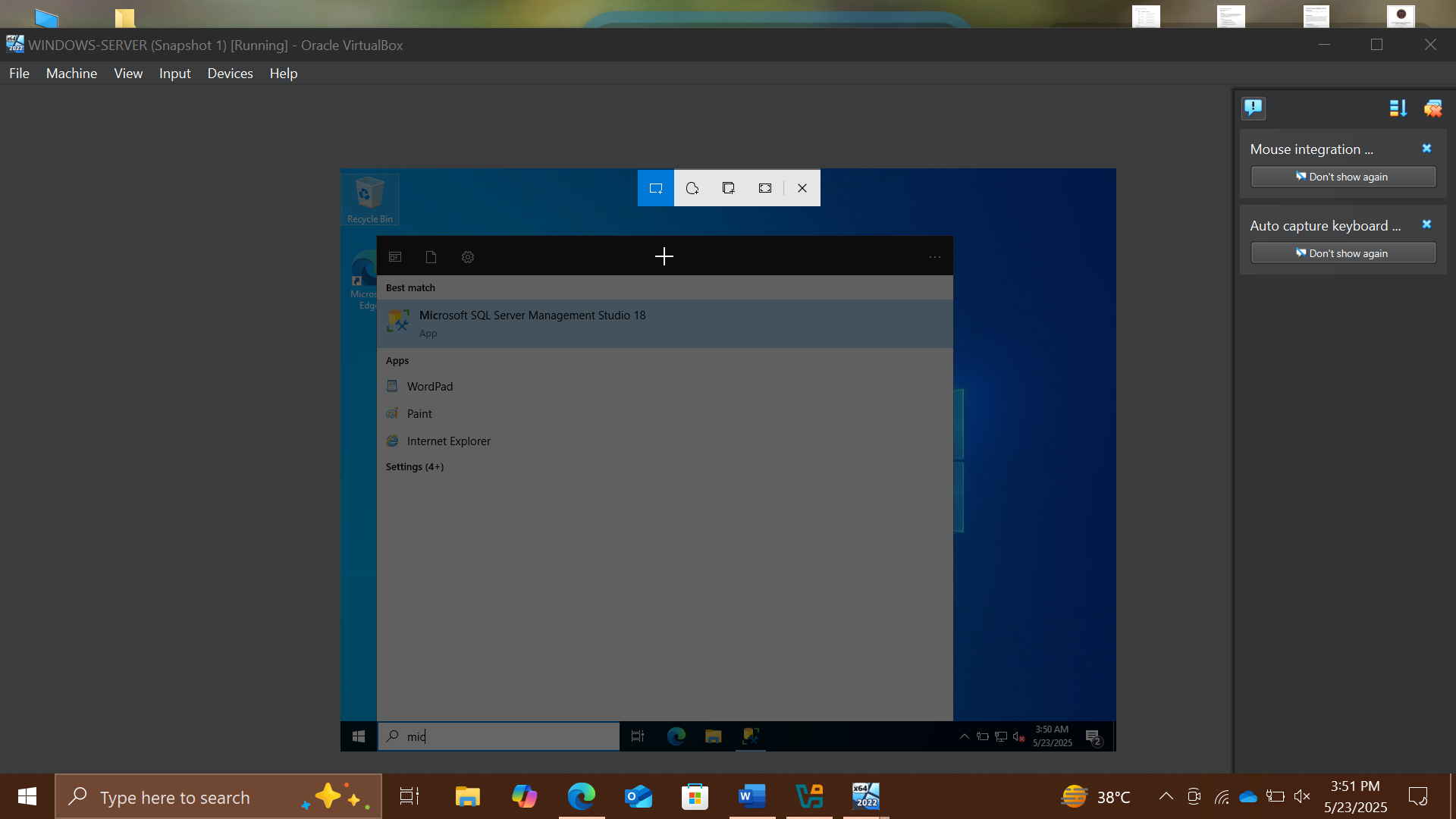


1. You are now logged into the website with a fake login, even though your credentials are not valid. Now, you can browse all the site’s pages as a registered member. After browsing the site, click Logout from the top-right corner of the webpage.

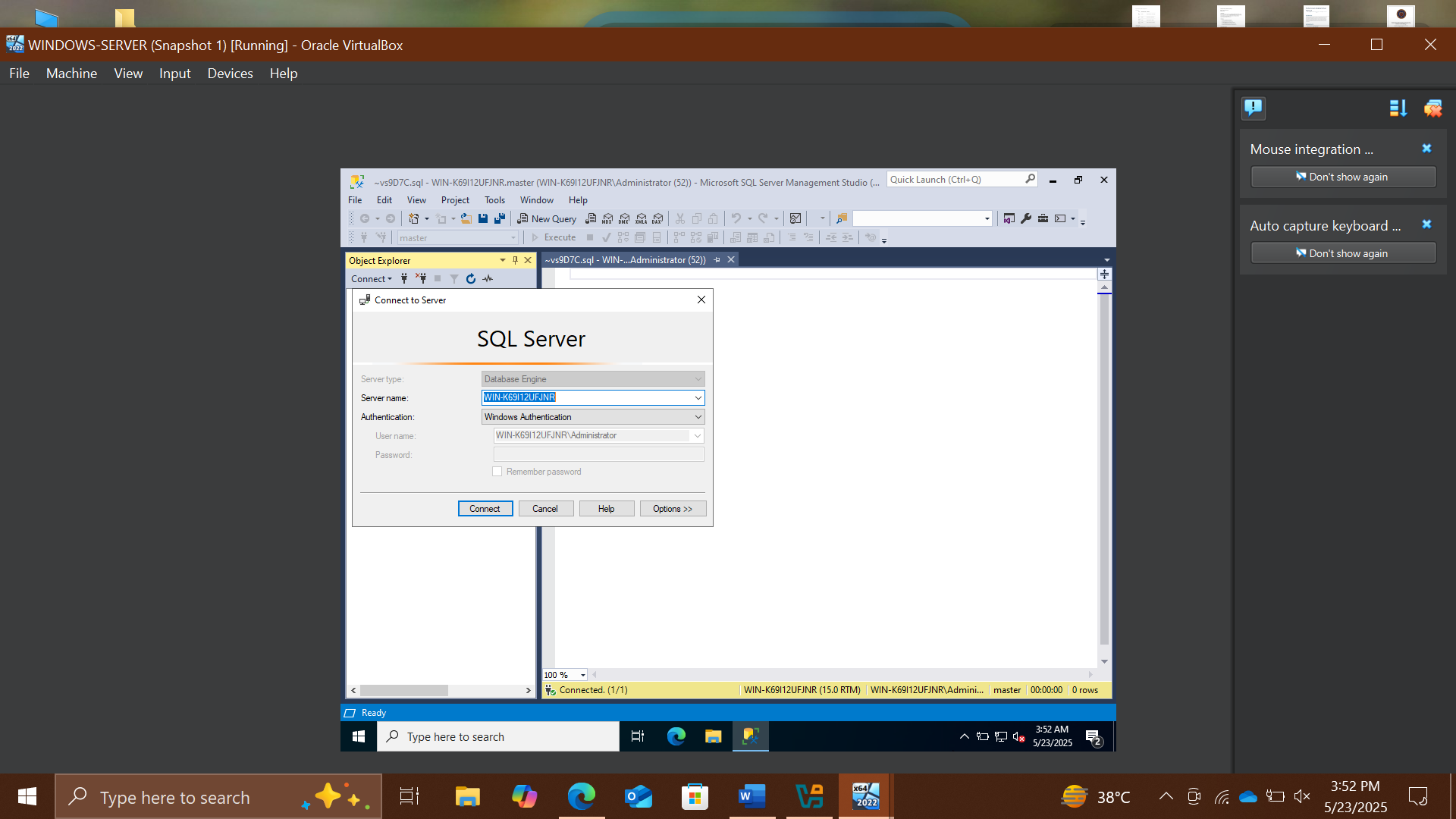
A screenshot of a computer

AI-generated content may be incorrect.

1. Now, we shall create a user account using the SQL injection query. Before proceeding with this sub-task, we shall first examine the login database of theGruyere website. Switch to the Windows Server 2019 virtual machine. In this task, we are logging into the Windows Server 2019 machine as a victim.



6. Microsoft SQL Server Management Studio opens, along with a Connect to Server pop- up. In the Connect to Server pop-up, leave the default settings as they are and click the Connect button.



1. In the left pane of the Microsoft SQL Server Management Studio window, under the Object Explorer section, expand the Databases node. From the available options, expand the Gruyere node, and then the Tables node under it.
2. Under the Tables node, right-click the dbo.Login file and click Select Top 1000 Rows from the context menu to view the available credentials.

A computer screen with a computer screen

AI-generated content may be incorrect.

1. You can observe that the database contains only one entry with the username and password alice and password123, respectively.

A screenshot of a computer

AI-generated content may be incorrect.

1. After browsing the required pages, click Logout from the top-right corner of the webpage.
2. You will observe that a new user entry has been added to the website’s login database file with the username and password as john and apple123, respectively. Note down the available databases.
3. Switch back to the Windows 11 virtual machine and the browser where the Gruyere website is open.
4. Click LOGIN on the menu bar and type the query blah';create database mydatabase; -- in the Username field (as your login name) and leave the password field empty. Click the Log in button.
5. In the above query, mydatabase is the name of the database.

## A screenshot of a computer AI-generated content may be incorrect.

## 

1. If no error message (or any message) displays on the webpage, it means that the site is vulnerable to SQL injection and a database with the name mydatabase has been created on the database server.
2. Switch back to the Windows Server 2019 virtual machine.
3. In the Microsoft SQL Server Management Studio window, un-expand the Databases node and click the Disconnect icon fa) and then click Connect Object Explorer icon & to connect to the database. In the Connect to Server pop-up, leave the default settings as they are and click the Connect button. By Fie
4. Expand the Databases node. A new database has been created with the name mydatabase, as shown in the screenshot.
5. Switch back to the Windows 11 virtual machine and the browser where the Gruyere website is open.
6. Click LOGIN on the menu bar and type the query blah'; DROP DATABASE mydatabase; - - in the Username field; leave the Password field empty and click Log in. Note: In the above query, you are deleting the database that you created in Step 15 (mydatabase). In the same way, you could also delete a table from the victim website database by typing blah’; DROP TABLE table\_name; -- in the Username field.

## A screenshot of a computer AI-generated content may be incorrect.

## A computer screen with a computer screen AI-generated content may be incorrect.

## Close the Microsoft SQL Server Management Studio window. 36. Switch back to the Windows 11 virtual machine and the browser where the GoodShopping website is open.

## Click LOGIN on the menu bar and type the query blah';exec master..xp\_cmdshell ‘ping www.certifiedhacker.com -I 65000 -t'; -- in the Username field; leave the Password field empty and click Log in. Note: In the above query, you are pinging the www.certifiedhacker.com website using an SQL injection query. -l is the sent buffer size and -t refers to pinging the specific host.

## 

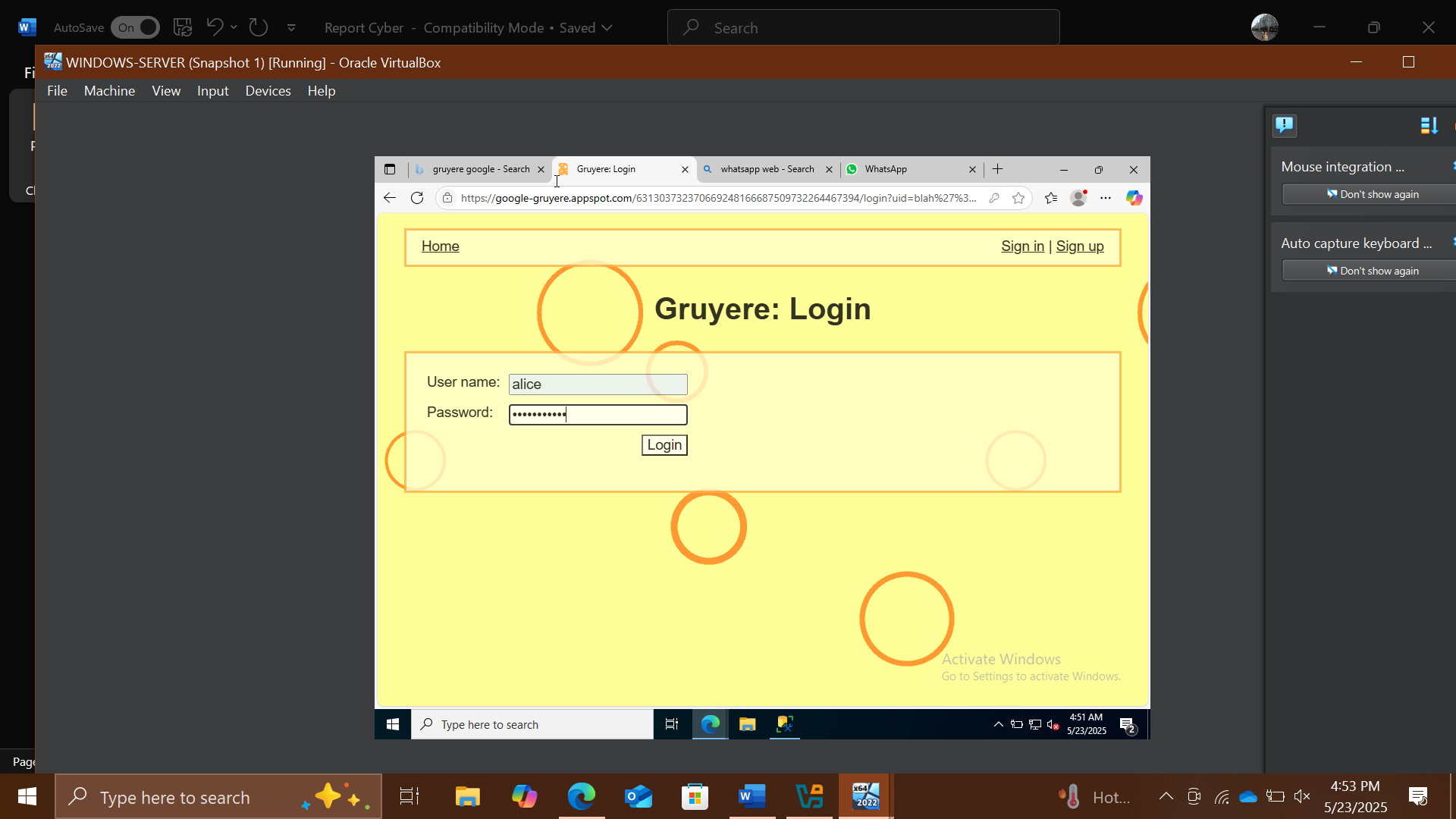
## 

## Part 2

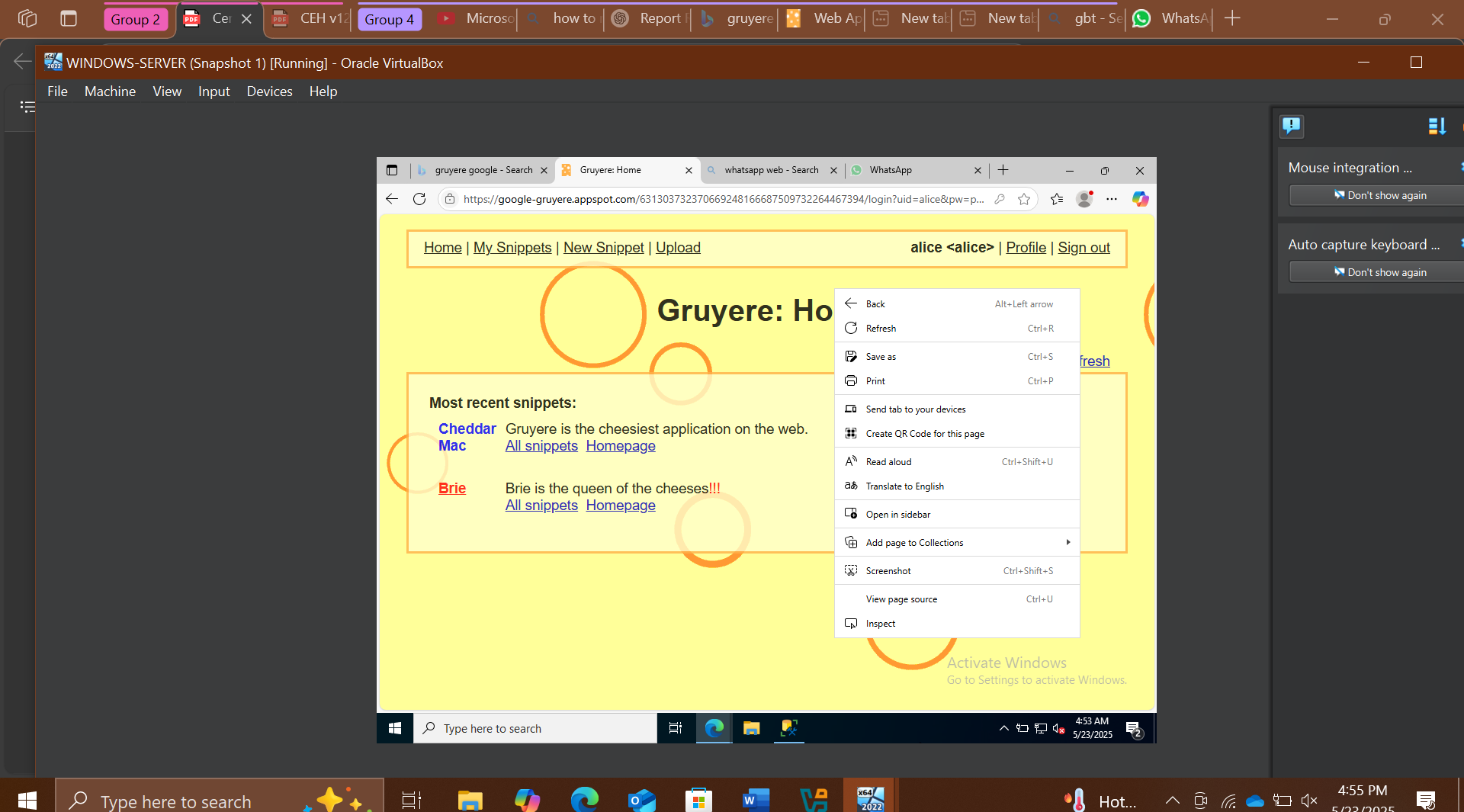
In this task, you will pretend that you are a registered user on the http://www.gruyere.com website, and you want to crack the passwords of the other users from the website’s database.

Note: Ensure that the Windows Server 2019 virtual machine is running.

1. Turn on the Parrot Security virtual machine.
2. In the login page, the attacker username will be selected by default.
3. Enter password as toor in the Password field and press Enter to log in to the machine. Note: If a Question pop-up window appears asking you to update the machine, click No to close the window.
4. Click the Mozilla Firefox icon from the menu bar in the top-left corner of Desktop to launch the web browser. Type http://www.gruyere.com/ and press Enter. A Login page loads; enter the Username and Password as alice and password, respectively.
5. Click the Login button.



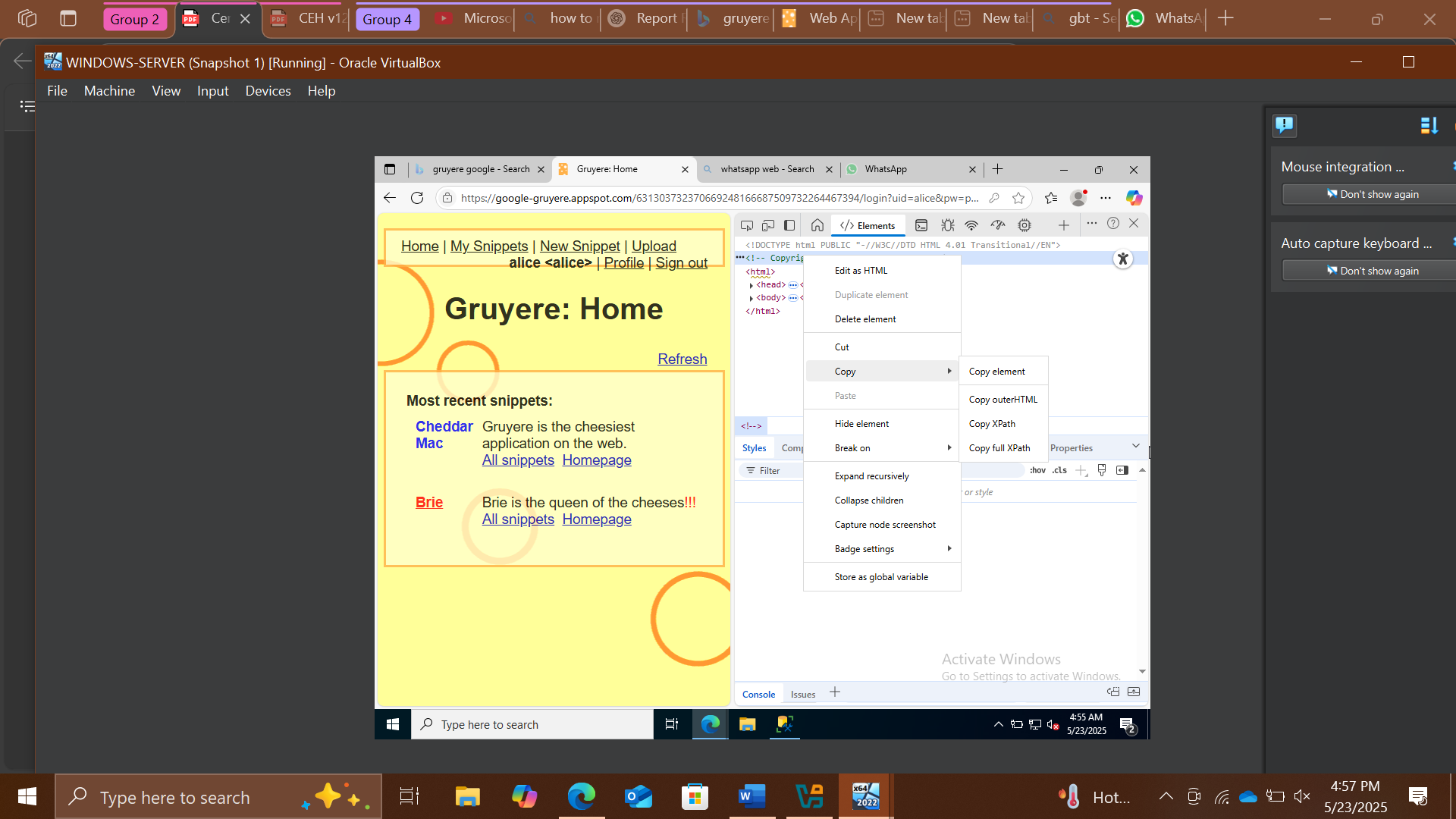
1. Once you are logged into the website, click the View Profile tab from the menu bar; and when the page has loaded, make a note of the URL in the address bar of the browser.
2. Right-click anywhere on the webpage and click Inspect Element (Q) from the context menu, as shown in the screenshot.



A screenshot of a computer

AI-generated content may be incorrect.

1. The Developer Tools frame appears in the lower section of the browser window. Click the Console tab, type document.cookie in the lower-left corner of the browser, and press Enter.
2. Select the cookie value, then right-click and copy it, as shown in the screenshot. Minimize the web browser.

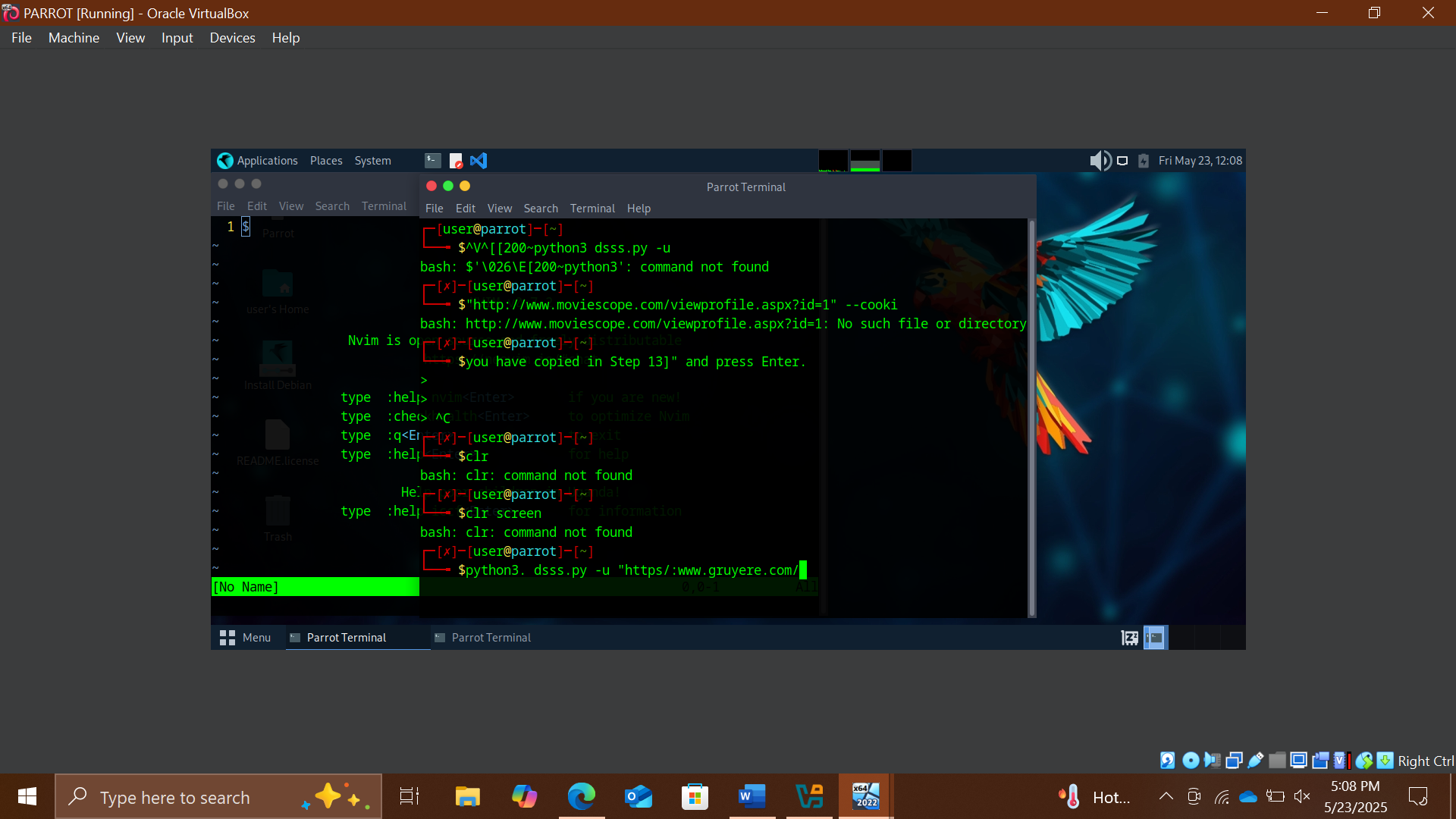


10. Select the cookie value, then right-click and copy it, as shown in the screenshot. Minimize the web browser.

11. Switch to a terminal window and type python3 dsss.py -u "http://www.gruyere.com/viewprofile.aspx?id=1" --cooki you have copied in Step 13]" and press Enter. " [cookie value which Note: In this command, -u specifies the target URL and --cookie specifies the HTTP cookie header value.

12. The above command causes DSSS to scan the target website for SQL injection vulnerabilities.

13. The result appears, showing that the target website is vulnerable to blind SQL injection attacks. The vulnerable link is also displayed, as shown in the screenshot.

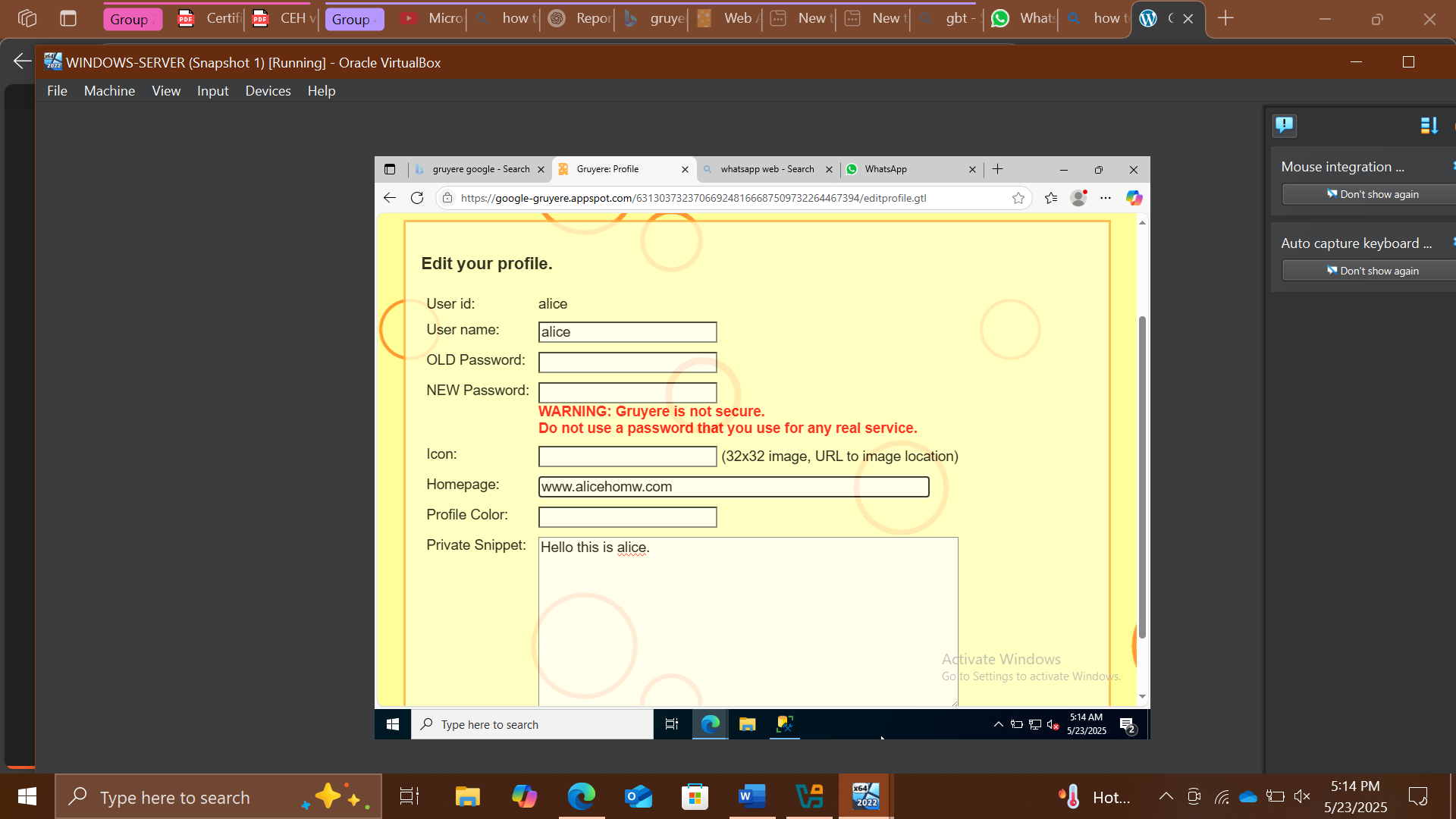


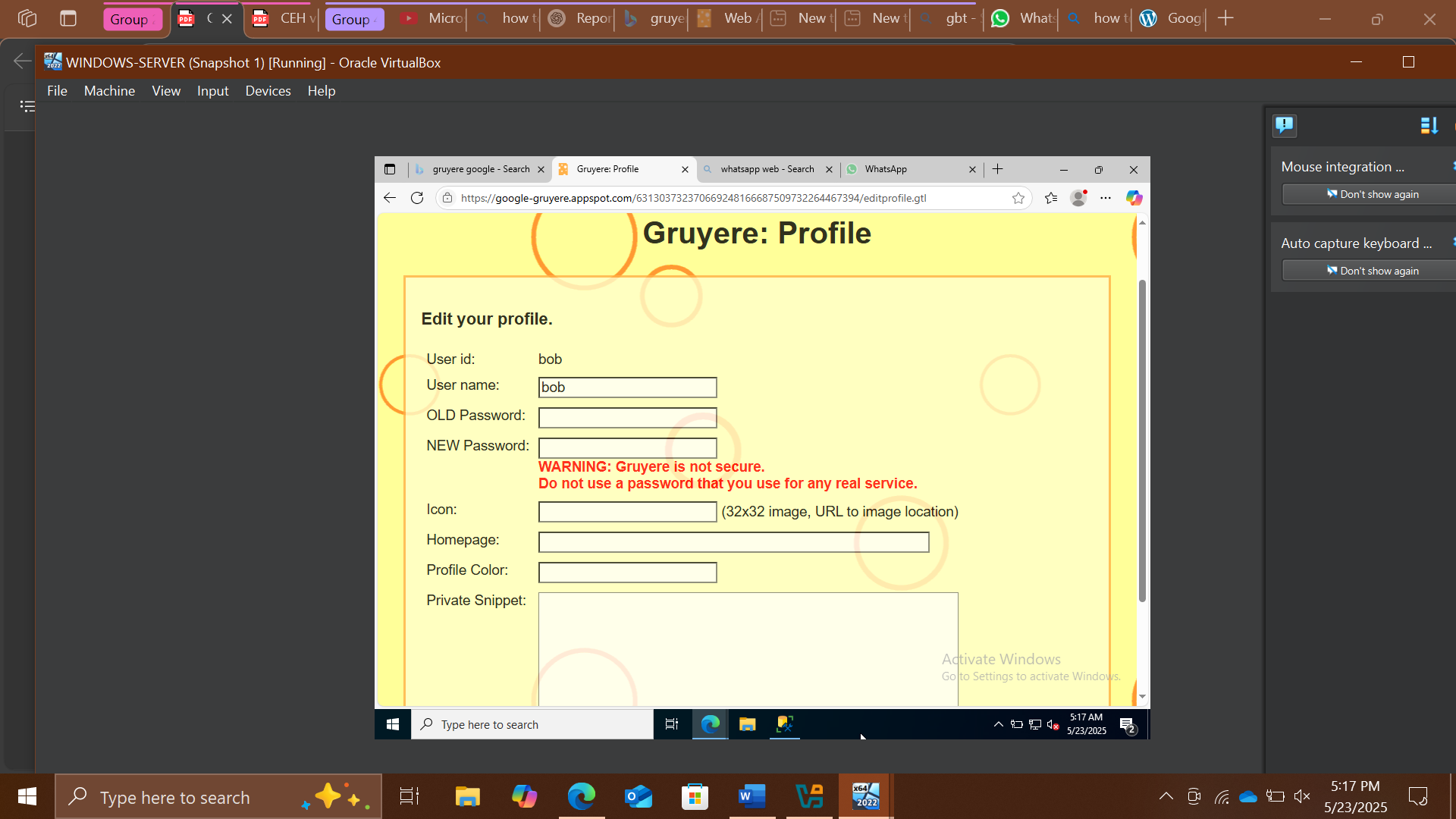
14. The above command causes DSSS to scan the target website for SQL injection vulnerabilities.

15. The result appears, showing that the target website Gruyere is vulnerable to blind SQL injection attacks. The vulnerable link is also displayed, as shown in the screenshot.

16. Switch to Mozilla Firefox; in a new tab, paste the copied link in the address bar and press Enter.

17. You will observe that information regarding available user accounts appears under the View Profile tab.





18. This concludes the demonstration of how to detect SQL injection vulnerabilities using DSSS.

19. Close all open windows and document all the information acquired.

20. Turn off the Parrot Security virtual machine.

----------------------------------------END LAB 1--------------------------------------

5. Methodology

- Identified vulnerable input fields on DVWA.  
- Used manual techniques to perform SQL injection (e.g., ' OR '1'='1).  
- Used sqlmap to automate detection and exploitation.  
- Documented behavior of application and data returned.

## 6. Types of SQL Injection Explored

- Classic SQLi: Basic payloads to bypass login forms.  
- Error-based SQLi: Observing SQL errors to infer information.  
- Union-based SQLi: Using UNION SELECT statements.  
- Blind SQLi: No error messages; inference through Boolean or Time delays.

## 7. Prevention Techniques Studied

- Input Validation and Sanitization  
- Use of Prepared Statements (Parameterized Queries)  
- ORM frameworks that abstract SQL layers  
- Disabling detailed error messages

## 8. Results and Observations

I saw that without proper input validation, the input field for user is very dangerous and I being a Cybersecurity student saw how to exploit this vulnerability and also learned to protect against it.

So for the final result. I can say that I successfully exploited the vulnerability in my Task.

## 9. Challenges and Errors Encountered

We faced numerous difficulties from the very beginning and until the very end. But throught out this I managed to complete the task assigned to me.

## 10. Conclusion

SQL Injection remains a dangerous but preventable vulnerability. Through this project, we explored various SQLi techniques, tools, and defenses. This hands-on practice strengthened our understanding of attack surfaces and security hardening in web applications.

## 11. References

- EC-Council, CEH v12 Module 15  
- OWASP SQL Injection: https://owasp.org/www-community/attacks/SQL\_Injection  
- DVWA GitHub Repository  
- sqlmap Documentation: https://sqlmap.org