# Họ và tên: Trần Minh Triết

# MSSV: SE172241

# Class: IA1702

# Question:

**What is a Union-Based SQL Injection attack, and how does it differ from other types of SQL injection attacks?**

Classic SQL Injection:

In classic SQL Injection, attackers manipulate the structure of the original query by injecting malicious SQL code. This might involve commenting out parts of the original query, adding conditions that are always true, or using techniques like Boolean-Based Injection.

Time-Based Blind SQL Injection:

Time-Based Blind SQL Injection involves injecting malicious SQL code that causes the application to delay its response, revealing information about the database based on the delay. It doesn't rely on the UNION operator and is typically used when the application doesn't directly display the results of the query.

Error-Based SQL Injection:

Error-Based SQL Injection exploits error messages generated by the database to extract information about the database structure. This type of attack doesn't necessarily involve the UNION operator but focuses on causing the database to reveal information through error messages.

Stored and Reflected SQL Injection:

Stored SQL Injection occurs when the malicious input is stored persistently in the database, affecting subsequent users. Reflected SQL Injection involves injecting malicious code into a URL or form input, and the injected code is reflected back in the application's response.

Union-Based SQL Injection is just one variant of SQL Injection attacks, and its success relies on the ability to inject a UNION SELECT statement into the original query. Application developers can prevent SQL Injection by using parameterized queries or prepared statements, input validation, and proper user input sanitation. Additionally, web application firewalls (WAFs) can help detect and block SQL Injection attempts.

**Explain the role of the UNION SQL operator in this type of attack and how it can be used to extract data from a database that is not typically accessible through the intended use of a web application.**

# The UNION SQL operator plays a crucial role in Union-Based SQL Injection attacks by allowing an attacker to combine the results of the original SQL query with the results of a crafted query. This operator is typically used to merge the columns from two SELECT statements into a single result set. Here's a step-by-step explanation of how the UNION operator is leveraged in a Union-Based SQL Injection attack and how it helps extract data that is not typically accessible:

# Identifying the Number of Columns:

# The first step for an attacker is to determine the number of columns in the original SELECT statement. This is achieved by injecting an ORDER BY clause with a large number, and then incrementing the number until an error is encountered or until a successful union is formed.

# Crafting the UNION SELECT Statement:

# With the number of columns known, the attacker crafts a UNION SELECT statement with the same number of columns as the original query. The data types of the columns in the injected query should match the data types of the corresponding columns in the original query.

# Extracting Data:

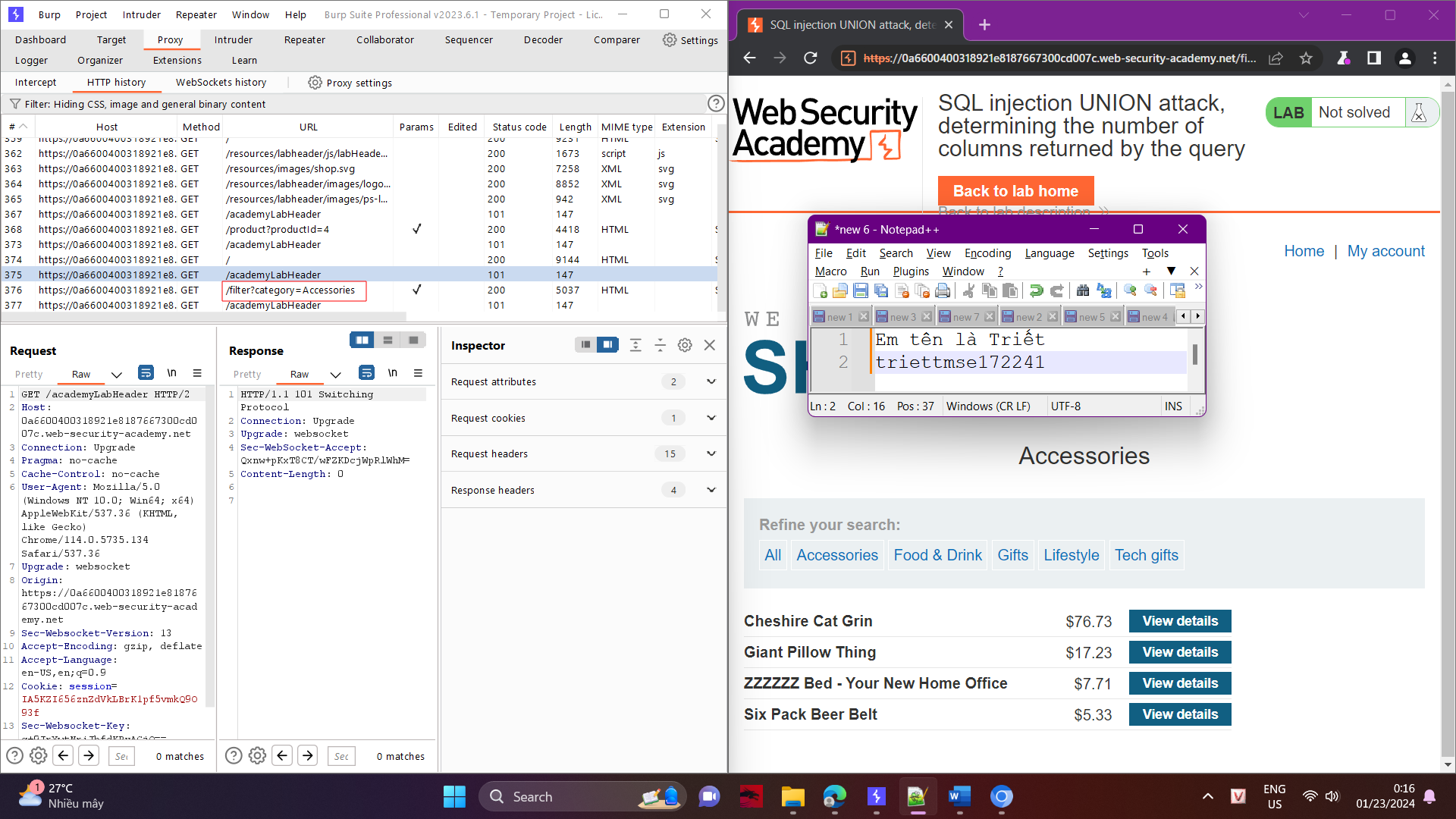
# When the malicious payload is executed, the result set will include rows from both the original query and the injected query. The attacker can then examine the response to extract sensitive information.

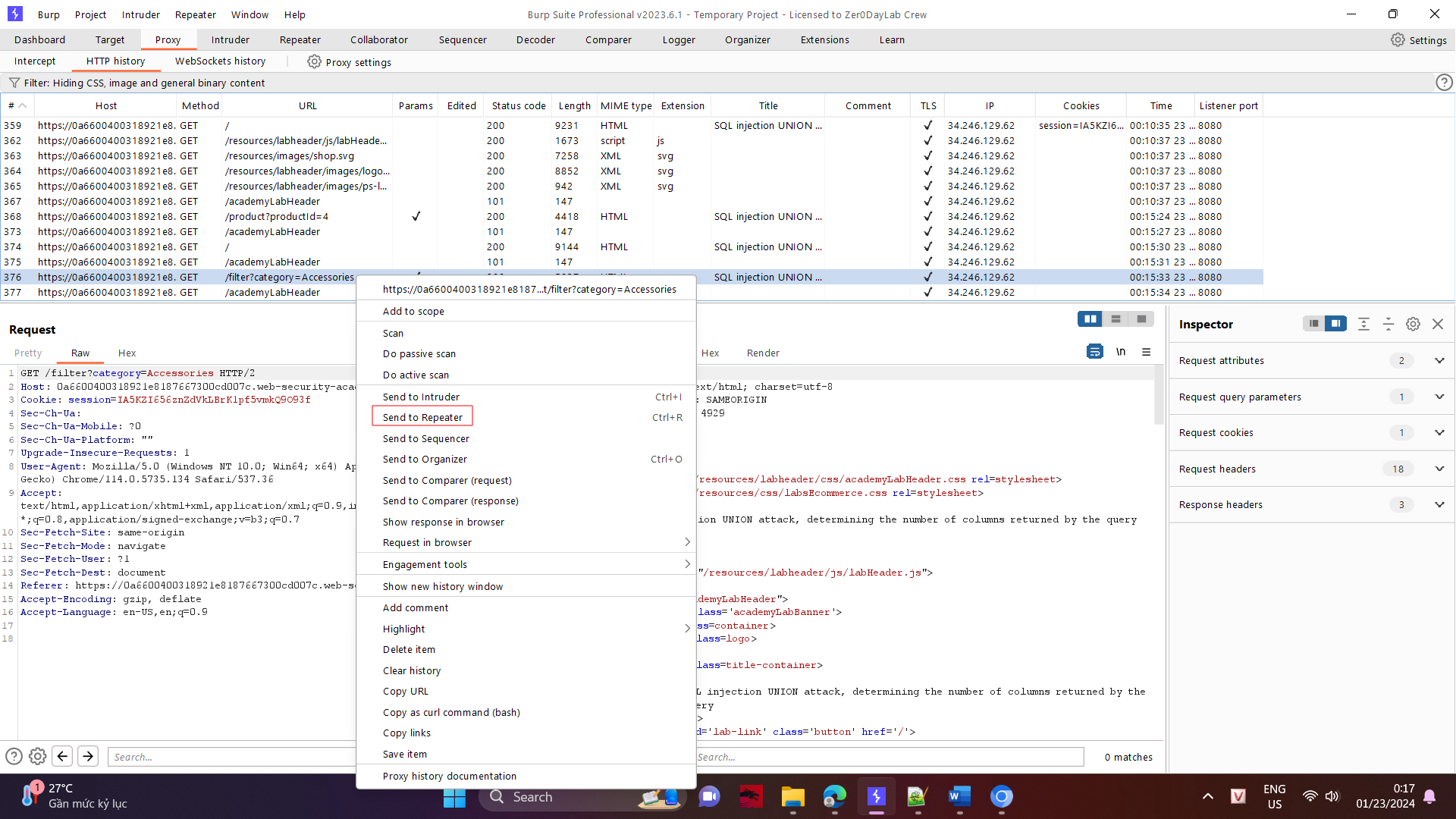
# For example, if the original query retrieves user information, the attacker might inject a UNION SELECT statement to retrieve information from another table, such as usernames and passwords.

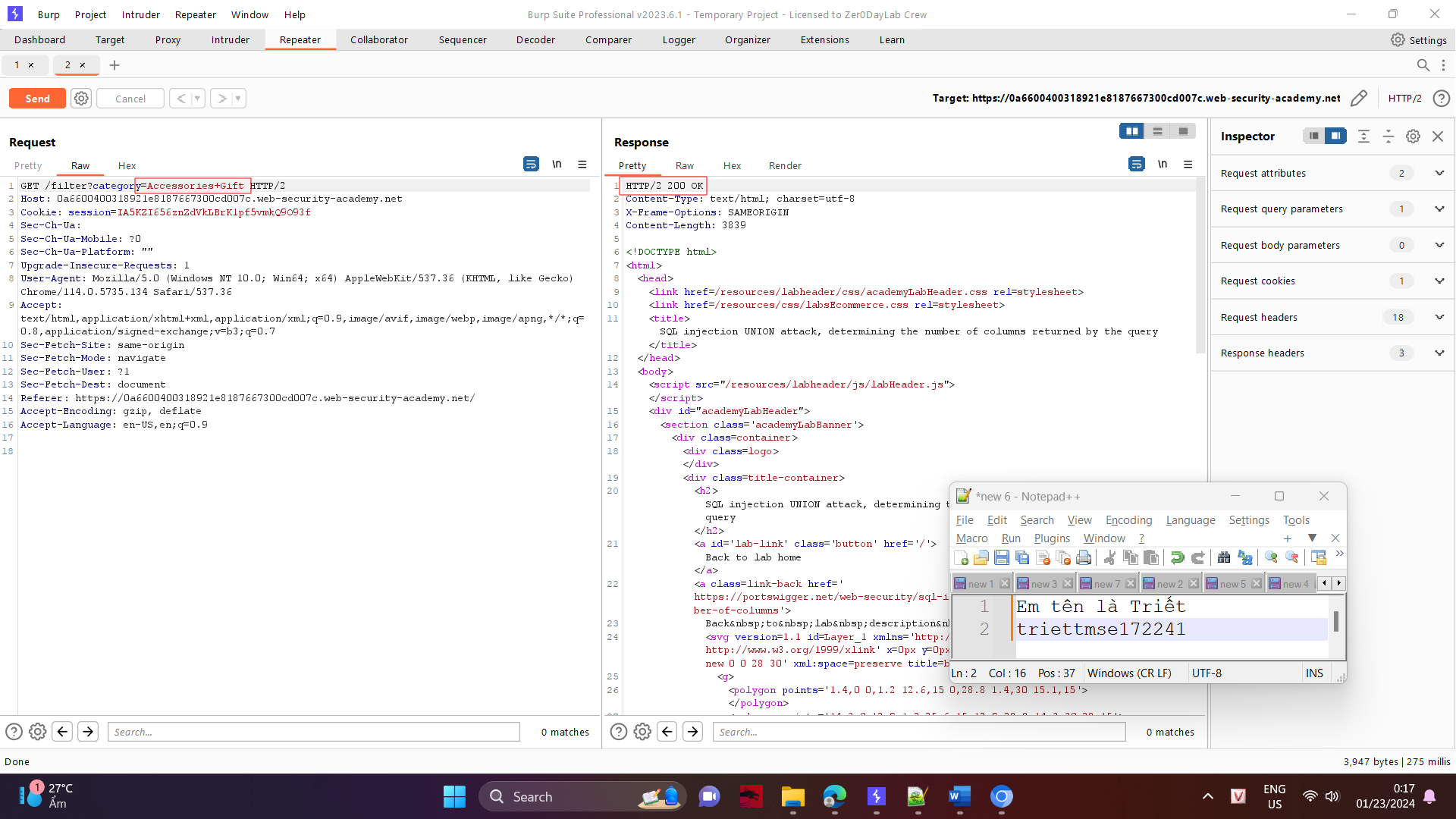
# Lab 9: SQL injection UNION attack, determining the number of columns returned by the query

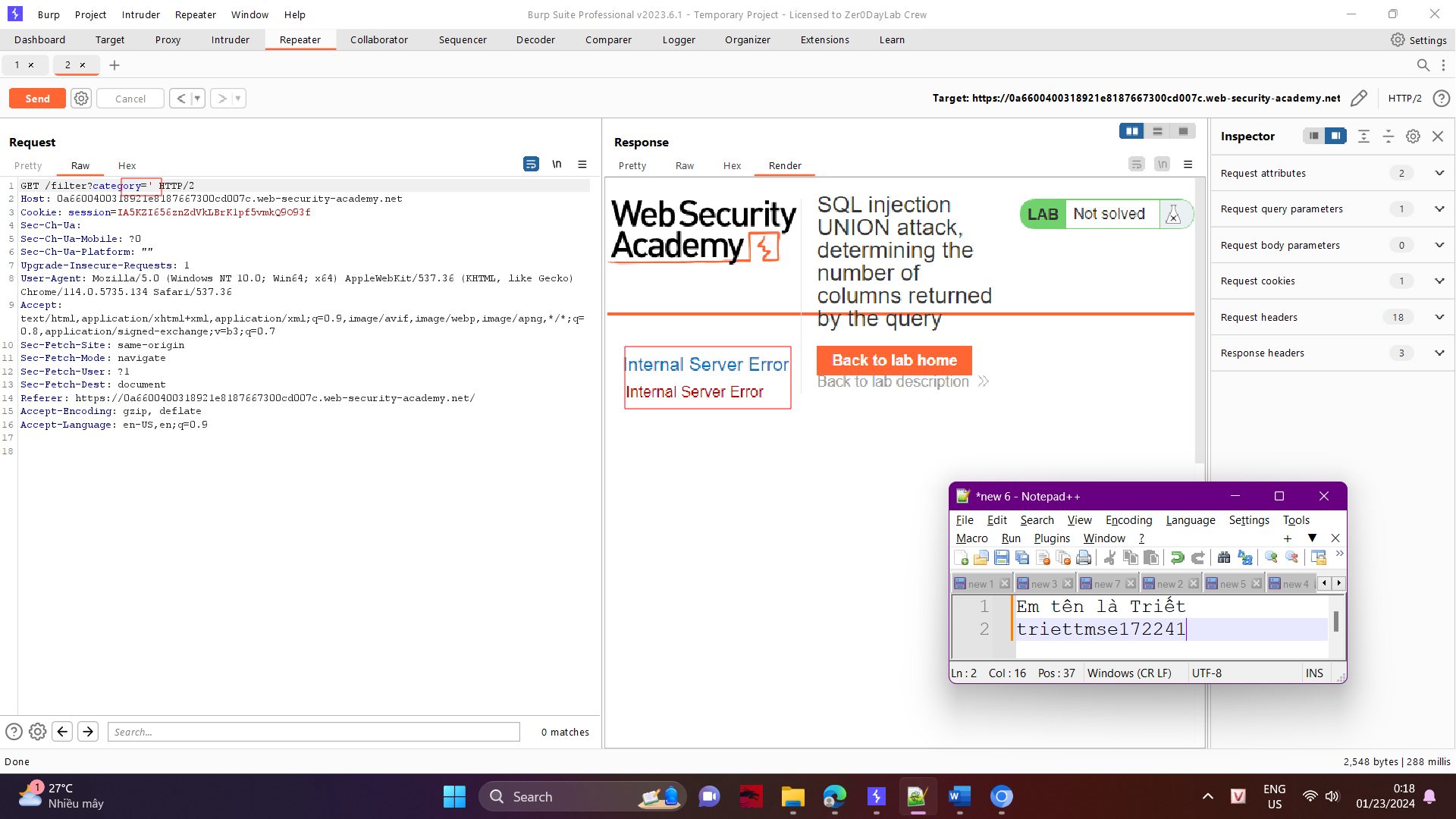
# 

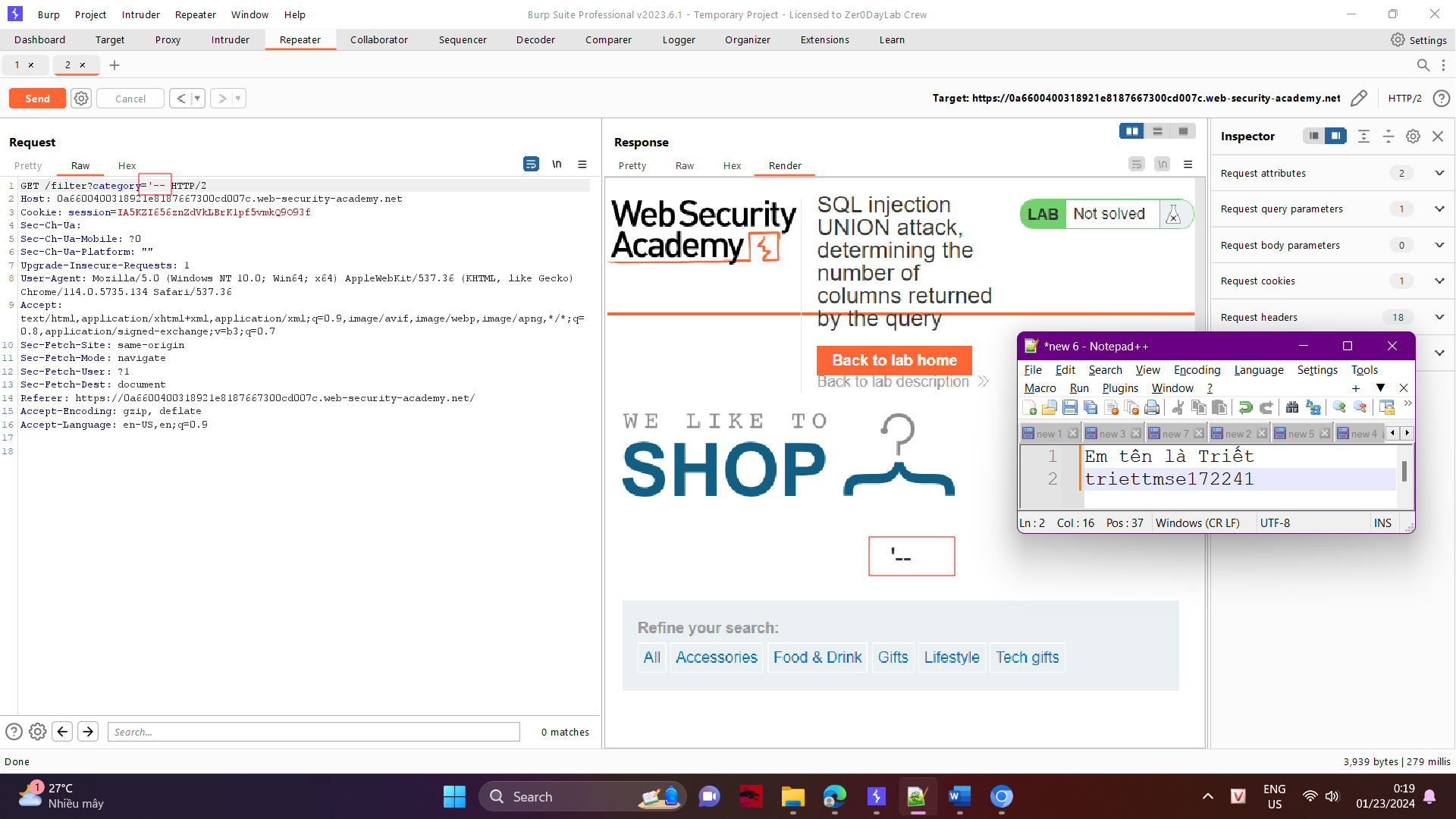
1. Use Burp Suite to intercept and modify the request that sets the product category filter.



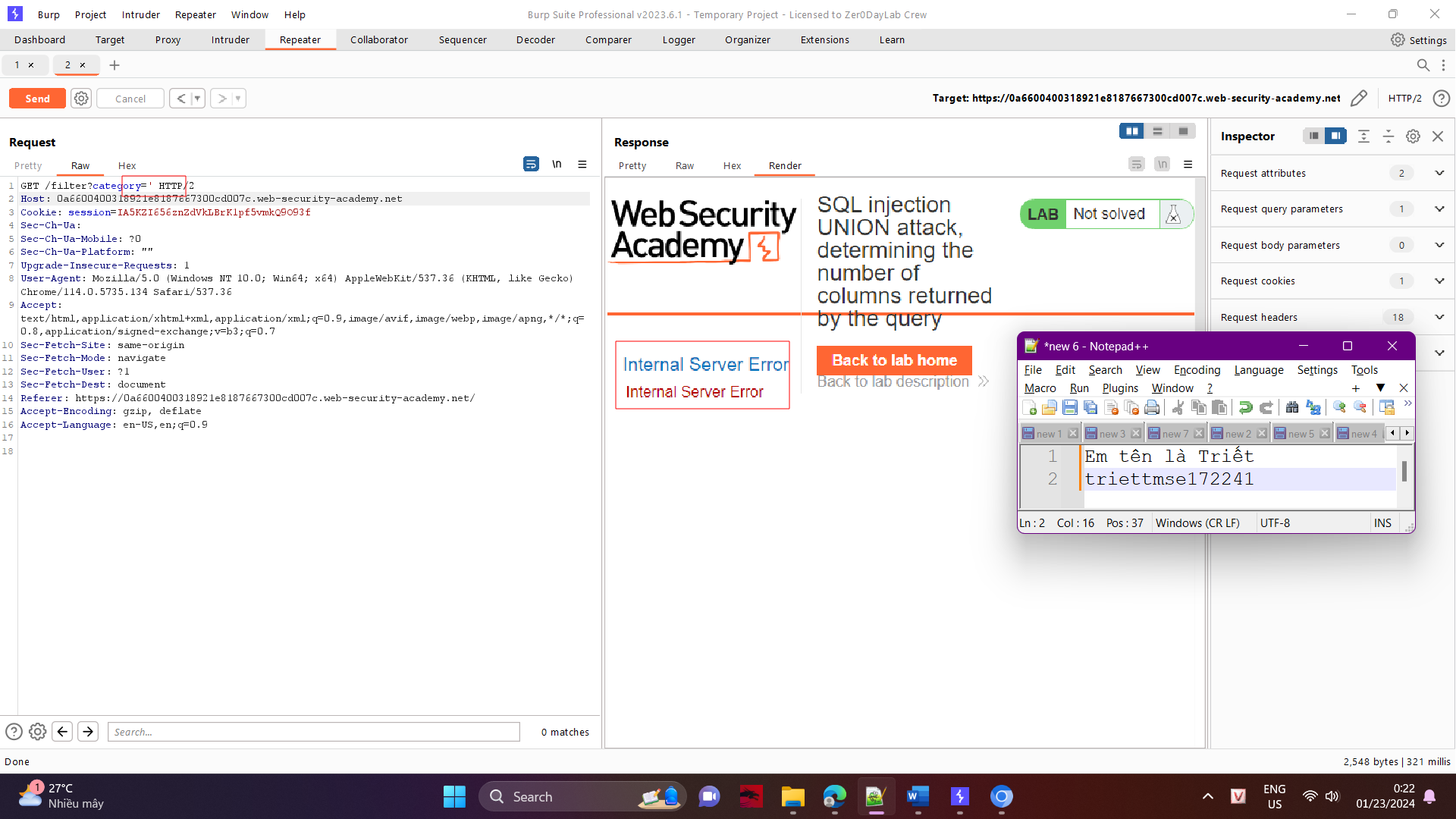






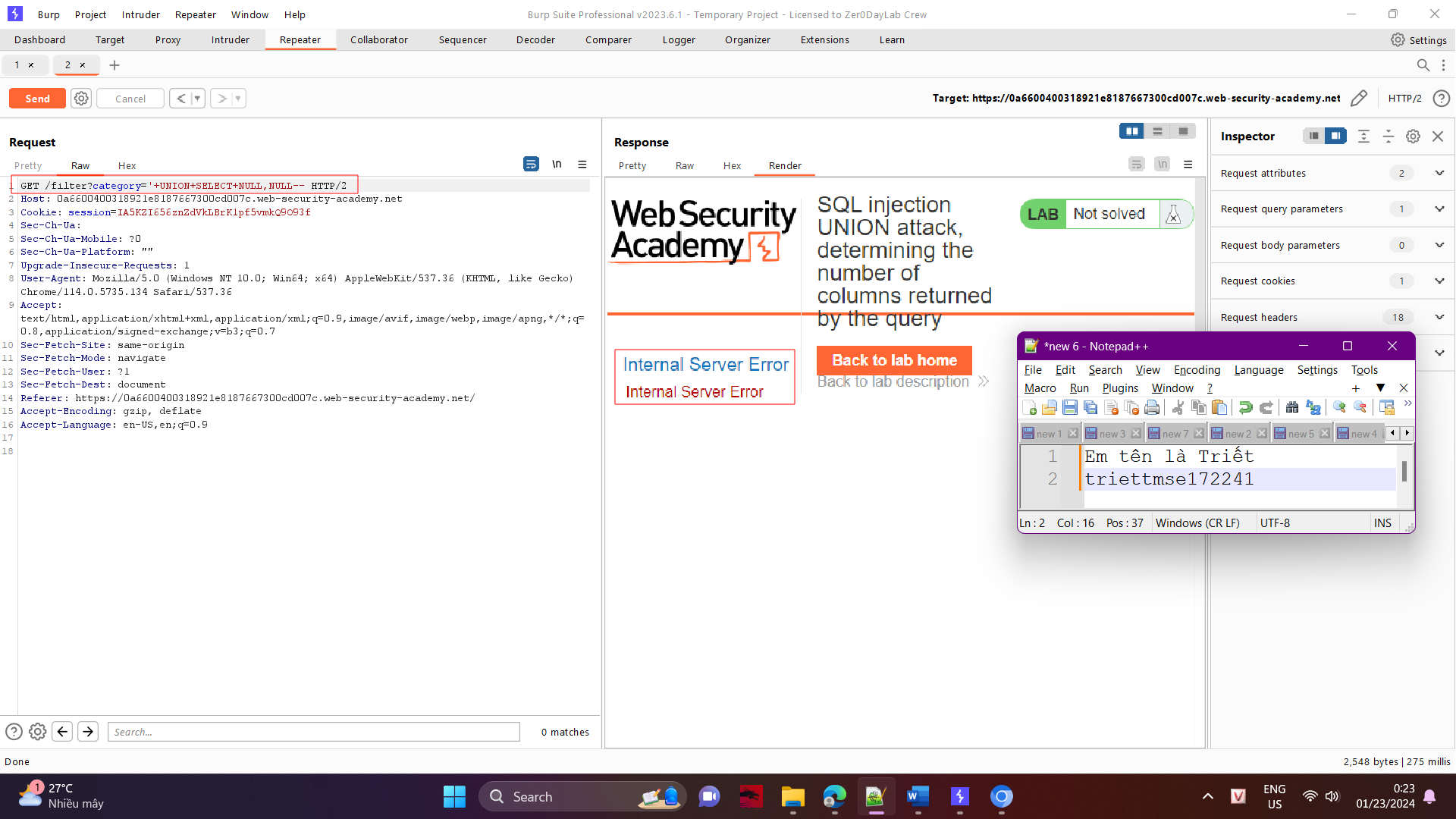


1. Modify the category parameter, giving it the value '+UNION+SELECT+NULL--. Observe that an error occurs.

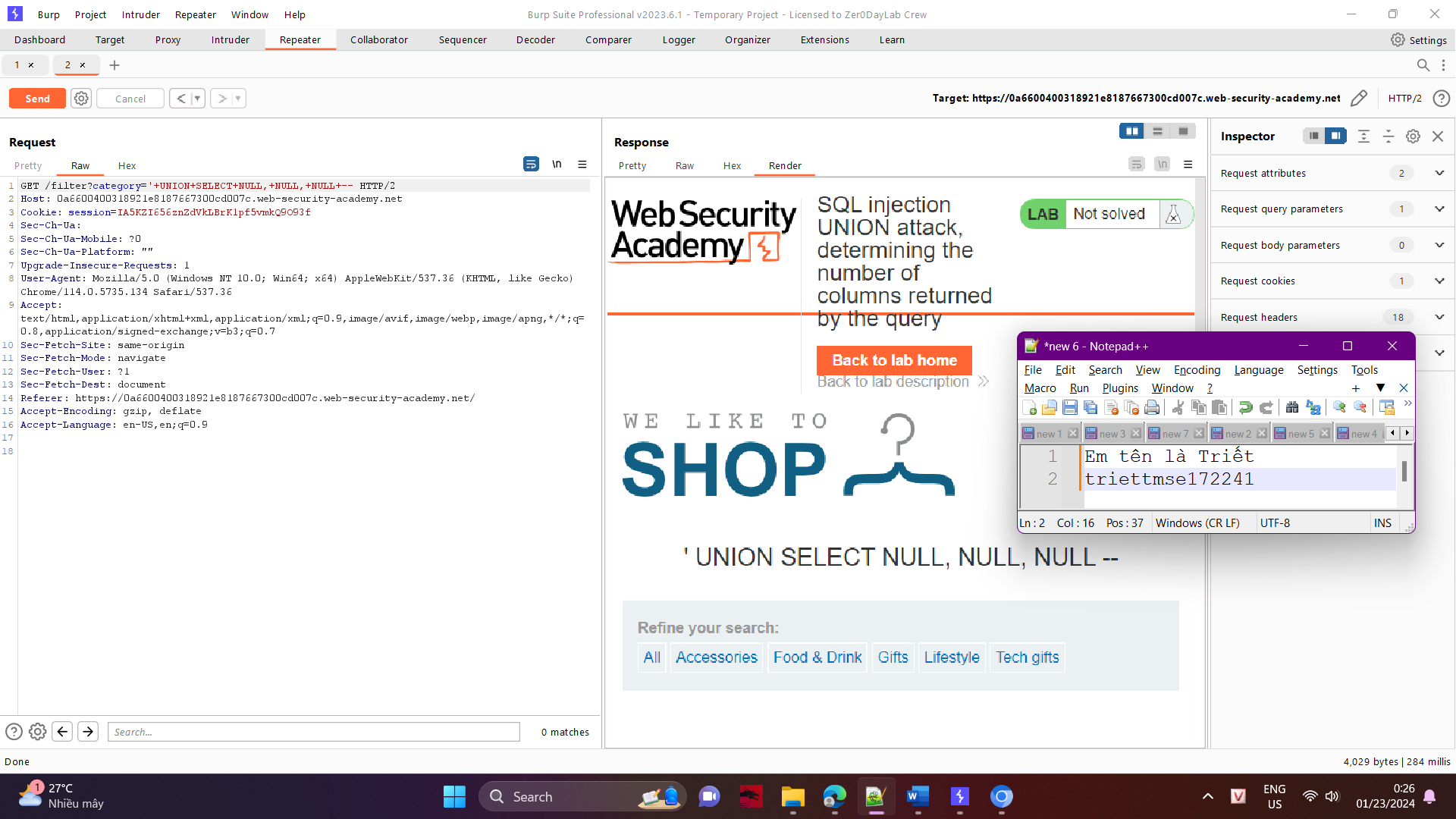


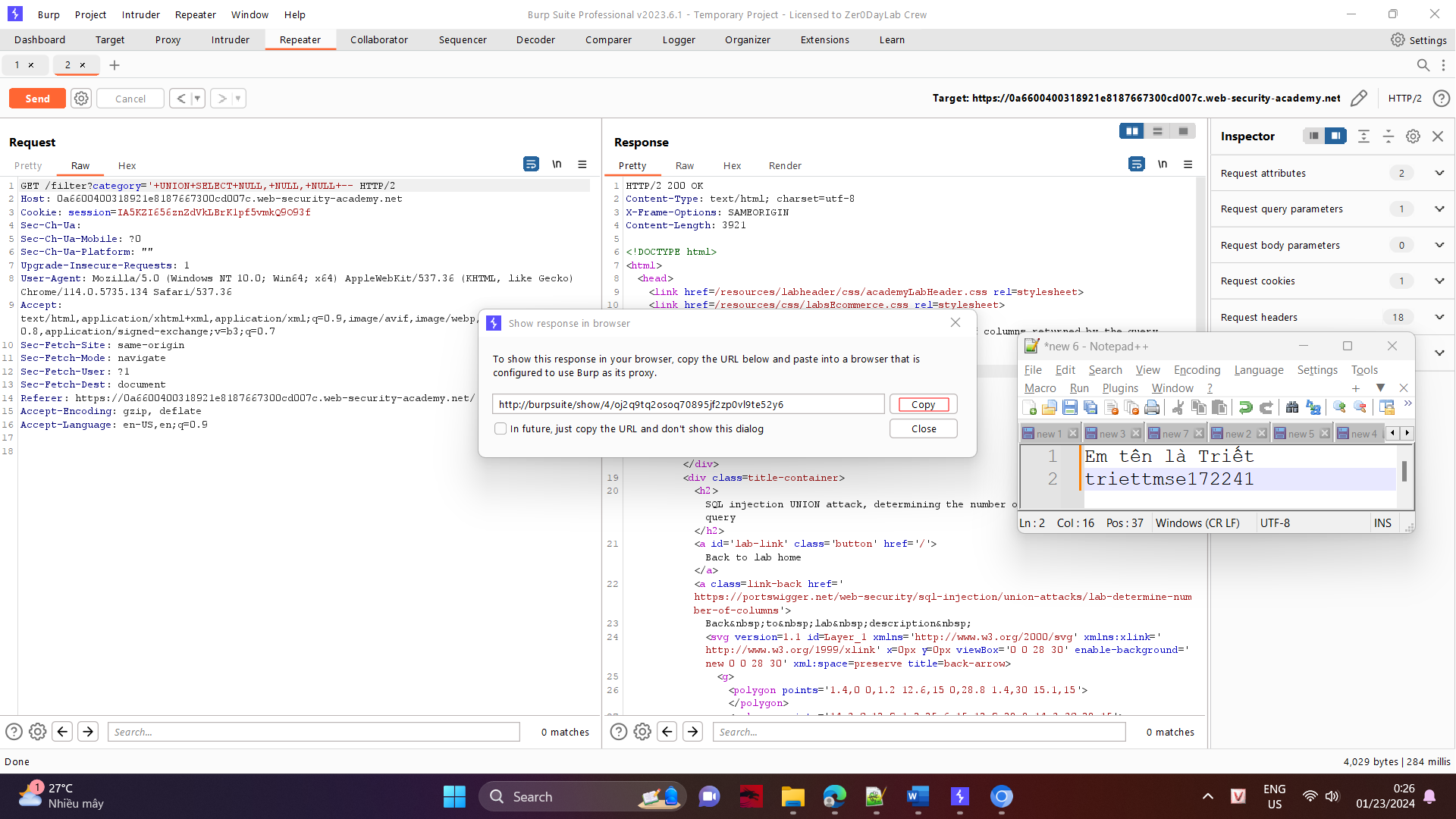
1. Modify the category parameter to add an additional column containing a null value:

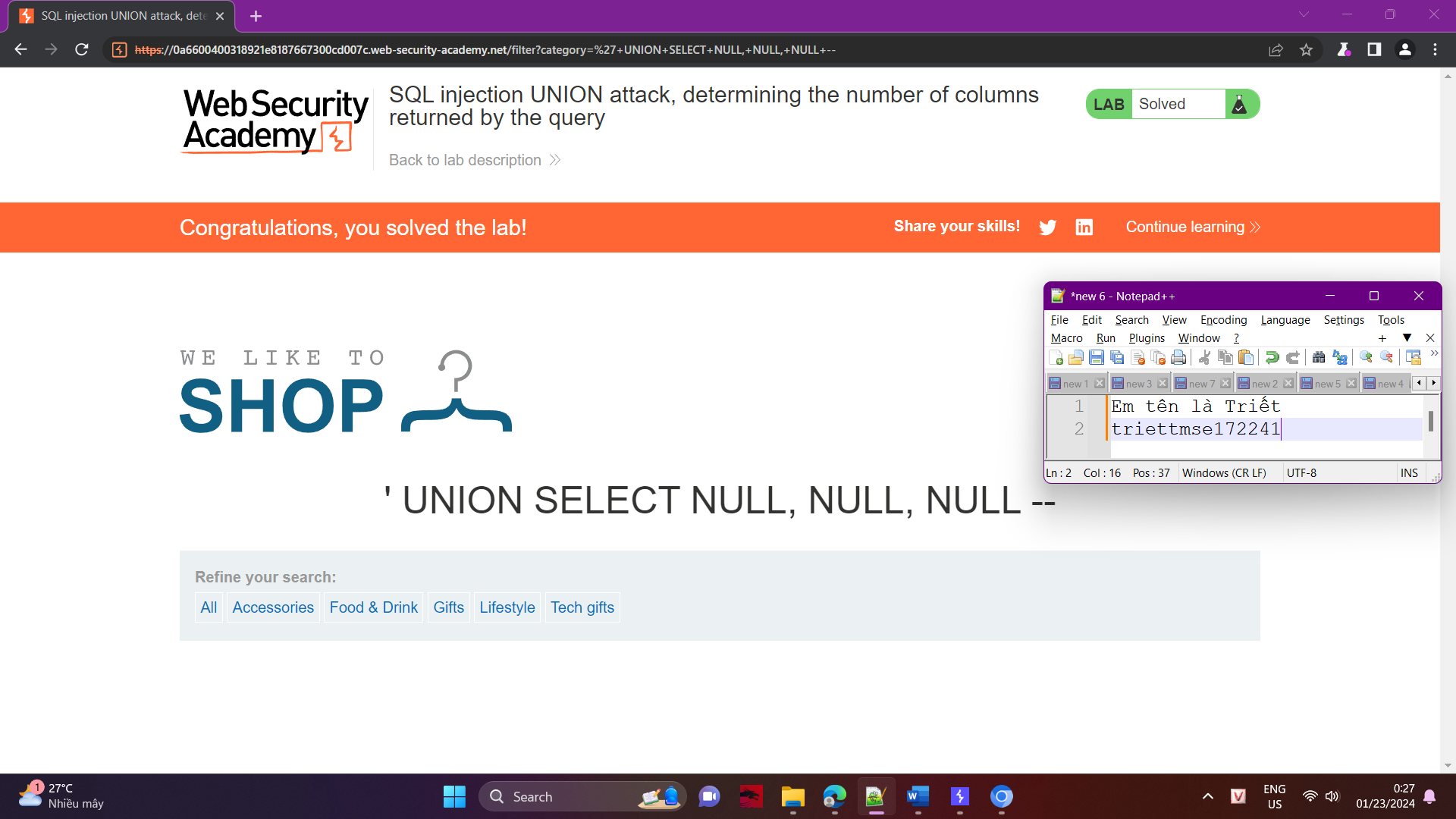
'+UNION+SELECT+NULL,NULL—



1. Continue adding null values until the error disappears and the response includes additional content containing the null values.







Done!