Tên: Trần Chí Hào

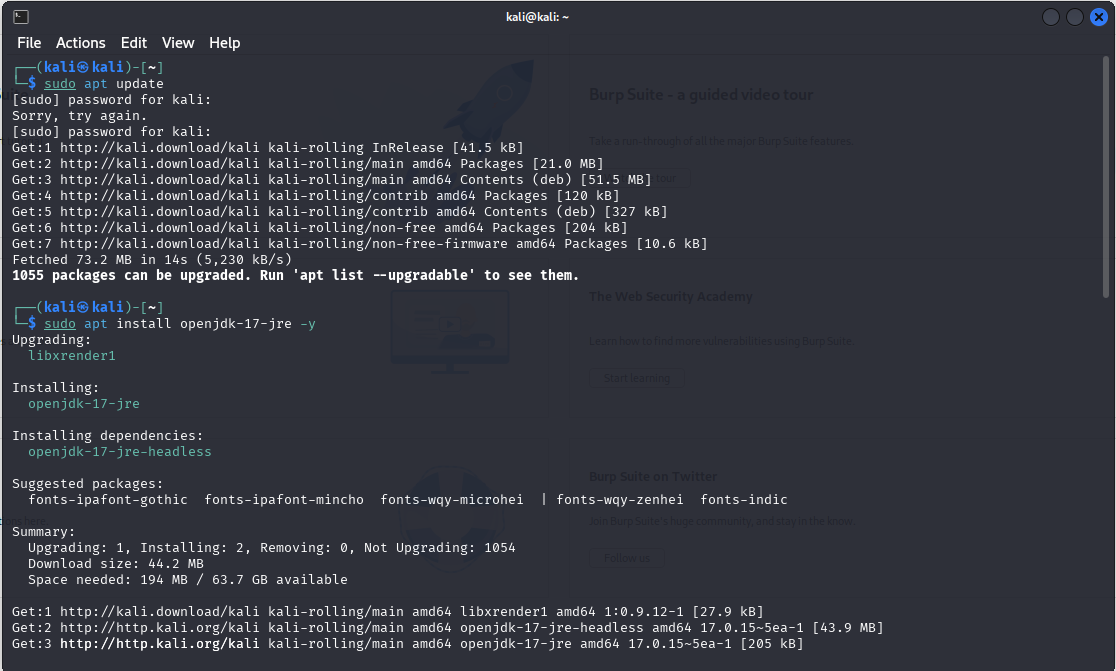
Lớp: 10\_ĐH\_CNPM1

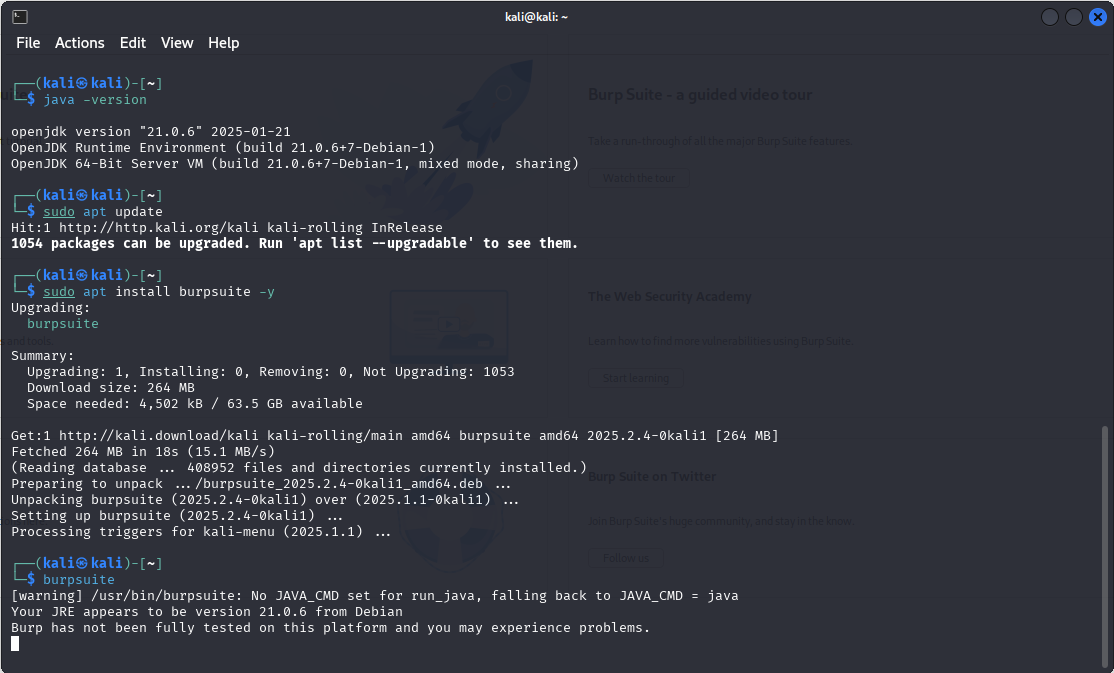
MSSV: 1050080012

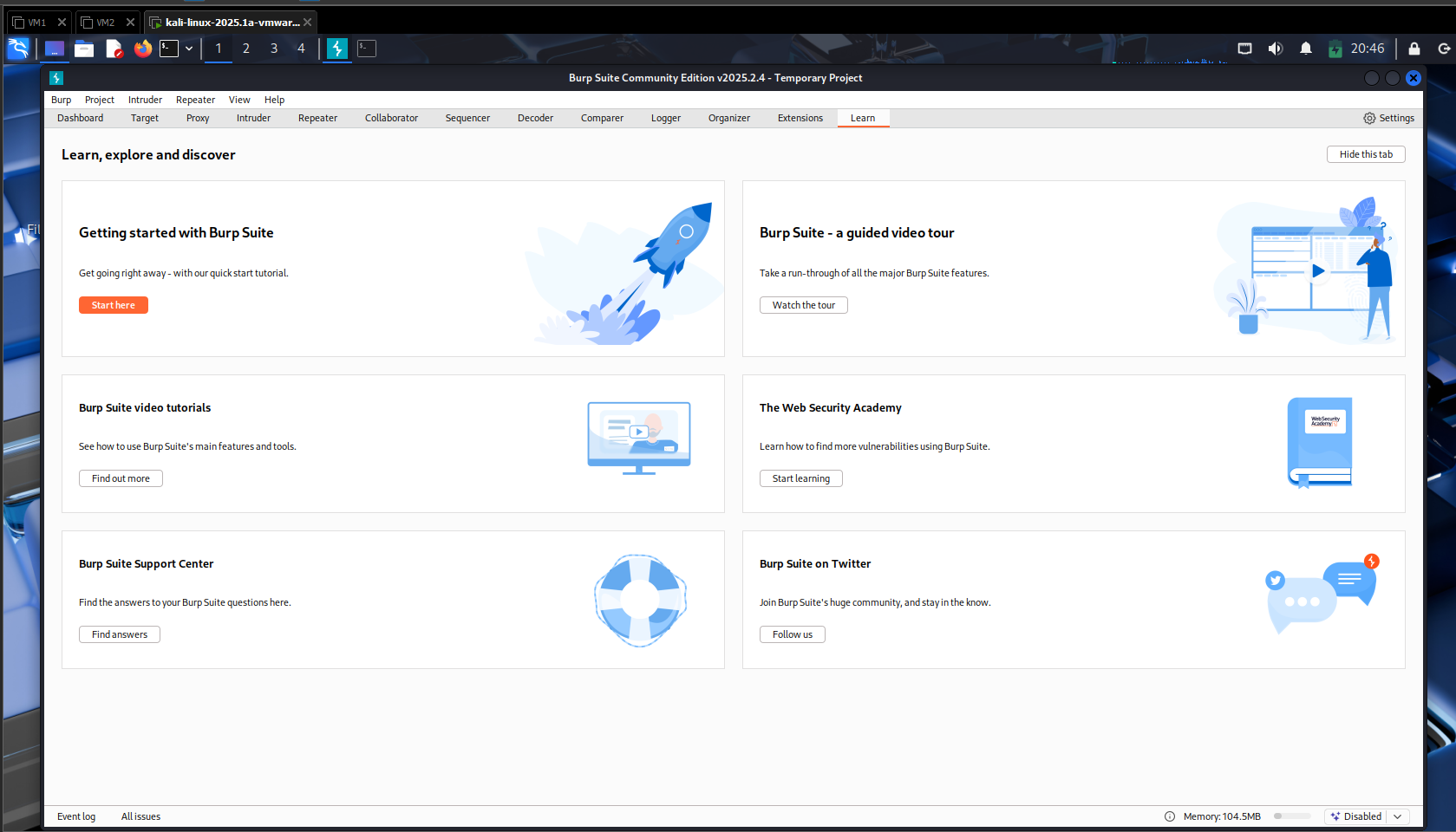
Link Youtube: https://youtu.be/NcjjXHi7kTs

LAB 8:

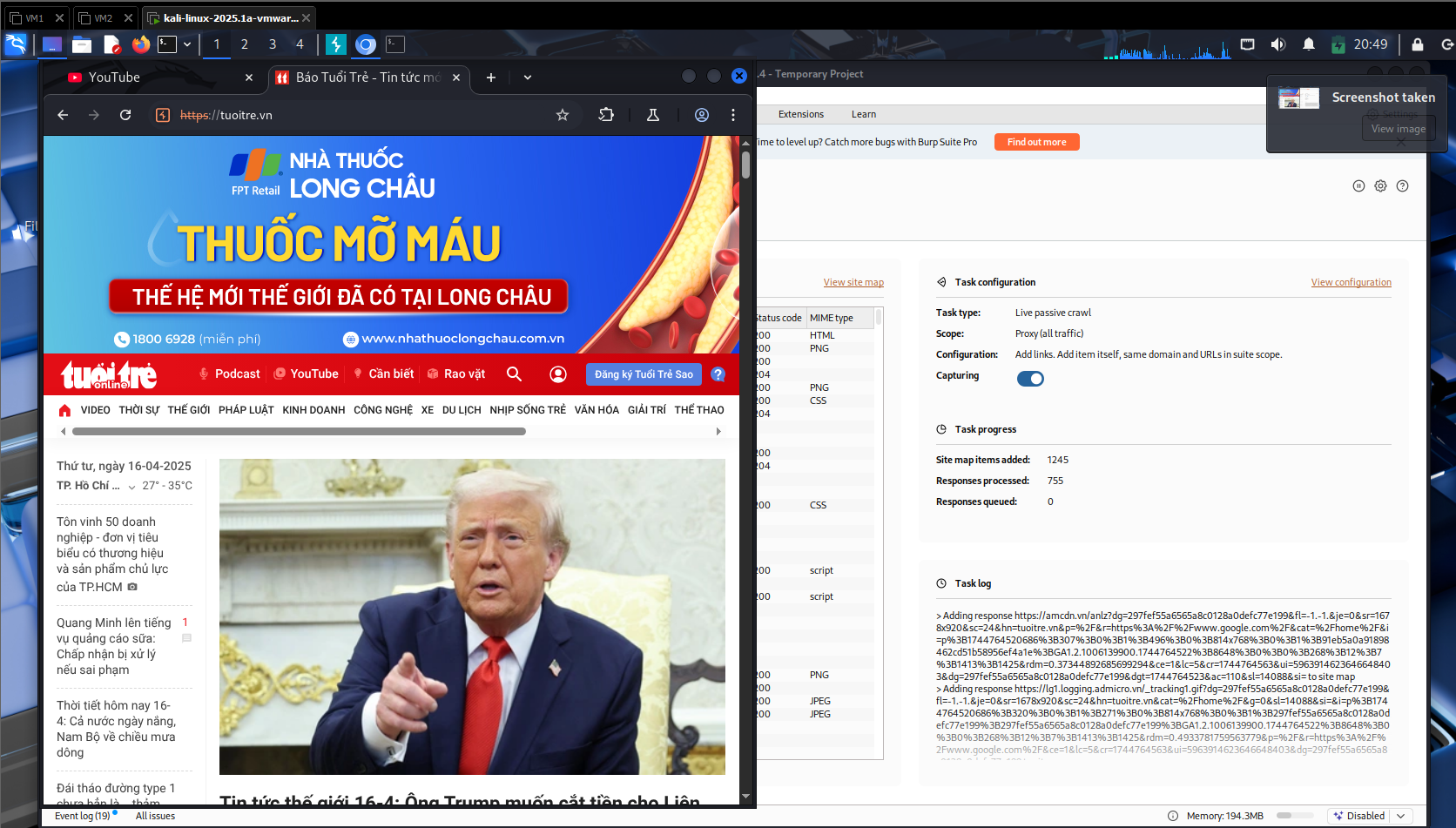
Cài đặt công cụ hỗ trợ kiểm thử Burpsuite

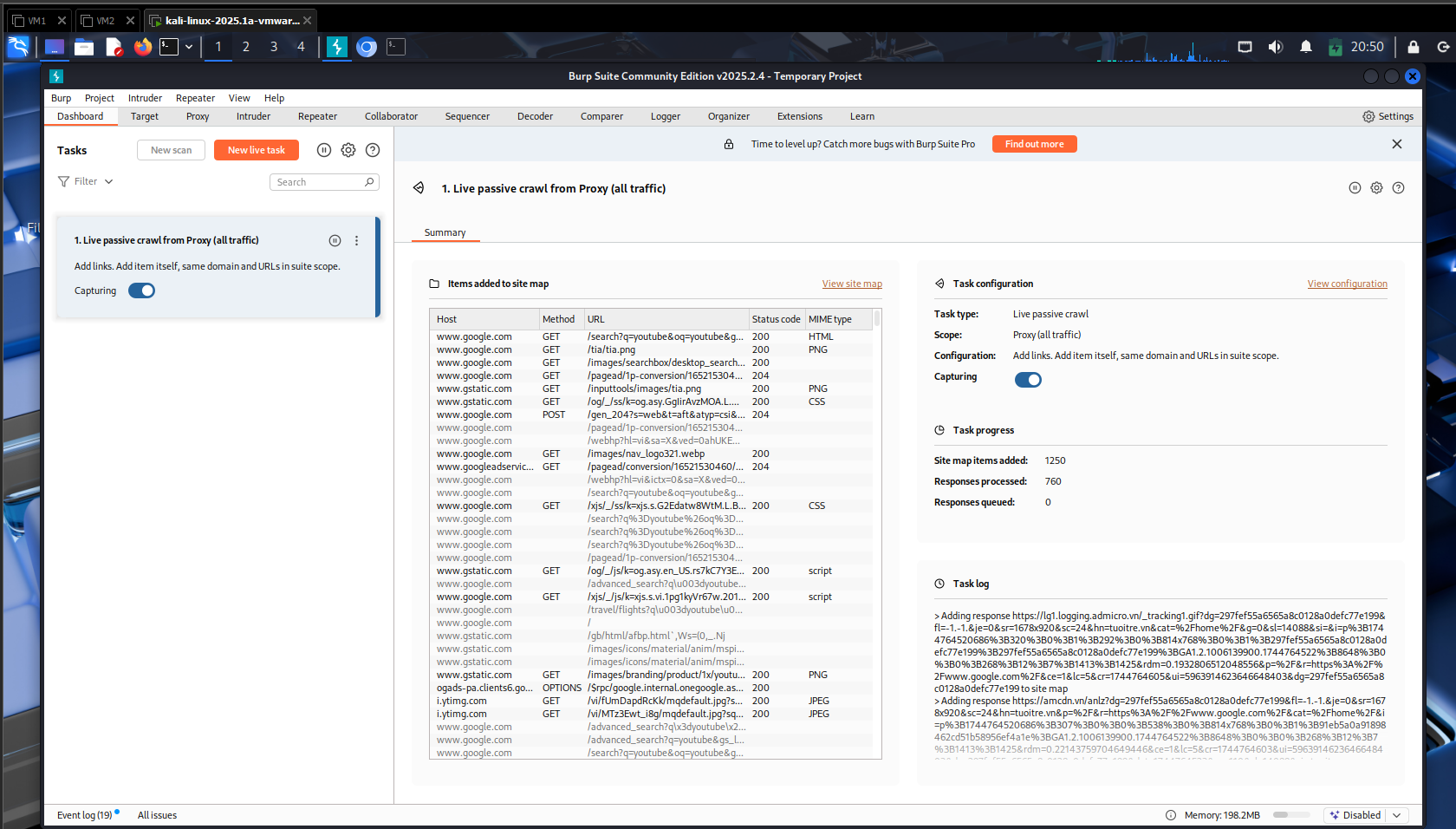






Truy cập trang google.com





**Câu 1*(3.5 điểm)***

Kiểm thử lỗ hỏng SQL Ịnection trên trang web:

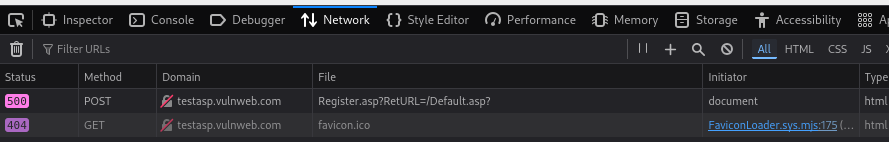
<http://testasp.vulnweb.com/Register.asp?RetURL=%2FDefault%2Easp%3F>

* Xác định các tham số đầu vào và cách thức chúng được gửi tới server **(0.25 điểm)**

Các tham số đầu vào bao gồm

Username, Real name, Email, Password

Cách thức chúng gửi tới server là: GET và POST

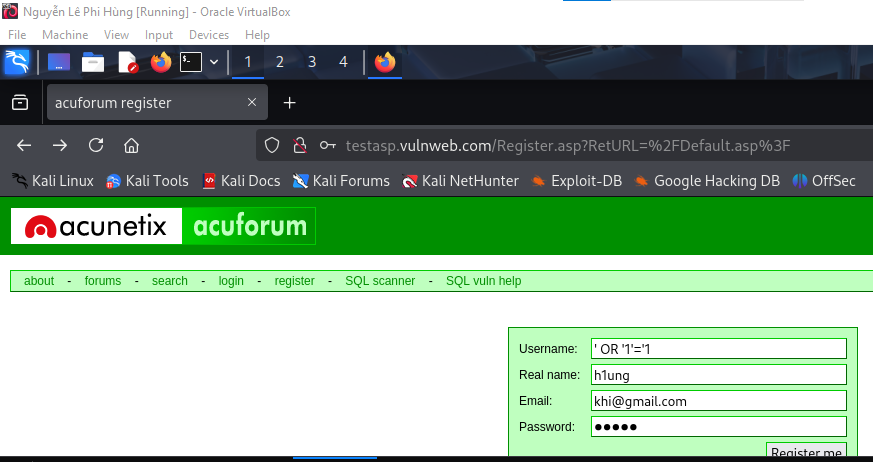
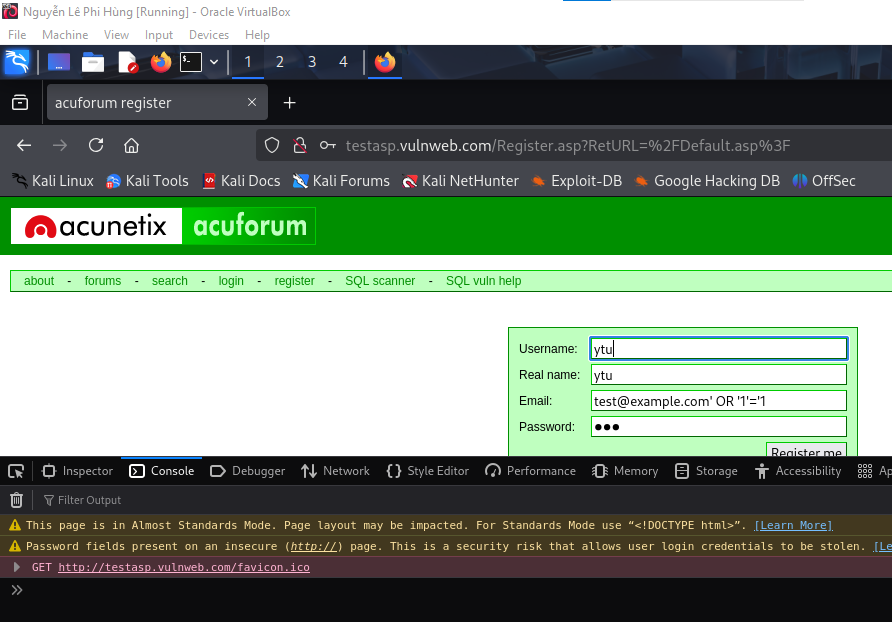
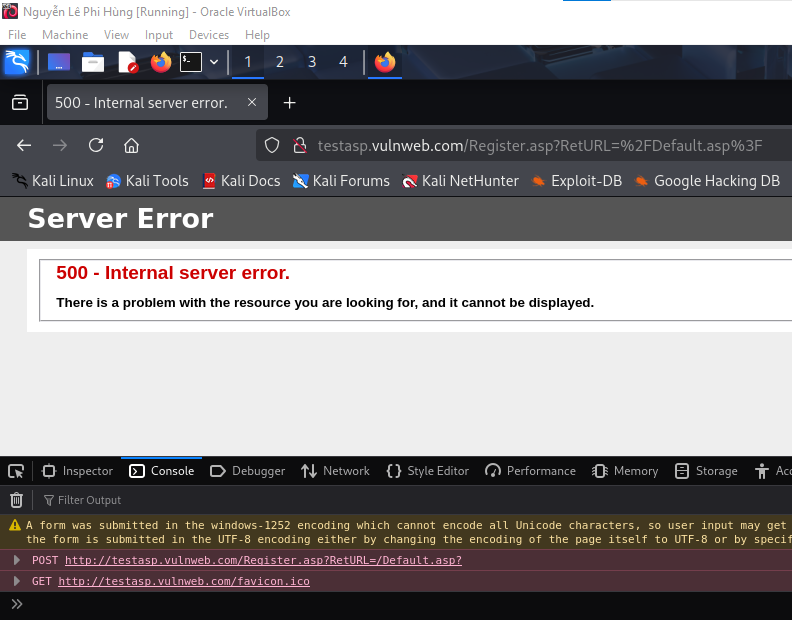


* Phân tích và phán đoán kiểu truy vấn **(0.25 điểm)**

Dựa trên các thành phần là Create và Insert Into

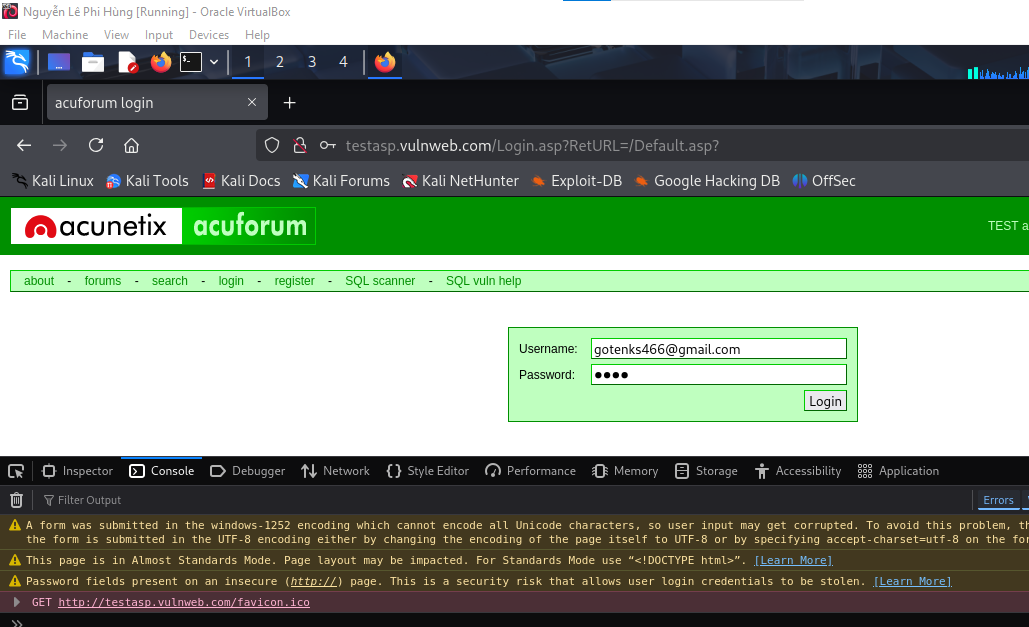
* Phân tích và phán đoán vị trí của các giá trị tham số đầu vào được sử dụng trong câu truy vấn **(0.5 điểm)**
* Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào **(2.5 điểm)**

Tiến hành thử payload SQL injection vào từng trường

* 
* 
* 
* Và kết quả lỗi không thể bypass

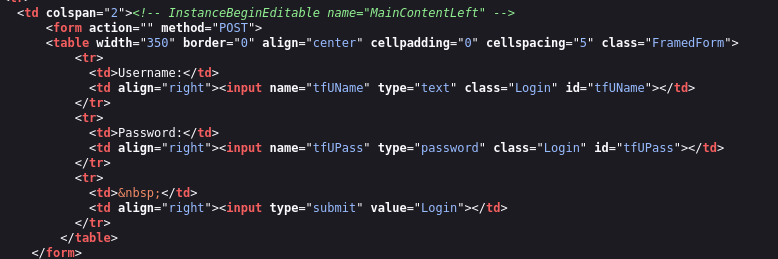
**Câu 2 *(3.5 điểm)***

Kiểm thử lỗ hổng SQL Injection trên trang web http://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp?

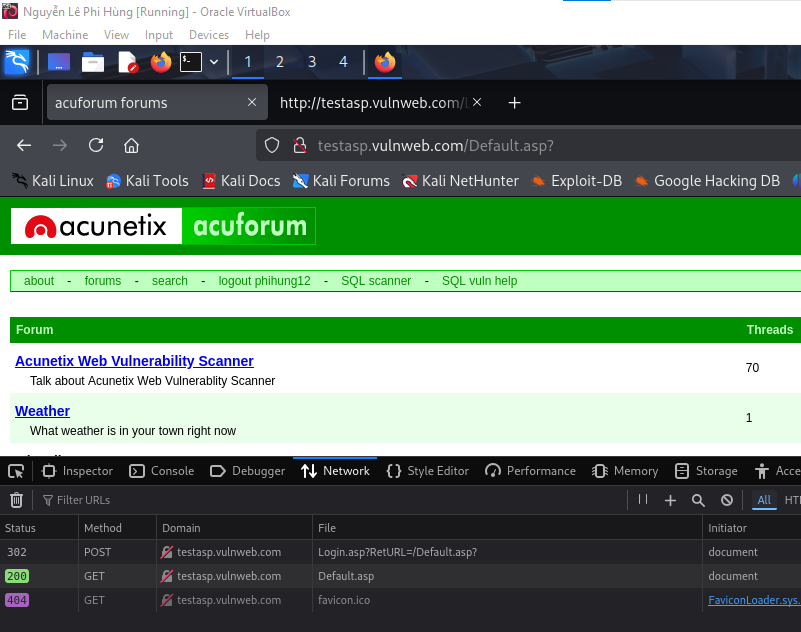


* Xác định các tham số đầu vào và cách thức chúng được gửi tới server **(0.25 điểm)**

Tiến hành vào Login.asp để xác định các tham số đầu vào



Cách thức chúng được gửi tới server là POST và GET

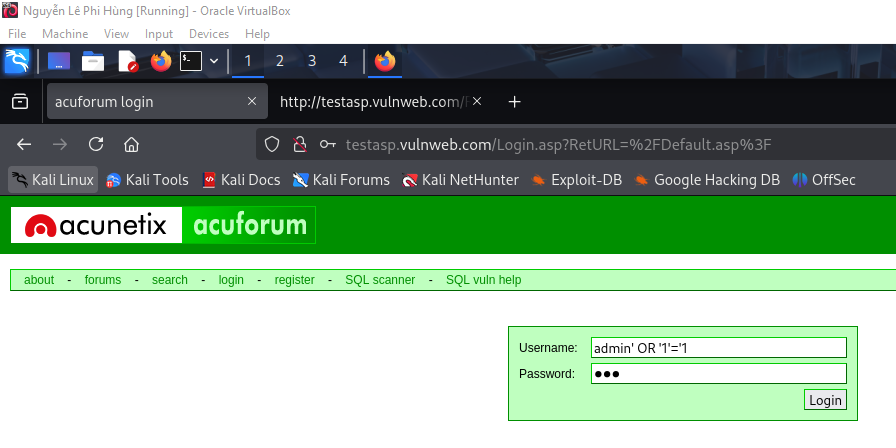


* Phân tích và phán đoán kiểu truy vấn, phân tích và phán đoán vị trí của các tham số đầu vào được sử dụng trong câu truy vấn **(0.25 điểm)**

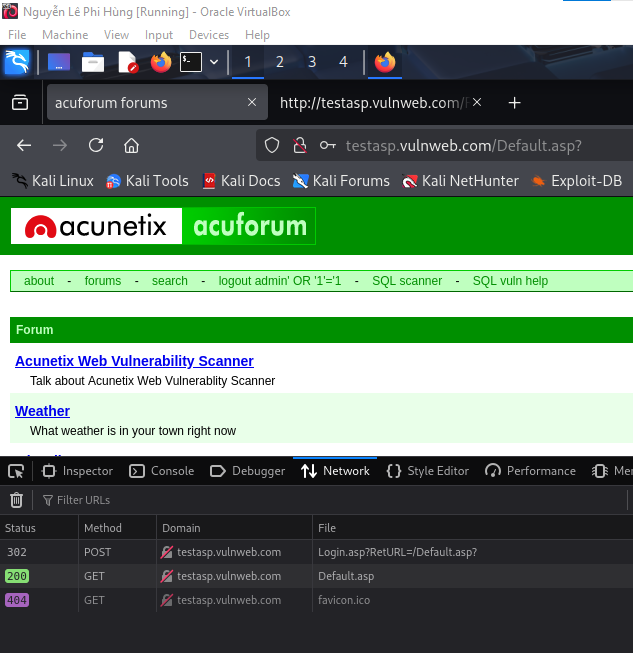
Vị trí tham số các đầu vào là: tfUName, tfRName, tfEmail, tfUPass

* Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào **(3 điểm)**

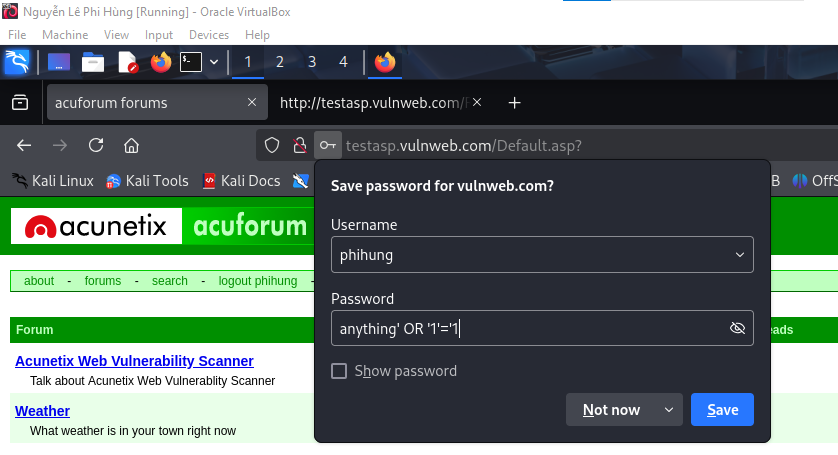
Khai thác tham số tfUname



Khi sử dụng username là admin’OR ‘1’1=’1 thì nó sẽ tự động bỏ qua mật khẩu giúp bypass



Tương tự với TfUpass nhập password là anything' OR '1'='1, thì nó sẽ tự động bỏ qua kiểm tra mật khẩu



**Câu 3 *(3 điểm)***

Kiểm thử lỗ hổng SQL Injection trên trang web http://testasp.vulnweb.com/

* Xác định các tham số đầu vào và cách thức chúng được gửi tới server **(0.25 điểm)**

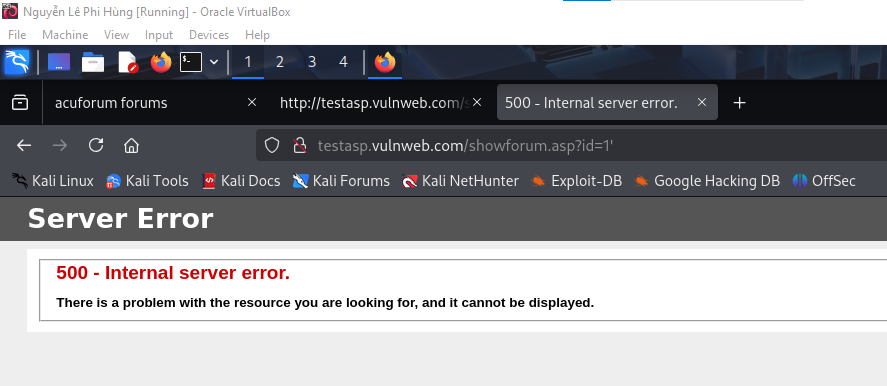
Là id



Và chúng được gửi bằng phương thức GET thông qua url

* Phân tích và phán đoán kiểu truy vấn, vị trí của các tham số đầu vào được sử dụng trong câu truy vấn **(0.25 điểm)**

Chạy code http://testasp.vulnweb.com/showforum.asp?id=1' sẽ xuất hiện lỗi



* Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào **(2.5 điểm)**

Với id mức độ ảnh hưởng là cao dễ rò rỉ dữ liệu

Search mức độ ảnh hưởng trung bình khi lọc input

Prod cao vì có UNION-based SQLi