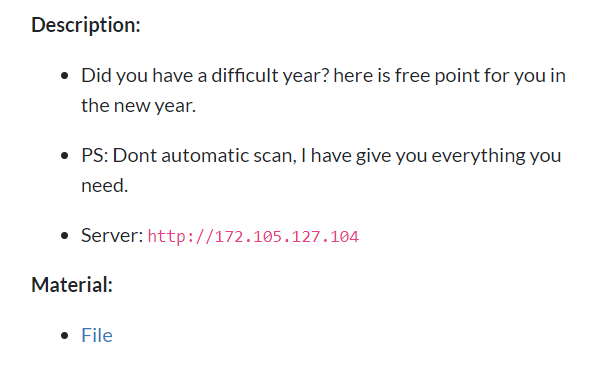
**PHP Magic CTF Writeups**

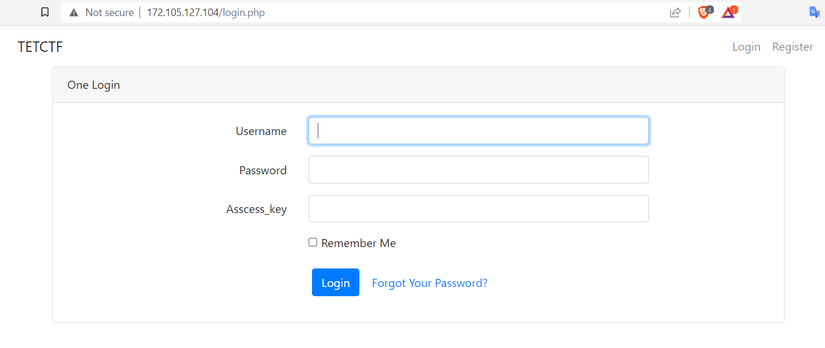
Sự kỳ diệu của PHP đã thúc đẩy mình viết bài này. Bài này mình sẽ tổng hợp và cập nhật các bài CTF khai thác các lỗ hổng "ảo ma" của PHP, mà đâu đó sẽ hơi khó khăn để tìm kiếm trên Google nếu với các cụm từ phổ thông kiểu "File Inclusion to RCE PHP", hoặc "Object Initialization to RCE PHP".

**1. File Inclusion dẫn tới RCE bằng pearcmd.php (hoặc peclcmd.php)**

**1.1. GIFT - TetCTF 2023**

**0x01 Mô tả challenge**

Phần mô tả của challenge GIFT không mở ra bất kỳ gợi ý nào. Ít nhất thì mình biết là không cần đoán mò cái gì cả.

Truy cập vào URL của web challenge, hiện ra 1 login form. Bước đầu nghịch ngợm thì thấy chỉ có nút login là dùng được.

Ngoài ra, challenge có đính kèm 1 file .sql được dump ra từ cơ sở dữ liệu MySQL . Cảm giác như đây là gợi ý duy nhất có thể liên quan gì đó tới SQL Injection (mà đúng thực là vậy).

-- MySQL dump 10.13 Distrib 5.6.51, for Linux (x86\_64)

--

-- Host: localhost Database: main

-- ------------------------------------------------------

-- Server version 5.6.51

DROP TABLE IF EXISTS `access\_key`;

CREATE TABLE `access\_key` (

`id` int(11) NOT NULL,

`key\_cc` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

LOCK TABLES `access\_key` WRITE;

INSERT INTO `access\_key` VALUES (2,'Censored');

UNLOCK TABLES;

DROP TABLE IF EXISTS `pictures`;

CREATE TABLE `pictures` (

`id` int(11) NOT NULL,

`url` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

LOCK TABLES `pictures` WRITE;

INSERT INTO `pictures` VALUES (1,'http://example.com/');

INSERT INTO `pictures` VALUES (2,'http://localhost/media/messi.jpg');

UNLOCK TABLES;

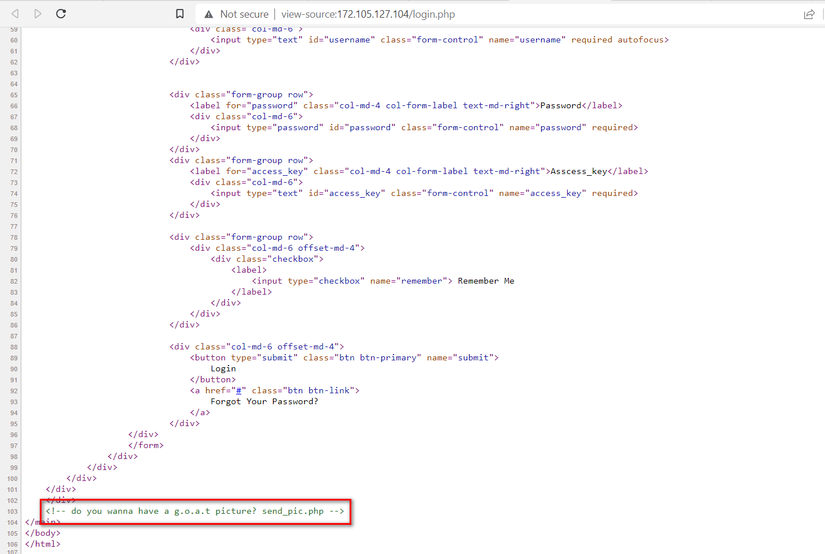
**0x02 Hướng giải**

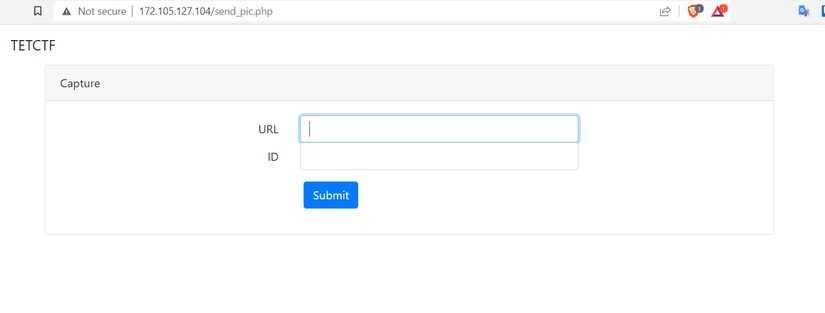
**SQL Injection lấy access key**

Nếu thực sự tồn tại SQL Injection, mình đoán khả năng cao nó không nằm ở chỗ form login. Có 2 lý do cho việc này:

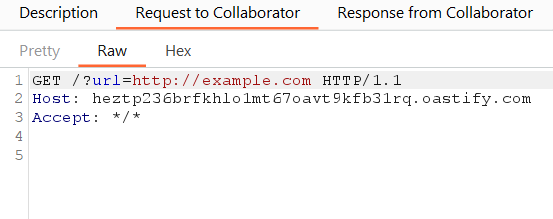
1. Mình không có bất kỳ thông tin gì về form login, việc trích xuất dữ liệu từ SQL Injection buộc phải dùng kỹ thuật time-based. Theo kinh nghiệm bản thân, thường CTF không ra kiểu time-based SQL Injection.
2. Trong trường hợp SQL Injection cho phép bypass login, thì việc đưa file .sql khá là vô nghĩa. Mình tin là tác giả không trick người chơi vô "rabbit hole" bằng file như vậy đâu.

Tóm lại, hướng đi là phải tìm ra một injection point khác. Trong trường hợp bí bách, Ctrl + U vẫn luôn là quân tốt dí điển hình. Vô thì thấy có thêm file send\_pic.php



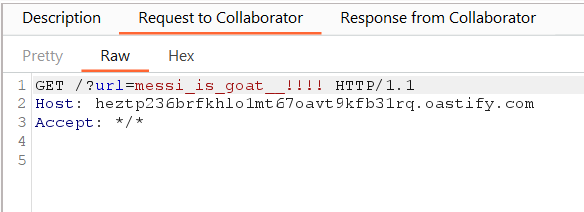


Nhìn thấy URL và ID, mình đoán ngay khả năng cao SQL Injection sẽ đánh vào tham số ID. Sau một thời gian mọc mạch sử dụng, mình biết được nó sẽ gửi data lấy bảng pictures theo ID tương ứng. Ví dụ với URL là Burp Collaborator Domain và ID là 1, mình thu được thông tin như ảnh sau:



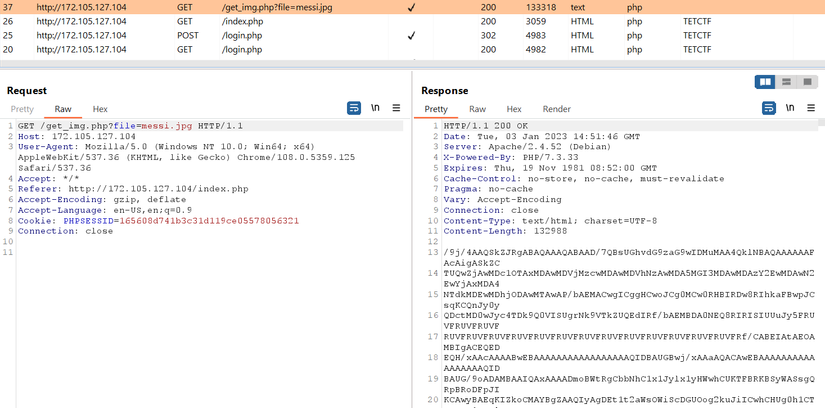
Mình xác nhận được lỗi SQL Injection khi inject và tham số ID các giá trị 1, 3-2, 100-99 cho về cùng một kết quả giống nhau. Mục tiêu là cần lấy key\_cc từ bảng access\_key, thế nên mình sử dụng UNION để lấy data ra. Web có lọc cụm select x from y, và chỉ cần dùng chữ hoa để bypass được (Uppercase bypass).

Tóm lại, với id=0 UNION SELECT key\_CC FROM access\_key WHERE id LIKE "2", mình retreive được access key của hệ thống



**Path traversal (hoặc thực ra File Inclusion)**

Với access key lấy được từ bước trước, mình quay lại trang login và đăng nhập. Username và password thì nhập bừa là được. Login xong thấy trong một loạt request có một cái lấy file ảnh để hiện ra.



Nhìn tham số file thì thường là nghĩ ngay tới path traversal rồi. Bằng việc fuzzing thử bằng một đường dẫn không tồn tại ../media2/messi.jpg, mình phát hiện ra đây là lỗ hổng file inclusion từ warning trả về trong response. Nói chung không có chuyện phải đoán tên file flag rồi path traversal để ra 😅.

Trong lúc fuzzing nghịch ngợm, mình cũng phát hiện ra web có lọc cụm ký tự ../../. Cái này thì chỉ cần thay ../../ thành .././../ là bypass được ngon lành. Thế nhưng để biết được flag nằm đâu thì buộc phải RCE được. Đoạn sau mới độc lạ PHP.

**File Inclusion vào peclcmd.php để lên RCE**

Bằng kinh nghiệm, hoặc bằng khả năng Google, dễ dàng tìm thấy mấy bài dạng PHP Inclusion lên RCE bằng phương pháp Log Poisoning. Mình cũng đã thử, và tìm ra /proc/self/fd/10 nhưng rồi không biết input vào từ đâu ra cái file log. Đành chịu phép 🫥

Tới khi hết TetCTF rồi, mình mới thấy các team chia sẻ về file inclusion vào peclcmd.php.

Không biết nó là gì, mình google thử các bài về pearcmd.php và peclcmd.php (2 file này tương tự và thay thế cho nhau được), thì thấy toàn mấy bài của các pháp sư Trung Hoa giải thích. Mọi người có thể xem giải thích tại [đây](https://w4rsp1t3.moe/2021/11/26/%E5%85%B3%E4%BA%8E%E5%88%A9%E7%94%A8pearcmd%E8%BF%9B%E8%A1%8C%E6%96%87%E4%BB%B6%E5%8C%85%E5%90%AB%E7%9A%84%E4%B8%80%E4%BA%9B%E6%80%BB%E7%BB%93/" \t "_blank), và có thể tham khảo luôn một loạt payload web CTF của một pháp sư nào đó tại [đây](https://github.com/w181496/Web-CTF-Cheatsheet" \t "_blank) luôn.

Tóm gọm lại, thì pearcmd.php (hay peclcmd.php) là một công cụ quản lý tiện ích mở rộng bằng command line. Thông thường thì nó nhận tham số từ Command Prompt (hoặc các trình tương tự). Tuy nhiên, nếu tham số register\_argc\_argv là ON, thì mình có thể truyền tham số từ web thông qua $\_SERVER['argv'].

Mình dùng payload lấy ra từ trong cheetsheet của pháp sư Trung Hoa luôn. Sửa lại được file cho hợp lệ, thành .././.././.././.././../usr/local/lib/php/peclcmd.php.

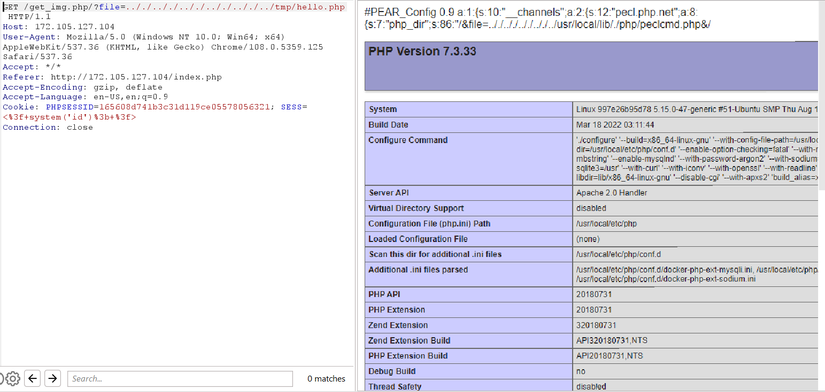


Tuy nhiên, khi mình thử file inclusion vào /tmp/hello.php thì lại hiện ra warning không tồn tại file. Xem chừng payload không thành công.

Tiếp tục fuzzing một hồi, mình phát hiện ra cụm /usr/local/lib/php cũng bị chặn lọc (super filtered challenge). Áp dụng cách bypass giống hệt với path traversal ở trên là qua được. Lần này ra được cái response uy tín rồi!



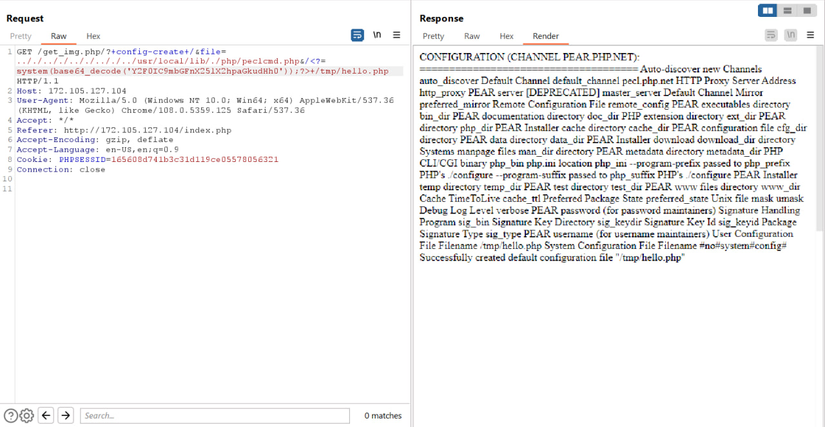
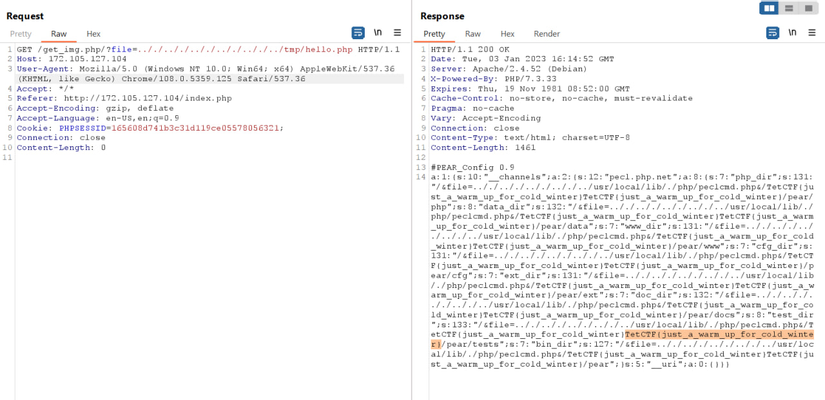
Thử file inclusion vào /tmp/hello.php coi sao.



Thế là code injection thành công. Giờ chỉ còn việc exploit lấy ra flag là xong.

**0x03 Lấy flag**

Mình không có cách nào để truyền được dấu cách trực tiếp trong đoạn code PHP, nên mình đẩy vào trong 1 đoạn base64 của payload tương ứng. Lòng vòng một hồi, mình ls thư mục / ra thì thấy có file flag. Sau đó cat file flag đó ra là xong.

1 - GET /get\_img.php/?+config-create+/&file=.././.././.././.././../usr/local/lib/./php/peclcmd.php&/<?=system(base64\_decode('Y2F0IC9mbGFnX25lX2hpaGkudHh0'));?>+/tmp/hello.php2 - GET /get\_img.php/?file=.././.././.././.././.././../tmp/hello.php

**0x04 Tóm tắt lại quá trình giải**

Tóm lại bài này giải qua 2 bước:

1. Union SQL Injection vào ID để lấy access\_key. Dùng uppercase payload để bypass filter select x from y.
2. File inclusion vào peclcmd.php để tạo 1 file config PHP. Dùng ký hiệu current directory ./ để bypass filter /../../ và /usr/local/lib/php.