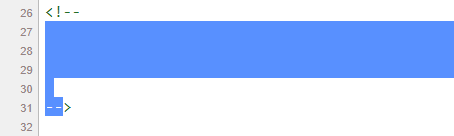
1. **HTML- Source code**  
   Ctrl U -> paste lên trình duyệt

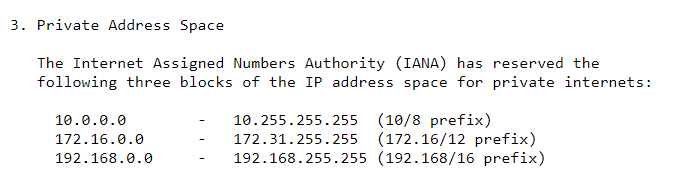




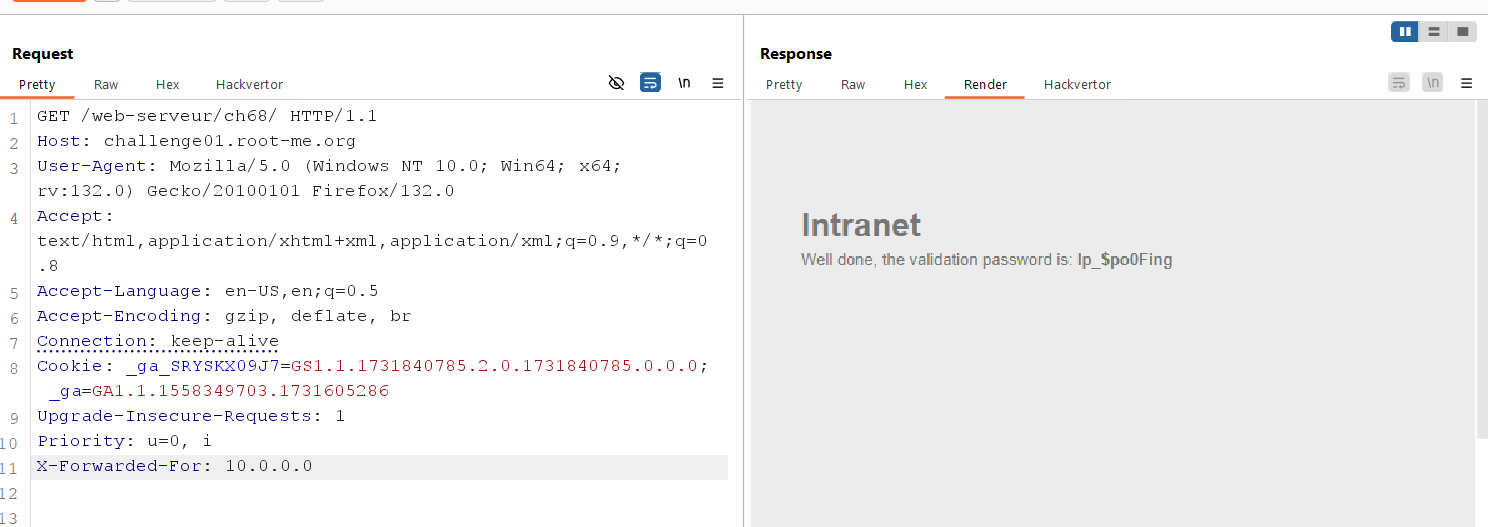
=> Flag: nZ^&@q5&sjJHev0

# HTTP - IP restriction bypass

Ở bài này cần tìm private IP để truy cập được vào mạng nội bộ



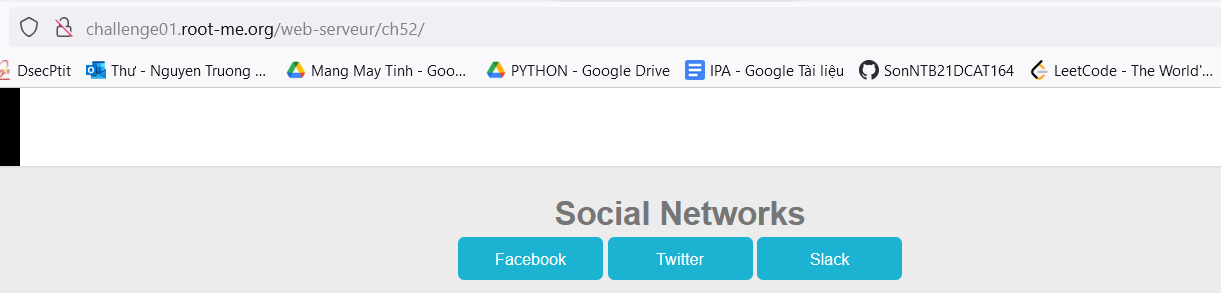
Sử dụng BurpSuite spoof IP thuộc 1 trong 3 dải trên bằng X-Forwarded-For



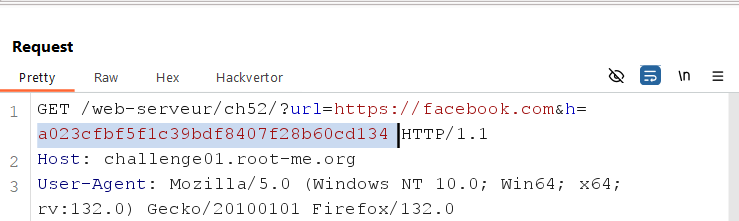
=> Flag: Ip\_$po0Fing

# HTTP - Open redirect

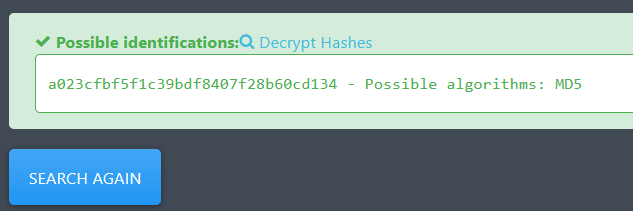
Khi vào chall thì thấy có 3 button dẫn tới các social network



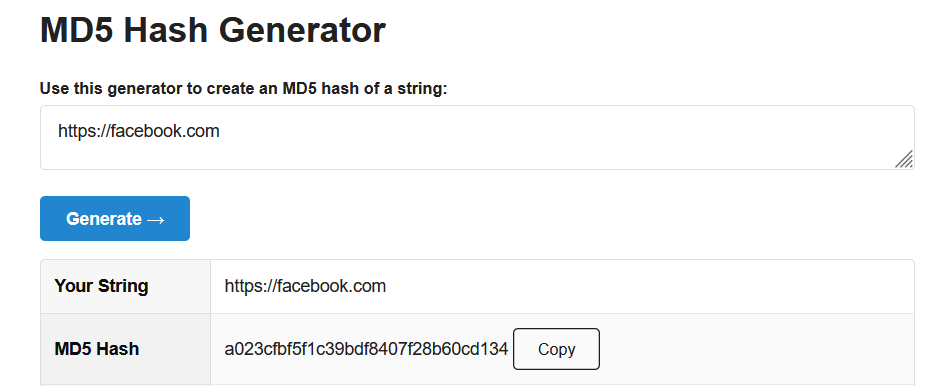
Sử dụng BurpSuite bắt request, thấy rằng khi click vào các đường dẫn thì ngoài url chứa link còn có 1 giá trị khả năng là mã hash



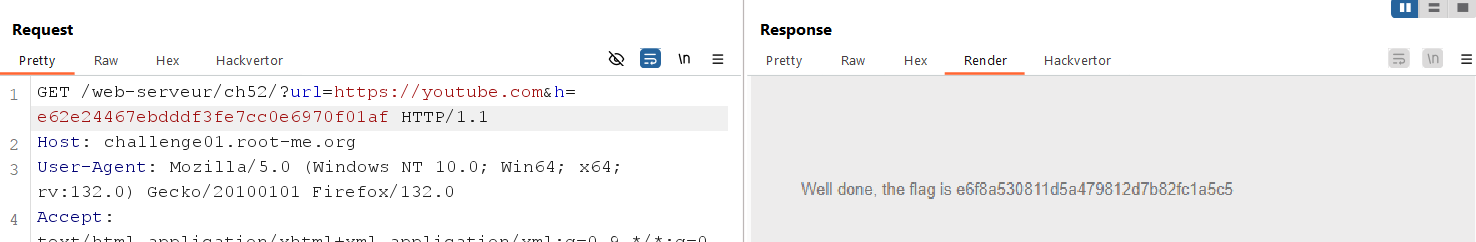
Sử dụng hash identifier thì thấy rằng đây là md5



Vậy thì có khả năng đây là md5 của url tương ứng, kiểm tra lại thì kết quả đúng như dự đoán



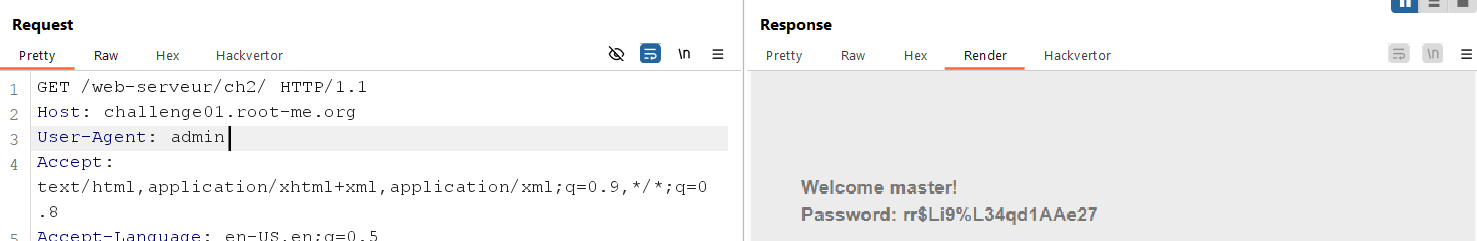
Nhiệm vụ của chall là redirect đến 1 url khác ngoài 3 url đề bài cho, vậy chúng ta sẽ thử với url <https://youtube.com> với md5 tương ứng là e62e24467ebdddf3fe7cc0e6970f01af



=> Flag: e6f8a530811d5a479812d7b82fc1a5c5

# HTTP - User-agent

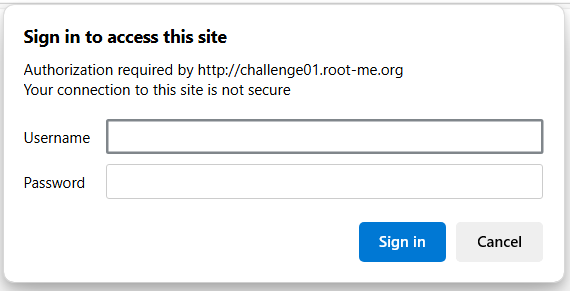
Khi vào chall thì thấy browser chúng ta đang sử dụng không phải “admin”. Vậy thì sửa lại header User-agent thành admin



=> Flag: rr$Li9%L34qd1AAe27

# Weak password

Khi vào chall thì có popup xác thực hiện lên

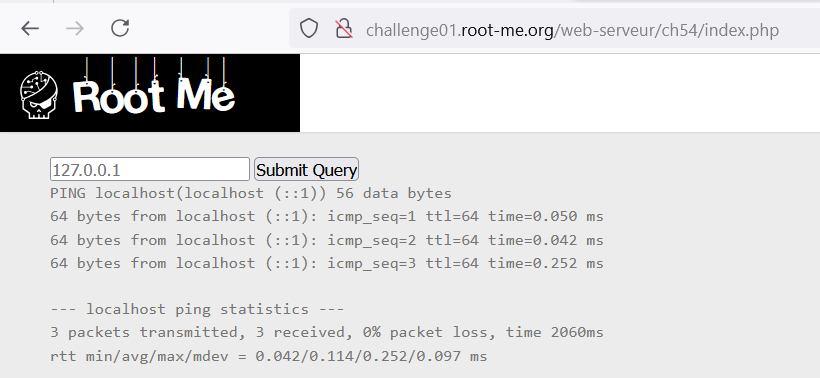


Nhập username và password là “admin” thì Sign in thành công

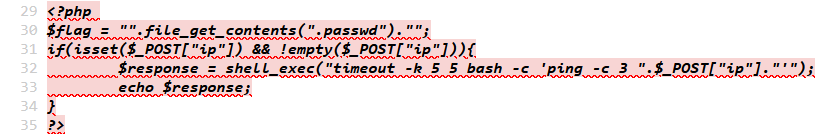
=> Flag: admin

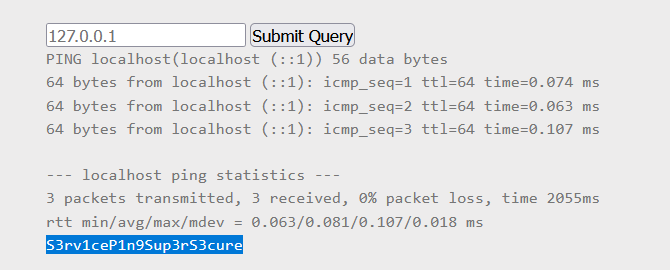
# PHP - Command injection

Khởi động chall và ping thử localhost -> ping thành công



Chuyển lệnh ping thành “localhost; cat index.php” và Ctrl U xem source -> Flag nằm ở file .passwd

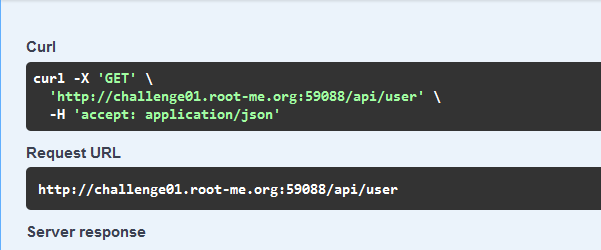


Sửa lệnh ping là thành “localhost; cat .passwd” 

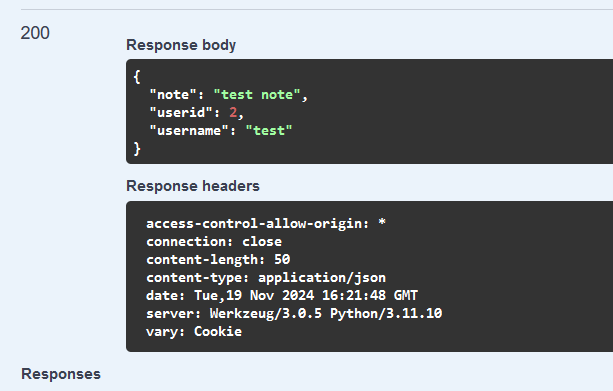
=> Flag: S3rv1ceP1n9Sup3rS3cure

# API - Broken Access

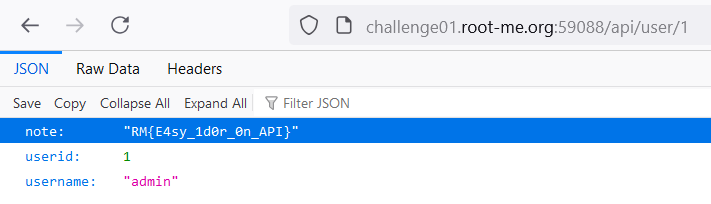
Thử nghiệm hết các api mà chall cung cấp, nhận thấy rằng với api /api/user, chúng ta có thể xem được response body bằng Request URL



Khi nhập user\_id bằng 1, response body trả về userid là 2, vậy có khả năng userid=1 chứa điều gì đó bất thường



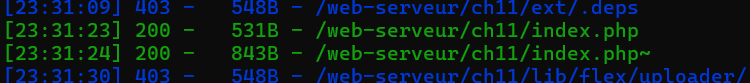
Copy Request URL lên trình duyệt và thêm đường dẫn tới userid=1 => Tồn tại IDOR và tìm thấy flag



=> Flag: RM{E4sy\_1d0r\_0n\_API}

1. **Backup file**

Sử dụng tool dirsearch trên Kali tìm đường dẫn đến file backup => index.php~



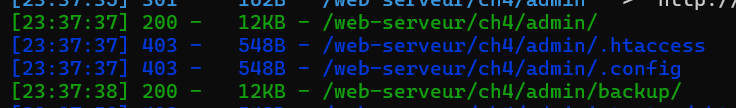
Nhập đường dẫn vào thì có thể tìm thấy password và username



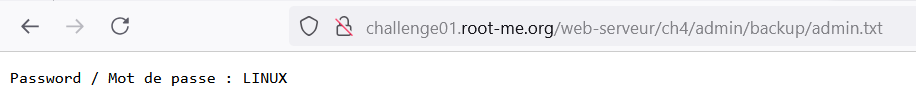
=> Flag: OCCY9AcNm1tj

# HTTP - Directory indexing

Sử dụng tool dirsearch trong Kali tìm các đường dẫn => Thấy xuất hiện đường dẫn backup



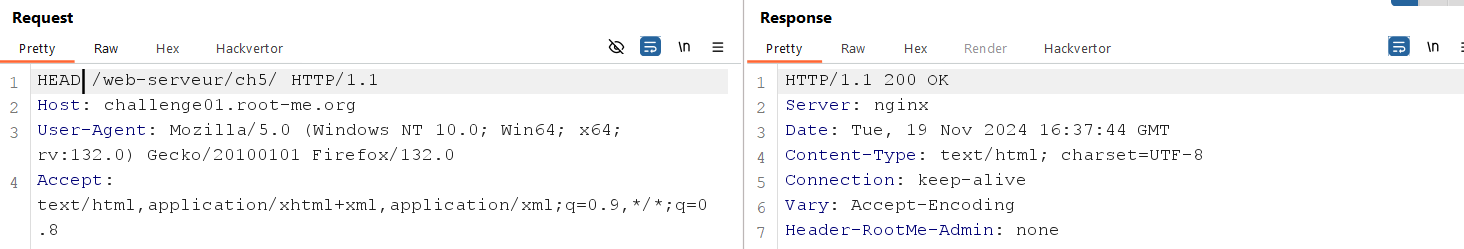
Truy cập vào và tìm thấy flag



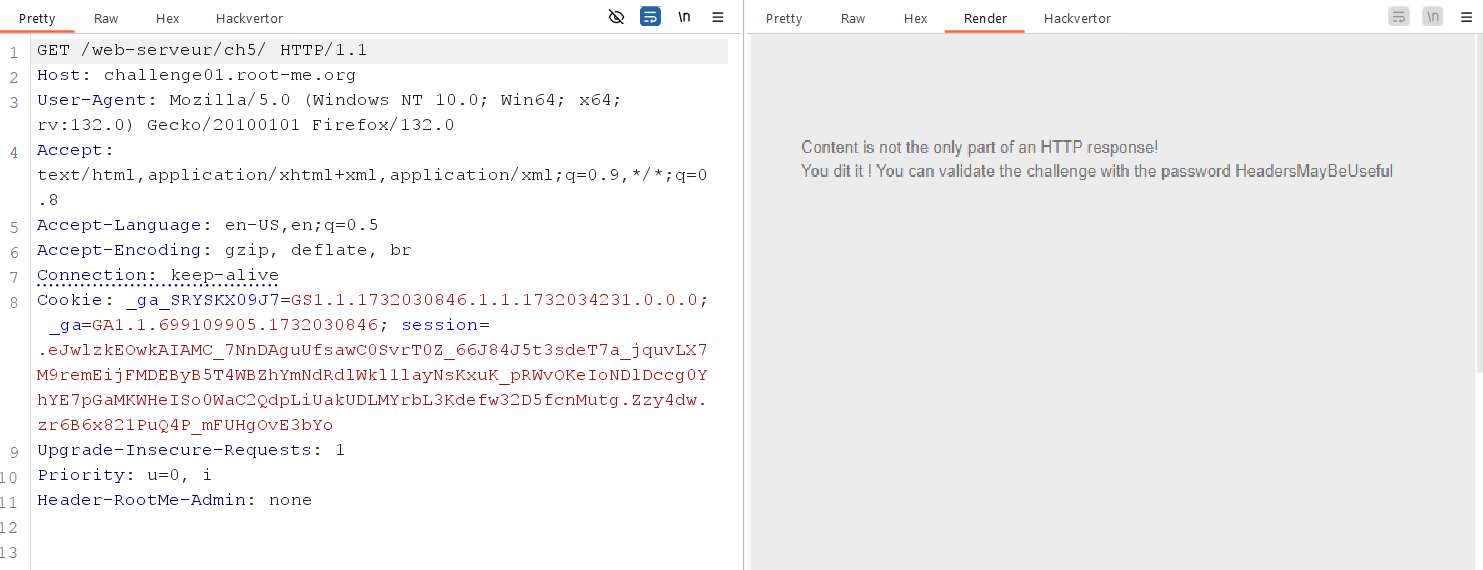
=> Flag: LINUX

1. **HTTP - Headers**

Sửa method GET thành HEAD và send, nhận thấy có 1 header là Header-RootMe-Admin: none xuất hiện



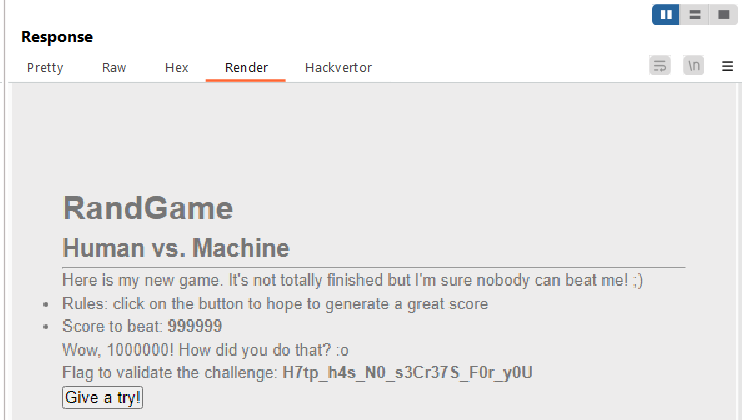
Send method GET với header vừa thêm



=> Flag: HeadersMayBeUseful

1. **HTTP - POST**

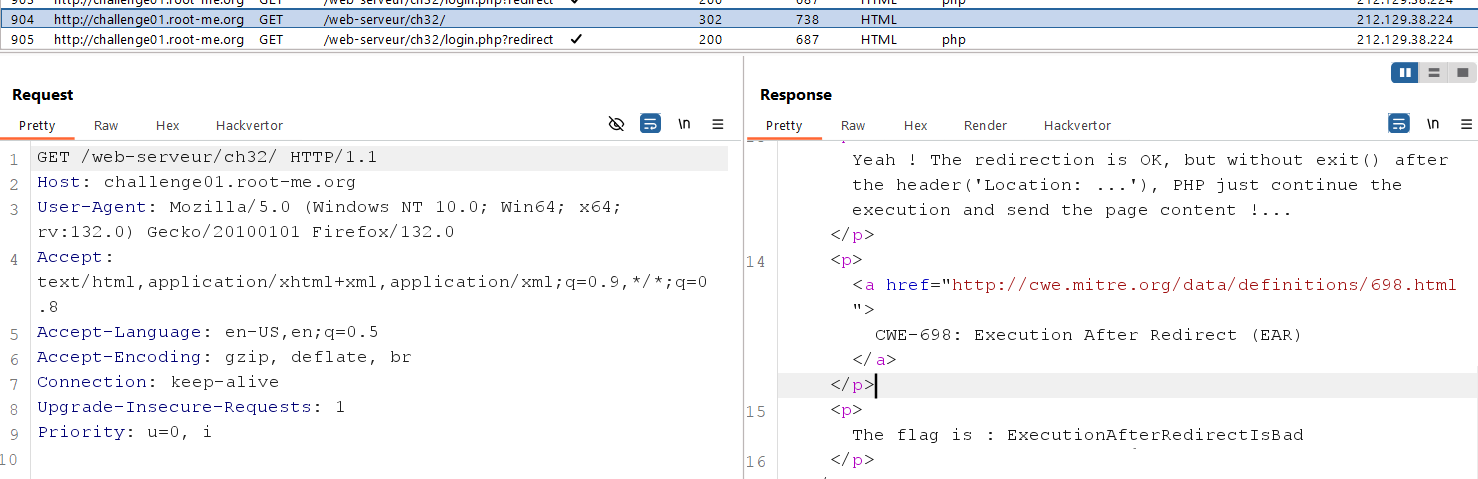
Chall này yêu cầu phải random được 1 số nguyên có giá trị lớn hơn 999.999. Vậy chúng ta chỉ cần gửi 1 POST request với score là 1.000.000



Flag: H7tp\_h4s\_N0\_s3Cr37S\_F0r\_y0U

1. **HTTP - Improper redirect**

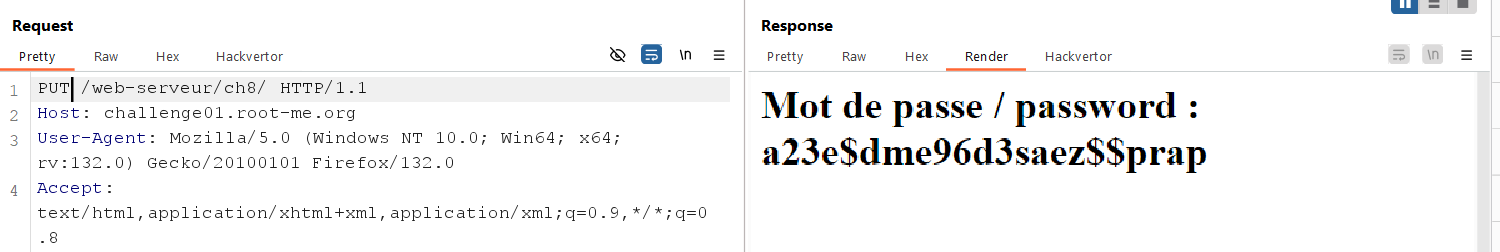
Khi khởi động chall thì dùng BurpSuite bắt request, có thể thấy flag xuất hiện ở response của request /web-serveur/ch32/



=> Flag: ExecutionAfterRedirectIsBad

1. **HTTP - Verb tempering**

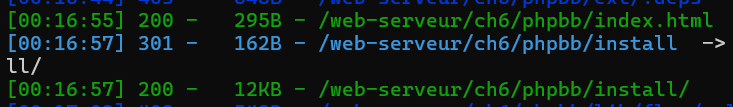
Thử thay method GET thành 1 method bất kỳ



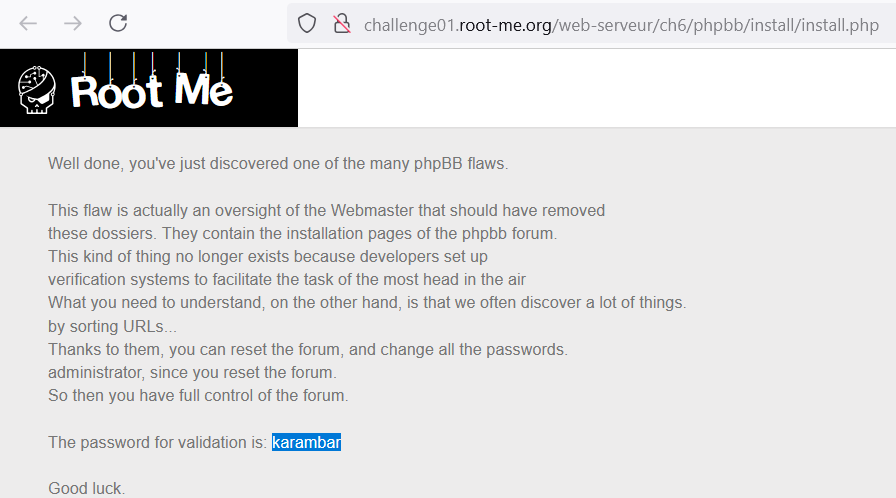
Flag: a23e$dme96d3saez$$prap

1. **Install files**

Dùng tool dirsearch trong Kali tìm thấy đường dẫn install



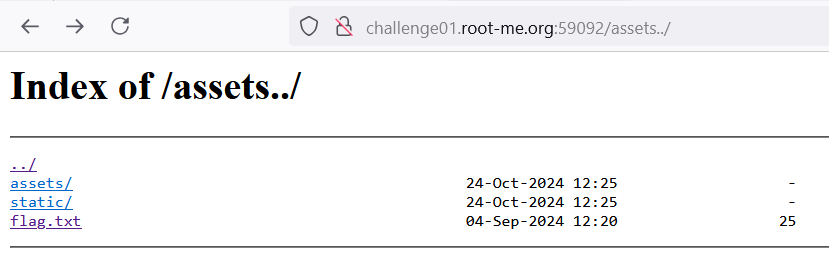
Truy cập và tìm được flag



=> Flag: karambar

# Nginx - Alias Misconfiguration

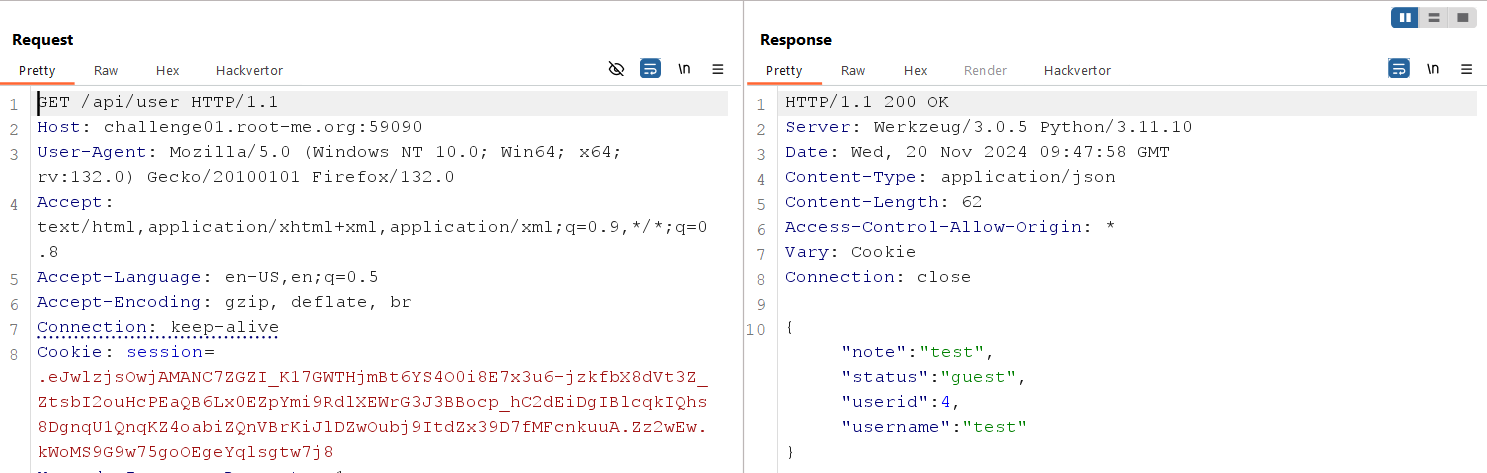
Truy cập vào đường dẫn assets..

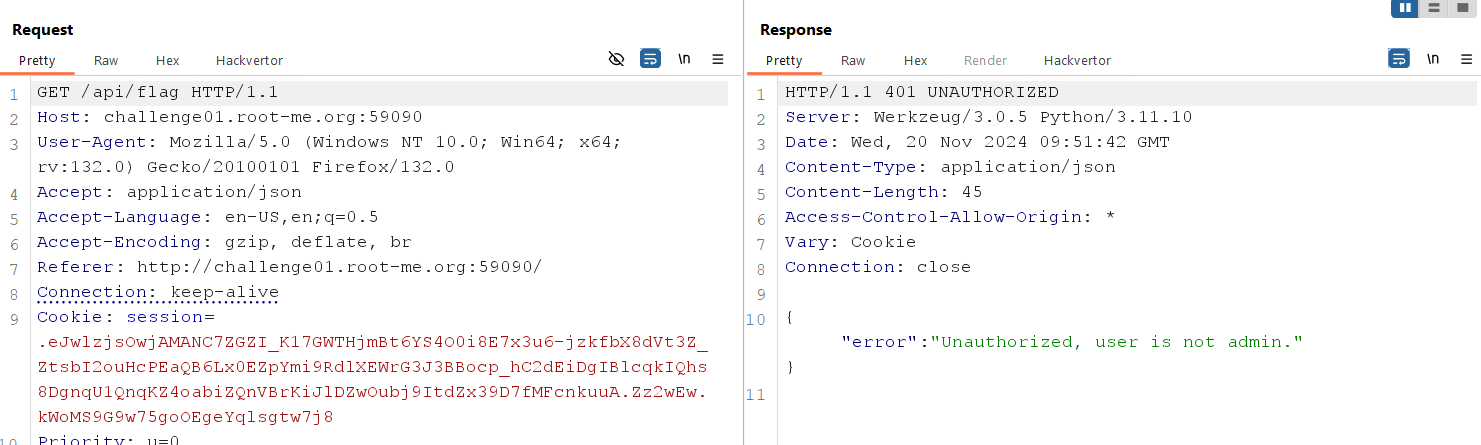


=> Flag: RM{4lias\_M1sC0nf\_HuRtS!}

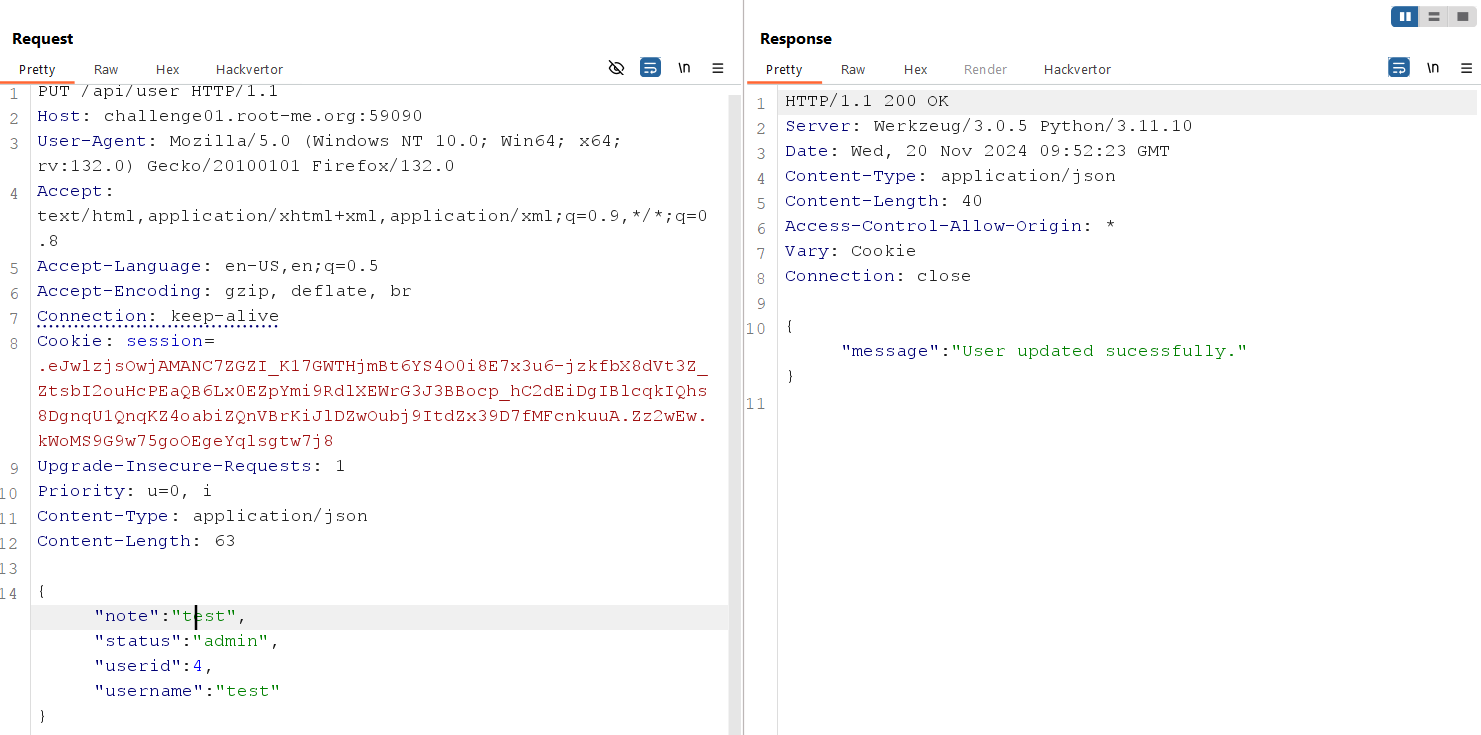
1. **API - Mass Assignment**

Chall này không tồn tại IDOR ở api/user như bài API trước. Khi truy cập vào /api/user thì response trả về chứa cặp key-value là “status”: “guest”, lúc này không có quyền để lấy flag từ api/flag

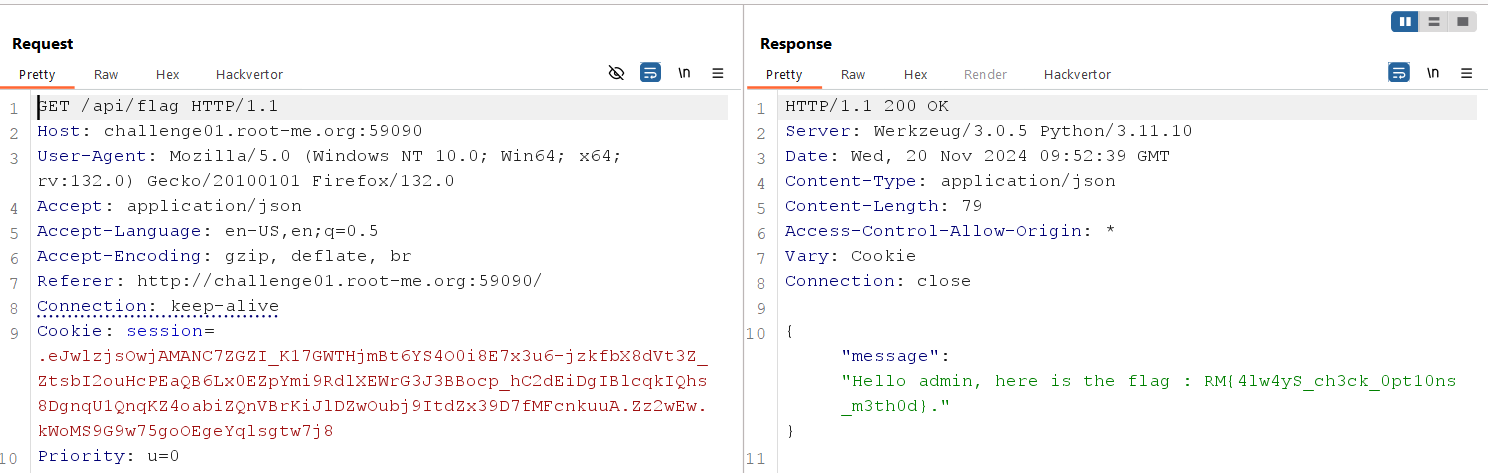




Vậy thì thử gửi lại yêu cầu PUT với Content-Type dạng json, guest chuyển thành admin



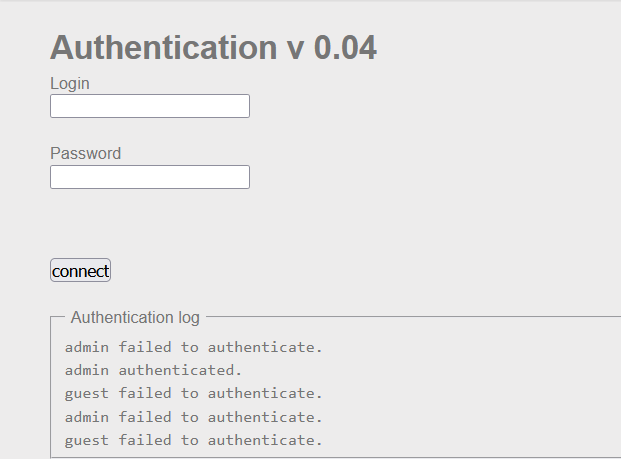
Sau đó gửi lại request GET /api/flag

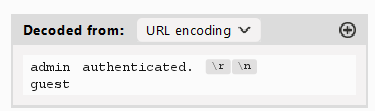


=> Flag: RM{4lw4yS\_ch3ck\_0pt10ns\_m3th0d}

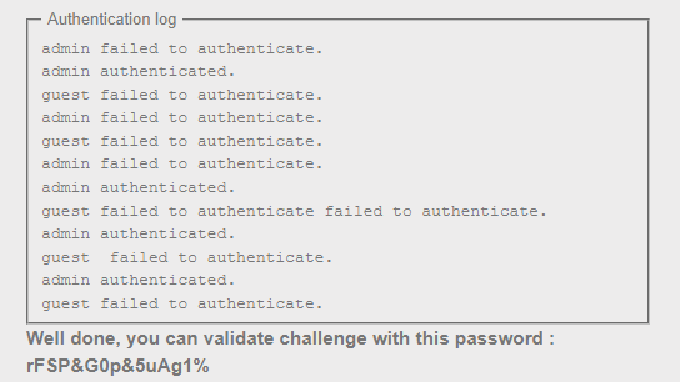
1. **CRLF**

Khi login thì log được ghi lại, quan sát thấy nếu login thành công thì có log là “username + authenticated”, còn nếu thất bại thì “username + failed to authenticate”



Log ghi trên các dòng khác nhau và lùi vào đầu dòng (CRLF) nên có thể sửa request để đáp ứng điều này. Thay đổi username như sau và send request  


Điều này sẽ dẫn đến admin login thành công và lấy được cờ



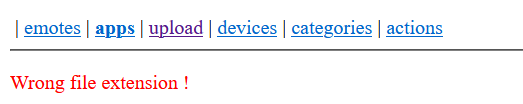
=> Flag: rFSP&G0p&5uAg1%

# File upload - Double extensions

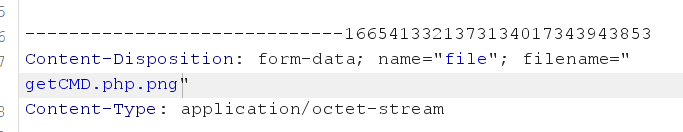
# Upload file getCMD.php như sau, mục đích là thực hiện được các command trên server

# 

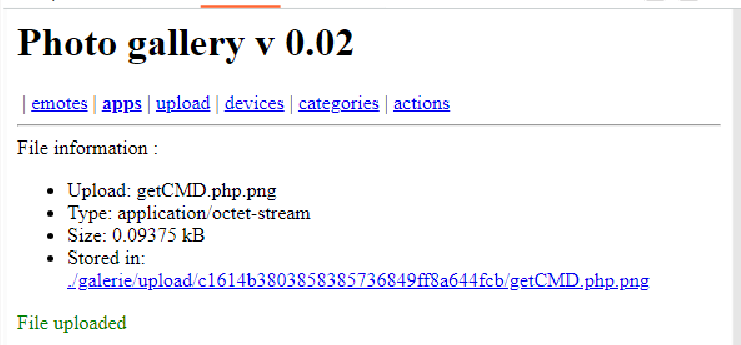
Nhận được thông báo sai extension khi upload



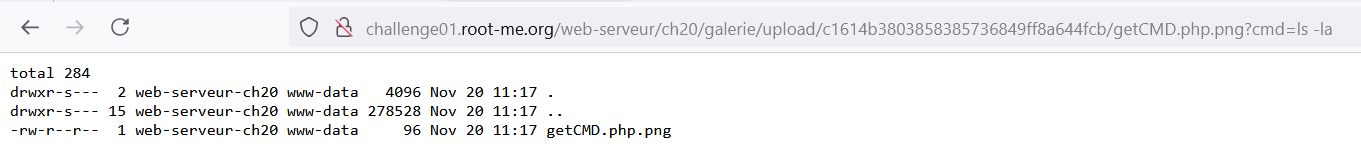
Do chỉ các extension có định dạng ảnh mới được chấp nhận nên chỉ cần thêm .png vào filename



Đã upload thành công

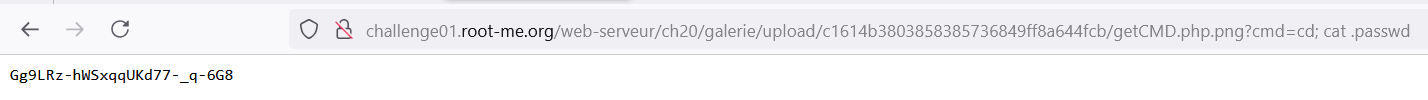


Chọn Show Response In browser, thực hiện các lệnh ngay trên URL



Nhiệm vụ là đọc file .passwd, vậy chỉ cần tìm file đó và cat nội dung

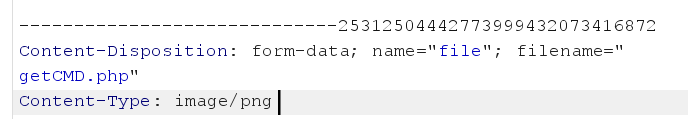




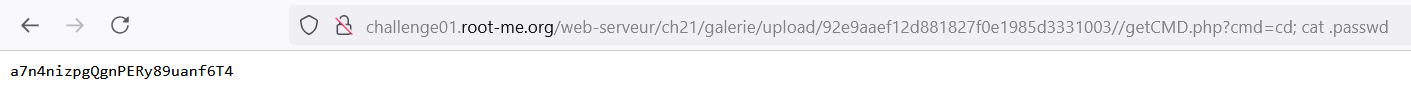
=> Flag: Gg9LRz-hWSxqqUKd77-\_q-6G8

# File upload - MIME type

Tương tự bài trên, tuy nhiên cần sửa Content-Type thành image/png



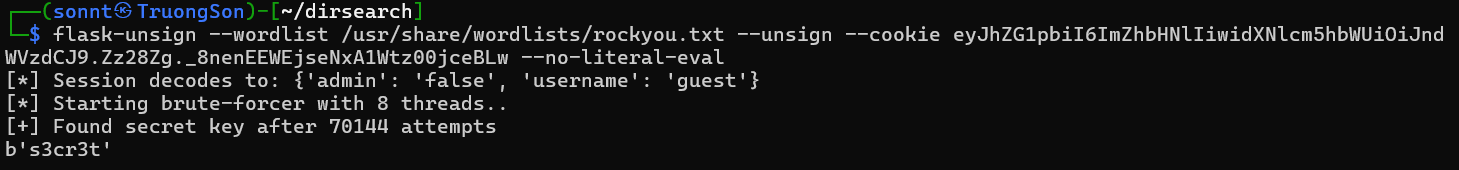
Sau đó thực hiện tương tự bài trên



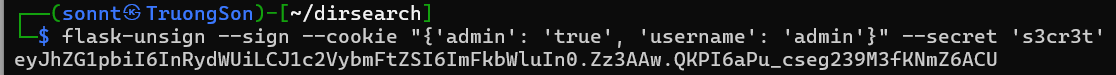
=> Flag: a7n4nizpgQgnPERy89uanf6T4

# 20.Flask - Unsecure session

Chall này sử dụng Flask để lưu thông tin session, vậy chỉ cần sử dụng flask-unsign trên Kali để brute-force secret key là được



Secret key tìm được là 's3cr3t', bây giờ chúng ta cần sign lại với {‘admin’: ‘true’, ‘username’: ‘admin’} với key này



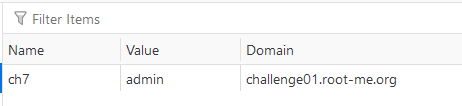
Thay thế giá trị session cũ thành giá trị ở trên, ta lấy được flag do lúc này chúng ta đã là admin

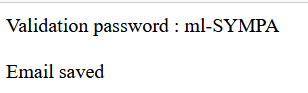


Flag: Fl4sK\_mi5c0nfigur4ti0n

**21. HTTP - Cookies**

Sửa đổi value thành admin

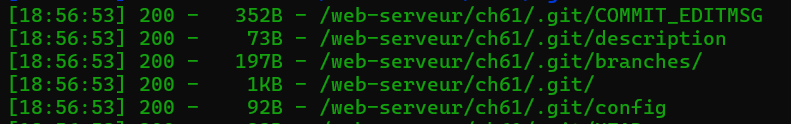




=> Flag: ml-SYMPA

# Insecure Code Management

Sử dụng dirsearch trên Kali thấy rằng chall sử dụng .git để lưu file backup



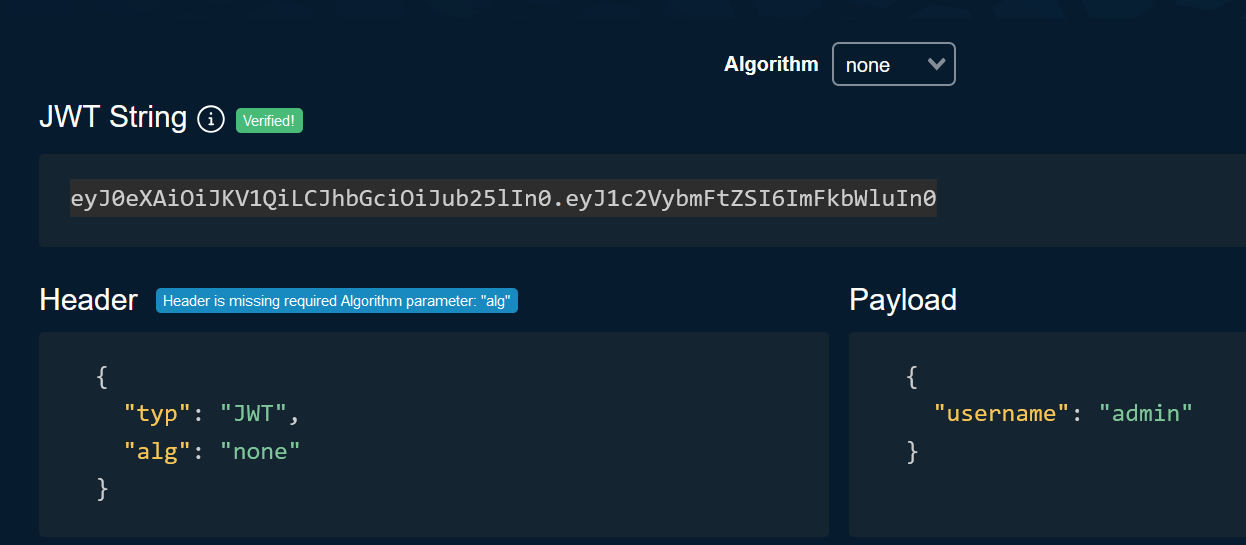
Clone thư mục đó về và kiểm tra lịch sử git, thấy rằng có tài khoản admin có mật khẩu là s3cureP@ssw0rd

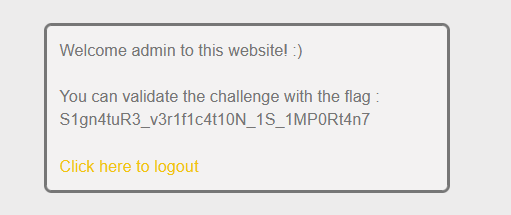
=> Flag: s3cureP@ssw0rd



**23.JWT - Introduction**

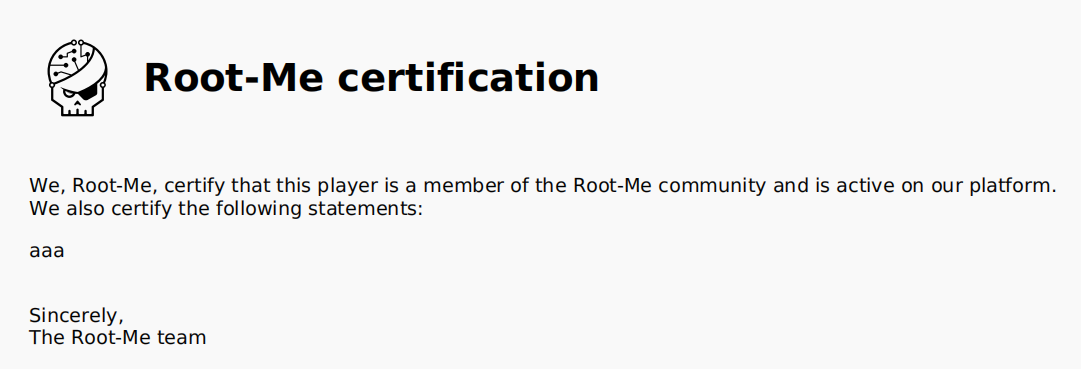
Login bằng tài khoản guest và thay đổi chuỗi jwt với username thành admin => Không vào được trang admin => Đổi thuật toán thành none và thêm dấu ‘.’ cuối chuỗi jwt => Tìm được flag



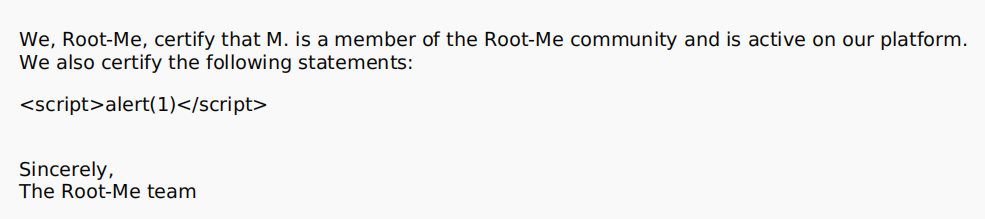


**24. XSS - Server Side**

Ngay khi vào chall ta thấy rằng có 1 text area, thử nhập 1 chuỗi bất kỳ “aaa” thì nhận được phản hồi sau dưới dạng 1 file pdf

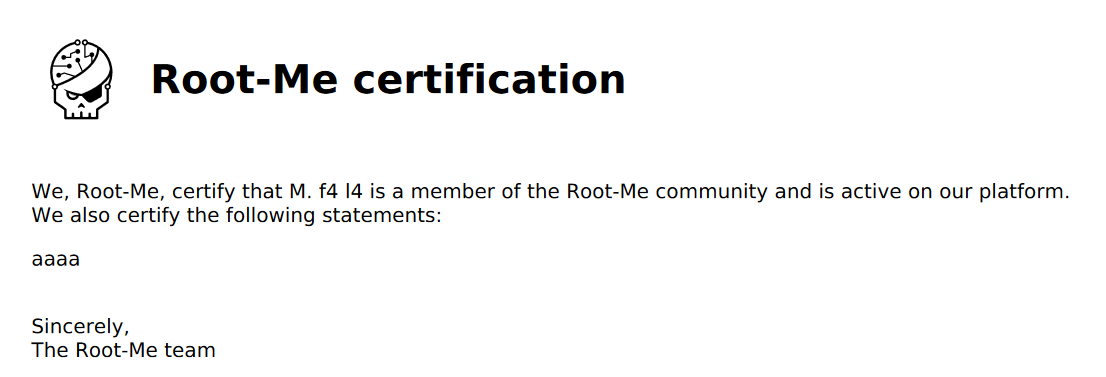


Chuỗi “aaa” được trả về nguyên vẹn, vậy rất có thể khai thác được Reflected XSS. Chúng ta thử chèn <script>alert(1)</script> xem có nhận được cửa sổ popup không, kết quả là script được giữ nguyên vẹn



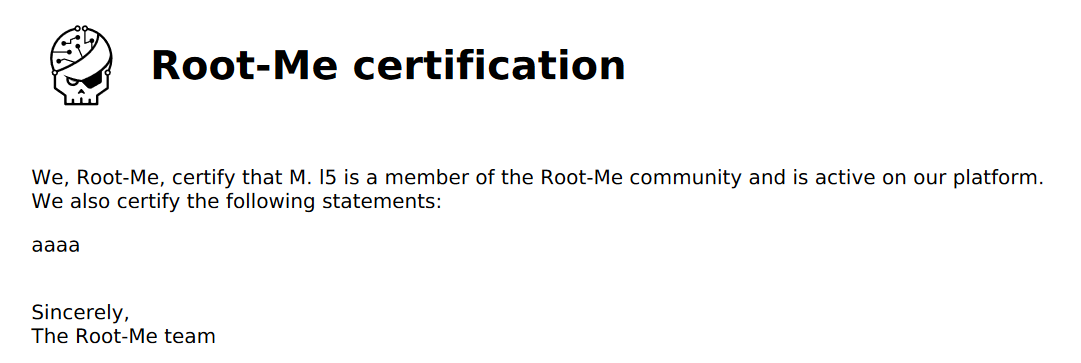
Vậy phần text area này khả năng không khai thác XSS được theo cách thường hoặc liên quan đến việc lọc input. Chúng ta thấy rằng trên web còn 2 button là Sign up và Login, vậy thử tạo tài khoản như sau  


Sau đó thử lại với phần text area và nhận được file sau

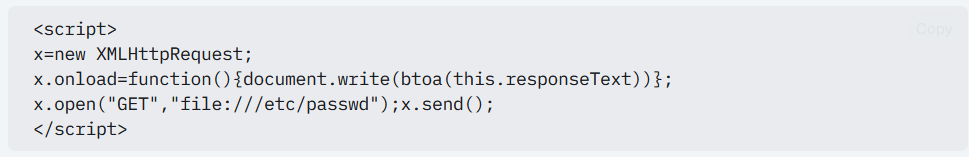


Có thể thấy rằng phần First name và Last name đã bị reflect lại ở sau chuỗi ký tự “M. ”, vậy ta thử chèn script vào phần First name hoặc Last name lúc Sign up để xem có gì xảy ra không

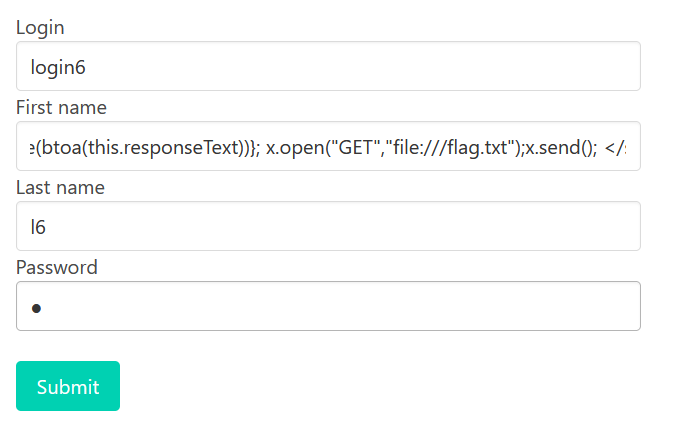




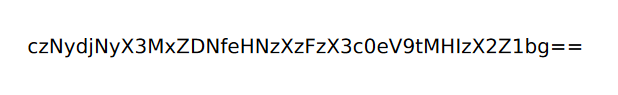
Có thể thấy First name đã được thực thi, nhưng việc thực thi đó khả năng cao được thực hiện ở phía máy chủ, vậy thay vì alert(), chúng ta có thể chèn script đọc nội dung file /flag.txt. Sign up lại và thực hiện điều này với script sau, được lấy từ <https://book.hacktricks.xyz/pentesting-web/xss-cross-site-scripting/server-side-xss-dynamic-pdf>.



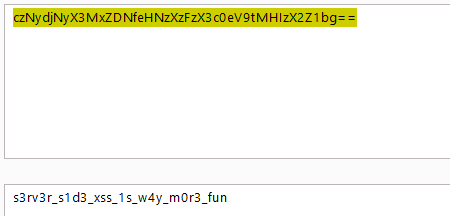
Thay /etc/passwd thành /flag.txt



Nhập nội dung vào text area và generate, lúc này chúng ta nhận được nội dung /flag.txt dạng base64, giờ chỉ cần giải mã và lấy cờ



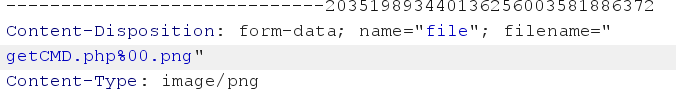
Giải mã



=> Flag: s3rv3r\_s1d3\_xss\_1s\_w4y\_m0r3\_fun

# File upload - Null byte

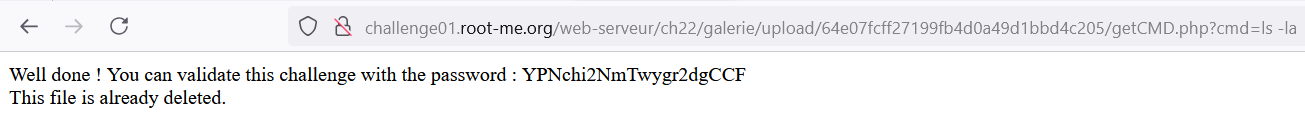
Chall này chỉ chấp nhận các file dạng ảnh, nếu sửa extension thành .png hay sửa Content-Type thành image/png như bình thường thì không bypass được, vậy thì thử chèn null byte vào sau extension .php và sửa Content-Type thành image/png



Kết quả đã upload thành công



File upload lên là webshell dùng để thực thi command, làm tương tự như 2 bài file upload trước là lấy được cờ



=> Flag: YPNchi2NmTwygr2dgCCF

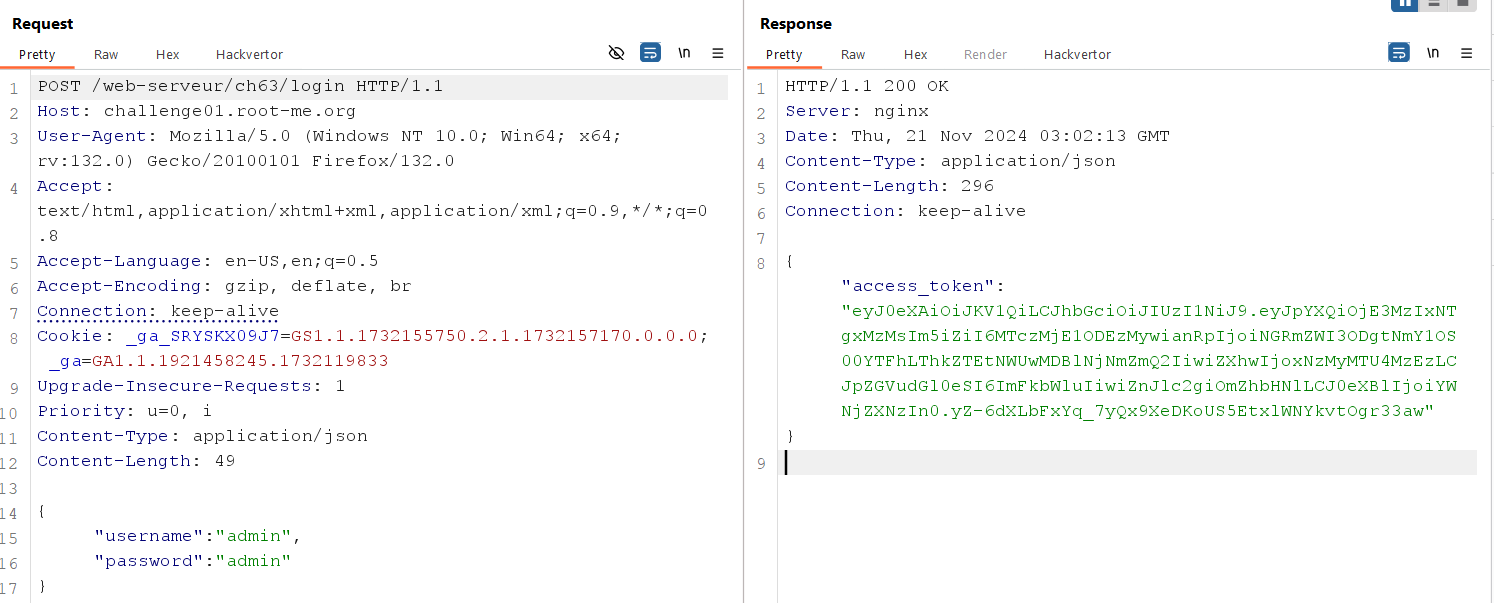
# JWT - Revoked token

Chall này yêu cầu gửi 2 request:

POST: login vào admin

GET: lấy flag từ admin

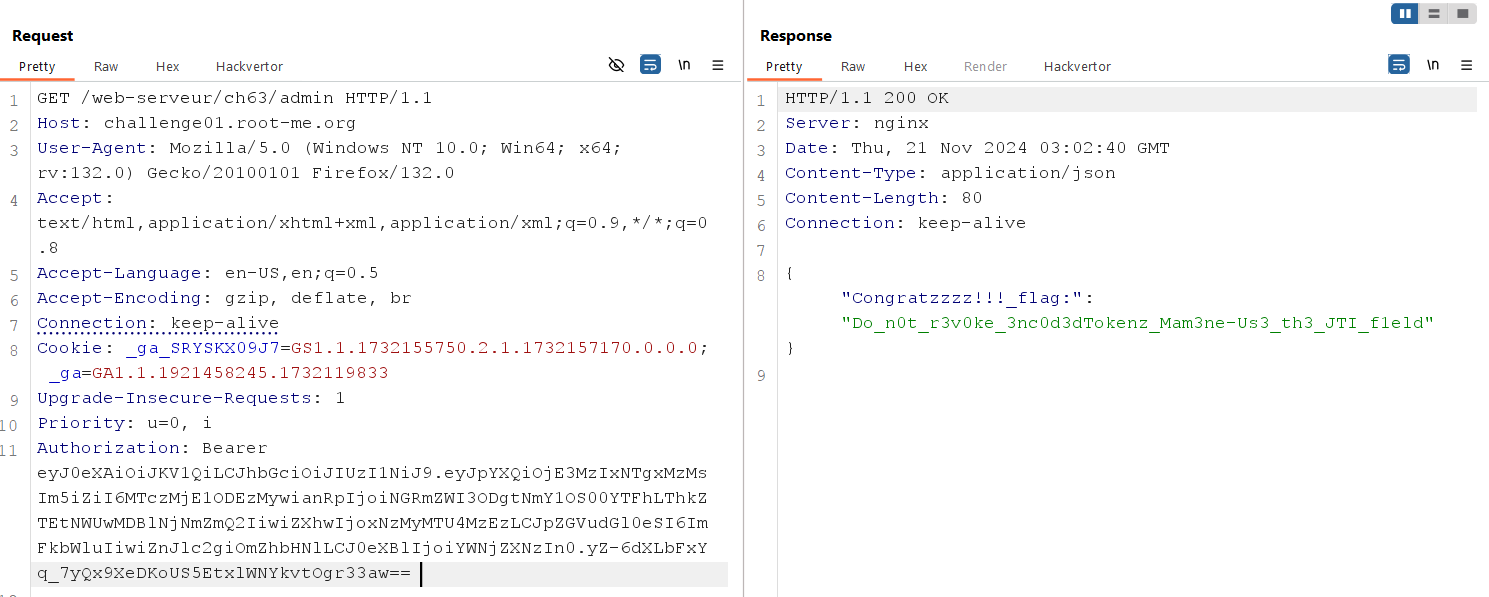
Gửi rq POST đăng nhập vào admin



Lấy token ở response paste vào header Authorization: Bearer, thấy rằng token đã bị revoke. Xem source code thì thấy do token nằm trong blacklist



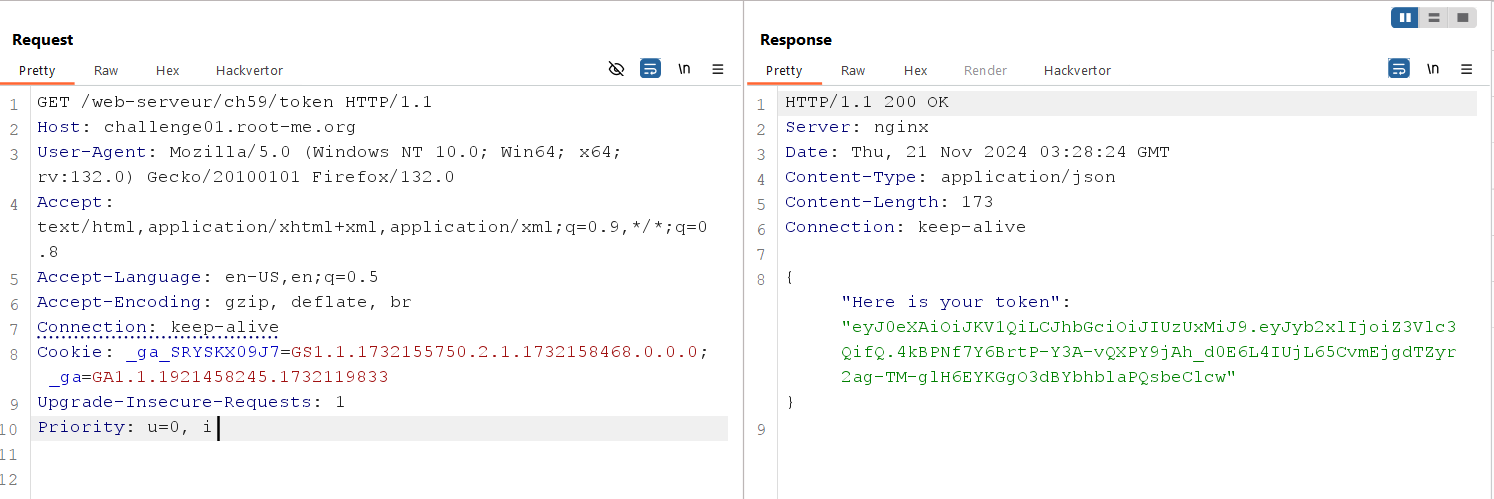
Vậy ta cần thêm chuỗi “==” ở cuối để lấy flag



=> Flag: Do\_n0t\_r3v0ke\_3nc0d3dTokenz\_Mam3ne-Us3\_th3\_JTI\_f1eld

# JWT - Weak secret

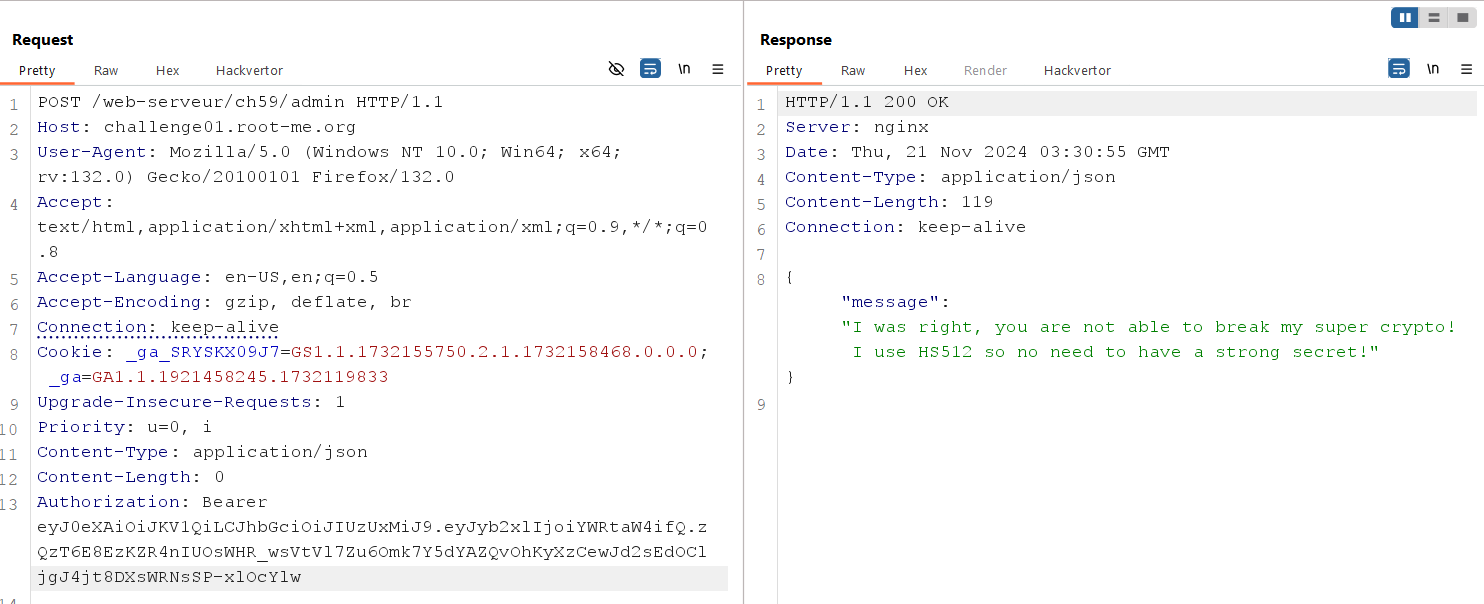
Chall này yêu cầu gửi request đến /admin kèm theo 1 token để lấy flag. Trước hết cần lấy token



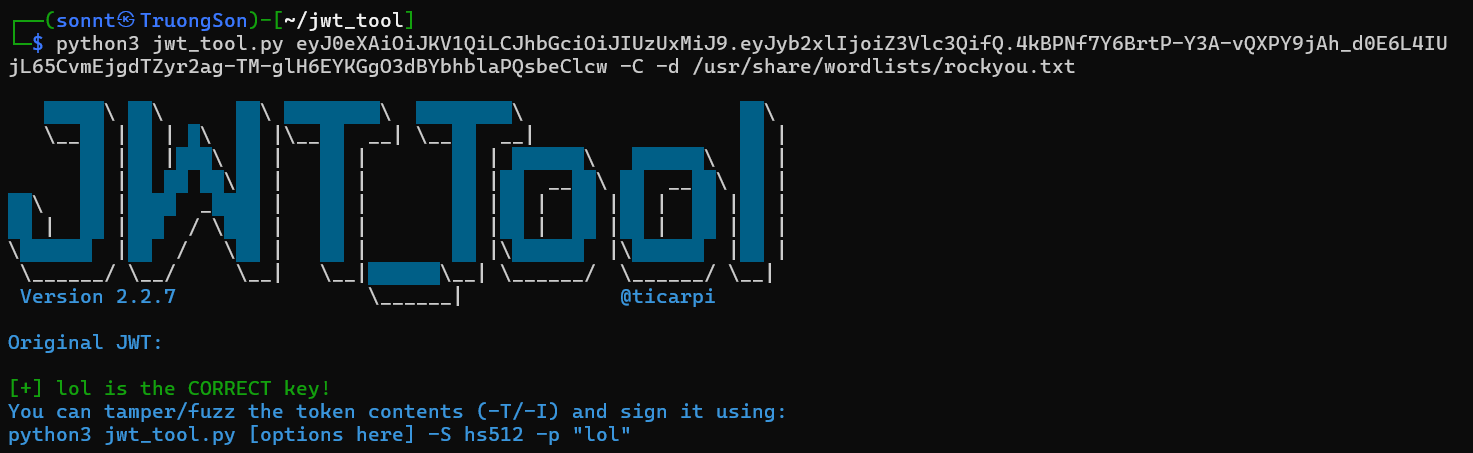
Token này sử dụng algo HS512, phần payload là guest



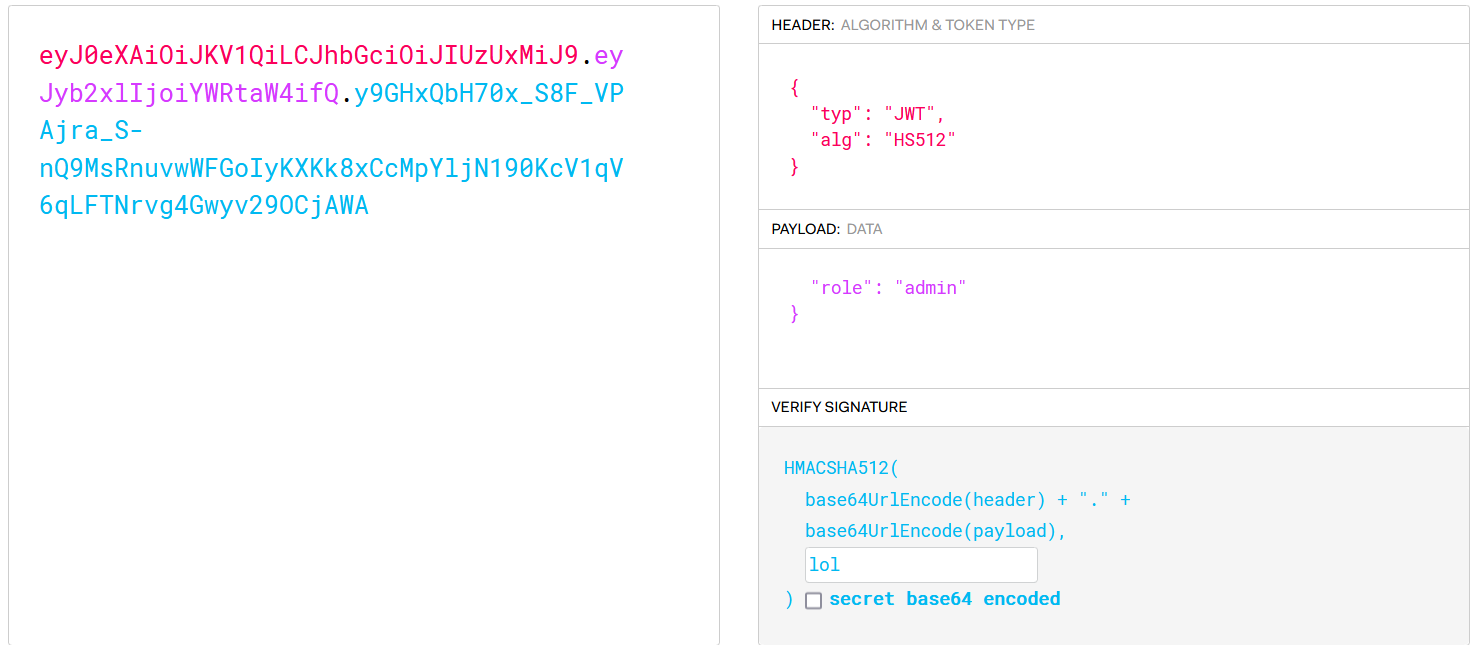
Thay lại role thành admin rồi dán token vào request admin



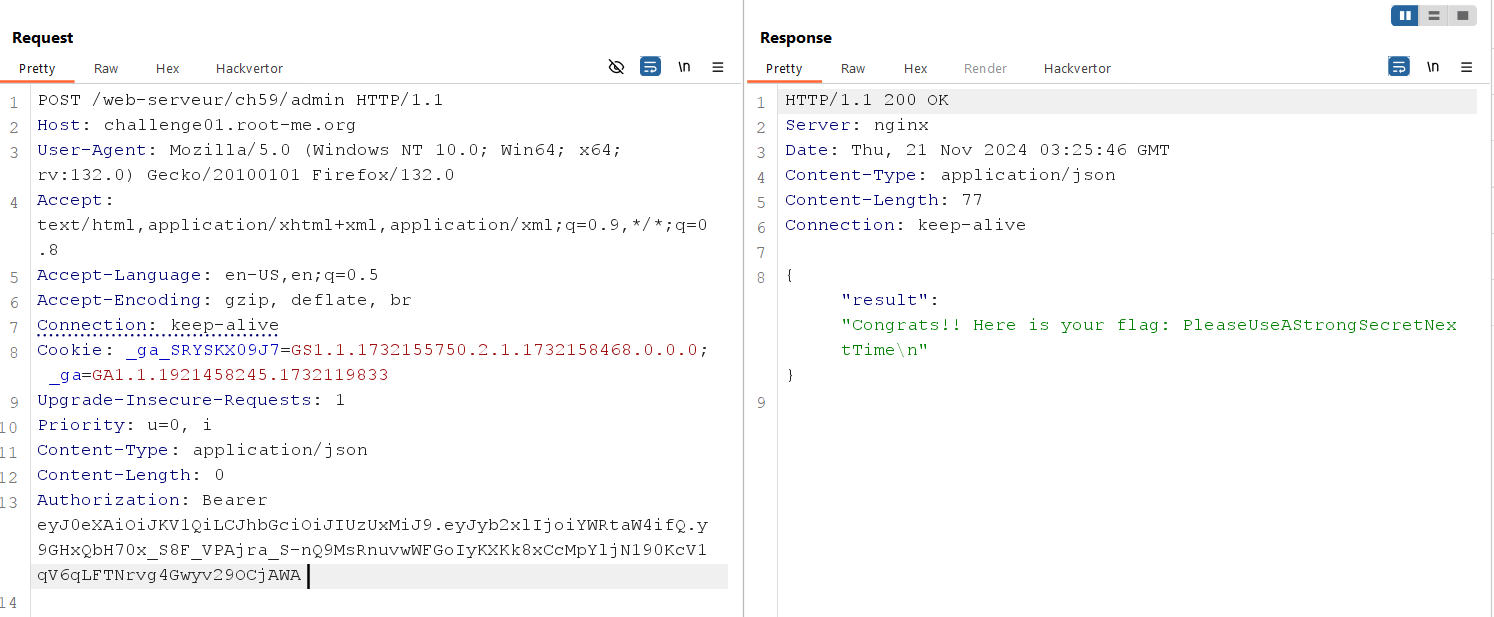
Vẫn chưa lấy được cờ, thử lại bằng cách đổi algo thành none nhưng vẫn không được, lúc này ta nghĩ đến brute-force tìm secret key



Secret key tìm được là “lol”, vậy ta thu được token mới như ảnh dưới



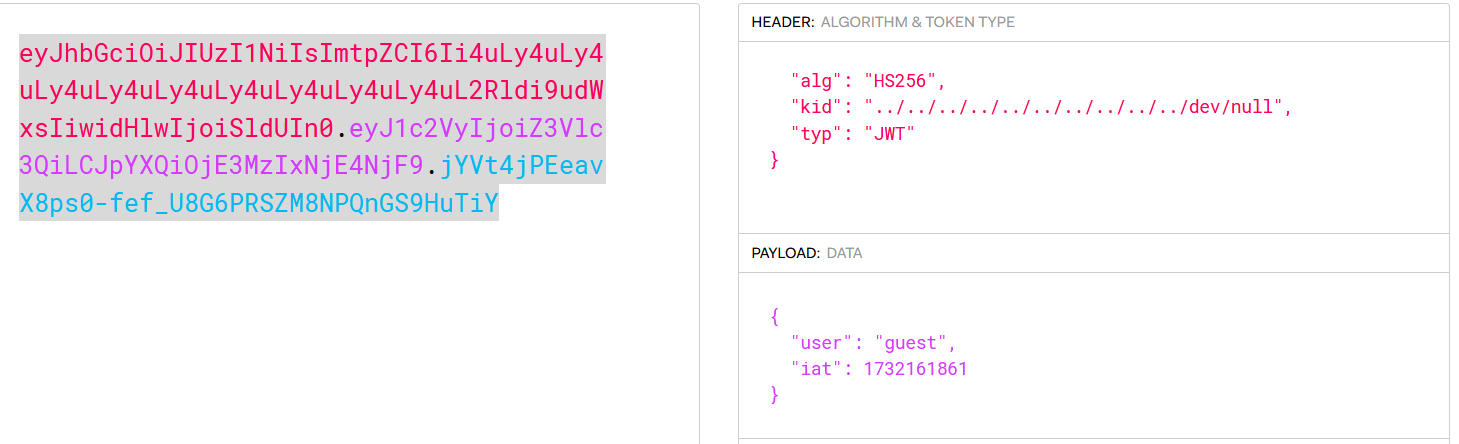
Thay đổi token cũ thành token mới tìm được, chúng ta lấy được flag



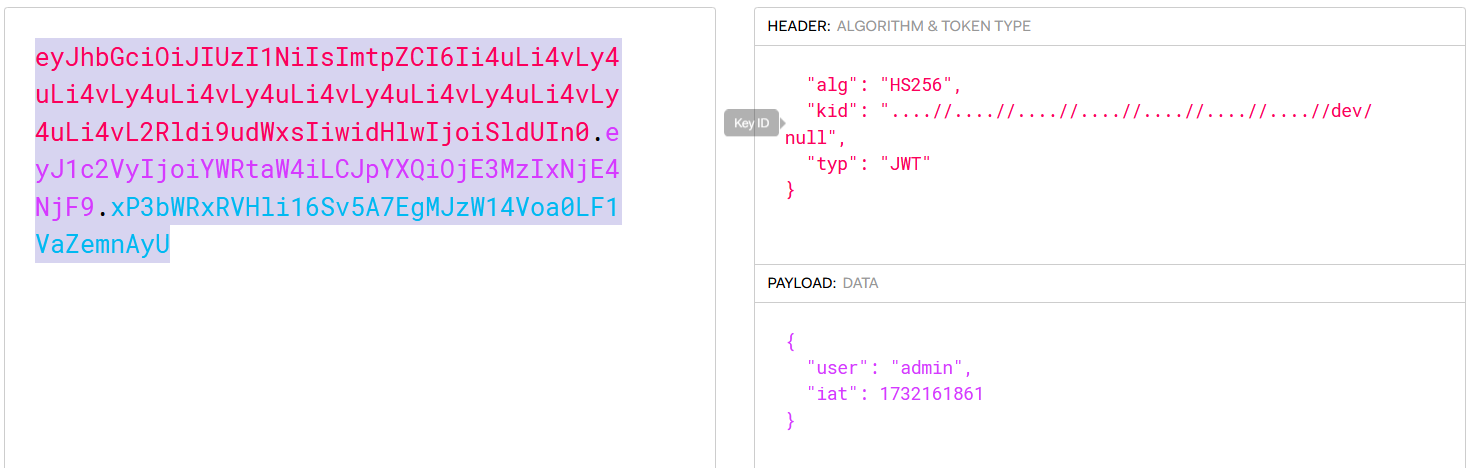
=> Flag: PleaseUseAStrongSecretNextTime

# JWT - Unsecure File Signature

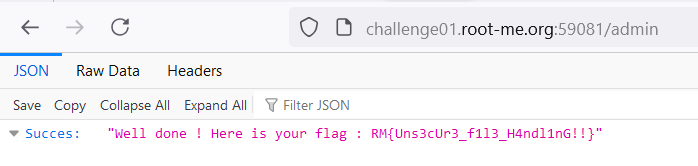
Chall này có jwt được thêm giá trị kid ở header, kid dùng để xác định đường dẫn key dùng để mã hóa hoặc giải mã token, giá trị này có thể bypass bằng path traversal, sử dụng file /dev/null - 1 file rỗng trong linux để khiến kid trở thành rỗng. Thử bypass bằng path traversal thông thường



Có vẻ như sử dụng ../ không đem lại hiệu quả (không vào được admin), vậy chúng ta hãy thử cách bypass khác, sử dụng ….// (signature hiện tại để rỗng)



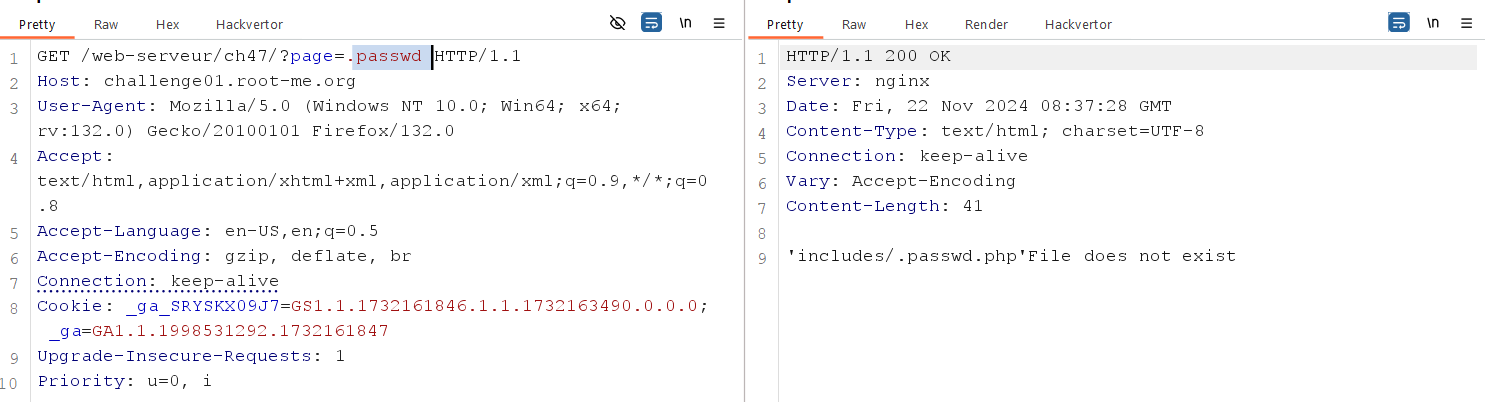
Bypass thành công



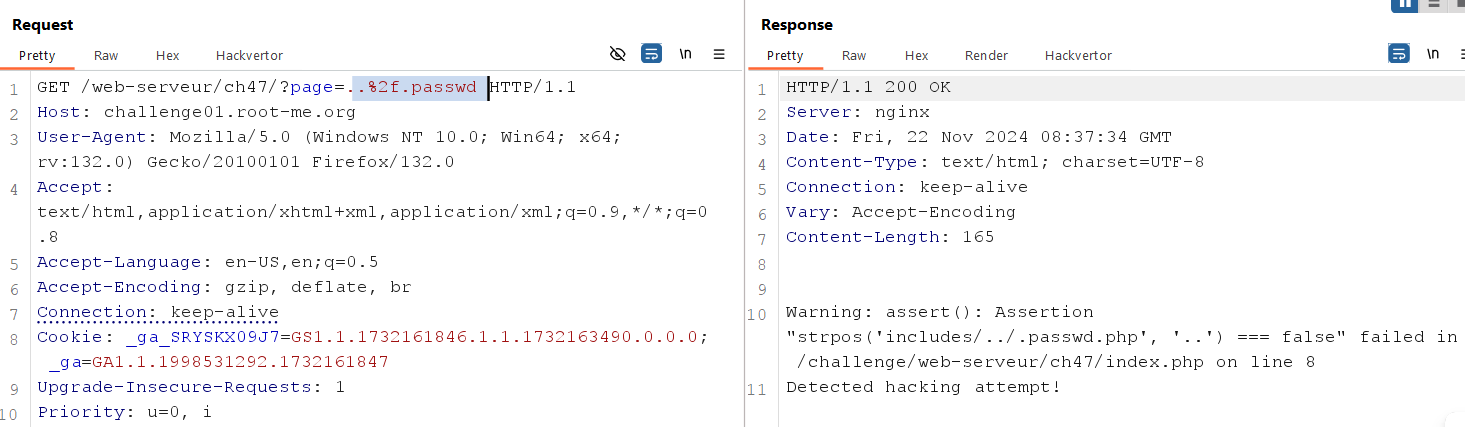
Flag: RM{Uns3cUr3\_f1l3\_H4ndl1nG!!}

1. PHP - assert()

Mục tiêu chall này là đọc được nội dung file .passwd, để ý rằng url chứa parameter có thể khai thác bằng path traversal. Khi thay thế home (hay contact, about) thành .passwd, kết quả là không tìm thấy file



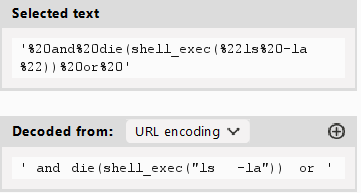
Vậy thì thử bằng cách thêm ../

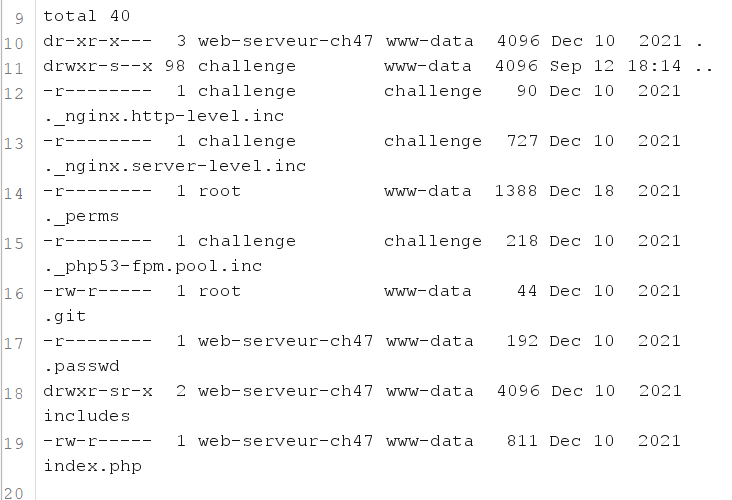


Có thể thấy rằng sử dụng server phát hiện chuỗi ký tự ‘..’ có xuất hiện và thực hiện ngăn chặn, có thể thử lại bằng cách encode nhưng vẫn bị filter. Tìm trên google về bypass assert() trong PHP thì thu được kết quả

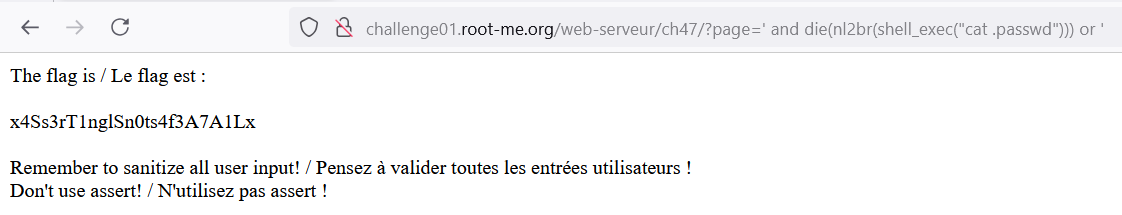


Vậy thử áp dụng vào trường hợp này thì bypass thành công





Tiến hành đọc file .passwd

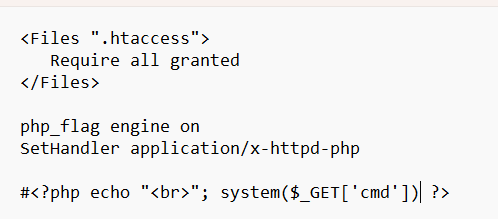


=> Flag: x4Ss3rT1nglSn0ts4f3A7A1Lx

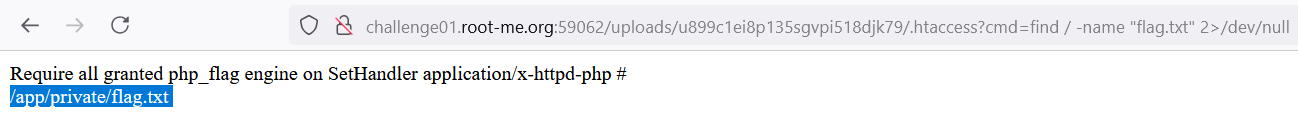
# PHP - Apache configuration

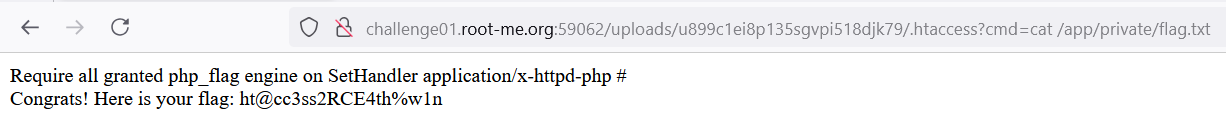
Chall này chặn tất cả các file php, nếu thử bypass bằng thêm extension, nullbyte, exiftool… thì đều không được.

Nhận thấy rằng trong response trả về, server sử dụng apache, vậy ta thử can thiệp vào tệp .htacess cho phép kích hoạt engine php



Sau đó tìm kiếm và đọc file flag.txt

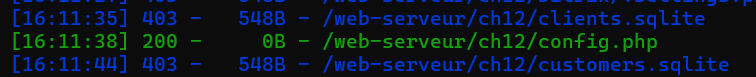




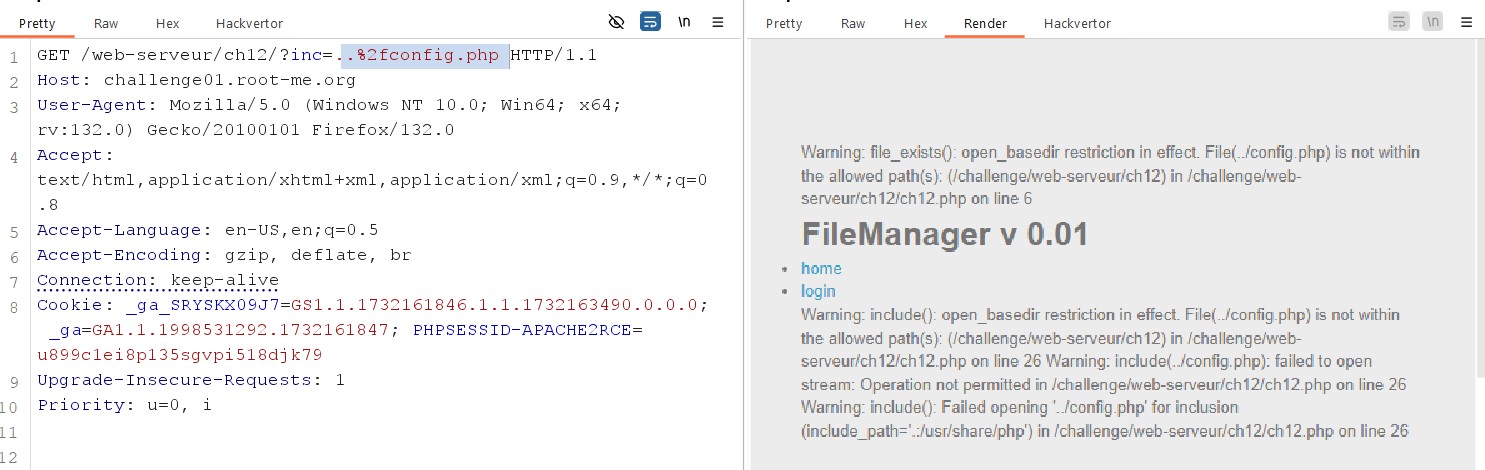
=> Flag: ht@cc3ss2RCE4th%w1n

# PHP - Filters

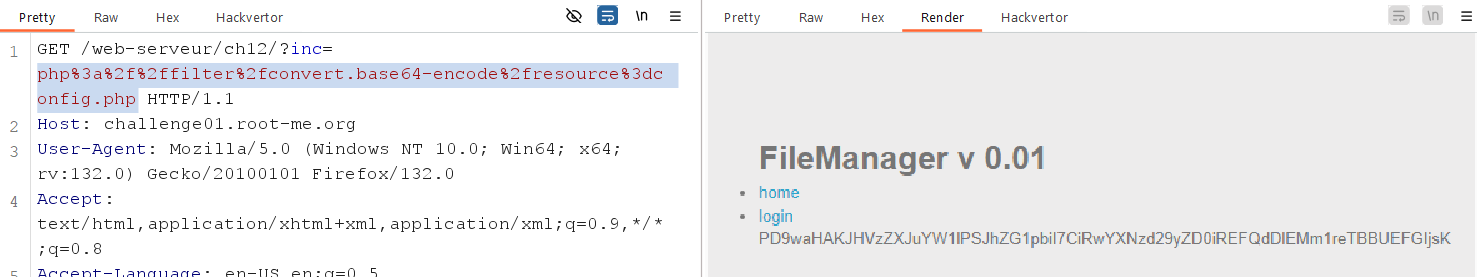
Chall này cần đăng nhập vào admin để lấy flag. Sau khi thử SQLi không thành công, thử quét đường dẫn bằng dirsearch thì thu được file config



Sau khi thử bypass bằng cách thêm ../, nhận thấy rằng đường dẫn không hợp lệ



Nhận thấy rằng response trả về có chứa hàm include(), thường xuất hiện khi có lỗ hổng LFI, và có thể sử dụng các kỹ thuật như encode bằng base64 để bypass



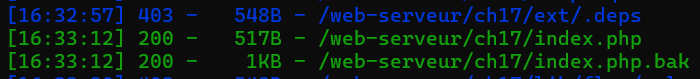
Gửi lại request đã encode bằng base64, thu được kết quả như trên, giờ chỉ cần giải mã là thu được username và password



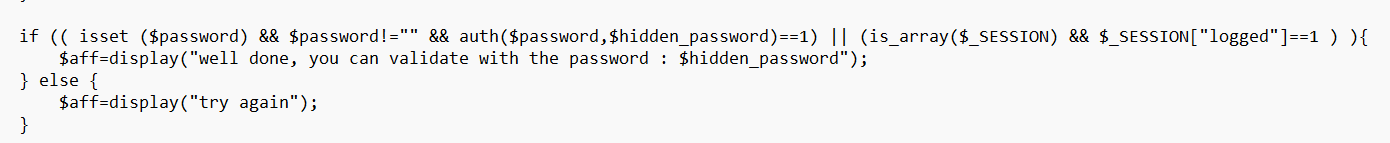
=> Flag: DAPt9D2mky0APAF

1. **PHP - register globals**

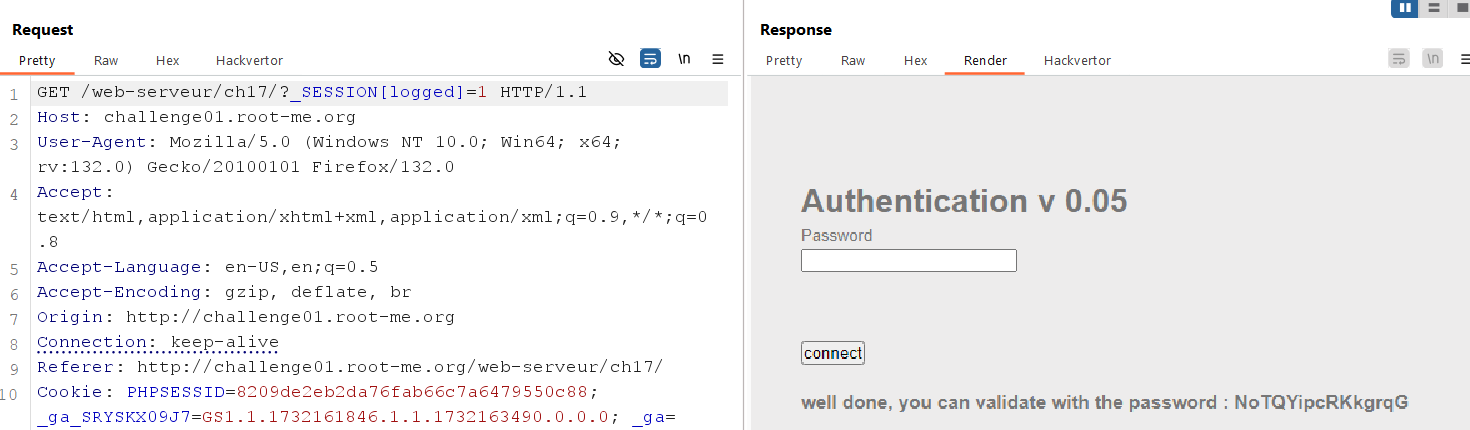
Thử bypass bằng SQLi không được => Quét dir tìm file backup



Khi truy cập vào đường dẫn trên, 1 file được tải về, giá trị cần chú ý là hidden\_password



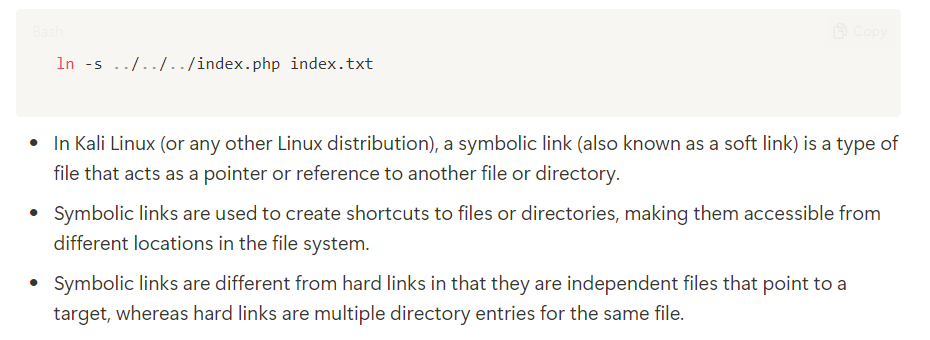
Dựa vào code, có thể thấy rằng lấy được hidden password nếu \_SESSION[“logged”]=1



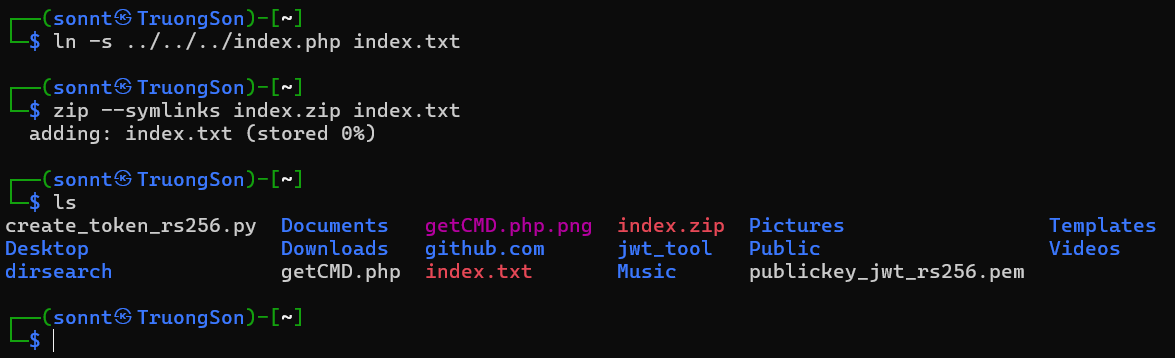
=> Flag: NoTQYipcRKkgrqG

1. **File upload - zip**

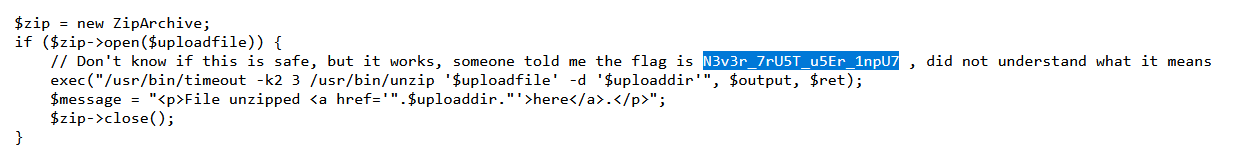
Ở chall này, khi upload file zip lên, chúng ta sẽ không truy cập được vào file vừa up. Do đó cần sử dụng soft link



Mục tiêu là đọc file index.php, vậy cần thực hiện tạo file index.txt rồi zip lại



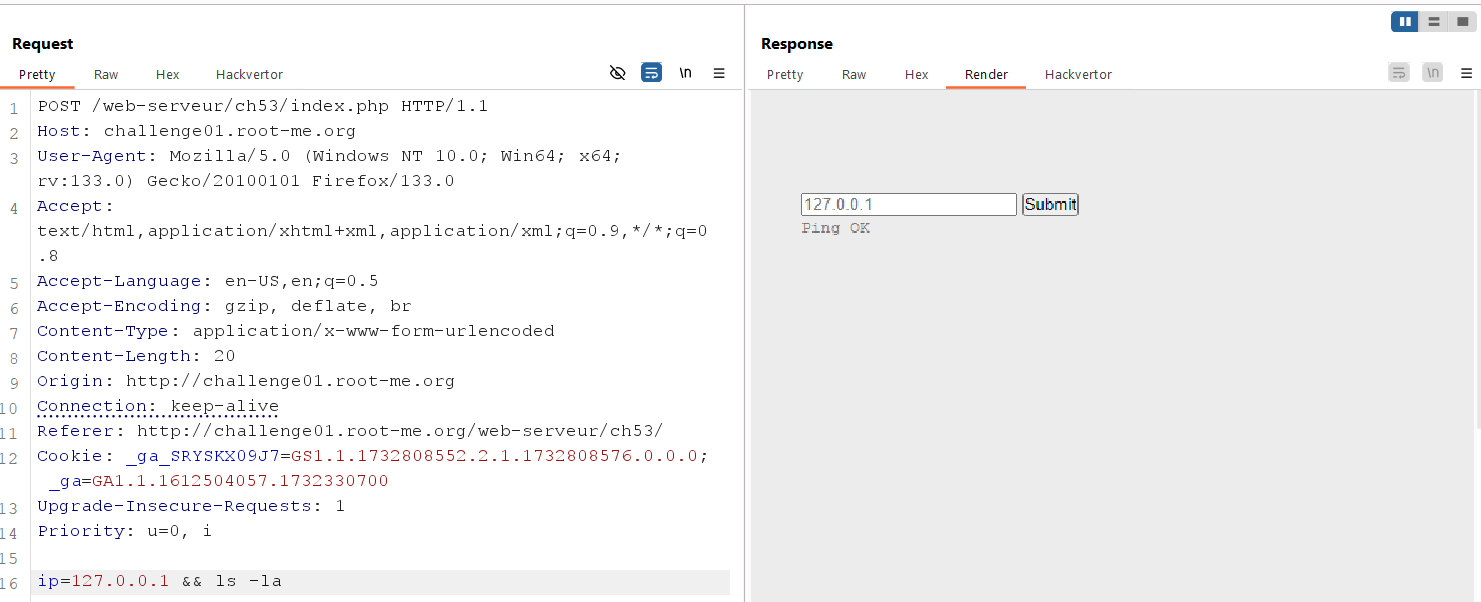
Up file index.zip lên rồi lấy flag



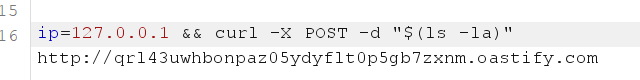
=> Flag: N3v3r\_7rU5T\_u5Er\_1npU7

# Command injection - Filter bypass

Chall này filter đi dấu “;” vốn dùng để ngắt lệnh, do đó thử && thì Ping OK



Tuy nhiên kết quả không hiện gì nên đây là dạng blind, vậy thử dùng burp collaborator đọc kết quả qua lệnh curl

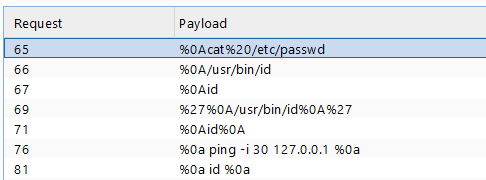


Tuy nhiên kết quả cũng không hiện, vậy ta thử cách bypass khác, sử dụng Intruder với từ điển sau <https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Command%20Injection/Intruder/command-execution-unix.txt>

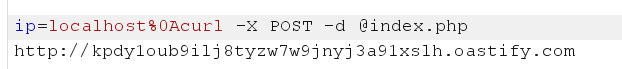
Position là đoạn ký tự sau localhost



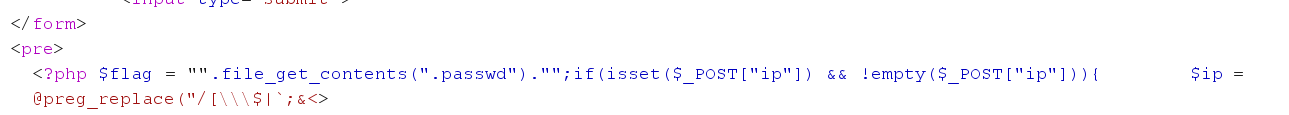
Kết quả thu được gồm các payload bắt đầu bằng %0A với dấu hiệu là length nhỏ hơn các payload còn lại



Thử lại bằng %0A



Kết quả là Poll thành công



Flag lấy từ file .passwd, vậy chỉ cần gửi lại nội dung file .passwd đến burp collab

=> Flag:

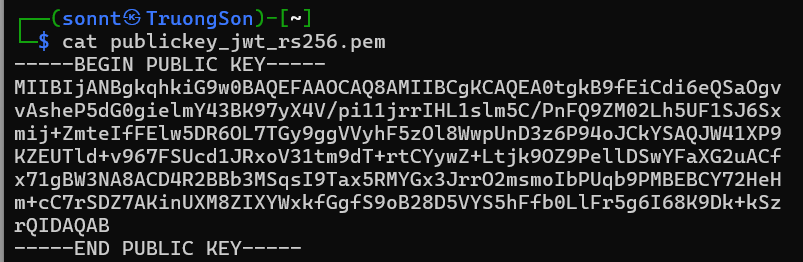


1. **JWT - Public key**

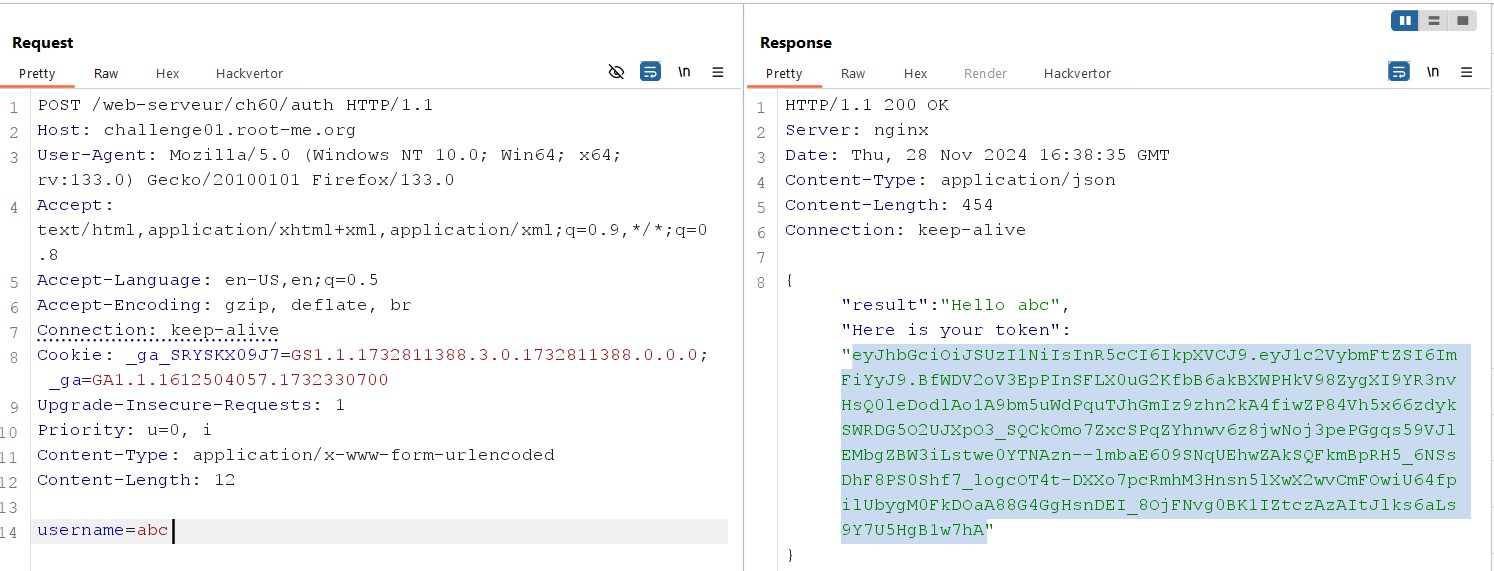
Gửi GET request và nhận public key



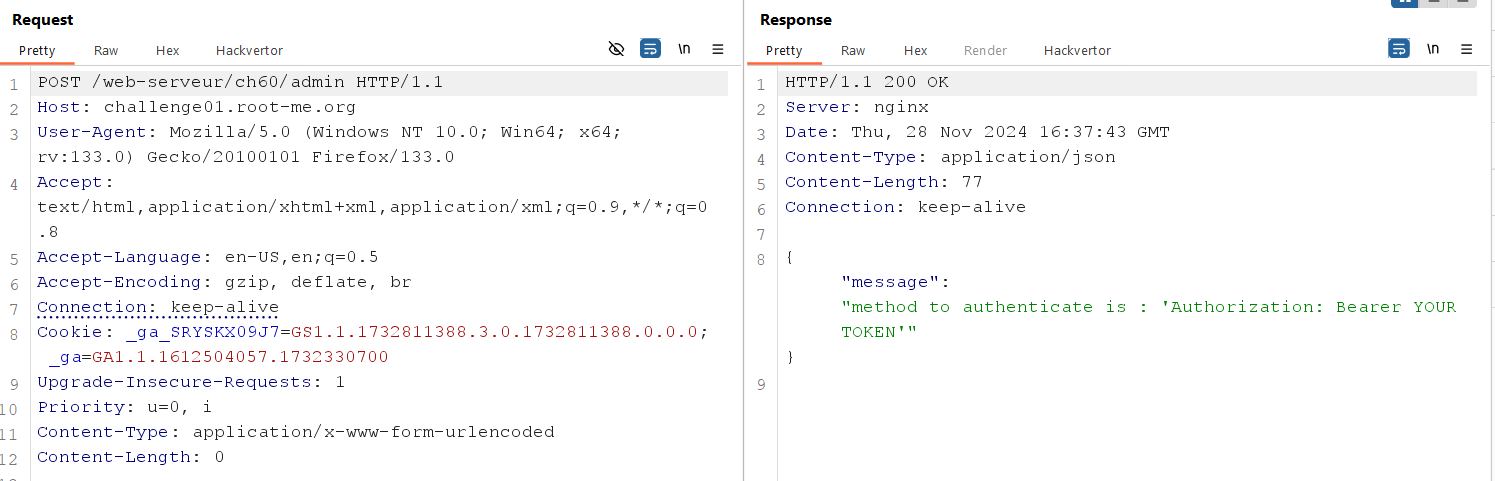
Chỉnh lại format rồi lưu lại vào 1 file



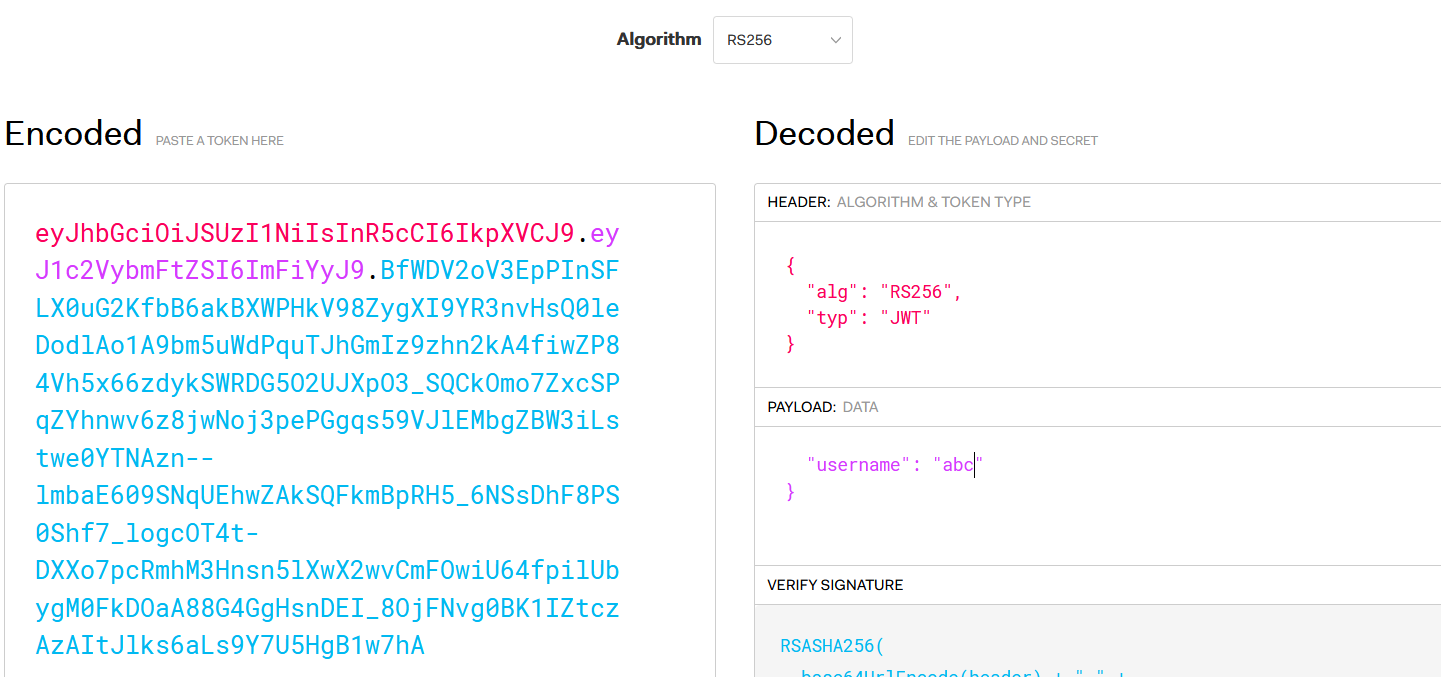
Khi gửi POST request /auth thì server bắt phải nhập thêm username, thử nhập admin thì không được nên thử lại với 1 username bất kỳ, lúc này nhận được 1 token



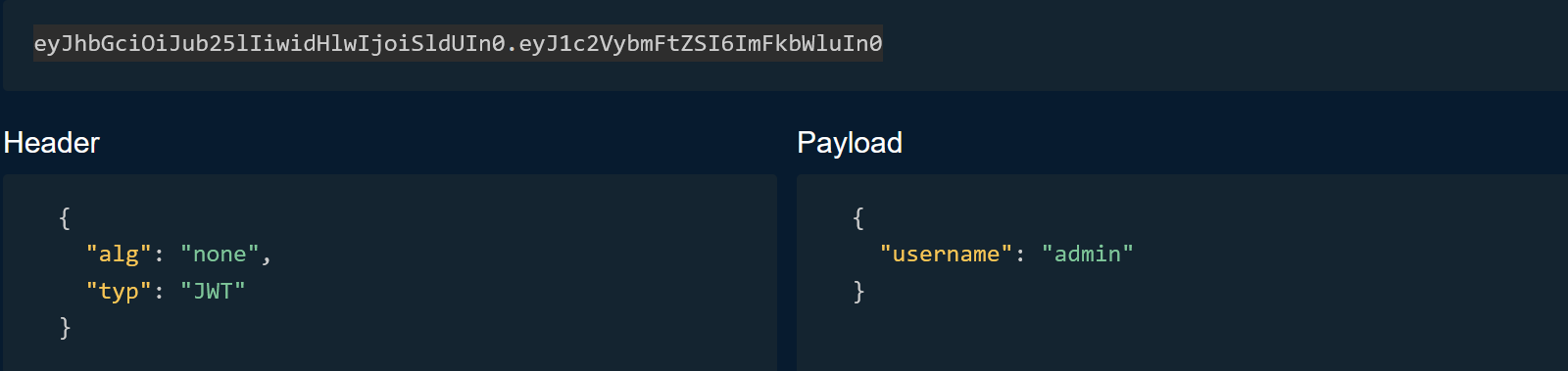
Có thể token này dùng để xác thực phía admin, gửi POST /admin thì thấy đúng là cần token này



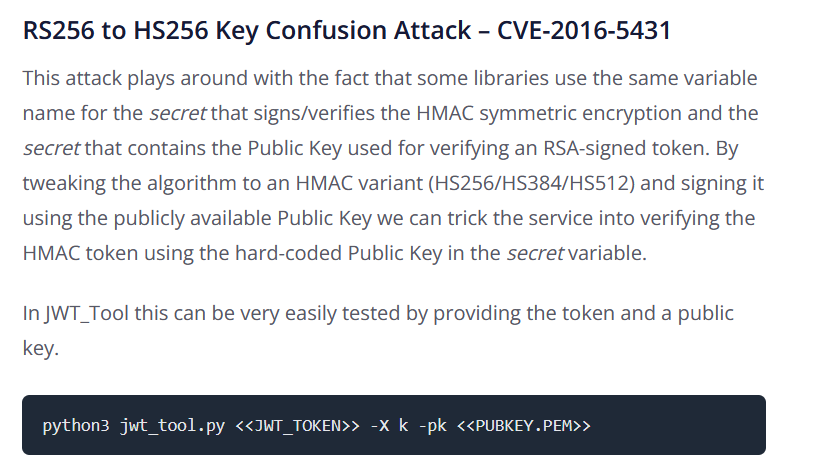
Khi gửi lại POST /admin thì thấy rằng token lấy được không hợp lệ, điều này là hiển nhiên vì token này dùng algo RS256 với khóa public dùng mã hóa và khóa private dùng giải mã



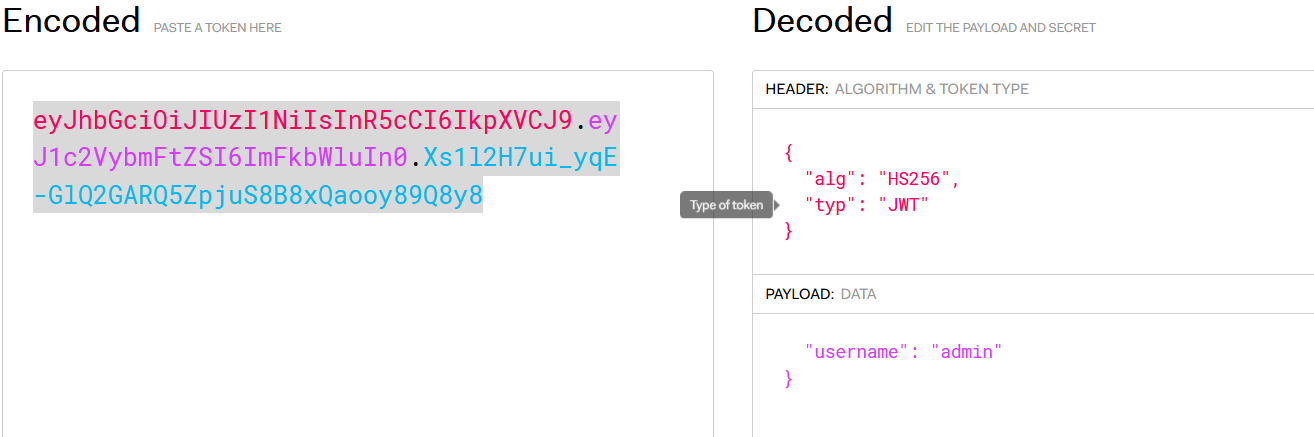
Ta thử sửa lại algo thành none và username thành admin



Tuy nhiên vẫn không xác thực được admin. Tìm trên google thì thấy có 1 CVE liên quan đến RS256 như sau



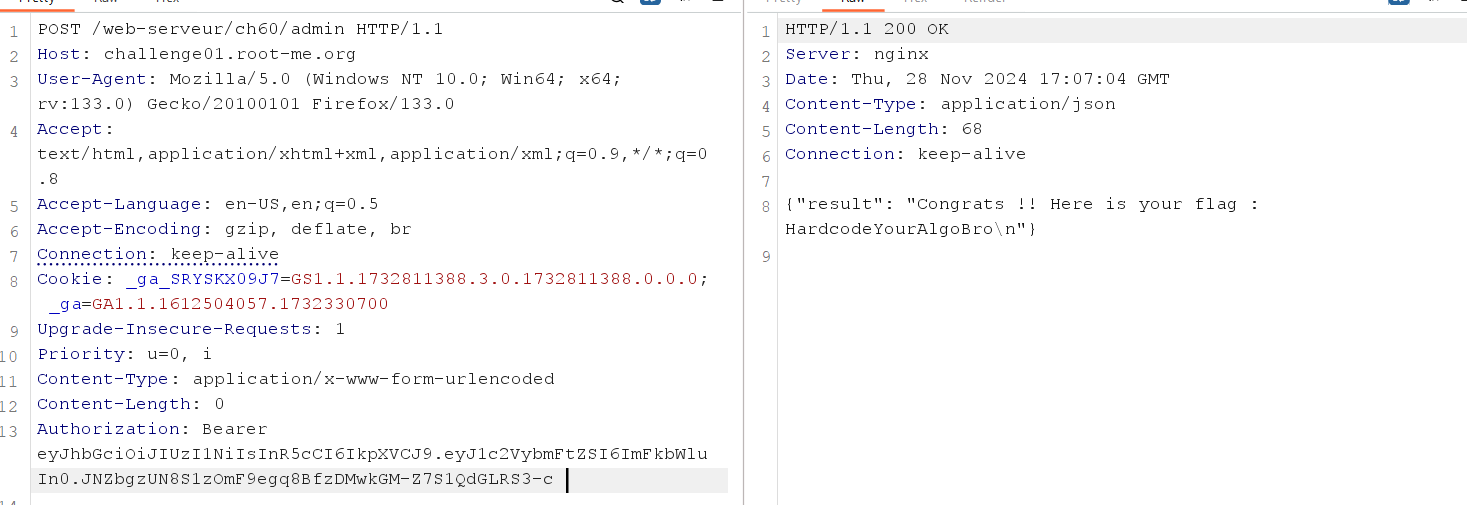
Vậy chúng ta chỉ cần chuyển thuật toán từ RS256 thành HS256 (hoặc 384, 512) rồi dùng jwt tool, sau đó dùng chính public key để ký. Trước hết đổi algo thành HS256 và username thành admin



Sau đó ký bằng public key



Gửi lại POST /admin với token mới thu được



=> Flag: HardcodeYourAlgoBro