

SQL injection - Authentication - GBK

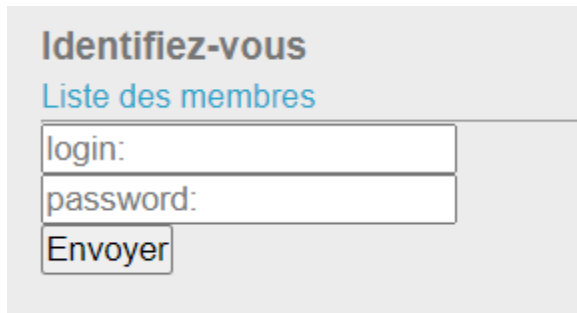
Title: Do you speak chinese ?

Point: 30 Points

Description: Get an administrator access.

Solution:

Vào trang web, có một login form cần bypass.



Identifiez-vous

Liste des membres

login:

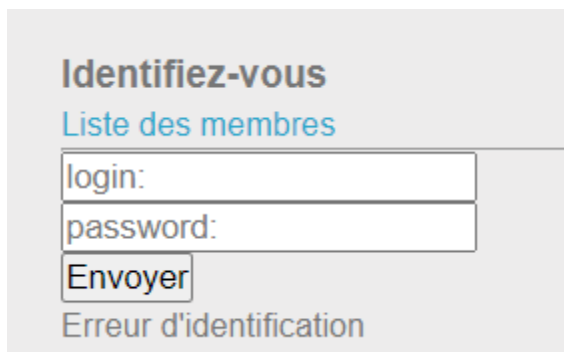
password:

Envoyer

NOTE: Trong chall lần này, chúng ta tấn công vào ô input đầu tiên (login). Còn ô input thứ 2 thì sau khi test thấy không bị vulnerable

Thử một đoạn attack nhỏ:

admin' or 1=1 --



Identifiez-vous

Liste des membres

login:

password:

Envoyer

Erreur d'identification

Một message từ chối báo lỗi. Hãy nhớ lại title nói gì không "Do you speak chinese ?" và tên của challenge liên quan gì đó đến **GBK**. Sau khi google thì chúng ta biết **GBK** là một dạng encode của ký tự Trung Quốc.

[Main reference](#). Và cái **GBK** này liên quan đến một thủ thuật có tên là "Bypassing the addslashes()". Bây giờ mọi thứ đã rõ, bên server back-end sẽ có một hàm hay cách nào đó và sẽ "làm sạch" dữ liệu input của chúng ta bằng cách thêm backslash "\" vào trước ký tự đặc biệt như ", ' , \ và sẽ khiến chúng bị escape và query của chúng ta sẽ bị sai.

Nhiệm vụ của ta là bypass nó. Challenge đã gợi ý về việc sử dụng mã hóa **GBK**, cái này có liên quan đến từ khóa "Multibyte character set", là dạng mã hóa nôm na là có độ dài đến tận 2 bytes. Một số loại người ta thường dùng là **Shift-jis, jis, euc-jp, euc-kr, ...** và Challenge đề xuất chúng ta dùng **GBK**.

Ý tưởng của chúng ta là sẽ thêm một byte ký tự (có chọn lọc) để sau khi server thêm ký tự backslash “\” thì sẽ kết hợp với ký tự chúng ta tạo thành 2 bytes ký tự Tung Của. Tức là ta sẽ xóa gián tiếp ký tự backslash bằng việc tạo thành ký tự Tung Hoa. Ví dụ:

Lưu ý về url-encoded: ' -> %27 , \ -> %5C

Payload mình sẽ có dạng:

%xx%27 or 1=1 -- -> Tương đương với **%xx' or 1=1 --**

Server thêm **backslash \ (%5C)**:

(%xx%5C)%27 or 1=1 -- -> Tương đương **%xx\' or 1=1 --**

Và **%xx%5C** sẽ tạo thành **ký tự GBK**. Bên server sau khi hoàn thành xong việc “làm sạch” dữ liệu input : **{character}' or 1=1 --**

Vậy thì mình sẽ thêm byte ký tự nào? Tham khảo bảng dưới (từ link bên trên):

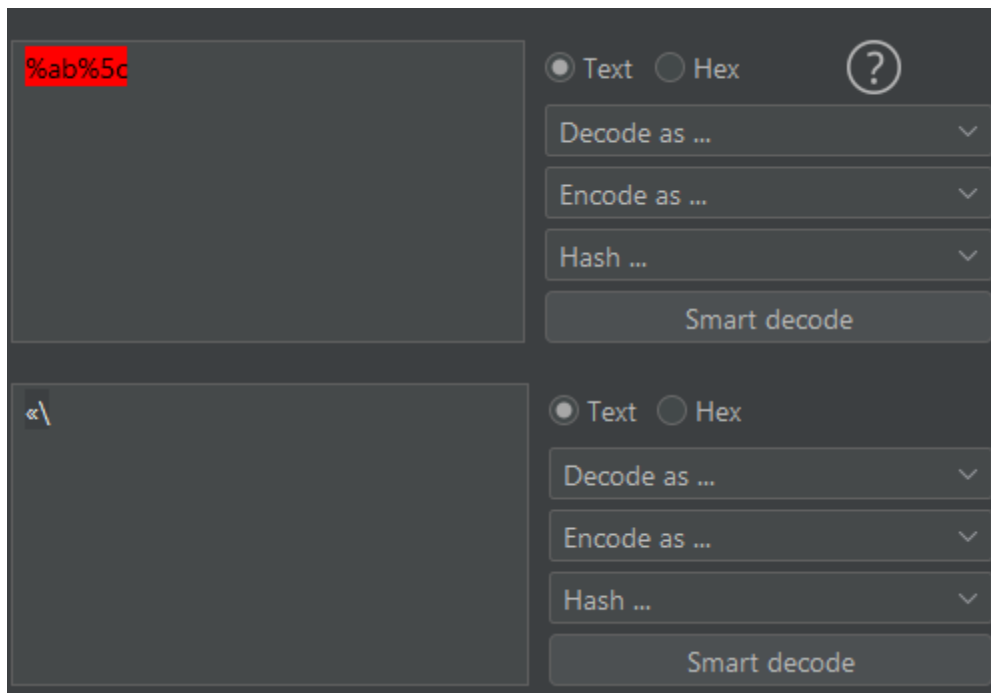
range	byte 1	byte 2	code points	characters			
				GB 18030	GBK 1.0	Codepage 936	GB 2312
Level GBK/1	A1 – A9	A1 – FE	846	718 ^[6] 8–10	717	715	682
Level GBK/2	B0 – F7	A1 – FE	6,768		6,763	6,763	6,763
Level GBK/3	81 – A0	40 – FE except 7F	6,080		6,080	6,080	
Level GBK/4	AA – FE	40 – A0 except 7F	8,160		8,160	8,080	
Level GBK/5	A8 – A9	40 – A0 except 7F	192		166	153	
user-defined 1 ^[6]	AA – AF	A1 – FE	564				
user-defined 2	F8 – FE	A1 – FE	658				
user-defined 3	A1 – A7	40 – A0 except 7F	672				
total:			23,940	21,887	21,886	21,791	7,445

1. Optional Choosing

2. %5C is in this range

%5C ** sẽ được thêm vào trước nháy đơn **%27 ' và sẽ ở vị trí byte thứ 2 . Byte đầu tiên thì theo bảng, miễn sao mình chọn đúng vùng giá trị là OK. Các bạn có thể thay đổi payload của mình bằng cách chọn vùng giá trị khác.

Ví dụ , Mình chọn **%AB** . Mình có thể check xem thử giá trị **%AB%5C** có hợp lệ hay không bằng cách chuyển đổi **%AB%5C -> «** (Unicode char). Sử dụng BurpSuite).



Tiếp theo sử dụng [This site](#) :

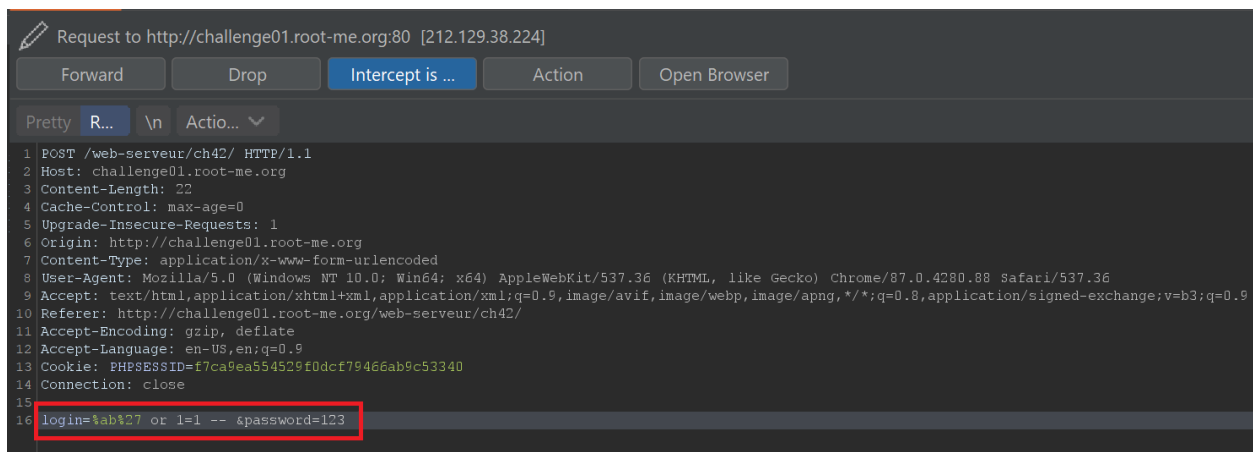
Chúng ta sẽ thấy «\ được encode thành 玦. Vậy nên payload trên của chúng ta sẽ thành : 玦' or 1=1 -- . Cái này sẽ “thông chốt” được login form.

Sử dụng "Burp Suite" để thực hiện. Sử dụng trực tiếp browser sẽ “hơi khó” cho việc mình nhập ký tự thực sự (Unicode) vào input , những ký tự như AB" hoặc "%27". Bên cạnh đó, form sử dụng POST để request nên mình chỉ có thể thao tác dữ liệu vào phần body của request. Nhập vào các giá trị sau và bật **intercept** lên.

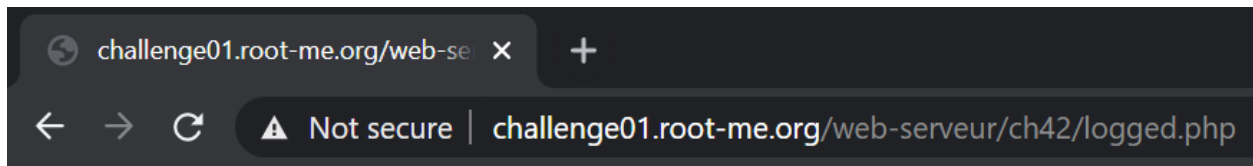
Login: 123

Password: 123

Gửi giá trị. Trong **request đã bị chặn**, thay đổi giá trị trường đầu thành **%ab%27 or 1=1 --**



Sau đó forward cái request đi:



Congratz! The validation password is: iMDaFlag1337!

OK. Bài khá đơn giản, nhưng mình giải thích nhiều vì đa số các wu khác giải vẫn tắt kinh khủng, giống như họ chôm thao tác từ nguồn nào và không hiểu vậy. Mình chèn vào việc giải thích nguồn gốc việc bypass challenge với việc xuất hiện **GBK** là ý tưởng để bypass nên hơi dài. Mong các bạn thông cảm.

Flag: **iMDaFlag1337!**