M10915027 石成峰 HW5

[Guess Time]

1. 在GET中加入參數ANSWER覆蓋掉原有的ANSWER，突破$guess == $ANSWER的檢查。
2. 利用md5及sha1對於陣列都會回傳NULL的特性突破md5($token1) == sha1($token2)的檢查後獲得flag。

[Shark]

1. 利用Array有indexOf及length的特性判斷出可以使用Array當作command的值。
2. 利用JS可以自動將Array轉成字串比對的特性突破command == "Baby shark, doo doo doo doo doo doo"後得到flag。

[Copy Generator]

1. 首先使用LFI並用php://filter將index.php的原始碼調出看，找到flag的第一段。
2. 發現調用template的方式是使用直接include檔案。
3. 找出存放session的路徑並在登入時將代碼<?php system($\_GET["cmd"]);?>當作username輸入。
4. 使用LFI調用session檔案便可以透過cmd參數進行sh的操作。
5. 在根目錄底下找到flag，用cat顯示獲得。

[Base32]

1. 透過原始碼可以知道能夠使用linux的Command Injection在指令中寫入其他指令。
2. linux允許使用$()當作變數進行輸入，因此先將base32 decode前先包一層encode，便可以在中間執行真正的指令cmd，整合後為string=$(base32 <<< $(cmd))。
3. 使用ls在根目錄底下找到flag，用cat顯示獲得。

[Meow]

1. 在id參數上輸入一個單引號後發現回傳500，可以得知有SQL Injection。
2. 使用union及information\_schema.tables找到所有的table中有一個叫做flag的table。
3. 之後用information\_schema.columns找出它裡面有2個column叫id及the\_flag。
4. 直接使用union select調出the\_flag中第一筆資料得到flag。

[(#°д°)]

1. 根據php可以對字串做bit operation的特性，使用~及非數字和字母的字元得出\_GET的變數名稱，可以獲得其他的GET參數。
2. 之後再透過GET的其他參數來輸入system這個字以及想要執行的cmd。
3. 使用ls在根目錄底下找到flag，用cat顯示獲得。

[VISUAL BASIC 2077]

1. 由於有res['username'] == username and res['password'] == password的條件所以可以推斷出，必須要能夠自由控制query的回傳值，所以必須使用union。
2. 由於要做到SQL Injection同時回傳的字串又必須要根query長的一樣因此可以推斷出必須要使用SQLite中能夠增加字串長度的function，符合條件的只有找到replace。
3. 透過調整replact函數的參數值以及使用quote函數解決整體單引號數量的問題，可以達到回傳的query值能與傳入的username及password完全匹配。
4. 發現類別Flag在轉成字串時會去判斷session中is\_admin的值，繞過判斷的方法為在username中加入直接調用Flag類別中的flag變數，因此username寫法為{flag.flag}。
5. 將SQL Injection的部分調整為password，並重新調整到字串匹配後得到username為{flag.flag}、password為' union select "{flag.flag}",replace(quote("\_")||"\_"||quote(")"),"\_",' union select "{flag.flag}",replace(quote("\_")||"\_"||quote(")"),"\_",')'。
6. 登入後即可在Hello,的後面原本是username的地方得到flag。