

NiteCTF 2021 writeups

r10921a03 張承遠

MISC

Hashes Hashes we all fall

Observation

這題題目敘述冗長，但大意是在說給我們一個hash值(使用sha-256)，要想辦法還原出hash前本來的值(密碼)。此外還有給個提示為密碼可能是Bee Movie這部電影的某個單字。

Exploitation

一開始想說使用Brute-force，若單字長度沒有太長的話可能有機會寫出來(共 $(26*2)^n$ 組合)，但Brute-force經嘗試後找不到。

後來有試著用常見英文單字去暴力解，但依然找不到相對應的hash值。最後是異想天開直接去下載Bee Movie的Transcript，並且做一些簡單處理後把Transcript裡每個單字都算hash值比對，結果Oinnabon這個單字比對成功，成功拿到Flag。

Let's be Artistic

Observation

1. 提供一個文字檔，裡面是一串密文

Exploitation

這題是隊友想到的，密文的字母對應到的是鍵盤位置，像是密文87yhnmkj 對應到的就是字母G。最後將所有密文在鍵盤上畫一遍就知道Flag了。

Prisoners

Observation

1. 題目為一個監獄裡有100個人，每個人各對應到一個編號，我們要依序猜出每個人對應到的編號是多少。
2. 在猜每個人的編號時最多只能猜50次，如果猜錯第50次程式會直接shutdown，猜對的話要猜下一個人。
3. 猜錯編號時會顯示猜錯的編號對應到哪個人。

Exploitation

根據Observation，這題可以使用greedy策略，去最大化能夠試的數量。

首先隨機猜第一個人的編號，並且在猜錯的過程中將其他人對應的編號存進hash map裡面。接著在猜後續的人的編號時，若發現此人的編號已經有存在hash map裡，一樣繼續猜到第49次，最後一次再提交正確的編號，這樣可以多存其他人的編號。

透過此Greedy策略，若前幾次成功猜到(第一次猜到機率:1/50)則後面猜到的機率會增加許多，成功拿到Flag。

Web

Welcome to nitectf

Observation

1. 題目給了一個網站，並且說flag在裡面

Exploitation

最後用滑鼠在網站下面的某句話點兩下後Flag就顯示出來了

BATCHEST

Observation

1. 題目只給了一個網站，要求你輸入一個動物名稱，輸入正確的動物名稱後會跟你說通過，反之則不通過。
2. 用簡單的payload ‘ OR 1=1 -- 可以順利登入。

3. 沒有給source code

Exploitation

由 Observation 可以猜測這是一個blind sql injection的題目，類似作業的log me in: final, 因為輸入後只會顯示結果是否正確。

由於我們沒有任何資訊，因此嘗試先leak出table name, column name等資訊，但是發現query table_schema相關關鍵字不會成功。後來猜想或許他的database不是用MySQL，試了 '`or sqlite_version() like '%' --`' 後發現成功通過，因此開始查sqlite找schema的語法。

最後使用此[網站](#)的payload，成功leak出table name(flag_tbl) 以及column name(flag_cln)，成功拿到藏在flag_cln下面的Flag。

JWT

Observation

- 一開始進到網站，裡面可以輸入東西並根據輸入產生 jwt token
- 產生 token 後會存在cookie，接著按下網站上的verify會跳出 not admin。

Exploitation

首先把 token 複製丟到 jwt.io 後可以解出以下內容。

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "typ": "JWT",  
  "alg": "HS256"  
}
```

PAYOUT: DATA

```
{  
  "user": "guest",  
  "admin_cap": "false",  
  "kid": "http://localhost:3000/secret.txt"  
}
```

“guest”就是剛剛在input輸入的東西，"admin_cap"感覺是跟前面的錯誤訊息有關，可能要把它改成 True 才能過，"kid"感覺是某個放密鑰的地方。

由於接下來要試著去更改訊息內容，因此必須要想辦法拿到 secret 重新簽章才能產出新的 jwt token 並通過審核。一開始想到 kid 有紀錄跟鑰匙有關的資訊，因此試了curl {website_domain}:3000/secret.txt 但是沒有反應。

後來去查詢後，發現 kid 參數是個optional data，可能會記錄存放密鑰的地方。因此想到一個可能的解法為：把 kid 的資訊換成自己架設的網站，網站存放自定義的 secret，並且在製作jwt token時也用此secret。這樣可以讓網站在審核的時候使用我網站的secret去做審核，而我傳入的token也確實用此secret去做簽章，因此會順利通過審核且可以更改為任意內容。

最後使用ngrok架設一個網站，內容為自己設定的密碼1234，接著用 jwt.io 更改payload，由於現在的密碼是可控的，因此我們可以偽造任何內容。更改payload內容為[**kid: {my_domain}** , **user: admin**]，並且用secret=1234加密產出JWT token。最後用此token覆蓋掉網站上存在cookie的token後按下verify，verify會根據kid的資訊拿到我們自己設的密碼，最後成功偽造token拿到Flag。

Crypto

VariableZZ

Observation

1. 這題給了一個array的密文，密文是把Flag的每個字元經過一個多項式運算後產生
2. 加密的多項式不會很複雜，是一個三次多項式

Exploitation

這題要想辦法解出加密時使用的多項式的常數項(a,b,c,d)。由於我們可以知道Flag的格式為`nite{...}`，這樣就代表我們有六個等式可以解這四個常數，非常足夠。

最後解出常數項後再去解密其他字元，就可以順利拿到Flag

Osint

Mailman 1

Observation

1. 這題給了一個人的id, 要我們肉搜這個人的email, email就是Flag
2. 提示有說他有在discord裡面

Exploitation

搜尋Discord id並查看他的profile page，可以發現他的[github](#)。

Github裡有兩個專案，一個包含一串密文跟一張狗的照片，另一個是類似自我介紹的專案，並包含一個上了鎖的zip檔案。

知道Github後搜尋關鍵字github find email, 參考此[網站](#)，利用使用者提交commit時的email去找。然而透過此方法找到的email為：

88324251+replierNite@users.noreply.github.com, 很明顯不是正確的email, 猜測email可能藏在上鎖的zip檔裡面。

接著仔細觀察該專案有提到跟commit hash相關的資訊，因此有試了最新一次的commit hash當作zip的密碼，但是錯誤的，賽後發現密碼是前一個的commit hash，因為該

commit 訊息為

This will be of some help
 replierNite committed on 11 Oct 2021