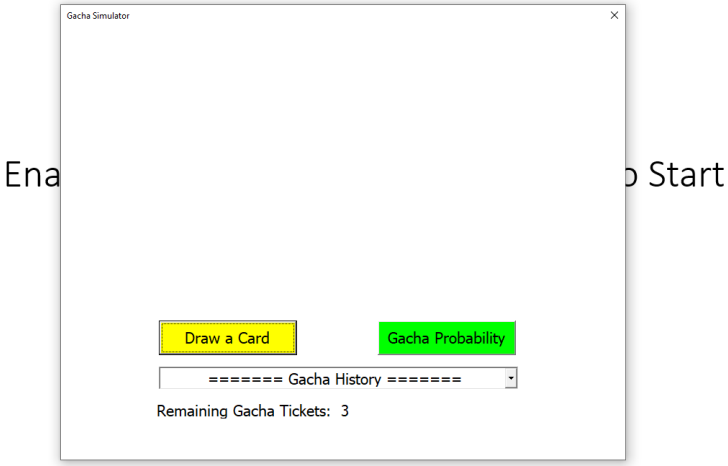
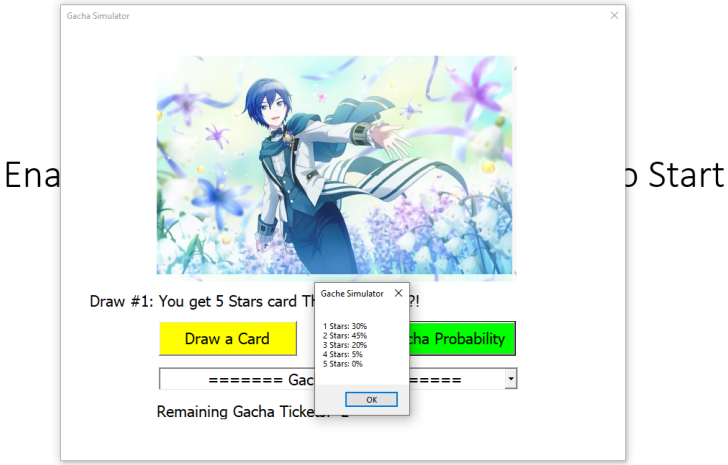
HKcert - Gatcha Machine

We are given only a .pptm file
This is a powerpoint file that has macros enabled.



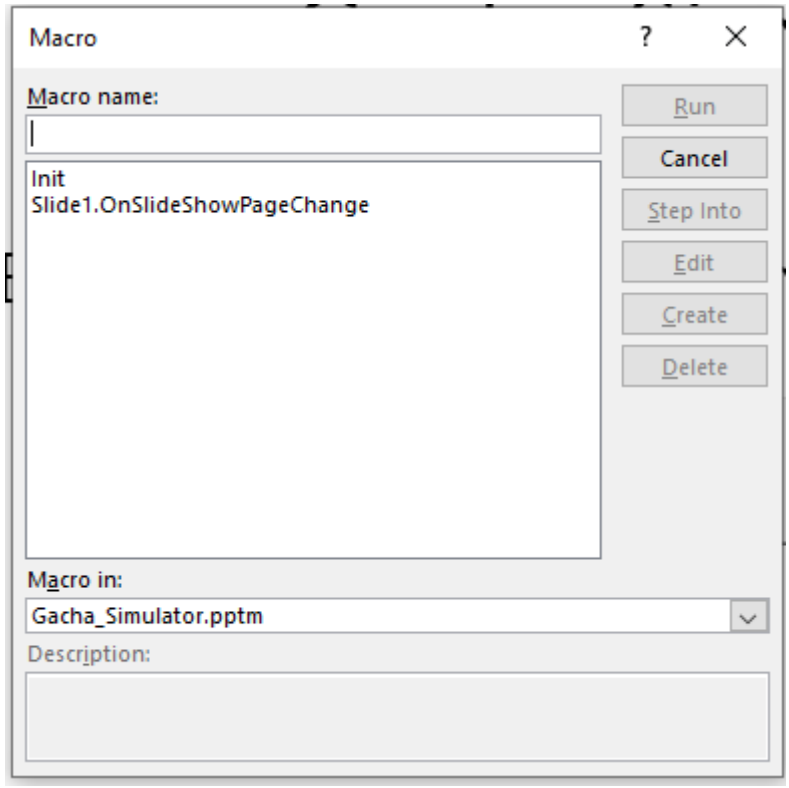
When running the PowerPoint we are given a Card game where we can draw new cards, view our previous cards, and also see the probability of each card.



When drawing a card you get the message

> Draw #-number-: You get -star_number- Stars card -name-

When viewing the probability of each card rarity, we can see that 5 star cards have a 0% chance of being drawn.



I then went on the try and edit the code behind this functionality; however, I was unable to edit the files and was presented with an enter password dialog box.

On the challenge description we are given a hint.

轉蛋模擬器 / Gacha Simulator (Reverse)

Partial Guide

This challenge is best viewed with .NET Framework
This is a pptm file, which is Macro Enabled. Make sure you check out the VBA code under the Developer tab.
https://support.microsoft.com/en-au/office/show-the-developer-tab-e1192344-5e56-4d45-931b-e5fd9bea2d45
Can't see source code? Try this
https://stackoverflow.com/questions/1026483/is-there-a-way-to-crack-the-password-on-an-excel-vba-project
Then there are many ways to solve, like changing the probability, rearrange the encrypted data, decrypt the encrypted data directly, etc. Good luck!

The first link is a guide on how to enable the developer tab in powerpoint.

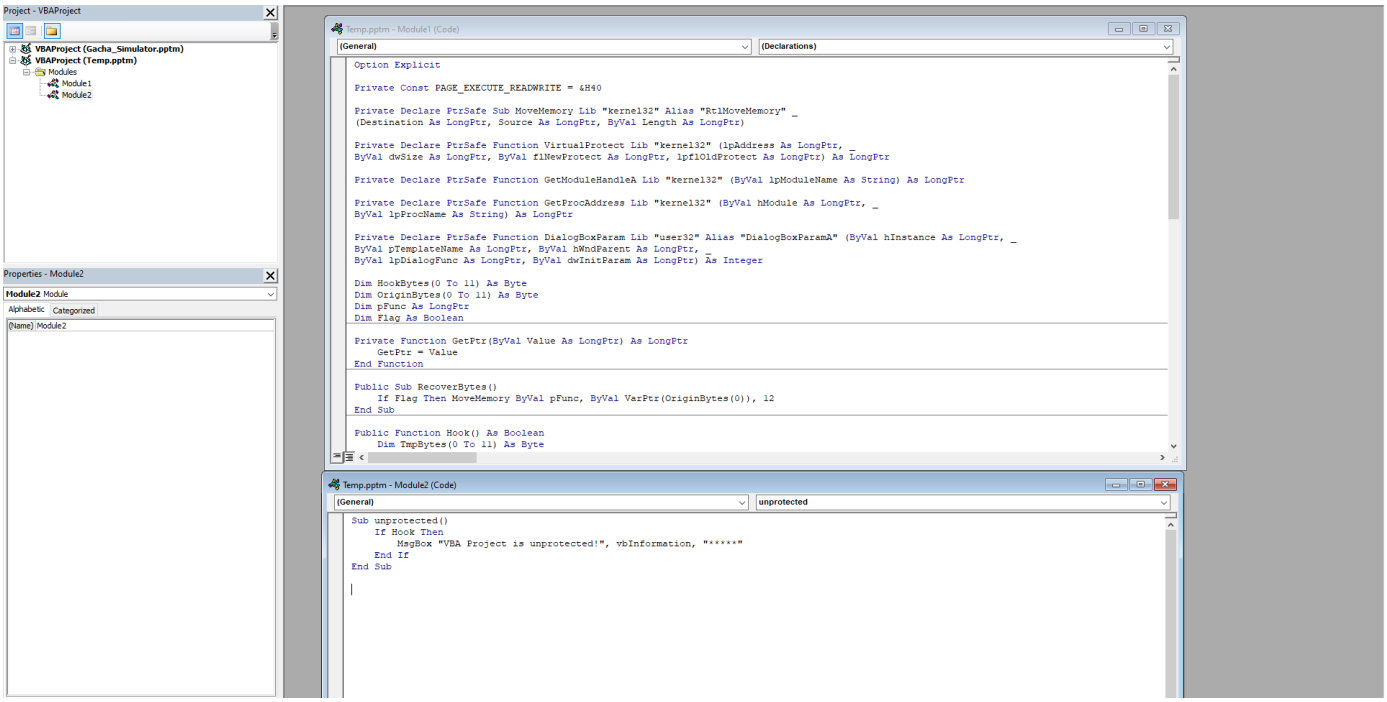
The second link is a stack overflow question on how to bypass Macro Password Protection in Microsoft Office.

- This works by exploiting how the check password function works in these applications, if the password is correct it will return 1 and if not returns 0.
- So, the given script on Stack Overflow will overwrite the memory location of this variable with 1 so that the application thinks the correct password was given.
    - The VBE (visual basic editor) will check this location, so if we run the Stack Overflow script it will set the location to 1 and we will be able to bypass the password check.

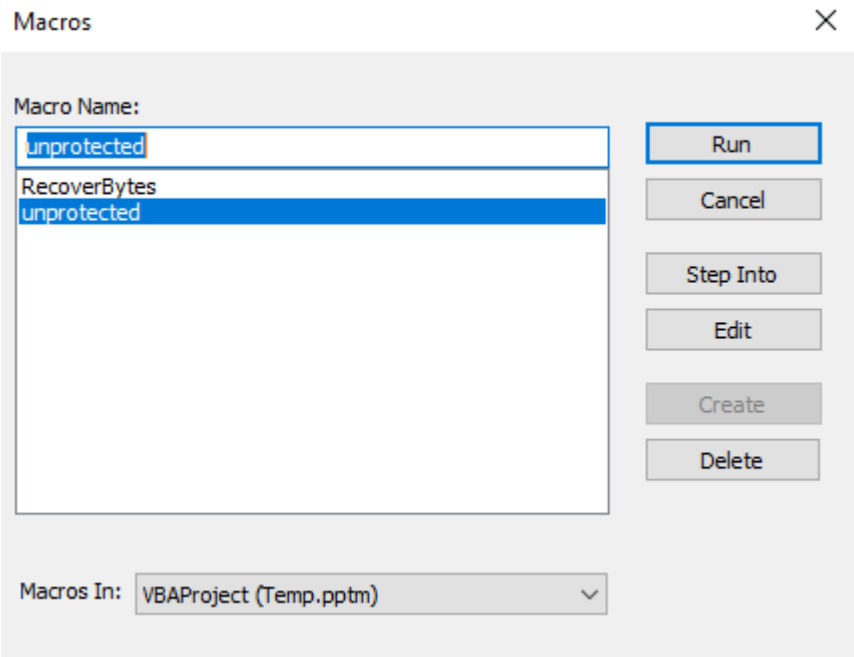
To use the script, we need to make a new PowerPoint file with macros enabled, and then create a new Module to run the given script.
We need to have both the new PowerPoint and the given PowerPoint open at the same time.
When I first ran this script, it said it was incompatible with 64bit systems.
I added the phrase 'SafePtr' to each declaration, as this was the solution given in the error log; however, after executing the script it would just crash my PowerPoint files.

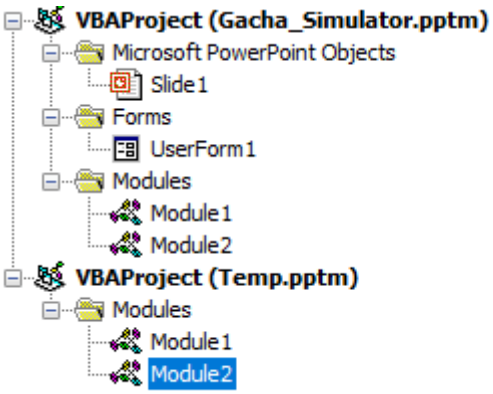
Going back to Stack Overflow I saw that there was a reponse to the original answer that was adapted to work on 64bit systems.
The fix changed push/ret instructions to mov/jmp, this is because push/ret are limited to 32bit addresses.

Now, the script was working.



Following the instructions on Stack Overflow, I created a new PowerPoint file with Macros enabled and then created two modules with the corresponding 64bit code to overwrite the memory address of the password authentication variable.





After running the Module 2 function 'unprotected' I could now view and edit the macros of the original PowerPoint given.

Gacha_Simulator.pptm	11/11/2023 1:24 PM	Microsoft PowerP...	66 KB
Gacha_Simulator_Backup.pptm		Microsoft PowerP...	64 KB
Module1.bas	11/11/2023 1:01 PM	BAS File	2 KB
Module2.bas	11/11/2023 1:01 PM	BAS File	2 KB
Slide1.cls	11/11/2023 1:01 PM	CLS File	1 KB
Temp.pptm	11/11/2023 12:54 PM	Microsoft PowerP...	38 KB
UserForm1.frm	11/11/2023 1:01 PM	FRM File	3 KB
UserForm1.frx	11/11/2023 1:01 PM	FRX File	6 KB

Before editing any of the files, I exported all of them to my local machine so that I would have a backup in case I lost access to the macros.

Gacha_Simulator.pptm - Slide1 (Code)

(General) | OnSlideShowPageChange

```
Sub OnSlideShowPageChange()
    Module2.Init
End Sub

Private Sub CommandButton1_Click()
    Module2.Init
End Sub
```

Initially I saw that on slide one, the Module2.Init function gets called when viewing the slide.

Gacha_Simulator.pptm - Module1 (Code)

(General)

```
Function Min(a, b)
    Min = a
    If b < a Then Min = b
End Function

Function B64Encode(bytes)
    On Error Resume Next
    Set b64Enc = CreateObject(
    Set utf8 = CreateObject("S
    blockSize = b64Enc.InputBl
    For Offset = 0 To LenB(byt
        Length = Min(blockSize
        b64Block = b64Enc.Tran
        result = result & utf8
    Next
    B64Encode = result
End Function

Function B64Decode(b64Str)
    On Error Resume Next
    Set b64Dec = CreateObject(
    Set utf8 = CreateObject("S
    bytes = utf8.GetBytes_4(b6
    B64Decode = b64Dec.Transfo
End Function

Function Encrypt(plaintext, ae
    On Error Resume Next
    Set aes = CreateObject("Sy
    Set utf8 = CreateObject("S
    aes.IV = utf8.GetBytes_4("
    aesKeyBytes = B64Decode(ae
    Set aesEnc = aes.CreateEnc
    plainBytes = utf8.GetBytes
    cipherBytes = aesEnc.Trans
```

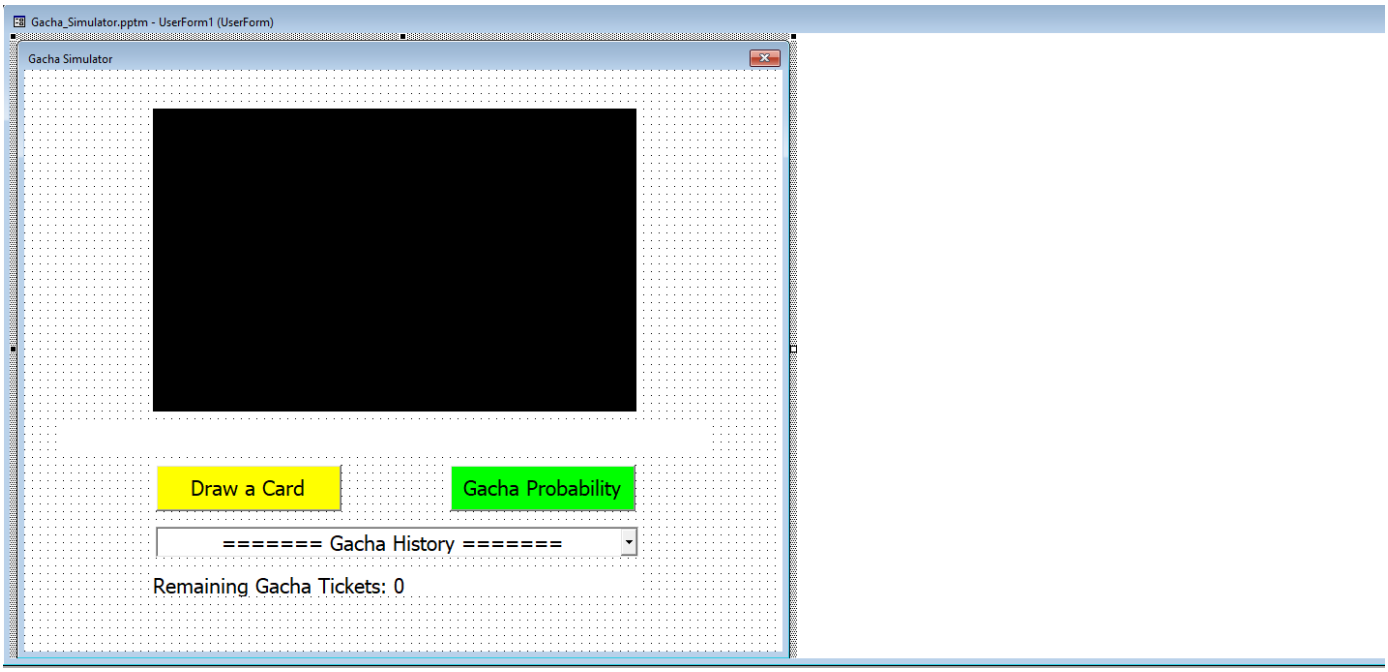
Gacha_Simulator.pptm - Module2 (Code)

(General) | (Declarations)

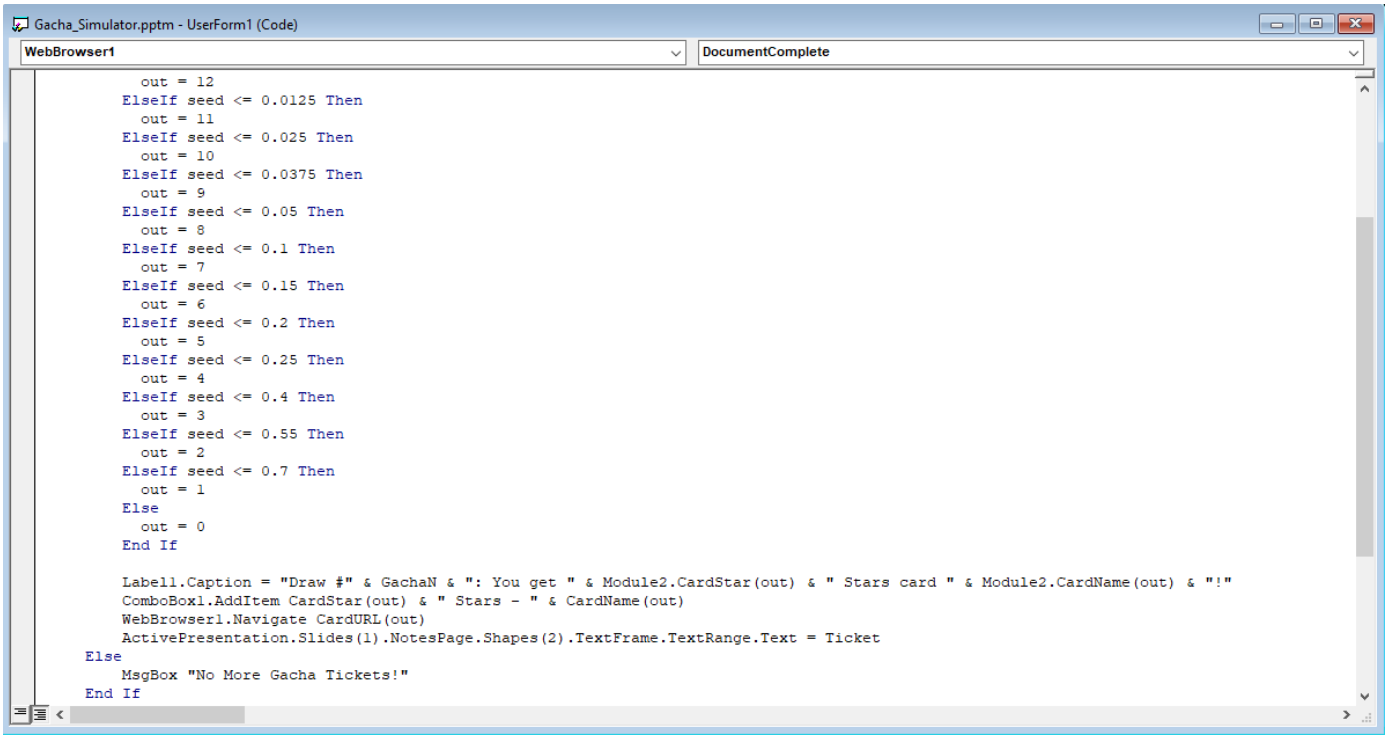
```
Global Key As String
Global Ticket As Integer
Global CardList() As String
Global CardStar() As String
Global CardName() As String
Global CardURL() As String
Global GachaN As Integer

Sub Init()
    On Error Resume Next
    GachaN = 0
    If Not IsNumeric(Application.ActivePresentation.Slides(1).NotesPage.Shapes(2).TextFrame.TextRange.Text) Then
        Ticket = 10
    Else
        Ticket = Int(Application.ActivePresentation.Slides(1).NotesPage.Shapes(2).TextFrame.TextRange.Text)
    End If

    UserForm1.Label2.Caption = "Remaining Gacha Tickets: " & Str(Ticket)

    Dim arrCards() As String
    arrCards = Split(Application.ActivePresentation.SlideMaster.Shapes(1).TextFrame.TextRange.Text, vbCr)
    Dim i As Long
    For i = LBound(arrCards) To UBound(arrCards)
        Row = Split(arrCards(i), "<>")
        ReDim Preserve CardStar(i)
        CardStar(i) = Row(1)
        ReDim Preserve CardName(i)
        CardName(i) = Row(2)
        ReDim Preserve CardURL(i)
        CardURL(i) = Module1.Decrypt(Row(3), Row(2))
    Next
    UserForm1.Show
End Sub
```

The Module1 and Module2 code appeared to just have encryption and decryption code, but no functionality for the Gatcha system was in these files.

In the VBE I saw that there was also a UserForm in the given powerpoint.



When opening the form initially, I did not see any code but just the GUI presented to the user when the script runs.



However, when I right clicked the User Form there was an option to view the code.
In the code I saw that a random seed was being generated to choose the card given to the user.

- In this, there was a card given only when the seed was <= 0. (out 12)
  Another interesting finding was that the picture being displayed was not a file, but rather a web browser rendering a given URL.

```
Gacha_Simulator.pptm - UserForm1 (Code)
CommandButton1                                   Click
        ElseIf seed <= 0.15 Then
          out = 6
        ElseIf seed <= 0.2 Then
          out = 5
        ElseIf seed <= 0.25 Then
          out = 4
        ElseIf seed <= 0.4 Then
          out = 3
        ElseIf seed <= 0.55 Then
          out = 2
        ElseIf seed <= 0.7 Then
          out = 1
        Else
          out = 0
        End If
        out = 12

        Label1.Caption = "Draw #" & GachaN & ": You get " & Module2.CardStar(out) & " Stars card " & Module2.CardName(out) & "!"
        ComboBox1.AddItem CardStar(out) & " Stars - " & CardName(out)
        WebBrowser1.Navigate CardURL(out)
        MsgBox CardURL(out)
        ActivePresentation.Slides(1).NotesPage.Shapes(2).TextFrame.TextRange.Text = Ticket
    Else
        MsgBox "No More Gacha Tickets!"
    End If
End Sub

Private Sub CommandButton2_Click()
    MsgBox "1 Stars: 30%" & vbCr & "2 Stars: 45%" & vbCr & "3 Stars: 20%" & vbCr & "4 Stars: 5%" & vbCr & "5 Stars: 0%", vbOKOnly, "Gache Sim
End Sub

Private Sub WebBrowser1_DocumentComplete(ByVal pDisp As Object, URL As Variant)
    WebBrowser1.Document.Body.Scroll = "no"
End Sub
```

I then edited the script, so that the card (out=12) was always generated by placing a new statement after the if checks that sets the variable out to equal 12 in every case.
Then I had the presentation output the URL that it was rendering card 12 from.

As we can see the flag is output and the card name is 'The S3cr3t Flag?!'
I had the presentation output the URL because the web browser scrolls the flag horizontally across the screen and it is hard to decipher the individual characters because of the font color as well as the similarity between I l and | (capital i, lowercase L, and the pipe symbol).

So, with the URL I navigated to the page from my own local browser.

There I got the same output.
However, now I could view the source code to get the flag in plaintext.

```
<body style="font-family:Arial;color:blue;background:url(https://i.imgur.com/23QDGvV.png)"><marquee><p> <h1>Congratulation!<h2>You drew a 5 stars card~<h3>Here is the flag: <h4>hkcert23{FIl1
```

I viewed the HTML source code by pressing CTRL U on the page in the local browser.
In the HTML code I found the flag in plaintext.
From here, I simply copied the text and pasted it into the challenge to submit the flag and received 250 points for the team.