CTF uppgift: The Cryptic Website

Gustav Löfqvist V210S

Uppgiften utmanar deltagarna att ‘hacka’/modifiera en webbplats för att upptäcka ett dolt python-skript. Deltagarna måste använda ledtrådar på webbsidan och sina kunskaper inom webbhantering och Python-programmering för att lösa de olika stegen och avslöja den gömda flaggan. Genom att inspektera källkoden, identifiera en nyckel och använda den för att dekryptera en gömd Python-kod, kommer deltagarna närmare lösningen på uppgiften. Uppgiften är utformad för att vara en lagom utmaning för nybörjare inom säkerhet och CTF.

Lösningsförslag

Uppgiften går självklart att lösa på flera olika sätt men här är mitt förslag:

1. Deltagarna besöker webbsidan och tar hjälp utav ledtrådarna* för att ta sig vidare till nästa steg.

2. Med hjälp av ledtrådarna* listar deltagarna ut att de behöver besöka hemsidans källkod och de öppnar ‘inspect element’.

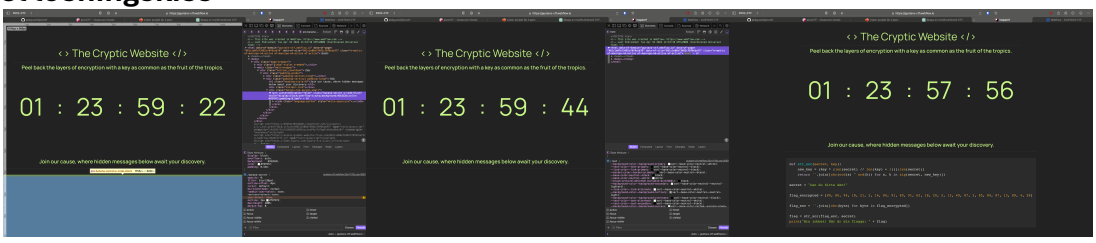
3. Efter att ha letat runt i inspectelement och hittat meddelanden I bland annat <Head> och <Body> så borde deltagarna hittat en python-script alt ‘secret-banana’ html-klass.

4. Studerar vi den här klassen kan vi modifera dess css och därmed extrahera ett python script. Python scriptet kommer gå att köra utan bekymmer men för att kunna hämta ut flaggan behöver deltagaren identifiera en key.

5. Scriptets key är ‘banana’ och det hintas på överallt i hemisdan. Bland annat i hemsidans html och texten som står tydligt ”Peel back the layers of encryption with a key as common as the fruit of tropics”. Efter att ha matat in banana så får deltagaren ut rätt flagga.

*När jag skriver ledtrådarna syftar jag på hints och liknande som finns på hemsidan. Inte ledtrådarna som befinner sig på githuben. Githubens ledtrådar är endast tänkt för de som fastnar och vill få en liten knuff i rätt riktning.

Enkel lösningskiss



Här kan vi se hur man exempelvis skulle kunna manipulera hemsidans källkod för att få fram den gömda python koden. Om vi kör pythonprogramet får vi outputen nedan:

```
gurra@VW21028 ~/Documents
python3 CrypticWebsite.py
Bra jobbat! Här är din flagga: ;102qw082$/_2pgc<b;y3xylu
```

Vi behöver justera python kodens secret till ‘banana’.

```python
def str_xor(secret, key):
    new_key = (key * (len(secret) // len(key) + 1))[:len(secret)]
    return ''.join([chr(ord(s) ^ ord(k)) for s, k in zip(secret, new_key)])

secret = "banana"

flag_encrypted = [80, 80, 94, 18, 21, 2, 16, 80, 91, 80, 91, 62, 18, 20, 2, 13, 49, 87, 3, 85, 89, 87, 13,

flag_enc = ''.join([chr(byte) for byte in flag_encrypted])

flag = str_xor(flag_enc, secret)
print("Bra jobbat! Här är din flagga: " + flag)
```

Genom att ta hjälp från hemisdans hints får vi fram key till banana och därmed printar pythonkoden ut rätt nyckel.

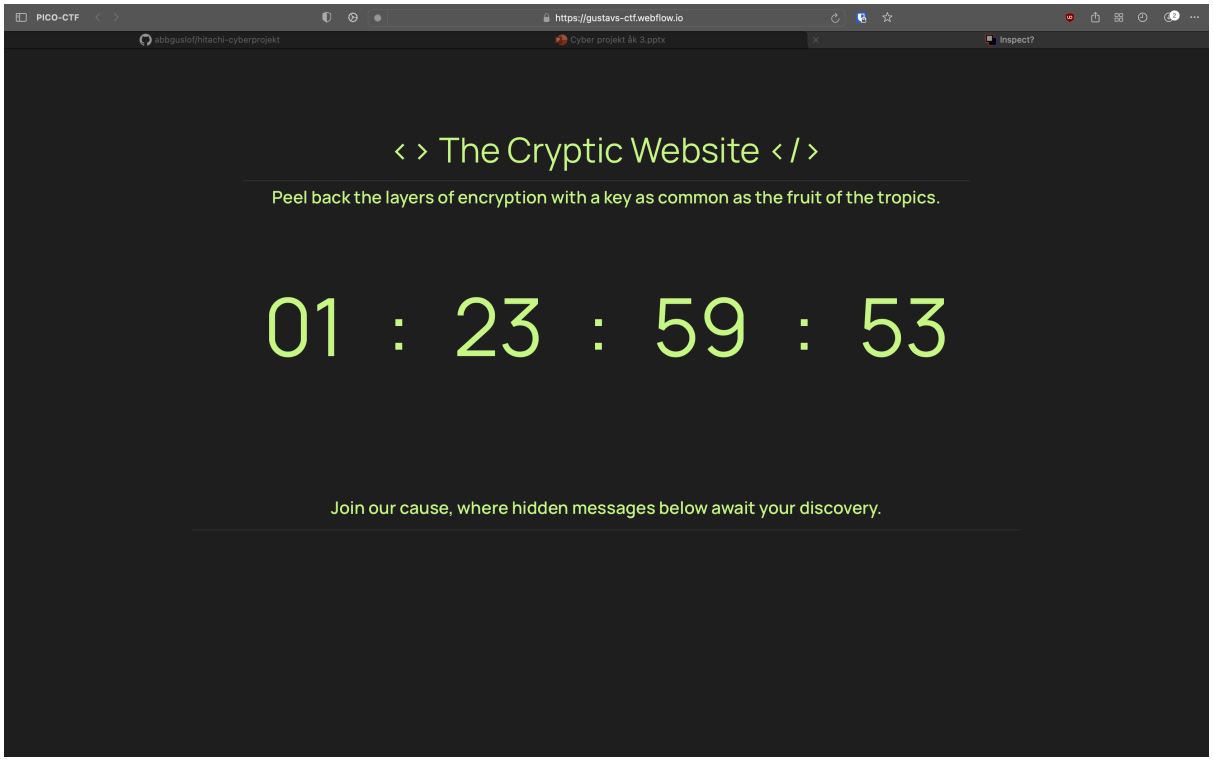
```
gurra@VW21028 ~/Documents
python3 CrypticWebsite.py
Bra jobbat! Här är din flagga: 210s{cr1515_pull_6a476c8f}
```

För en mer detaljerad lösningsskiss se githubens Solution-TheCrypticWebsite.pdf

Utförande

Uppgiften visade sig inte vara något krånglig att slutföra. Och den skiljer sig inte speciellt mycket från planeringen. Eftersom jag redan tidigt visste hur jag ville att mina uppgift skulle se ut kunde jag bara lätt både följ min planering och mina idéer.

Jag började med att skapa webbplatsen för uppgiften med hjälp av Webflow. Webflow är ett verktyg som tillåter mig att snabbt skapa professionella webbplatser på ett visuellt sätt. Genom att använda ett tredjepartsverktyg som är mindre kodbaserat och mer visuellt förlorar jag visserligen en del av flexibiliteten, men det accelererar skapnings processen. Eftersom webbplatsen var mycket enkel valde jag att använda ett sådant verktyg trots att jag förlorade en del av kontrollen. Jag valde specifikt Webflow för att de fortfarande erbjuder en stor flexibilitet med bland annat egen kodinmatning. Hemsidan utformades för att försöka ge deltagarna så mycket information som möjligt.



På toppen av hemsidan finns uppgiftens titel med <title> taggar runt sig. Detta är typiskt för HTML-kod, vilket antyder att det kan vara värt att inspektera webbsidans källkod. Titeln på hemsidan, 'inspect?', hintar också på att använda 'inspect element'-funktionen i webbläsaren för att undersöka webbsidans innehåll närmare.

Klockan på webbsidan har egentligen ingen relevans för uppgiften utan är avsiktligt inkluderad för att förvirra. Dock finns det en text under klockan som leder deltagarna till att söka 'below', det vill säga under texten. Detta är där Python-skriptet är gömt. Om användarna sedan hovrar över texten så blir den lite genomskinlig vilket skall försöka hinta mot att pythonskriptet är osynligt(genomskinligt).

När hemisdan var klar kunde jag lägga till egen kod och anpassa den för att passa uppgiften bättre. Jag lade till kommentarer i HTML-koden och modifierade CSS. CSS modifikationen består av ‘Opacity: 0’, ‘User-Select: none’ och ett blur filter. Detta är för att göra skriptet osynligt och se till att användarna inte kan markera det utan att först modifiera css.

Python-skriptet är en mycket enkel XOR-chifferimplementation. XOR-cipher (Exklusiv Eller-kryptering) är en symmetrisk krypteringsmetod som bygger på den logiska/binary operatorn XOR. XOR tillämpas på binär data. Jag valde att använda ett sådant eftersom det är mycket svårt att knäcka utan rätt nyckel. XOR-cipher har samma nyckel för både kryptering och dekryptering. Genom att bara reverse-engineera skriptet kommer man alltså inte långt. Deltagarna måste därmed hitta rätt nyckel till XOR-chiffern för att få ut rätt flagga.

Svårighetsgrad: 0.85 (ca 85% som borde kunna lösa uppgiften)

Uppgiften kräver grundläggande kunskaper inom python och webbhanterring. Det är något som alla borde ha bemästrat. Det kårngliga skulle kunna vara kombinationen av dessa kunskaper. Uppgiften är lagom för nybörjare inom säkerhet och CTF.