

Beståndsdelar

- 1 Att flera hemsidor använder sig av olika typer av optimering kommer väl knappast som en överraskning för de flesta.
- 2 Att analysera information om detta kan dock ibland vara intressant.
- 3
- 4 Mellan vilka datum var google analytics tag UA-3738754 aktiv för sok.aftonbladet.se?

Det finns en rad olika hemsidor som kan nyttjas för att utforska hur en webbsida är uppbyggd.

För denna uppgift används lämpligen <https://builtwith.com/>

Log In · Signup for Free

built with

Tools ▾

Features ▾

Plans

Customers

Resources ▾

Website, Tech, Keyword

Lookup

Home / sok.aftonbladet.se Technology Profile

SOK.AFTONBLADET.SE

Technology Profile Detailed Technology Profile Meta Profile Relationship Redirect Recommendations Company

Mobile

View Global Trends

Viewport Meta

Viewport Meta Usage Statistics · Download List of All Websites using Viewport Meta

This page uses the viewport meta tag which means the content may be optimized for mobile content.

iPhone / Mobile Compatible

iPhone / Mobile Compatible Usage Statistics · Download List of All Websites using iPhone / Mobile Compatible

Profile Details

Change Layout

Link to this page. This profile will be updated 6th June 2023.

BuiltWith Top Site Rank

aftonbladet.se is ranked 77,770th in our top sites list. View BuiltWith Top Site Rank.

Log In · Signup for Free

built with

Tools ▾

Features ▾

Plans

Customers

Resources ▾

Website, Tech, Keyword

Lookup

Home / aftonbladet.se Technology Profile / aftonbladet.se Historical Website Relationship Profile

AFTONBLADET.SE

Technology Profile Detailed Technology Profile Meta Profile Relationship Redirect Recommendations Company

AFTONBLADET.SE Tag History

UA-3738754	Oct 2008	Aug 2018
HJ-9766	May 2016	Aug 2018
HJ-185963	Jul 2018	Jul 2018
UA-73834518	May 2016	Jun 2018

AFTONBLADET.SE Connected Websites

[aftonbladet.se](#)

[tv.aftonbladet.se](#)

[sok.aftonbladet.se](#)

[www.aftonbladet.se](#)

[boom.aftonbladet.se](#)

[film.aftonbladet.se](#)

Log In · Signup for Free

built with

Tools ▾

Features ▾

Plans

Customers

Resources ▾

Website, Tech, Keyword

Lookup

Home / UA-3738754 Google Analytics Tag Usage History

UA-3738754

Google Analytics Tag Usage and History

UA-3738754 Connected Domains		
Domain	First Detected	Last Detected
sok.aftonbladet.se	March-13	October-17
aftonbladet.se	October-08	August-18
mobil.aftonbladet.se	March-13	March-13
aftonbladet-cdn.se	October-13	August-17

BlackAndYellowMaybe?

Uppgiften består av en PDF med manuset till Bee movie, där en del rader är överstrukna i olika färger. Detta är en modifierad variant av en uppgift som förekom på tidigare miniCTF.

I detta fall gömmer sig flaggan inbäddad i hexdatan till filen, och hittas enklast genom att använda ett kommando i stil med `xxd BlackAndYellowMaybe.pdf | grep -i -A 2 "hex"`

där `xxd` genererar en hexdump på filen och outputn skickas till `grep` som söker efter strängen "hex", `cAsEiNsEnSiTiVe` och inkluderar två rader efter en träff.

Detta ger flaggan **HEX{i_am_not_a_bee}**

```
pontus@Slapptop:~/Downloads/miniCTF2$ xxd BlackAndYellowMaybe.pdf | grep -i -A 2 "hex"
000139d0: 4807 ec34 002a 4845 587b 695f 616d 5f6e  H..4.*HEX{i_am_n
000139e0: 6f74 5f61 5f62 6565 7d69 0465 b4e7 ae73  ot_a_bee}i.e...s
000139f0: f8ee 7a36 369a 1f89 b816 7170 0513 29e7  ..z66.....qp..).
pontus@Slapptop:~/Downloads/miniCTF2$
```

Dagens ...någon

Dagens ...någon

Det finns få saker som förvirrar oss så mycket som högtider, men tydligen finns det i alla fall ett simpelt system för att räkna ut när påsken ska infalla:



Enligt denna enkla formel, hur många dagar skulle det vara mellan nyårsafton och skärtorsdagen?

Den här uppgiften skulle egentligen specificerat att det var år 2022 som åsyftades, men en felaktig skärmbild laddades upp pga slarvig korrläsning.

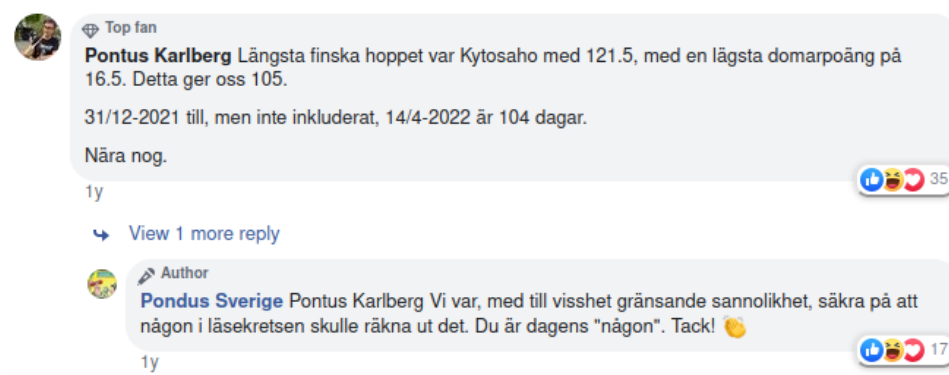
Jag ber om ursäkt till alla som lagt mycket tid och fått fram felaktigt svar!

Med information om rätt årtal kan korrekt statistik googlas fram, exempelvis

<https://medias1.fis-ski.com/pdf/2022/JP/3854/2022JP3854RL1.pdf>

Varefter en beräkning kan göras.

Alternativ lösning är att googla på uppgiftens namn och "Pondus", hitta strippen bland bilder och följa länk till Facebook, där följande beräkning gömmer sig bland kommentarerna:



En svunnen tid

Förfluten tid

Den här bilden togs 2022 på en plats i Sverige.

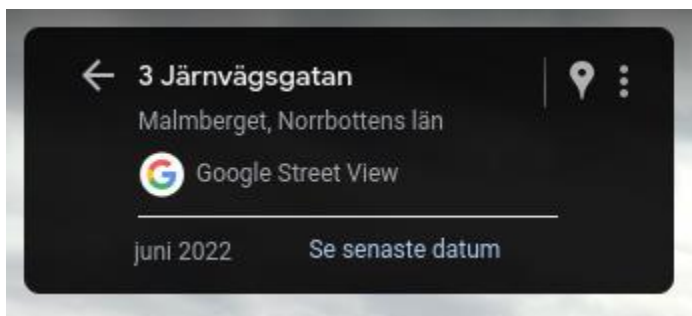


2011 stod det tvärsöver gatan två bilar, vilka färger hade dessa – och vilket var numret till Dannes?

Steg 1 är att identifiera var byggnaden finns.

Göres exempelvis genom att bildgoogla på “Nordan byggnad”, vilket visar att byggnaden finnes i Malmberget.

Steg 2 är att identifiera från vilken vinkel bilden är tagen för att säkerställa att en står på rätt gata, då Google Streetview i vissa fall har bilder från olika årtal för olika gator.



Genom att klicka på “Se senaste datum” väljes juli 2011 från möjliga bilder.

Numret till Dannes Gatukök syns i ett skyltfönster, och om en använder Streetview för att se bakom där bilden togs står där två bilar.



Flugit sin kos?

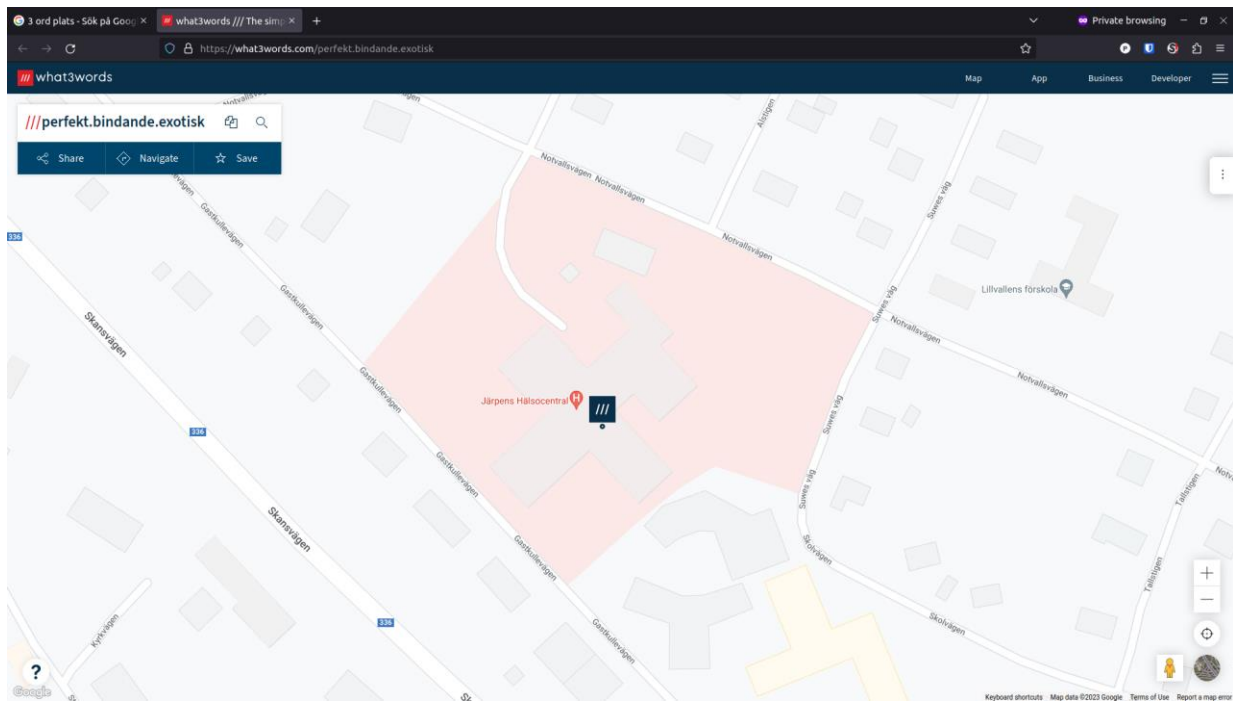
Flugit sin kos?

Vi har försökt komma i kontakt med en person, men varje gång vi frågar var hen är svarar hen endast med "perfekt bindande exotisk"

Var kan hen befinna sig?

Genom att söka efter "3 ord plats", "3 ord adress" eller liknande kan en få information om <https://what3words.com/> vilket är ett alternativt sätt att ange positionering, där världen delats in i kvadrater och varje kvadrat tilldelats tre ord.

Genom att på hemsidan och ange orden "perfekt bindande exotisk" kan exakt plats finnas, vilket i detta fall var **Järpens vårdcentral**.



MathsIsFun

På grund av bristfällig kvalitetskontroll verkar programmet ej ha fungerat på allas maskiner, något som absolut tas i beaktning till nästa gång.

Ber om ursäkt för detta!

Efter att ZIP-filen packats upp körs programmet, vilket slumpar ett antal mattefrågor och fortsätter skapa frågor till dess att användaren svarat rätt på alla - alternativt svarar fel på en fråga.

Medan programmet är igång skapas även en fil vid namn secret.txt i parent directory till där scriptet körs ifrån, vilken sedan tas bort när programmet stängs ned (om det ej avbryts vid körning), innehållandes flaggan **HEX{oH_You_Sne4ky_7h1nG}**

```
(swef@swef)-[~/CTF/mathsIsFun]
$ unzip mathsIsFunDist.zip
Archive:  mathsIsFunDist.zip
  creating: dist/
  inflating: dist/mathsIsFun.py
  creating: dist/pyarmor_runtime_000000/
  inflating: dist/pyarmor_runtime_000000/pyarmor_runtime.so
  inflating: dist/pyarmor_runtime_000000/__init__.py

(swef@swef)-[~/CTF/mathsIsFun]
$ cd dist

(swef@swef)-[~/CTF/mathsIsFun/dist]
$ python3 mathsIsFun.py
What is 9 - 9? 0
What is 6 - 6? 0
What is 9 + 7? 16
What is 9 - 7? ^Z
zsh: suspended  python3 mathsIsFun.py

(swef@swef)-[~/CTF/mathsIsFun/dist]
$ cat ../secret.txt
HEX{oH_You_Sne4ky_7h1nG}
```


RandomFiles

När ZIP-filen packats upp finns det två mappar, med 40 textfiler med gibberish i vardera, ytterligare en låst ZIP-fil samt ett textdokument innehållandes nonsens-text.

Genom att rekursivt kontrollera innehållet i filerna i mapparna, och ta filnamnen (utan ändelse) från alla filer där något skiljer sig och kombinera detta till en sträng får en fram lösenordet till locked.zip.

locked.zip innehåller endast en fil, flag.txt - som i sin tur innehåller flaggan

HEX{sH4rP_eYe5_0r_go0D_t00IS?}

```
pontus@Slapptop:~/Downloads/miniCTF2/randomFiles$ ls
folderOne folderTwo locked.zip theRealPassword.txt
pontus@Slapptop:~/Downloads/miniCTF2/randomFiles$ diff -qr folderOne/ folderTwo/
Files folderOne/bgluqvqogt.txt and folderTwo/bgluqvqogt.txt differ
Files folderOne/ckmhksbkog.txt and folderTwo/ckmhksbkog.txt differ
Files folderOne/dgppczxrkg.txt and folderTwo/dgppczxrkg.txt differ
Files folderOne/dkgugjicxx.txt and folderTwo/dkgugjicxx.txt differ
Files folderOne/kkysublllyf.txt and folderTwo/kkysublllyf.txt differ
Files folderOne/kymgsooivq.txt and folderTwo/kymgsooivq.txt differ
Files folderOne/ocxoduwnef.txt and folderTwo/ocxoduwnef.txt differ
Files folderOne/qanexysesj.txt and folderTwo/qanexysesj.txt differ
pontus@Slapptop:~/Downloads/miniCTF2/randomFiles$ unzip locked.zip
Archive:  locked.zip
[locked.zip] flag.txt password:
  extracting: flag.txt
pontus@Slapptop:~/Downloads/miniCTF2/randomFiles$ cat flag.txt
HEX{sH4rP_eYe5_0r_go0D_t00IS?}
```

Secret Images?



HexMastah
@elonmusk

...

There's apparently some cool dude called "Kanelbuss" who shares images with hyper cool secrets on social media!

Can you find them?

6:31 PM · Apr 29, 2023 · Twitter for CoolKids

12.3K Retweets 494.8K Quote Tweets 132.9K Likes



Genom att använda ett verktyg som går genom användare på olika sociala media, såsom exempelvis Sherlock, kan användaren "Kanelbuss" hittas på sociala media.

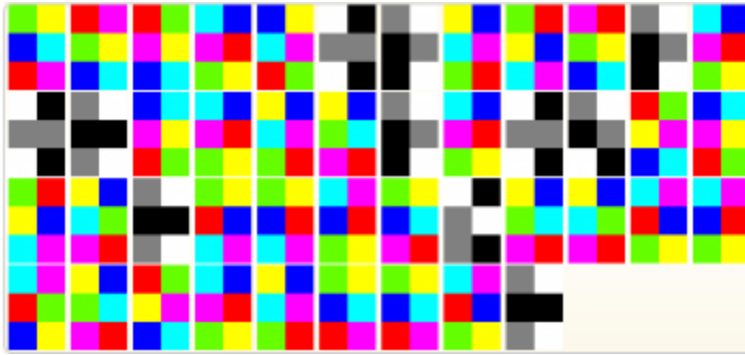
```
pontus@Slapptop:~/Downloads/miniCTF2/sherlock$ python3 sherlock Kanelbuss > output.txt
pontus@Slapptop:~/Downloads/miniCTF2/sherlock$ cat output.txt
[*] Checking username Kanelbuss on:

[+] Enjin: https://www.enjin.com/profile/Kanelbuss
[+] G2G: https://www.g2g.com/Kanelbuss
[+] Imgur: https://imgur.com/user/Kanelbuss
[+] Quizlet: https://quizlet.com/Kanelbuss
[+] Reddit: https://www.reddit.com/user/Kanelbuss
[+] Twitch: https://www.twitch.tv/Kanelbuss
[+] Virgool: https://virgool.io/@Kanelbuss
[+] ebio.gg: https://ebio.gg/Kanelbuss

[*] Search completed with 8 results
pontus@Slapptop:~/Downloads/miniCTF2/sherlock$
```

Går en genom dessa kan en på <https://imgur.com/user/Kanelbuss>

Finna följande bild:



Detta är HexaHue och ger avkodat strängen

“JBCVQ630GA3V62SVOM3V64DSGN2HIWK7MNXWYMDVOJJX2”

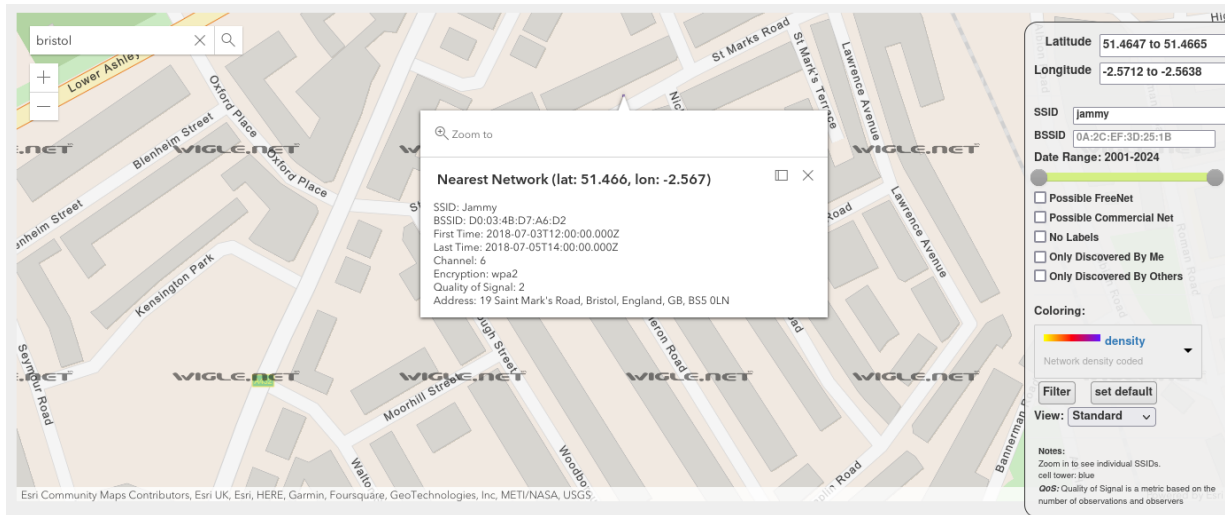
Vilket avkodat från base32 (med CyberChef, kan tydligen vara problematiskt på andra sätt om inte ‘===’ läggs till i slutet) ger flaggan **HEX{n07_jUs7_pr3ttY_col0urS}**.

Trafikspaning

1 För ett antal år sedan fanns det ett nätverk med SSID i Bristol, "Jammy" - men på vilken gata kunde en finna detta nätverk?

Genom att googla på exempelvis "find address for SSID" kan en hitta till <https://wigle.net/index> som tillåter en att söka efter inrapporterade nätverk över världen.

Därefter väljs Bristol som plats och SSID "jammy" som filter, vilket ger två träffar i Bristol -- bägge på **Saint Mark's Road**.



Notera dock att specifik information om nätverket endast är tillgänglig om en är inloggad, vilket i detta fall ej krävdes, då gatan markeras på kartan.

VariousWeb

```
1  Fyra superspännande flaggor gömmer sig på denna hemsida, kan du hitta alla?
2
3  https://ideweb2.hh.se/~pontka22/test/index.php
```

Genom att kontrollera källkoden, eller markera hela sidan, på about.php kan flaggan **HEX{h1dd3N_1n_pla1N_slghT}** hittas.

```
11
12 <h1>About this website</h1>
13 <p>This is a work in progress :)</p><br><br><div id="w3-section-w3-banana"><p>HEX{h1dd3N_1n_pla1N_slghT}</p></div></div>
```

På products.php finns denna knapp, som flyttar sig om muspekaren närmar sig.

Product	Description	Colour	Price
Cotton T-shirt	100% cotton from free range sheep in the Himalayas	Gray	€40.00
Wool T-shirt	100% cashmere wool from free range goats		€80.00
Cargo pants	100% made from renewable hemp plants	Green	€50.00
Banana suit	Dress like a funky banana	Banana yellow of course	€25.00

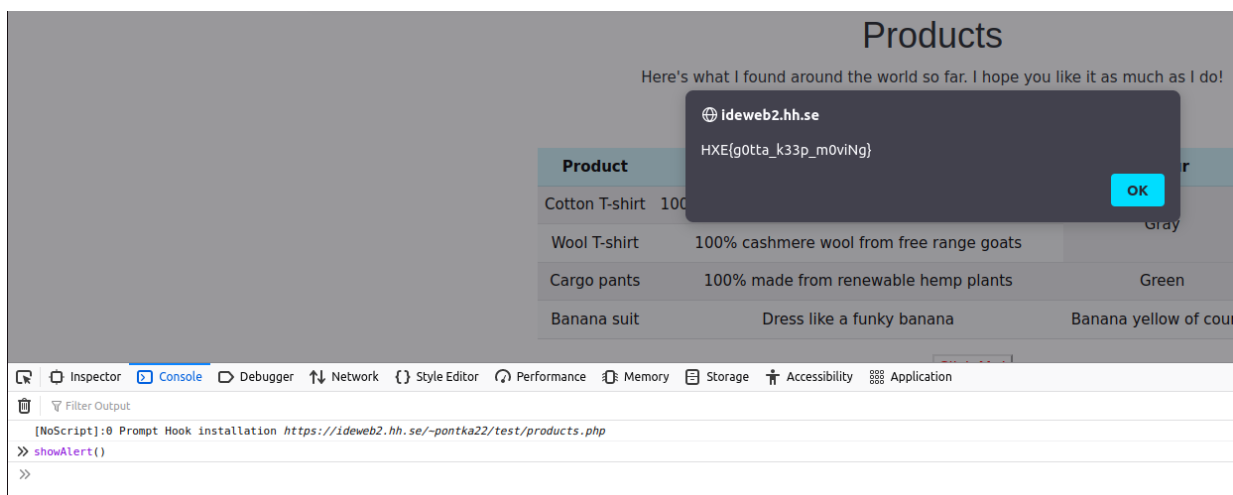
Click Me!

Genom att kontrollera källkoden kan en finna följande kodsnuitt:

```
<button id="movingButton" onmouseover="moveButton()" onclick="showAlert()">Click Me!</button>
```

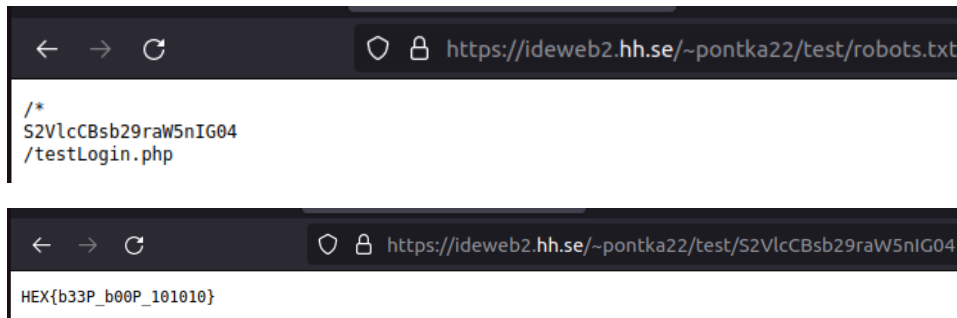
Genom att i webbläsarens konsol kalla på funktionen showAlert() finnes flaggan

HXE{g0tta_k33p_m0viNg} (felstavad och allt!)

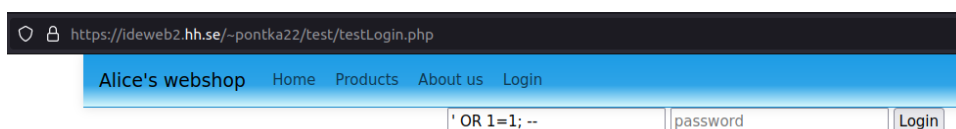


I robots.txt finns strängen “S2VlcCBsb29raW5nIG04”, vilken översatt från base64 ger “Keep looking m8”. Det är dock även namnet på en faktisk fil på servern, vilken innehåller flaggan

HEX{b33P_b00P_101010}



Robots.txt innehåller även referens till `/testLogin.php` vilket är en loginfunktion som är sårbar för SQL-injections.



Efter att ha loggat in via denna funktion och återgå till `index.php` laddas en (egentligen två) bild på hemsidan.

Inspekteras denna, eller källkoden, kan en se att `div`-taggen faktiskt innehåller två bilder som ligger på varandra.

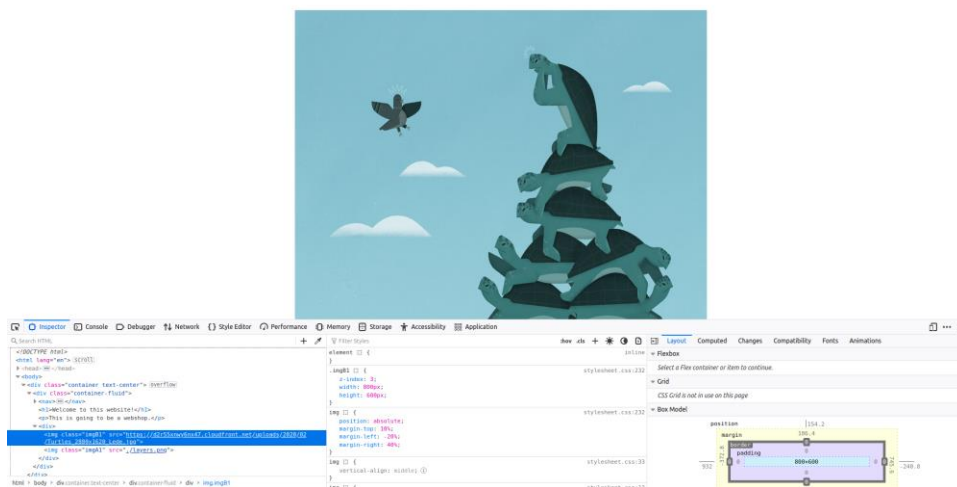


Bild nummer två kan visas antingen genom att modifiera CSS eller direkt navigera till bildens adress.

TQJ{fgDfX3e_4XX_ft3_i4K}

Bild nummer två är text med ett rotationsskiffer, vilket dechiffrerat ger flaggan
HEX{tuRtL3s_4LL_th3_w4Y}

Recipe

ROT13

☒ Rotate lower case chars

☒ Rotate upper case chars

☐ Rotate numbers

Amount
-12

Input

TQJ{fgDfX3e_4XX_ft3_i4K}

Output

HEX{tuRtL3s_4LL_th3_w4Y}