

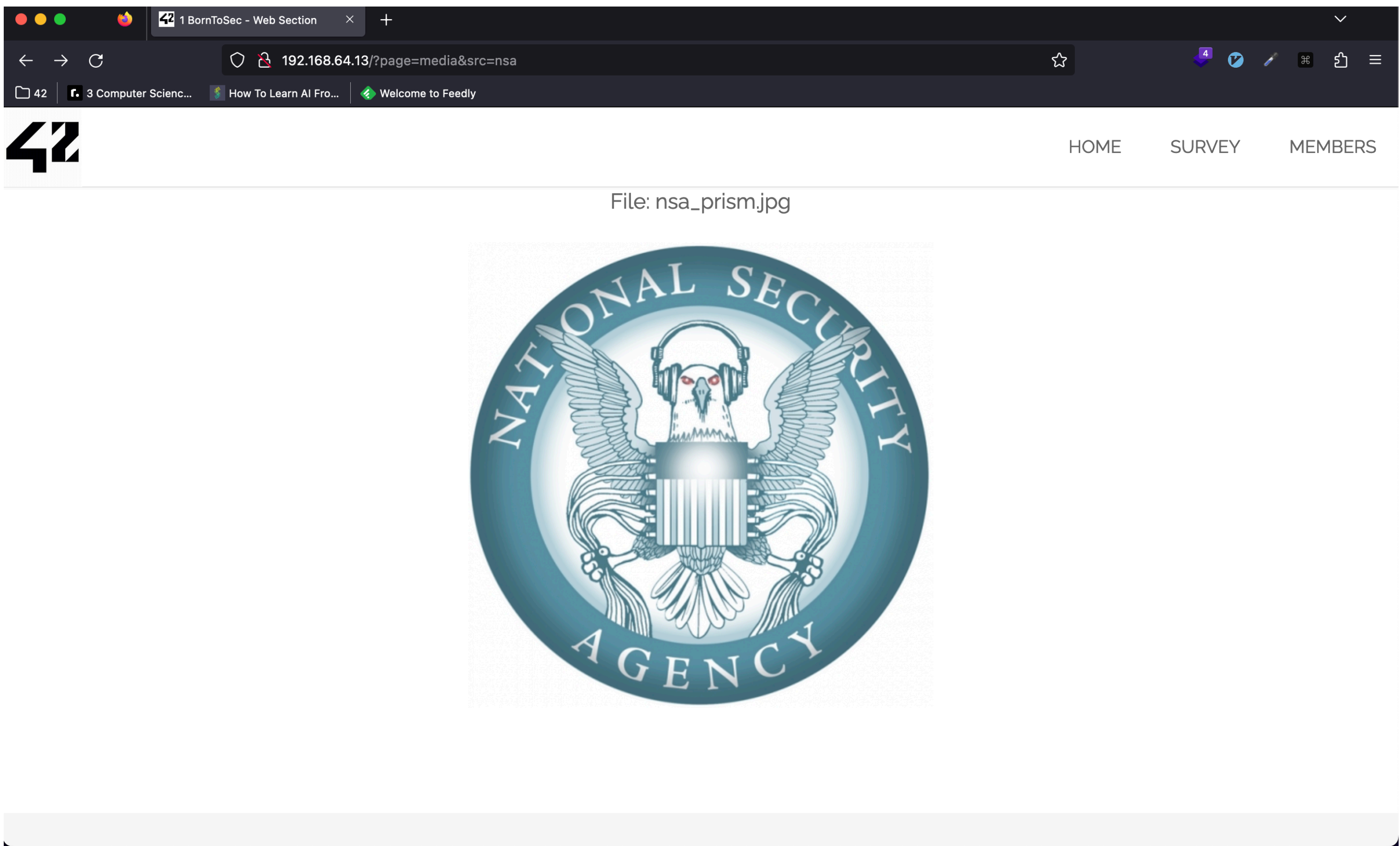
XSS Injection

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

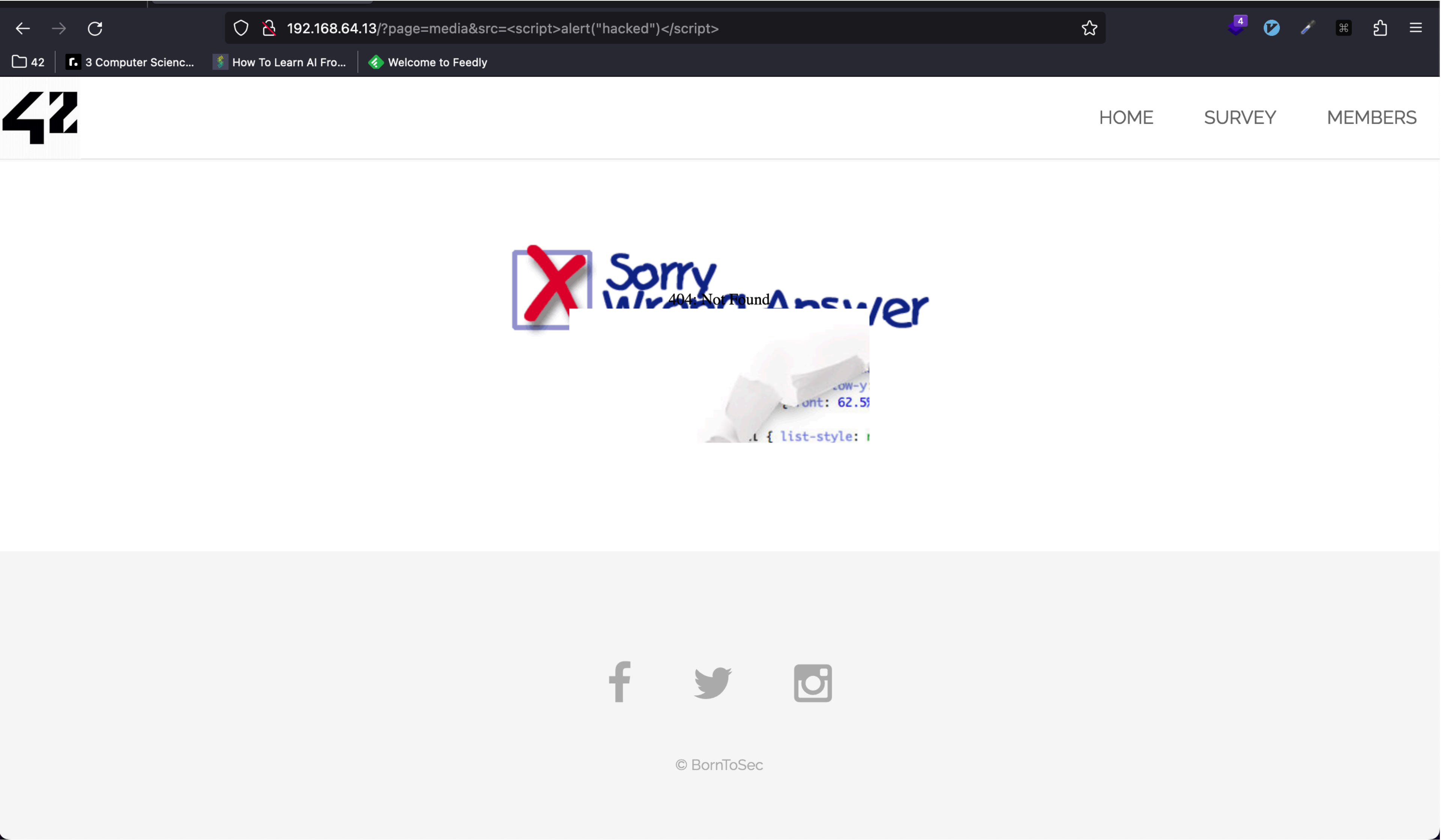
An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page. For more details on the different types of XSS flaws, see: [Types of Cross-Site Scripting](#).

Walk through

Step 1: Go to the nsa image reader page



Step 2: Try injecting XSS in url, that doesn't totally work :/



Step 3: We saw with inspecting code page that the image is loaded by data object

1 BornToSec - Web Section

192.168.64.13/?page=media&src=nsa

423 Computer Scienc...How To Learn AI Fro...Welcome to Feedly


42

HOME

SURVEY

MEMBERS

File: nsa_prism.jpg



Inspecteur

Rechercher dans le HTML

<section id="main" class="wrapper">

|

<object data="http://192.168.64.13/images/nsa_prism.jpg">

#document</div></object></div></td></tr></tbody></table></div></div></section><!--Footer--><footer id="footer"></footer>

élément :: {

html, body, div, span, applet, object, inline:1

iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li, fieldset, form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td, article, aside, canvas, details, embed, figure, figcaption, footer, header, hgroup, menu, nav, output, ruby, section, summary, time, mark, audio, video :: {

margin: 0;

padding: 0;

border: 0;

font-size: 100%;

font: inherit;

vertical-align: baseline;

}

::before,::after :: {

margin: 0;

padding: 0;

border: 0;

font-size: 100%;

font: inherit;

vertical-align: baseline;

}

| |

Élément sélectionné

Aucun problème de compatibilité trouvé.

Tous les problèmes

text-size-adjust (expérimental, préfixe nécessaire)

font-smooth

Paramètres

Step 4: So try the same payload used in step 2 but encoded in base64 ! Done :)

42

3 Computer Scienc...

How To Learn AI Fro...


Welcome to Feedly

HOME

SURVEY

MEMBERS

THE FLAG IS : 928D819FC19405AE09921A2B71227BD9ABA106F9D2D37AC412E9E5A750F1506D



f

© BornToSec

How to fix

- Escape user input for example in php we got htmlspecialchars() function for this purpose
- Use validation for all types of user-supplied input