Exercise - CSP Playground

1. Introduction

> Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement to distribution of malware.

Please use the CSP demo service under RESOURCES and apply

1. CSP
2. Exploit

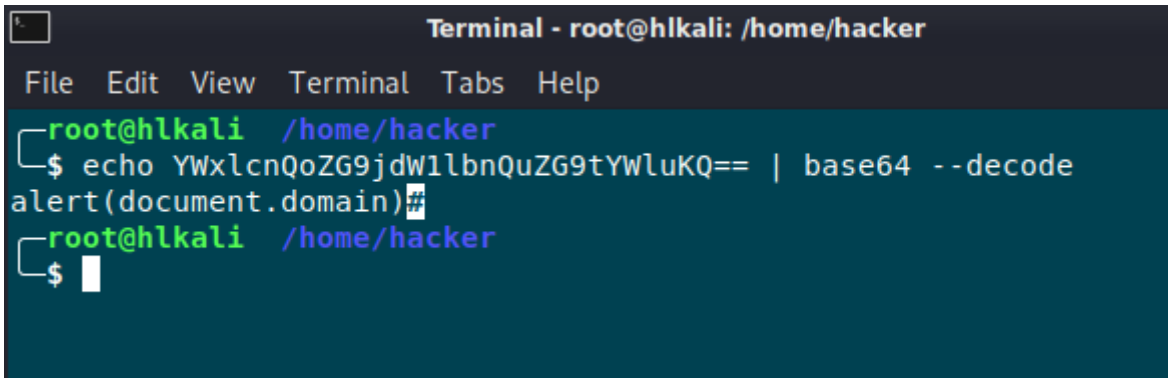
Find out what is allowed and what is denied

Please respond to the following questions

1. what is the meaning of data in
   `<script src="data:;base64,YWxlcnQoZG9jdW1lbnQuZG9tYWluKQ==></script>`
2. what is the difference between hash and nonce
   `<script nonce="ABC">alert(document.cookie)</script>`
   `<img onerror=alert("HASH") src=x>`
3. explain the following csp
   `script-src 'unsafe-hashes' 'self' 'sha256-bnQkgwAfjTxnZSlFxZe1ogJadBHLnRuuL54WC+v+tMY='`

2. Anwers

1. `data` allows loading resources via the data scheme (example Base64 encoded images)
   In the case above it loads a base64 encoded script with the following content:



2. `nonce-` allows an inline script or CSS to execute if the script (example: `<script nonce="r@nd0m">`) tag contains a nonce attribute matching the nonce specifed in the CSP header. The nonce should be a secure random string, and should not be reused.

   `sha256-` allows an inline script or CSS to execute if its hash matches the specified hash in the header. Currently supports SHA256, SHA384 or SHA512

3. `script-src 'self'` allows loading resources from the same origin (same scheme, host and port) `'unsafe-hashes' 'sha256-abc...'` allows you to enable scripts in event handlers (eg onclick). Does not apply to javascript: or inline `<script>`

3. Further Ressources

https://content-security-policy.com/
https://medium.com/@bhaveshthakur2015/content-security-policy-csp-bypass-techniques-e3fa475bfe5d