

Vaycons:~\$ Web-Sistema pentesting



Informazio Sistemen Segurtasuna Kudeatzeko Sistemak

3. Zereginak

Lier Laiseka Merino

Asier Fernández Beraza

Hodei Castro Peral

Andoni Ortiz de Zarate Urruela

AURKIBIDEA

HTTP Segurtasun Goiburuak (CSP, Anti-Clickjacking, etab.)

Konponbidea:

Zerbitzariaren Informazio Ihesak

Konponbidea

Cookie Seguruen Konfiguraziona

Konponbidea:

Anti-CSRF Tokenen Ezartzea

Konponbidea:

CSP: style-src unsafe-inline

Erreferentziak

HTTP Segurtasun Goiburuak (CSP, Anti-Clickjacking, etab.)

ZAP-ek detektatu zuen gure zerbitzariak ez zituela HTTP segurtasun-goiburu kritikoak bidaltzen.

Falta de cabecera Anti-Clickjacking

URL:	http://localhost:81/
Riesgo:	⚠️ Medium
Confianza:	Medium
Parámetro:	x-frame-options
Ataque:	
Evidencia:	
CWE ID:	1021
WASC ID:	15
Origen:	Pasivo (10020 - Cabecera Anti-Clickjacking)
Referencia de Alerta:	10020-1
Vector de Entrada:	
Descripción:	La respuesta no protege contra ataques de "ClickJacking". Debes incluir Content-Security-Policy con la directiva "frame-ancestors" o X-Frame-Options.

Cabecera Content Security Policy (CSP) no configurada

URL:	http://localhost:81/robots.txt
Riesgo:	⚠️ Medium
Confianza:	High
Parámetro:	
Ataque:	
Evidencia:	
CWE ID:	693
WASC ID:	15
Origen:	Pasivo (10038 - Cabecera Content Security Policy (CSP) no configurada)
Referencia de Alerta:	10038-1

Falta encabezado X-Content-Type-Options

URL:	http://localhost:81/
Riesgo:	⚠️ Low
Confianza:	Medium
Parámetro:	x-content-type-options
Ataque:	
Evidencia:	
CWE ID:	693
WASC ID:	15
Origen:	Pasivo (10021 - Falta encabezado X-Content-Type-Options)

Konponbidea:

Lehenik, Apache-ren *mod_headers* modulua gaitu dugu gure **Dockerfile** fitxategian. Modulu hau ezinbestekoa da Apache-k goiburu pertsonalizatuak bidali ahal izateko.

Behin modulua gaituta, **app/.htaccess** fitxategiaren hasieran *<IfModule mod_headers.c>* blokea erabili dugu.

Ondorioz, honako kode hau gehitu egin dugu:

Header always set Content-Security-Policy ...: Honek nabigatzaileari agintzen dio baliabideak (script-ak, estiloak) gure domeinutik ('self') soilik kargatzeko.

Header always set X-Frame-Options "SAMEORIGIN": "Anti-Clickjacking" ahultasuna konpontzen du.

Header always set X-Content-Type-Options "nosniff": MIME motako erasoak saihesten ditu.

Zerbitzariaren Informazio Ihesak

ZAP-ek detektatu zuen zerbitzariak bere bertsio zehatza erakusten zuela, erasotzaileei informazio baliotsua emanez.

El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By""

URL: http://localhost:81/login
Riesgo: Low
Confianza: Medium
Parámetro:
Ataque:
Evidencia: X-Powered-By: PHP/7.2.2
CWE ID: 497
WASC ID: 13
Origen: Pasivo (10037 - El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By"")
Vector de Entrada:
Descripción: El servidor de la web/aplicación está divulgando información mediante uno o más encabezados de respuesta HTTP ""X-Powered-By"". El acceso a tal información podría facilitarle a los atacantes la identificación de otros marcos/componentes de los que su aplicación web depende y las vulnerabilidades a las que pueden estar sujetos tales componentes.

Otra información:

Solución: Asegúrese de que su servidor web, servidor de aplicaciones, balanceador de carga, etc. está configurado para suprimir las cabeceras "X-Powered-By".

El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP

URL: http://localhost:81/main.js
Riesgo: Low
Confianza: High
Parámetro:
Ataque:
Evidencia: Apache/2.4.25 (Debian)
CWE ID: 497
WASC ID: 13
Origen: Pasivo (10036 - Cabecera de Respuesta del Servidor HTTP)
Referencia de Alerta: 10036-2
Vector de Entrada:
Descripción: El servidor web/aplicación está filtrando información de versión a través de la cabecera de respuesta HTTP "Server". El acceso a dicha información puede facilitar a los atacantes la identificación de otras vulnerabilidades a las que está sujeto su servidor web/aplicación.

Otra información:

Solución: Asegúrese de que su servidor web, servidor de aplicaciones, balanceador de carga, etc. está configurado para suprimir la cabecera "Server" o proporcionar detalles genéricos.

Konponbidea:

Dockerfile fitxategia aldatu dugu informazio hori ezkutatzeko.

RUN echo "ServerTokens Prod" ... eta RUN echo "ServerSignature Off" ... komandoek Apache-ren bertsio zehatza ezkutatzen dute.

RUN echo "expose_php = Off" ... komandoak PHP-k X-Powered-By goiburua bidaltzea eragozten du.

Cookie Seguruuen Konfigurazioa

ZAP-ek gure saio-cookiea (PHPSESSID) segurua ez zela adierazi zuen, bi arrazoirengatik:

Cookie Sin Flag HttpOnly : Cookie-a JavaScript bidez irakur zitekeen, XSS erasoen bidez saioa lapurtzeko arriskua areagotuz.

Cookie sin el atributo SameSite : Cookie-a CSRF erasoen aurrean zaurgarria zen.

Cookie Sin Flag HttpOnly

URL:	http://localhost:81/
Riesgo:	Low
Confianza:	Medium
Parámetro:	PHPSESSID
Ataque:	
Evidencia:	Set-Cookie: PHPSESSID
CWE ID:	1004
WASC ID:	13
Origen:	Pasivo (10010 - Cookie Sin Flag HttpOnly)

Cookie sin el atributo SameSite

URL:	http://localhost:81/
Riesgo:	Low
Confianza:	Medium
Parámetro:	PHPSESSID
Ataque:	
Evidencia:	Set-Cookie: PHPSESSID
CWE ID:	1275
WASC ID:	13
Origen:	Pasivo (10054 - Cookie sin el atributo SameSite)

Konponbidea:

Arazoa `app/config.php` fitxategian konpondu dugu, `session_start()` deia egin baino lehen. Gure proiektuak PHP 7.2.2 bertsioa erabiltzen duenez (honek ez du SameSite zuzenean onartzen), `ini_set()` eta `session_set_cookie_params()` konbinazio bat erabili dugu bateragarritasuna mantentzeko.

Anti-CSRF Tokenen Ezartzea

ZAP-ek gure formularioek CSRF (Cross-Site Request Forgery) erasoen aurkako babesik ez zutela adierazi zuen .

Ausencia de Tokens Anti-CSRF

URL:	http://localhost:81/login
Riesgo:	Medium
Confianza:	Low
Parámetro:	
Ataque:	
Evidencia:	<form action="procesar_login.php" method="POST" id="login_form" onsubmit="return validateLoginForm()>
CWE ID:	352
WASC ID:	9
Origen:	Pasivo (10202 - Ausencia de Tokens Anti-CSRF)

Konponbidea:

Saioetan oinarritutako token sistema bat implementatu dugu:

header.php fitxategian, saio bakoitzeko token bakar bat sortzen dugu (`$_SESSION['csrf_token']`).

Token hau ezkutuko eremu gisa (`<input type="hidden" name="csrf_token" ...>`) gure POST formulario guztiengatik sartzen da.

GET bidezko ekintza sentikorretan (adib. liburu pertsonalak gehitu/kendu) tokena URLan gehitzen da (`&token=...`).

Gure logika fitxategi guztien (procesar_...) hasieran `validar_csrf_token()` funtzioari deitzen zaio. Tokena bat ez badator, script-ak huts egiten du.

CSP: style-src unsafe-inline

ZAP-ek 'style-src unsafe-inline' alerta ematen du. Hau mantendu dugu, gure proiektuaren CSS estiloak HTML fitxategien barruan txertatuta daudelako (<style> blokean eta style="..." atributuetan). 'unsafe-inline' kentzeak alertak konponduko lituzke, baina webgunearen diseinu guztia apurtuko luke. Konponbide 'perfektua' CSS guztia kanpoko .css fitxategi batera eramatea litzateke.

CSP: style-src unsafe-inline

URL:	http://localhost:81/robots.txt
Riesgo:	 Medium
Confianza:	High
Parámetro:	Content-Security-Policy
Ataque:	
Evidencia:	default-src 'self'; script-src 'self' ; style-src 'self' 'unsafe-inline'; img-src 'self' data:; object-src 'no ne'; frame-ancestors 'self'; form-action 'self';
CWE ID:	693
WASC ID:	15
Origen:	Pasivo (10055 - CSP)
Referencia de Alerta:	10055-6

Erreferentziak

- https://owasp.org/www-community/controls/Content_Security_Policy
- <https://owasp.org/www-project-secure-headers/>
- <https://developer.mozilla.org/es/docs/Web/HTTP/Reference/Headers/X-Frame-Options>
- <https://httpd.apache.org/docs/2.4/mod/core.html>
- <https://www.php.net/manual/es/ini.core.php>
- <https://owasp.org/www-community/HttpOnly>
- <https://owasp.org/www-community/SameSite>
- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Content-Security-Policy/style-src>