

Ari Hendriatno

20230801114

Laporan penyelesaian masalah

Permasalahan

[1] Missing security header: Referrer-Policy

- Risk Level: 1 (Low)

Vulnerability Details:

- Evidence 1:

- URL: <https://herself-refused-office-hospitality.trycloudflare.com/>

- Evidence: Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with name 'referrer' is not present in the response. Request / Response

- Description: We noticed that the target application's server responses lack the <code>Referrer-Policy</code> HTTP header, which controls how much referrer information the browser will send with each request originated from the current web application.

- Recommendation: The Referrer-Policy header should be configured on the server side to avoid user tracking and inadvertent information leakage. The value `no-referrer` of this header instructs the browser to omit the Referer header entirely.

[2] Missing security header: Strict-Transport-Security

- Risk Level: 1 (Low)

Vulnerability Details:

- Evidence 1:

- URL: <https://herself-refused-office-hospitality.trycloudflare.com/>

- Evidence: Response headers do not include the HTTP Strict-Transport-Security header Request / Response

- Description: We noticed that the target application lacks the HTTP Strict-Transport-Security header in its responses. This security header is crucial as it instructs browsers to only establish secure (HTTPS) connections with the web server and reject any HTTP connections.

- Recommendation: The Strict-Transport-Security HTTP header should be sent with each HTTPS response. The syntax is as follows: `Strict-Transport-Security: max-age=<seconds>[; includeSubDomains]` The parameter `max-age` gives the time frame for requirement of HTTPS in seconds and should be chosen quite high, e.g. several months. A value below 7776000 is considered as too low by this scanner check. The flag `includeSubDomains` defines that the policy applies also for sub domains of the sender of the response.

[3] Missing security header: Content-Security-Policy

- Risk Level: 1 (Low)

Vulnerability Details:

- Evidence 1:

- URL: <https://herself-refused-office-hospitality.trycloudflare.com/>

- Evidence: Response does not include the HTTP Content-Security-Policy security header or meta tag
Request / Response

- Description: We noticed that the target application lacks the Content-Security-Policy (CSP) header in its HTTP responses. The CSP header is a security measure that instructs web browsers to enforce specific security rules, effectively preventing the exploitation of Cross-Site Scripting (XSS) vulnerabilities.

- Recommendation: Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

[4] Missing security header: X-Content-Type-Options

- Risk Level: 1 (Low)

Vulnerability Details:

- Evidence 1:

- URL: <https://herself-refused-office-hospitality.trycloudflare.com/>

- Evidence: Response headers do not include the X-Content-Type-Options HTTP security header
Request / Response

- Description: We noticed that the target application's server responses lack the `X-Content-Type-Options` header. This header is particularly important for preventing Internet Explorer from reinterpreting the content of a web page (MIME-sniffing) and thus overriding the value of the Content-Type header.

- Recommendation: We recommend setting the X-Content-Type-Options header such as `X-Content-Type-Options: nosniff`.

[5] Server software and technology found

- Risk Level: 1 (Low)

Vulnerability Details:

- Evidence 1:

- Software / Version: Alpine.js
- Category: JavaScript frameworks

- Evidence 2:

- Software / Version: Bunny
- Category: CDN

- Evidence 3:

- Software / Version: Bunny Fonts
- Category: Font scripts

- Evidence 4:

- Software / Version: Axios
- Category: JavaScript libraries

- Evidence 5:

- Software / Version: Livewire
- Category: Web frameworks, Miscellaneous

- Evidence 6:

- Software / Version: Laravel

- Category: Web frameworks

- Evidence 7:

- Software / Version: PHP 8.3.23

- Category: Programming languages

- Evidence 8:

- Software / Version: Cloudflare

- Category: CDN

- Description: We noticed that server software and technology details are exposed, potentially aiding attackers in tailoring specific exploits against identified systems and versions.

- Recommendation: We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

[6] Robots.txt file found

- Risk Level: 1 (Low)

Vulnerability Details:

- Evidence 1:

- URL: <https://herself-refused-office-hospitality.trycloudflare.com/robots.txt>

- Description: We found the robots.txt on the target server. This file instructs web crawlers what URLs and endpoints of the web application they can visit and crawl. Website administrators often misuse this file while attempting to hide some web pages from the users.

- Recommendation: We recommend you to manually review the entries from robots.txt and remove the ones which lead to sensitive locations in the website (ex. administration panels, configuration files, etc).

cara penyelesaian

Langkah-langkah:

A. Buat Middleware Kustom

bash

CopyEdit

php artisan make:middleware SecurityHeaders

B. Edit app/Http/Middleware/SecurityHeaders.php

php

CopyEdit

```
namespace App\Http\Middleware;
```

```
use Closure;
```

```
use Illuminate\Http\Request;
```

```
use Symfony\Component\HttpFoundation\Response;
```

```
class SecurityHeaders
```

```
{
```

```
    public function handle(Request $request, Closure $next): Response
```

```
    {
```

```
        $response = $next($request);
```

```
        $response->headers->set('Referrer-Policy', 'no-referrer');
```

```
        $response->headers->set('Strict-Transport-Security', 'max-age=31536000; includeSubDomains');
```

```
        $response->headers->set('Content-Security-Policy', "default-src 'self'");
```

```
        $response->headers->set('X-Content-Type-Options', 'nosniff');
```

```
        $response->headers->set('X-Frame-Options', 'DENY');
```

```
        return $response;
```

```
    }
```

```
}
```

C. Tambahkan ke Middleware Global

Edit file app/Http/Kernel.php:

php

CopyEdit

```
protected $middleware = [  
    // ...  
    \App\Http\Middleware\SecurityHeaders::class,  
];
```

✅ 2. Sembunyikan Informasi Server (Laravel + Docker + PHP)

A. Matikan Header PHP (Laravel on Nginx/Apache)

Jika Anda menggunakan **Apache** atau **Nginx** di Docker, tambahkan di konfigurasi php.ini:

ini

CopyEdit

```
expose_php = Off
```

Untuk Nginx atau Apache, pastikan juga header seperti ini diblokir:

- Server
- X-Powered-By

Contoh untuk **nginx.conf**:

nginx

CopyEdit

```
server {  
    ...  
    add_header X-Powered-By "";  
    add_header Server "";  
}
```

✅ 3. Revisi atau Batasi robots.txt

Pastikan tidak mengekspos direktori sensitif seperti /admin, /config, dsb.

Contoh **robots.txt** aman:

makefile

CopyEdit

```
User-agent: *
```

```
Disallow: /private/
```

```
Disallow: /admin/
```

Restart Laravel & Docker

Setelah semua konfigurasi dilakukan:

bash

CopyEdit

php artisan config:clear

php artisan optimize:clear

Jika menggunakan Docker, rebuild image:

bash

CopyEdit

docker-compose down

docker-compose up --build -d

Tambahan: Cek Ulang Security Headers

Setelah semuanya berjalan, pastikan header sudah muncul dengan tools seperti:

- <https://securityheaders.com/>
 - `curl -I https://yourdomain.com`
-

Jika kamu menggunakan **Cloudflare**, pastikan juga:

- Tidak ada header yang ditimpa oleh Cloudflare.
- Tambahkan header yang sama melalui **Cloudflare Rules (Transform Rules → Modify Response Header)** jika ingin full di level CDN.