1. KEAMANAN DALAM MODEL

Model Buyer dan Seller dalam sistem ini dirancang dengan fokus pada keamanan data pengguna, khususnya pada data sensitif seperti email, nomor telepon, dan alamat. Pengamanan dilakukan dengan menggunakan fitur enkripsi otomatis dari Laravel, yaitu Crypt::encryptString() untuk menyimpan data dalam bentuk terenkripsi ke dalam basis data, serta Crypt::decryptString() untuk mendekripsi data secara otomatis ketika diakses. Dengan mekanisme ini, data tidak disimpan dalam bentuk teks biasa (plain text), sehingga apabila terjadi pelanggaran keamanan atau pencurian data, informasi sensitif tetap terlindungi. Relasi antar model juga telah diterapkan dengan baik, seperti Buyer yang memiliki hubungan hasMany dengan model Transaction dan Review, serta Seller yang memiliki relasi serupa dengan model House. Meskipun pendekatan ini meningkatkan keamanan data, terdapat beberapa keterbatasan, seperti tidak dapat melakukan pencarian (searching) atau pengurutan (sorting) berdasarkan data terenkripsi, karena nilai yang disimpan bukan nilai asli. Selain itu, proses dekripsi sangat bergantung pada APP_KEY dari Laravel. Jika kunci ini hilang atau berubah, maka seluruh data terenkripsi tidak akan dapat diakses kembali, sehingga diperlukan sistem pengelolaan kunci yang aman dan backup secara rutin. Perlu dicatat pula bahwa model Buyer belum mengenkripsi field address, berbeda dengan model Seller yang sudah mengenkripsi field tersebut. Oleh karena itu, konsistensi dalam perlakuan terhadap field sensitif antar model perlu diperhatikan untuk menjaga integritas dan standar keamanan sistem secara keseluruhan.

2. PENETRATION TEST

[1] Missing security header: Referrer-Policy
- Risk Level: 1 (Low)

VULNERABILITY DETAILS:

- Evidence 1:
  - URL: https://respected-vehicles-comments-plots.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.

SOLUSI

Menambahkan refererpolicyheader

<?php

```
namespace App\Http\Middleware;
use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;
class ReferrerPolicyHeader
{
/**
* Handle an incoming request.
*
* @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) $next
*/
public function handle(Request $request, Closure $next): Response
{
$response = $next($request);
$response->headers->set('Referrer-Policy', 'no-referrer');
return $response;
}
}
```

dan menambahkan ke kernel

```
<?php
use App\Http\Middleware\SecurityHeaders;
use Illuminate\Foundation\Application;
use Illuminate\Foundation\Configuration\Exceptions;
use Illuminate\Foundation\Configuration\Middleware;
return Application::configure(basePath: dirname(__DIR__))
->withRouting(
web: __DIR__ . '/../routes/web.php',
commands: __DIR__ . '/../routes/console.php',
health: '/up',
```

```
api: __DIR__ . '/../routes/api.php',
)
->withMiddleware(function (Middleware $middleware) {
$middleware->append(\App\Http\Middleware\ReferrerPolicyHeader::class);
})
->withExceptions(function (Exceptions $exceptions) {
//
})->create();
```

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

Risk Level: 1 (Low)

VULNERABILITY DETAILS:

- Evidence 1:
  - URL: https://respected-vehicles-comments-plots.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.

SOLUSI

Menambahkan

```
if ($request->isSecure()) {
$response->headers->set('Strict-Transport-Security', 'max-age=15768000; includeSubDomains');
}
```

ke reffererpolicyHeader

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

- Risk Level: 1 (Low)

VULNERABILITY DETAILS:

- Evidence 1:
  - URL: https://respected-vehicles-comments-plots.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.

SOLUSI

Sudah menambahkan

```
$response->headers->set('Content-Security-Policy',
        "default-src 'self'; " .
        "script-src 'self'; " .
        "style-src 'self'; " .
        "img-src 'self' data:; " .
        "font-src 'self'; " .
        "object-src 'none'; " .
        "base-uri 'self'; " .
        "frame-ancestors 'none';"
    );
```

namun masih muncul vulnerability

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

- Risk Level: 1 (Low)

VULNERABILITY DETAILS:

- Evidence 1:
  - URL: https://respected-vehicles-comments-plots.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 <code>X-Content-Type-Options</code> 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`.

SOLUSI

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

ke ReffererPolicyHeader

[5] Server software and technology found

- Risk Level: 1 (Low)

VULNERABILITY DETAILS:

- Evidence 1:
  - Software / Version: Alpine.js 3.14.9
  - Category: JavaScript frameworks
- Evidence 2:
  - Software / Version: Bunny
  - Category: CDN
- Evidence 3:
  - Software / Version: Filamentphp
  - Category: Development
- Evidence 4:
  - Software / Version: Bunny Fonts
  - Category: Font scripts
- Evidence 5:
  - Software / Version: Livewire
  - Category: Web frameworks, Miscellaneous
- Evidence 6:
  - Software / Version: Laravel
  - Category: Web frameworks
- Evidence 7:
  - Software / Version: PHP
  - 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.

SOLUSI

Sudah mematikan expose_php = Off dan menambahkan
$response->headers->remove('X-Powered-By');
$response->headers->remove('Server')
Namun masoh muncul

[6] Robots.txt file found

- Risk Level: 1 (Low)

VULNERABILITY DETAILS:

- Evidence 1:
  - URL: https://respected-vehicles-comments-plots.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).

SOLUSI

menghapus file robots.txt

3. ANALISA

Berdasarkan hasil audit keamanan terhadap aplikasi yang dikembangkan menggunakan Laravel 12, ditemukan sejumlah kerentanan terkait absennya beberapa header keamanan penting. Untuk mengatasi hal tersebut, dilakukan penambahan middleware SecurityHeaders yang secara eksplisit mengatur header seperti Referrer-Policy, Strict-Transport-Security, Content-Security-Policy, dan X-Content-Type-Options. Header Referrer-Policy disetel ke no-referrer guna mencegah pengiriman informasi asal ke domain eksternal. Sementara itu, Strict-Transport-Security ditambahkan secara kondisional ketika koneksi HTTPS aktif, dengan nilai max-age=15768000; includeSubDomains, yang bertujuan memaksa penggunaan protokol HTTPS dan meningkatkan keamanan komunikasi data. Untuk memperkuat perlindungan terhadap injeksi konten, disisipkan pula Content-Security-Policy dengan nilai default-src 'self', disertai pembatasan terhadap sumber skrip, gambar, dan objek. Namun demikian, penggunaan 'unsafe-inline' pada style-src masih dipertahankan untuk menjaga kompatibilitas, meskipun disadari bahwa hal ini tetap membawa risiko dan dapat ditingkatkan dengan pendekatan nonce atau hash. Selain itu, header X-Content-Type-Options dengan nilai nosniff berhasil ditambahkan untuk mencegah browser melakukan MIME sniffing yang berpotensi dimanfaatkan oleh pihak tidak bertanggung jawab. Di luar masalah header, audit juga mengungkap bahwa informasi terkait perangkat lunak yang digunakan (seperti Laravel, PHP, Livewire, dan Alpine.js) dapat terdeteksi oleh pihak luar, yang berpotensi digunakan untuk fingerprinting. Untuk mengurangi risiko ini, disarankan agar server dikonfigurasi untuk menyembunyikan informasi seperti X-Powered-By dan Server, serta mempertimbangkan untuk menghindari penggunaan CDN yang secara eksplisit menyebutkan nama teknologi. Terakhir, ditemukan keberadaan file robots.txt yang berisi Disallow: untuk semua user-agent. Meskipun file ini tidak berisi direktori sensitif, penghapusannya dilakukan sebagai langkah preventif untuk mencegah pengungkapan tidak sengaja terhadap struktur direktori internal. Secara keseluruhan, langkah-langkah mitigasi yang telah dilakukan sudah memperkuat postur keamanan aplikasi web dan mengikuti rekomendasi dari OWASP serta praktik terbaik industri.