BETRIK
BESEMAH TEKNOLOGI INFORMASI & KOMPUTER
Editor Office : Pusat Penelitian & Pengabdian Pada Masyarakat (PPPM) ITPA
Phone : 0857-9716-9578
email : betrikitpa@itpa.ac.id
ISSN : 2339 - 1871

Analisis *Vulnerability Assessment* Sistem Informasi Pendidikan, Pelatihan PT Azure Samudera Karsa Menggunakan ZAP

Soni Ayi Purnama[1]
Fakultas Teknik, Program Studi Sistem Informasi, Universitas Bengkulu, Bengkulu, Indonesia[1]
Sur-el :*soni.mkom @unib.ac.id[1]

Penulis Korespondensi: Soni Ayi Purnama, soni.mkom@unib.ac.id

Abstrak: Penelitian ini bertujuan untuk menganalisis keamanan sistem informasi pendidikan dan pelatihan pada PT Azure Samudera Karsa dengan menggunakan metode *vulnerability assessment*. Pentingnya peningkatan keamanan sistem informasi menjadi fokus utama dalam rangka meningkatkan kredibilitas dan juga kualitas layanan pendidikan dan pelatihan yang diselenggarakan oleh PT Azure Samudera Karsa. Dalam era digital saat ini, sistem informasi yang rentan terhadap serangan siber dapat menimbulkan berbagai konsekuensi negatif, termasuk kebocoran data peserta pelatihan, manipulasi informasi, hingga gangguan operasional. Oleh karena itu, evaluasi keamanan menjadi aspek penting yang tidak boleh diabaikan. Alat yang digunakan dalam menganalisis keamanan sistem informasi pendidikan dan pelatihan PT Azure Samudera Karsa adalah *Zed Attack Proxy* (ZAP), sebuah aplikasi *open source* yang umum digunakan untuk mendeteksi celah keamanan pada aplikasi *web*. Hasil analisis *vulnerability assessment* menunjukkan bahwa terdapat 3 level peringatan, yaitu: 3 peringatan pada level medium, 6 peringatan pada level *low*, dan 3 pada level informational, dengan total keseluruhan sebanyak 13 peringatan. Temuan ini menjadi dasar penting bagi manajemen untuk segera melakukan perbaikan guna meminimalkan risiko dan meningkatkan perlindungan terhadap sistem yang digunakan.

Kata kunci : Sistem Informasi, *Vulnerabilty Assessment*, ZAP

Abstract: *This study aims to analyze the security of the educational and training information system at PT Azure Samudera Karsa using the vulnerability assessment method. Enhancing the security of information systems is a key priority in order to improve the credibility and quality of the educational and training services provided by PT Azure Samudera Karsa. In today's digital era, information systems that are vulnerable to cyberattacks can lead to various negative consequences, including data breaches, information manipulation, and operational disruptions. Therefore, security evaluation becomes a crucial aspect that must not be overlooked. The tool used to assess the security of the educational and training information system at PT Azure Samudera Karsa is Zed Attack Proxy (ZAP), an open-source application commonly used to detect security vulnerabilities in web applications. The results of the vulnerability assessment revealed three levels of alerts: 3 alerts at the medium level, 6 alerts at the low level, and 3 informational alerts, totaling 13 alerts. These findings serve as an important basis for management to take immediate corrective actions to minimize risks and enhance the protection of the system in use..*

Keywords: *Information System, Vulnerabilty Assement, ZAP*

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

1. PENDAHULUAN

Dalam era transformasi digital yang semakin pesat, sistem informasi dan keamanan jaringan komputer menjadi komponen vital yang menopang berbagai aktivitas dalam kehidupan modern. Jaringan komputer tidak hanya berfungsi sebagai sarana komunikasi dan pertukaran data, tetapi juga berperan penting dalam pengelolaan sumber daya secara efektif dan efisien. Akan tetapi, dengan meningkatnya kompleksitas infrastruktur jaringan, risiko terhadap aspek keamanannya pun kian meningkat. Ancaman-ancaman ini mencakup serangan malware, kejahatan siber (cybercrime), serta eksploitasi terhadap celah keamanan atau kerentanan sistem yang belum ditangani dengan baik.

Simons (2018) menyatakan bahwa keamanan sistem informasi merupakan usaha untuk mengidentifikasi dan mencegah tindakan tidak etis dalam sistem yang berbasis informasi, meskipun informasi tersebut tidak memiliki eksistensi fisik. Lebih lanjut, Perrin (2008) menekankan bahwa sistem informasi dibangun berdasarkan tiga prinsip fundamental yang dikenal sebagai CIA Triad, yaitu Confidentiality (kerahasiaan), Integrity (integritas), dan Availability (ketersediaan). Ketiga prinsip ini menjadi fondasi utama dalam perancangan kebijakan dan strategi keamanan informasi, serta digunakan sebagai acuan dalam mengidentifikasi risiko dan menentukan langkah mitigasi yang tepat.

Dalam praktiknya, sistem digital seperti situs web sering kali menjadi target utama serangan siber karena tingginya interaksi pengguna dan kompleksitas kode yang digunakan. Serangan yang umum terjadi meliputi Distributed Denial of Service (DDoS), brute force attack, Cross-site Scripting (XSS), SQL Injection, penyebaran malware, serta berbagai teknik eksploitasi lainnya. Untuk mengantisipasi hal tersebut, diperlukan pendekatan strategis dalam bentuk evaluasi keamanan sistem secara berkala dan menyeluruh.

Salah satu metode yang terbukti efektif dalam proses ini adalah penetration testing atau uji penetrasi. Metode ini dilakukan dengan cara melakukan simulasi serangan terhadap sistem untuk mengidentifikasi potensi kerentanan tanpa harus mengakses informasi sensitif seperti kredensial pengguna atau data pribadi. Penetration testing memungkinkan pengembang untuk memahami titik lemah sistem sebelum dimanfaatkan oleh pihak yang tidak bertanggung jawab, sehingga dapat menjadi langkah preventif dalam meningkatkan ketahanan dan keandalan sistem informasi digital. Dalam konteks pendidikan dan pelatihan berbasis teknologi, penerapan metode ini juga sangat relevan untuk menjamin integritas sistem yang digunakan serta melindungi data pengguna dari risiko kebocoran atau manipulasi.

Pentingnya menganalisis kerentanan kemanan *(vulnerability assessment)* sistem informasi pendidikan dan pelatihan pada PT. Azure Samudera Karsa yaitu untuk mencegah terjadinya serangan siber dan juga untuk meningkatkan kredibelitas dan kualitas layanan. Alat *penetration testing* yang digunakan untuk menganalisis kerentanan sistem informasi yaitu dengan menggunakan ZAP.

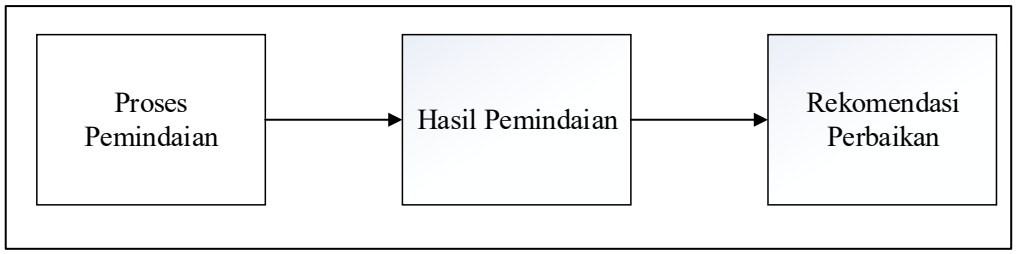
2. METODOLOGI PENELITIAN

Metode *Vulnerability Assessment* diterapkan melalui serangkaian tahapan sistematis untuk mengidentifikasi potensi risiko dan mengelompokkan temuan berdasarkan tingkat kerentanannya terhadap

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

All author: Soni Ayi Purnama

suatu sistem. Proses ini dilakukan dengan memanfaatkan alat bantu ZAP. Seluruh tahapan dalam penelitian ini disusun dan dilaksanakan sesuai dengan alur proses yang ditampilkan pada gambar 1.



Gambar 1. Tahapan Penelitian

2.1 Proses Pemindaian

Pada tahapan proses pemindaian terlebih dahulu dilakukan identifikasi website sistem informasi pendidikan dan pelatihan PT. Azure Samudera Karsa dan setelah dilakukan identifikasi website yaitu dilakukan *Vulnerability Assessment* dengan menggunakan ZAP.

2.2 Hasil Pemindaian

Hasil pemindaian merupakan sebuah peringatan tentang celah *(vuln)* yang terdistribusi, celah tersebut didistribusikan menjadi peringatan *high, medium low* dan sebuah *informational* sehingga analisis menjadi semakin mudah dan sangkil.

2.3 Rekomendasi Pemindaian

Takhir dari penelitian ini adalah penyusunan rekomendasi perbaikan. Rekomendasi tersebut berisi dokumentasi hasil evaluasi terhadap kerentanan sistem yang bertujuan memberikan masukan konstruktif bagi pengembang. Tujuannya adalah agar sistem yang dikembangkan memiliki perlindungan yang memadai dan tidak mudah dieksploitasi oleh pihak yang tidak bertanggung jawab. Pengembang diharapkan dapat merujuk pada rekomendasi ini untuk mengatasi dan memperbaiki potensi celah keamanan yang telah teridentifikasi.

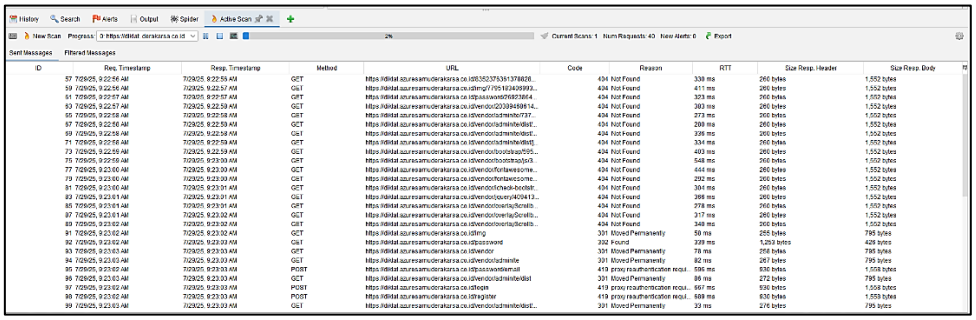
3. HASIL DAN PEMBAHASAN

3.1 Proses Pemindaian

Dalam tahapan ini dilakukan identifikasi terhadap website yang akan dilakukan *penetration testing*. Sistem yang akan dilakukan *penetration testing* yaitu Sistem Informasi Pendidikan dan Pelatihan PT. Azure Samudera Karsa yang dapat diakses melalui halaman web *https://diklat.azuresamuderakarsa.co.id/*. Sistem pendidikan dan pelatihan ini dikembangkan menggunakan bahasa pemrograman PHP versi 8, dengan memanfaatkan Framework Laravel versi 8, serta dijalankan pada server *Apache*.. Sistem pendidikan dan pelatihan ini memiliki fitur proses pendaftaran peserta pelatihan, proses pelatihan dan mengelola keuangan berupa arus kas. Proses pemindaian *(penetration testing)* yaitu dengan menggunakan *Zed Attack Proxy* (ZAP). Alat penetrasi ZAP yang digunakan yaitu ZAP versi 2.16.1.

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

All author: Soni Ayi Purnama



Gambar 2. Proses Pemindaian

3.2 Hasil Pemindaian

Berdasarkan hasil pengujian penetrasi yang dilakukan dengan menggunakan ZAP, analisis terhadap *Vulnerability Assessment* mencakup tahapan proses pemindaian (*scanning*), identifikasi peringatan yang muncul selama proses tersebut sebagai berikut:

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	1 (7.7%)	2 (15.4%)	0 (0.0%)	3 (23.1%)
	Low	0 (0.0%)	1 (7.7%)	5 (38.5%)	0 (0.0%)	6 (46.2%)
	Informational	0 (0.0%)	1 (7.7%)	1 (7.7%)	2 (15.4%)	4 (30.8%)
	Total	0 (0.0%)	3 (23.1%)	8 (61.5%)	2 (15.4%)	13 (100%)

Gambar 3. Hasil *Scanning* ZAP

Gambar 3. tersebut menampilkan tabel hasil *vulnerability assessment* yang mengkategorikan temuan berdasarkan dua dimensi, yaitu tingkat risiko *(Risk)* dan tingkat kepercayaan *(Confidence)*. Dalam tabel ini, tidak ditemukan kerentanan dengan risiko tinggi. Risiko medium tercatat sebanyak 3 kasus (23,1%), dengan sebagian besar memiliki tingkat kepercayaan medium. Risiko *low* merupakan yang paling dominan dengan total 6 kasus (46,2%), mayoritas juga berada pada tingkat kepercayaan medium. Sementara itu, informational memiliki 4 kasus (30,8%) yang tersebar di berbagai tingkat kepercayaan. Secara keseluruhan, dari total 13 temuan, sebagian besar memiliki tingkat kepercayaan medium (8 kasus atau 61,5%), diikuti oleh *low* (2 kasus atau 15,4%), *high* (3 kasus atau 23,1%), dan tidak ada temuan yang dikonfirmasi langsung oleh pengguna maupun yang berada pada tingkat kepercayaan tinggi terkait risiko tinggi. Temuan ini menunjukkan bahwa sebagian besar kerentanan memiliki tingkat risiko rendah hingga menengah, namun tetap memerlukan perhatian untuk mitigasi.

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

Alert type	Risk	Count
Content Security Policy (CSP) Header Not Set	Medium	5 (38.5%)
Missing Anti-clickjacking Header	Medium	4 (30.8%)
Vulnerable JS Library	Medium	1 (7.7%)
Big Redirect Detected (Potential Sensitive Information Leak)	Low	4 (30.8%)
Cookie No HttpOnly Flag	Low	7 (53.8%)
Cross-Domain JavaScript Source File Inclusion	Low	16 (123.1%)
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	8 (61.5%)
Strict-Transport-Security Header Not Set	Low	16 (123.1%)
X-Content-Type-Options Header Missing	Low	15 (115.4%)
Authentication Request Identified	Informational	1 (7.7%)
Information Disclosure - Suspicious Comments	Informational	2 (15.4%)
Re-examine Cache-control Directives	Informational	5 (38.5%)
Session Management Response Identified	Informational	12 (92.3%)
Total		13

Gambar 4. Daftar Kerentanan Berdasarkan Hasil *Scanning* ZAP

Dari hasil scanning yang dilakukan dengan menggunakan ZAP maka dapat dihasilkan bahwa terdapat 3 level peringatan yaitu: 3 peringatan pada level medium; 6 peringatan pada level low dan 3 pada level informational dengan total 13 peringatan. Berikut daftar kerentanan dan jenis serangan pada celah tersebut dituangkan pada tabel 1.

Tabel 1. Daftar Kerentanan dan Jenis Serangan

No	Kerentanan	Jenis Serangan
1	*Content Security Policy (CSP) Header Not Set*	*Cross-site Scripting* (XSS)
2	*Missing Anti-clickjacking Header*	*Clickjacking attack*
3	*Vulnerable JS Library*	*Cross-site Scripting* (XSS), *Prototype Pollution*, *Denil of Service* (Dos), *Remote Code Execution* (RCE),
4	*Big Redirect Detected (Potential Sensitive Information Leak)*	*Session hijacking*
5	*Cookie No HttpOnly Flag*	*Cross-site Scripting* (XSS)
6	*Cross-Domain JavaScript Source File Inclusion*	XSS, S*ession Hijacking, Phising, Man in the Middle* (MitM)
7	*Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*	*Targeting exploitation*, *Zero day targeting*
8	*Strict-Transport-Security Header Not Set*	*SSL strip attack, downgrade attack, man in the middle* (MitM)
9	*X-Content-Type-Options Header Missing*	*MIME Snifing Attack,* XSS
10	*Authentication Request Identified*	*Targeted expoloitation*
11	*Information Disclosure - Suspicious Comments*	
12	*Re-examine Cache-control Directives*	
13	*Session Management Response Identified*	

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

3.3 Rekomendasi Perbaikan

Tahap akhir dari penelitian ini adalah penyusunan rekomendasi perbaikan. Rekomendasi tersebut berfungsi sebagai bentuk dokumentasi hasil evaluasi terhadap kerentanan sistem, yang ditujukan sebagai acuan bagi pengembang dalam meningkatkan keamanan aplikasi. Dengan mengikuti rekomendasi ini, pengembang diharapkan dapat melakukan penyesuaian yang diperlukan guna melindungi sistem dari potensi serangan yang dilakukan oleh pihak yang tidak bertanggung jawab.

1. *Content Security Policy (CSP) Header Not Set*

   Melakukan konfigurasi pada *header* CSP untuk mengurangi risiko serangan *Cross-site Scripting* (XSS) yaitu dengan menambahkan kode program pada *file .htaccess* sebagai berikut:

```
Header set Content-Security-Policy "default-src 'self'; script-src 'self'; style-src 'self'; img-src 'self'"
```

2. *Missing Anti-clickjacking Header*

   Untuk menangani serangan yang dibuka dalam <*iframe*> dalam situs lain atau dinamakan *clickjacking attack* maka harus mengikuti langkah-langkah berikut ini:

   a. Membuat *middleware* baru

```
php artisan make:middleware XFrameOptionsHeader
```

   b. sebagai berikut:

```
namespace App\Http\Middleware;
use Closure;
use Illuminate\Http\Request;
class XFrameOptionsHeader
{
    public function handle(Request $request, Closure $next)
    {
        $response = $next($request);

        // Menolak semua framing
        $response->headers->set('X-Frame-Options', 'DENY');

        // Alternatif: Izinkan hanya domain tertentu
        // $response->headers->set('X-Frame-Options', 'SAMEORIGIN');

        return $response;
    }
}
```

3. *Vulnerable JS Library*

   Vulnerable JS *Library* rentan terhadap serangan *Cross-site Scripting* (XSS*), Prototype Pollution*, *Denil of Service* (Dos), *Remote Code Execution* (RCE), oleh karena itu perlu dilakukan pembaharuan versi pada JS Library

4. *Big Redirect Detected (Potential Sensitive Information Leak)*

   Kerentanan *big redirect detected* yaitu rentan terhadap serangan phising maupun *session hijacking* untuk mengantisipasi hal tersebut maka perlu dilakukan perbaikan sebagai berikut:

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

a. Menggunakan session untuk menyimpan token dengan tidak melalui *query string* sebagai berikut:

```
session()->put('auth_token', $token);
return redirect('/dashboard);
```

b. Menggunakan middleware untuk validasi redirect validasi redirect sebagai berikut:

```
// App\Http\Middleware\ValidateRedirectUrl.php

public function handle($request, Closure $next)
{
    $redirectTo = $request->input('redirect_to');

    if ($redirectTo && !Str::startsWith($redirectTo, config('app.url'))) {
        abort(403, 'Redirect URL not allowed.');
    }

    return $next($request);
}
```

5. *Cookie No HttpOnly Flag*

Kerentanan dari cookie no httponly flag yaitu rentan terhadap serangan XSS sehingga perlu dilakukan perbaikan sebagai berikut:

a. Pada route *file "config/session"*:

```
'http_only' => true,
```

b. Menggunakan *middleware agar menjamin cookie* aman

```
namespace App\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Cookie;

class ForceSecureCookies
{
    public function handle($request, Closure $next)
    {
        $response = $next($request);

        foreach ($response->headers->getCookies() as $cookie) {
            $response->headers->setCookie(
                new \Symfony\Component\HttpFoundation\Cookie(
                    $cookie->getName(),
                    $cookie->getValue(),
                    $cookie->getExpiresTime(),
                    $cookie->getPath(),
                    $cookie->getDomain(),
                    true, // secure
                    true  // httpOnly
                )
            );
        }
        return $response;
    }
}
```

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

6. *Cross-Domain JavaScript Source File Inclusion*

Kerentanan *Cross-Domain JavaScript Source File Inclusion* yaitu rentan terhadap serangan XSS, *session hijacking, phising, man in the middle* (MitM) oleh karena itu perlu dilakukan perbaikan dari penggunaaan javascript dari sumber yang tidak terpercaya. Berikut daftar langkah-langkah perbaikan untuk kerentanan *Cross-Domain JavaScript Source File Inclusion*:

Tabel 2. Langkah Perbaikan Kerentanan *Cross-Domain JavaScript Source File Inclusion*

Langkah Perbaikan	Tujuan dan Penjelasan
Memanfaatkan CDN yang terpercaya dan menggunakan protokol HTTPS	Menghindari pemuatan skrip dari sumber eksternal yang tidak dapat dijamin keamanannya
Menambahkan header Content Security Policy (CSP)	Membatasi dan mengontrol sumber asal JavaScript yang dapat dijalankan oleh aplikasi
Menerapkan Subresource Integrity (SRI)	Menghindari risiko manipulasi terhadap file JavaScript yang dimuat dari CDN
Menghindari penggunaan JavaScript eksternal jika tidak diperlukan	Mengurangi potensi vektor serangan dari pihak ketiga
Melakukan audit JavaScript secara berkala	Memastikan hanya skrip yang benar-benar dibutuhkan yang dimuat oleh aplikasi

7. *Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*

Kerentananan dari Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) yaitu rentan dari serangan *targeting exploitation*, *zero day targeting* oleh karena itu perlu dilakukan perbaikan sebagai berikut:

Tabel 3. Langkah Perbaikan Kerentanan *Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*

Lapisan	Tindakan yang Dilakukan
PHP	Nonaktifkan opsi *expose_php* dengan menetapkan nilainya ke *Off* pada berkas konfigurasi php.ini
Laravel	Implementasikan middleware khusus untuk menghapus header *X-Powered-By* dari setiap respons
Web Server	Konfigurasikan server dengan instruksi *Header unset* (untuk Apache) guna menghilangkan *header* terkait
Verifikasi	Lakukan pemeriksaan dengan menggunakan alat seperti *DevTools* pada *browser* atau perintah *curl* untuk memastikan header tidak muncul lagi

8. *Strict-Transport-Security Header Not Set*

Kerentanan dari *Strict-Transport-Security Header Not Set* yaitu rentan terhadap serangan *SSL strip attack, downgrade attack, man in the middle* (MitM). Berikut daftar perbaikan yang dapat dilakukan:

Tabel 4. Langkah Perbaikan Kerentanan *Server Strict-Transport-Security Header Not Set*

Langkah	Tujuan dan Penjelasan
Terapkan penggunaan HTTPS secara menyeluruh di Laravel	Mencegah pertukaran data dalam format plaintext melalui protokol HTTP
Konfigurasikan HSTS melalui *.htaccess* atau *virtual host Apache*	Menginstruksikan browser untuk selalu menggunakan koneksi HTTPS secara otomatis

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

Langkah	Tujuan dan Penjelasan
Aktifkan modul headers pada Apache	Memungkinkan penambahan header HTTP tambahan oleh server
Gunakan *URL::forceScheme('https')* di Laravel	Memastikan seluruh URL yang dihasilkan Laravel menggunakan protokol HTTPS
Lakukan verifikasi dengan perintah curl atau melalui DevTools browser	Menjamin bahwa header *Strict-Transport-Security* dikirim dalam setiap respons server

9. *X-Content-Type-Options Header Missing*

Kerentanan pada *X-Content-Type-Options Header Missing* yaitu rentan terhadap serangan *MIME Snifing Attack* ataupun XSS. Berikut daftar perbaikan yang dapat dilakukan sebagai berikut:

Tabel 5. Langkah Perbaikan Kerentanan *X-Content-Type-Options Header Missing*

Langkah Perbaikan	Tujuan dan Penjelasan
Konfigurasikan Apache untuk menyisipkan header *X-Content-Type-Options*	Mencegah browser melakukan penebakan atau perubahan jenis konten secara otomatis
Pastikan Laravel tidak memodifikasi nilai *Content-Type* yang valid	Menjaga agar header respons yang dihasilkan tetap sesuai dan konsisten
Terapkan *middleware* Laravel sebagai opsi tambahan	Menyediakan mekanisme cadangan untuk menambahkan header pada tingkat aplikasi
Lakukan verifikasi menggunakan *DevTools* atau perintah curl	Memastikan bahwa *header* dikirim bersama setiap respons dari server

10. *Authentication Request Identified*

Kerentanan *Authentication Request Identified* rentan terhadap serangan *brute force.* Berikut langkah-langkah perbaikan yang dapat dilakukan sebagai berikut:

Tabel 6. Langkah Perbaikan Kerentanan *Authentication Request Identified*

Langkah Perbaikan	Tujuan dan Penjelasan
Gunakan protokol HTTPS secara menyeluruh	Mencegah intersepsi data sensitif seperti kredensial atau token oleh pihak ketiga di jaringan yang tidak aman
Hindari pengiriman informasi sensitif melalui URL	Pastikan parameter seperti *token* dan *email* tidak dikirim melalui query string, melainkan melalui *body POST*
Standarisasi pesan kesalahan saat login	Gunakan pesan error yang bersifat umum agar penyerang tidak dapat membedakan penyebab kegagalan login
Terapkan pembatasan laju permintaan *(rate limiting)*	Batasi jumlah percobaan login dalam periode tertentu untuk mencegah serangan *brute force*
Aktifkan proteksi CSRF dan atur cookie secara aman	Gunakan atribut *HttpOnly, Secure*, dan SameSite untuk menjaga keamanan sesi pengguna
Lakukan pencatatan dan pemantauan aktivitas login	Deteksi dan analisis aktivitas mencurigakan seperti percobaan login massal atau dari lokasi yang tidak biasa
Gunakan middleware untuk validasi dan sanitasi data masuk	Memastikan bahwa input pengguna pada endpoint otentikasi aman dan tidak mengandung muatan berbahaya

11. *Information Disclosure - Suspicious Comments*

Kerentanan *Information Disclosure* yaitu rentan terhadap serangan *targeted expoloitation,* berikut langkah-langkah untuk menangani kerentanan tersebut:

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

Tabel 7. Langkah Perbaikan Kerentanan *Authentication Request Identified*

Langkah Perbaikan	Tujuan dan Penjelasan
Hindari komentar bersifat internal di *view* atau JS	Untuk mencegah penyebaran informasi pengembangan yang tidak semestinya dilihat pengguna.
Lakukan proses minifikasi terhadap *file frontend*	Menghapus komentar dari file *JavaScript* dan CSS saat proses *build* untuk rilis produksi.
Gunakan tool build seperti *Laravel Mix* atau *Vite*	Mendukung proses otomatisasi penghapusan komentar dalam pipeline pengembangan.
Audit manual *file view* dan sumber *frontend*	Melakukan pemeriksaan terhadap seluruh file HTML, Blade, dan JS untuk menghapus komentar sensitif.
Terapkan pemeriksaan dengan alat pemindai kode	Menggunakan tools seperti ZAP atau SonarQube untuk mengidentifikasi komentar mencurigakan.
Nonaktifkan mode debug dan proteksi server	Menonaktifkan APP_DEBUG serta mencegah Apache mengungkapkan struktur direktori aplikasi.

12. *Re-examine Cache-control Directives*

Kerentanan *Re-examine Cache-control Directives* yaitu rentan terhadap serangan *session hijacking*. Berikut langkah-langkah perbaikan yang dapat dilakukan:

Tabel 8. Langkah Perbaikan Kerentanan *Authentication Request Identified*

Tindakan Perbaikan	Tujuan dan Penjelasan
Konfigurasikan header Cache-Control untuk halaman privat	Menghindari penyimpanan halaman sensitif oleh browser maupun proxy
Tambahkan header Pragma dan Expires	Memberikan dukungan tambahan terhadap browser lama yang belum mendukung cache modern
Gunakan middleware Laravel untuk mengatur header	Menjamin seluruh respons terkait informasi pribadi menyertakan header pengendali cache
Atur header secara langsung pada server Apache	Untuk file HTML atau PHP statis, server dapat mencegah cache secara langsung
Lakukan verifikasi dengan curl atau DevTools	Memastikan header yang diharapkan benar-benar diterapkan pada halaman tertentu

13. *Session Management Response Identified*

Kerentanan *Session Management Response Identified* yaitu rentan dari serangan *session hijacking.* Berikut langkah-langkah perbaikan yang dapat dilakukan:

Tabel 9. Langkah Perbaikan Kerentanan *Authentication Request Identified*

Tindakan Perbaikan	Tujuan
Kustomisasi Nama Cookie Sesi Laravel	Menyamarkan teknologi yang digunakan agar tidak mudah dikenali oleh penyerang.
Gunakan Atribut Keamanan pada Cookie	Meningkatkan proteksi terhadap akses cookie oleh skrip pihak ketiga dan mencegah penyalahgunaan lintas situs.
Hapus Header HTTP yang Mengungkap Informasi	Mencegah pengungkapan informasi framework dan server pada header respons HTTP.
Enkripsi Data Sesi	Menjaga integritas dan kerahasiaan data yang tersimpan di sisi klien.
Regenerasi ID Sesi Setelah Autentikasi	Menghindari reuse ID sesi lama yang dapat disalahgunakan oleh pihak lain.
Batasi Durasi Sesi Aktif (Timeout)	Mengurangi risiko akses tidak sah dengan memperpendek masa aktif sesi.

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pengujian keamanan terhadap sistem informasi pendidikan dan pelatihan PT Azure Samudera Karsa dengan menggunakan perangkat *penetration testing* ZAP, berhasil mengidentifikasi sejumlah kerentanan pada aplikasi tersebut. Pengujian ini dilaksanakan secara sistematis melalui pendekatan *vulnerability assessment*, yang bertujuan untuk mengevaluasi potensi kelemahan pada aspek keamanan aplikasi web. Hasil yang diperoleh menunjukkan bahwa terdapat beberapa titik rawan yang memungkinkan pihak tidak berwenang untuk mengeksploitasi sistem, baik untuk mengakses data sensitif maupun melakukan manipulasi informasi.

Peringatan dari hasil *scanning* dengan menggunakan ZAP dihasilkan 3 level peringatan yaitu: 3 peringatan pada level medium; 6 peringatan pada level low dan 3 pada level *informational* dengan total 13 peringatan. Temuan ini menjadi landasan dalam menyusun rekomendasi strategis bagi pengembang sistem, agar dilakukan penguatan terhadap mekanisme keamanan secara proaktif guna menghadapi potensi serangan siber di masa mendatang

DAFTAR RUJUKAN

[1] BSSN, “Laporan Bulanan Publik,” no. 70, pp. 01–20, 2023, [Online]. Available: www.idsirtii.or.id
[2] S. Nurul, Shynta Anggrainy, and Siska Aprelyani, “Faktor-Faktor Yang Mempengaruhi Keamanan Sistem Informasi: Keamanan Informasi, Teknologi Informasi Dan Network (Literature Review Sim),” J. Ekon. Manaj. Sist. Inf., vol. 3, no. 5, pp. 564–573, 2022, doi: 10.31933/jemsi.v3i5.992.
[3] B. Harahap, “Penerapan Keamanan Owasp Terhadap Aplikasi GTFW Pada Website Universitas Battuta,” J. Inform. dan Teknol. Pendidik., vol. 1, no. 2, pp. 80–86, 2021, doi: 10.25008/jitp.v1i2.15.
[4] S. Hidayatulloh and D. Saptadiaji, “Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP),” J. Algoritm., vol. 18, no. 1, pp. 77–86, 2021, doi: 10.33364/algoritma/v.18-1.827.
[5] H. Sofyan, M. Sugiarto, and B. M. Akbar, “Implementation of Penetration testing on Websites to Improve Security of Information Assets UPN ‘Veteran’ Yogyakarta,” Telematika, vol. 20, no. 2, p. 153, 2023, doi: 10.31315/telematika.v20i2.7757.
[6] A. W. Kuncoro and F. Rahma, “Analisis Metode Open Web Application Security Project (OWASP) pada Pengujian Keamanan Website: Literature Review,” Automata, vol. 3, no. 1, pp. 1–5, 2021, [Online]. Available: https://www.sciencedirect.com
[7] F. Al Fajar, “Analisis Keamanan Aplikasi Web Prodi Teknik Informatika Uika Menggunakan Acunetix Web Vulnerability,” Inova-Tif, vol. 3, no. 2, p. 110, 2020, doi: 10.32832/inova-tif.v3i2.4127.
[8] G. Guntoro, L. Costaner, and M. Musfawati, “Analisis Keamanan Web Server Open Journal System (Ojs) Menggunakan Metode Issaf Dan Owaspv(Studi Kasus Ojs Universitas Lancang Kuning),” JIPI (Jurnal Ilm. Penelit. Dan Pembelajaran Inform., vol. 5, no. 1, p. 45, 2020, doi: 10.29100/jipi.v5i1.1565.
[9] Y. Yudiana, A. Elanda, and R. L. Buana, “Analisis Kualitas Keamanan Sistem Informasi E-Office Berbasi Website Pada STMIK Rosma Dengan Menggunakan OWASP Top 10,” CESS (Journal Comput. Eng. Syst. Sci., vol. 6, no. 2, p. 185, 2021, doi: 10.24114/cess.v6i2.24777.
[10] Riyan Farismana and Dian Pramadhana, “Perbandingan Vulnerability Assesment Menggunakan Owasp Zap dan Acunetix Pada Sistem Informasi Repositori Politeknik Negeri Indramayu,” J. Tek. Inform. dan Teknol. Inf., vol. 3, no. 2, pp. 26–32, 2023, doi: 10.55606/jutiti.v3i2.2853.
[11] I Made Adi Surya Permana, I. G. P. K. . Juliharta, and I. G. J. E. . Putra, “Analisis Keamanan Sistem Informasi Menggunakan Metode Vulnerability Assesment pada Aplikasi Web

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025

All author: Soni Ayi Purnama

Karangasem.go.id", remik, vol. 9, no. 2, pp. 466-473, Apr. 2025.
[12] Educational And Training Information System At PT. Azure Samudera Karsa, *JTIF*, vol. 1, no. 2, pp. 86–95, Nov. 2024, doi: 10.71251/jtif.v1i2a2.
[13] Pelatihan Penggunaan Sistem Informasi Pendidikan Dan Pelatihan Pada PT Azure Samudera Karsa, *ngabdimas*, vol. 8, no. 01 Juni, pp. 14–17, Jun. 2025, doi: 10.36050/1jassv94.

Received: 29-07-2025 | Accepted: 07-08-2025 | Published Online: 30-08-2025