JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 12 No. 3S1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v12i3S1.5236

ANALISIS KEAMANAN *WEBSITE* PEMERINTAH DESA CURUG MENGGUNAKAN *OPEN WEB APPLICATION SECURITY PROJECT* (OWASP)

Syahrul Dwi Hilda[1*], Nono Heryana[2], Azhari Ali Ridha[3]

[1,2,3]Universitas Singaperbangsa Karawang; Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361; (0267) 641177

Received: 22 Agustus 2024
Accepted: 5 Oktober 2024
Published: 12 Oktober 2024

Keywords:
vulnerabilities, OWASP, Curug Village Government.

Corespondent Email:
2010631250076@student.unsika.ac.id

Abstrak. Keamanan siber telah menjadi isu krusial di era digital saat ini, terutama bagi situs-situs pemerintah yang sering menjadi target serangan. Menurut Badan Siber dan Sandi Negara (BSSN), situs pemerintah rentan diretas. Tujuan penelitian ini bertujuan untuk menganalisis keamanan *website* Pemerintah Desa Curug menggunakan *Open Web Application Security Project* (OWASP). Analisis dilakukan terhadap sepuluh kategori utama kerentanan keamanan aplikasi web yang terdaftar dalam OWASP *Top* 10 2021, termasuk *Broken Access Control, Cryptographic Failures, Injection, Insecure Design, Security Misconfiguration, Vulnerable and Outdated Components, Identification and Authentication Failures, Software and Data Integrity Failures, Security Logging and Monitoring Failures, serta Server-side Request Forgery.* Hasil dari pengujian yang dilakukan yaitu 4 dari 8 kerentanan tersebut termasuk ke dalam daftar OWASP *Top* 10 tahun 2021 yaitu pada kerentanan *Injection* dan *Security Misconfiguration.* Rekomendasi perbaikan diberikan berdasarkan temuan-temuan tersebut, yang diharapkan dapat membantu Pemerintah Desa Curug dalam memperkuat keamanan siber mereka.

Abstract. *Cybersecurity has become a crucial issue in the current digital era, especially for government websites that are often targeted by attacks. According to the National Cyber and Crypto Agency (BSSN), government websites are vulnerable to hacking. This study aims to analyze the security of the Curug Village Government website using the Open Web Application Security Project (OWASP). The analysis was conducted on the ten main categories of web application security vulnerabilities listed in OWASP Top 10 2021, including Broken Access Control, Cryptographic Failures, Injection, Insecure Design, Security Misconfiguration, Vulnerable and Outdated Components, Identification and Authentication Failures, Software and Data Integrity Failures, Security Logging and Monitoring Failures, and Server-side Request Forgery. The results of the testing showed that 4 out of 8 vulnerabilities fall into the OWASP TOP 10 for 2021, particularly in the categories of Injection and Security Misconfiguration. Recommendations for improvements are provided based on these findings, which are expected to help the Curug Village Government strengthen their cybersecurity.*

1. PENDAHULUAN

Pemanfaatan teknologi informasi dan komunikasi saat ini telah menyebabkan terjadinya perubahan mendasar dalam berbagai sektor kehidupan sosial. Seiring perkembangannya, penggunaan teknologi informasi ini juga telah menimbulkan berbagai macam masalah dan ancaman terutama terkait isu keamanan data dan informasi akibat tindak kejahatan dunia maya [1].

Salah satu sektor yang saat ini mulai gencar memanfaatkan teknologi informasi adalah

pemerintah di level desa. Pemerintah Desa Curug contohnya telah mengembangkan situs desa yang bertujuan untuk mendukung layanan masyarakat dan tata kelola pemerintahan desa yang lebih baik dan transparan. Namun demikian, website desa yang terkoneksi ke internet ini menjadi rentan terhadap serangan siber yang dapat menyebabkan pencurian dan kerusakan data [2].

Keamanan siber telah menjadi isu krusial di era digital saat ini, terutama bagi situs-situs pemerintah yang sering menjadi target serangan. Menurut Badan Siber dan Sandi Negara (BSSN), situs pemerintah rentan diretas akibat berbagai faktor, termasuk kelemahan dalam sistem keamanan, kurangnya pembaruan rutin, serta kurangnya sumber daya manusia yang terlatih di bidang keamanan siber, tercatat bahwa sepanjang tahun 2021 terdapat 5940 kasus web defacement, dan 1574 kasus terjadi pada sektor pemerintahan diantaranya 1097 kasus pada situs pemerintah daerah dan 477 kasus pada situs pemerintah pusat[3]. Selain itu, ancaman serangan siber terus berkembang dengan teknik yang semakin kompleks dan canggih, membuat perlindungan terhadap data dan informasi publik menjadi sebuah tantangan yang semakin besar bagi para pihak terkait.

Tingkat keamanan pada setiap sistem informasi memiliki tingkat yang berbeda-beda antara satu organisasi dengan organisasi lainnya, hal ini bertujuan untuk mencegah, mendeteksi kerusakan pada sistem, dan terhindar dari serangan siber. Keamanan sistem informasi dapat dicapai dengan melakukan sebuah pengujian yang berkaitan dengan keamanan siber, contohnya adalah penetration testing. Penetration testing adalah suatu kegiatan untuk melakukan eksploitasi sistem dengan cara legal yang bertujuan untuk mendapatkan akses ke data sensitif. Seorang penguji harus memiliki sebuah keahlian khusus dan izin yang sudah disetujui institusi terkait sebelum melakukan penetration testing[4].

Salah satu kerangka analisis keamanan web yang sering digunakan adalah OWASP (Open Web Application Security Project). OWASP menyediakan kerangka dan rekomendasi praktik terbaik terkait keamanan aplikasi web dan menyusun 10 jenis kerentanan keamanan web teratas yang perlu menjadi perhatian utama. Beberapa peneliti terdahulu telah menerapkan panduan analisis OWASP untuk mengaudit keamanan website lembaga pemerintahan dan sukses mendeteksi berbagai bug dan kerentanan berisiko. Namun implementasi OWASP khusus untuk keamanan website desa masih minim dilakukan .

Mengacu pada fenomena tersebut, penulis tertarik untuk melakukan analisis tingkat keamanan website Pemerintah Desa Curug dengan menerapkan kerangka kerja OWASP Top 10 versi 2021. Dengan pengujian berbasis standar dan metodologi yang terdokumentasi, diharapkan dapat mendapatkan profil keamanan website desa serta kajian celah sistem yang mendalam sekaligus rekomendasi perbaikannya. Sehingga pengelola website dapat segera meremediasi risiko keamanan pada website untuk mencegah potensi eksploitasi di masa mendatang .

2. TINJAUAN PUSTAKA

2.1 Information Gathering

Information Gathering adalah proses mengumpulkan data yang relevan untuk mengetahui lebih lanjut tentang sasaran, sistem, dan jaringan yang menjadi target serangan siber[5]. Information Gathering adalah langkah penting dalam proses cybersecurity karena memberikan dasar yang kuat untuk merancang dan melaksanakan serangan siber yang efektif. Berikut merupakan metode dari Information Gathering :

1. Passive Information Gathering: teknik ini melibatkan pengumpulan informasi tanpa berinteraksi langsung dengan target. Contohnya adalah menggunakan mesin pencari, situs web publik, dan jaringan sosial untuk mengumpulkan data.
2. Active Information Gathering: Berbeda dengan metode pasif, metode ini melibatkan interaksi langsung dengan target, seperti melakukan pemindaian port atau pengecekan keamanan jaringan. Meskipun lebih berisiko terdeteksi, metode ini dapat memberikan informasi yang lebih detail.

2.2 Vulnerability Analysis

Setiap sistem informasi, termasuk aplikasi web, pada dasarnya memiliki kerentanan atau vulnerability yang menjadi celah bagi penyerang untuk melakukan aksi eksploitasi. Beberapa kasus keberhasilan hacking dan peretasan situs pemerintah

seringkali bermula dari adanya vulnerability di dalam sistem yang tidak tertutup dengan baik . Vulnerability adalah kelemahan desain, implementasi, atau konfigurasi sistem yang dapat dimanfaatkan untuk mengakses, memodifikasi, hingga merusak data atau fungsi sistem secara tidak sah. Jenis vulnerability yang umum ditemukan seperti celah injeksi (injection flaws), kontrol autentikasi yang lemah, konfigurasi keamanan yang tidak tepat, hingga keberadaan komponen sistem yang rentan[6].

Vulnerability Analysis adalah proses identifikasi, klasifikasi, dan penilaian kelemahan atau celah keamanan dalam sistem komputer, jaringan, atau perangkat lunak yang dapat dieksploitasi oleh ancaman[5] . Hal ini dibagi menjadi beberapa teknik sebagai berikut :

1. Static Analysis

Merupakan teknik pengujian keamanan di mana penguji tidak menjalankan skenario pengujian secara langsung. Dalam pendekatan ini, penguji melakukan pemeriksaan terhadap komponen sistem yang diuji, seperti struktur kode sumber, dokumentasi sistem, dan aset-aset lainnya. Pengujian dengan teknik ini tidak melibatkan tindakan eksploitasi, sehingga tidak memberikan dampak pada sistem yang sedang diuji. Meskipun demikian, kelemahan utama dari teknik ini adalah proses yang memakan waktu lama dan tingginya potensi terjadinya human error.

2. Manual Testing

Adalah teknik pengujian keamanan yang dilakukan secara manual oleh penguji tanpa melibatkan bantuan alat atau perangkat lunak khusus. Dalam pendekatan ini, penguji mengandalkan pengetahuan dan pengalaman mereka untuk mengidentifikasi area-area yang rentan pada aplikasi website. Meski dapat dilakukan secara menyeluruh, teknik Manual Testing membutuhkan banyak waktu dan upaya dari penguji.

3. Fuzz Testing

Adalah teknik pengujian di mana penguji memasukkan data acak yang tidak valid ke dalam website. Hasil input data acak tersebut kemudian dianalisis untuk menentukan apakah terdapat kesalahan atau kerentanan pada sistem yang dapat diidentifikasi dan dieksplorasi lebih lanjut.

3. Automated Testing

Adalah teknik pengujian keamanan yang dilakukan dengan memanfaatkan bantuan perangkat lunak atau alat otomatis. Perangkat lunak tersebut membantu penguji dalam mengidentifikasi kerentanan keamanan pada sistem aplikasi website secara lebih cepat dan efisien dibandingkan dengan pengujian manual.

2.3 Penetration Testing

Penetration Testing menawarkan banyak keuntungan bagi organisasi. Pertama, membantu mengidentifikasi kerentanan dan kelemahan dalam sistem dan jaringan organisasi. Dengan mensimulasikan serangan siber di dunia nyata, pengujian penetrasi memungkinkan bisnis untuk mendeteksi dan mengatasi secara proaktif potensi kelemahan keamanan sebelum peretas jahat dapat mengeksploitasinya. Pendekatan proaktif ini tidak hanya melindungi data sensitif tetapi juga melindungi data sensitif mencegah potensi kerugian finansial dan kerusakan reputasi. Selain itu, pengujian penetrasi memberikan wawasan berharga tentang efektivitas kontrol keamanan suatu organisasi, sehingga memungkinkan mereka untuk menyempurnakan kontrol keamanan mereka mekanisme pertahanan dan meningkatkan postur keamanan mereka secara keseluruhan [6].

Penetration Testing dibagi menjadi beberapa beberapa kategori. Berdasarkan hak akses atau informasi terhadap website atau aplikasi target, hal tersebut dibagi sebagai berikut :

1. Black Box Testing

Teknik ini memanfaatkan keahlian dan perspektif penguji untuk melakukan serangkaian serangan pada suatu sistem aplikasi. Dalam skenario pengujian ini, penguji bertindak layaknya peretas yang menyerang dari dalam maupun luar jaringan sistem. Penguji tidak memiliki akses atau informasi mengenai topologi jaringan, konfigurasi sistem, ataupun detail internal lainnya. Black Box Testing bisa dilakukan di area eksternal maupun internal sistem[7].

2. White Box Testing

Berbeda dengan Black Box, teknik White Box Testing mengharuskan penguji memiliki akses informasi mengenai sistem yang diuji. Informasi tersebut meliputi infrastruktur, arsitektur jaringan, kode program, dan detail teknis lainnya. Dengan informasi tersebut, penguji dapat melakukan pengujian di berbagai area sistem secara lebih menyeluruh.

3. Grey Box Testing

Pendekatan ini merupakan kombinasi dari dua teknik di atas. Dalam Grey Box Testing, penguji memperoleh informasi terbatas tentang sistem yang diuji dan umumnya hanya memiliki akses sebagai pengguna normal. Dengan cakupan informasi dan akses yang tidak terlalu terbatas namun juga tidak lengkap, penguji dapat melakukan pengujian dengan perspektif yang lebih luas.

2.4 Reporting

Reporting adalah proses dokumentasi dan pelaporan hasil dari analisis keamanan siber, termasuk penemuan kelemahan, hasil uji penetrasi, dan rekomendasi mitigasi. Tujuan dari reporting adalah untuk memberikan informasi yang jelas dan komprehensif kepada pemangku kepentingan tentang status keamanan sistem dan langkah-langkah yang perlu diambil untuk meningkatkan keamanan.
2.5 Open Web Application Security Project (OWASP)

Melihat maraknya insiden keamanan siber yang disebabkan oleh vulnerability pada aplikasi web, OWASP (Open Web Application Security Project) hadir sebagai organisasi nirlaba yang membantu komunitas global dalam mengamankan sistem mereka. OWASP memiliki misi untuk meningkatkan keamanan perangkat lunak dengan menyediakan berbagai tools, dokumen, dan standar keamanan yang dapat diakses secara gratis dan terbuka [8].

Salah satu kontribusi utama OWASP adalah daftar OWASP Top 10 yang merangkum sepuluh jenis kerentanan website paling kritis yang harus menjadi fokus perhatian pengembang dan administrator website. Daftar ini terus diperbarui oleh OWASP untuk mengikuti dinamika ancaman keamanan di dunia maya. Pada versi 2021, OWASP Top 10 mencakup kerentanan seperti Broken Access Control, Cryptographic Failures, Injection, Insecure Design, hingga Server-Side Request Forgery.

Berikut daftar 10 jenis kerentanan website yang telah di klasifikasikan oleh OWASP pada tahun 2021 :

1. Broken Access Control

Broken Access Control merupakan keadaan dimana sistem kontrol yang tidak memerlukan proses otorisasi untuk mengakses informasi, yang dapat menyebabkan bocornya informasi yang sensitif[9].

2. Cryptographic Failures

Cryptographic Failures terjadi ketika teknik enkripsi tidak digunakan atau digunakan secara tidak benar[10]. Beberapa alasan untuk Cryptographic Failures termasuk pada :

a. Informasi sensitif dapat dikirimkan dalam teks biasa melalui jaringan atau disimpan dalam database atau file dalam teks biasa.
b. Penggunaan algoritma enkripsi yang sudah tua atau lemah
c. Mismanagement pada kunci enkripsi

3. Injection

Celah injeksi, seperti SQL, dan LDAP injection terjadi ketika data yang berbahaya dikirim ke interpreter sebagai bagian dari perintah atau query. Data berbahaya milik penyerang dapat mengelabui interpreter untuk mengeksekusi perintah yang tidak diinginkan atau mengakses data secara ilegal[6].

4. Insecure Design

Desain yang tidak aman adalah sebuah representasi kategori yang luas dari banyak kelemahan yang berbeda, yang diekspresikan sebagai "desain kontrol yang tidak ada atau kurang efisien"[11].

5. Security Misconfiguration

Kelemahan ini dapat dimanfaatkan oleh penyerang untuk mendapatkan akses yang tidak sah atau bahkan membahayakan sistem. Kesalahan konfigurasi keamanan terkadang dideteksi oleh pemindai otomatis. Contoh skenario serangan yaitu ditemukannya list directory yang terbuka [12].

6. Vulnerable and Outdated Components

Peretas memiliki kemampuan untuk memasuki dan mengedit kode. Ini mungkin disebabkan oleh komponen pihak ketiga dan ketergantungan yang tidak aman. Serangan jenis ini dapat diatasi dari dalam sistem melalui analisis komposisi perangkat lunak. Analisis memungkinkan audit atau pemrogram untuk menemukan bagian yang tidak aman sebelum sistem merilis website atau aplikasi[12].

7. Identification and Authencation Failures

Kesalahan dalam proses identifikasi dan otentikasi pengguna dapat memungkinkan penyerang untuk mendapatkan akses tidak sah ke sistem atau data. Contoh kesalahan ini termasuk penggunaan kata sandi yang lemah, kurangnya verifikasi identitas pengguna yang cukup kuat, atau kegagalan dalam melindungi otentikasi dari serangan brute force. Penyerang dapat mengeksploitasi kelemahan ini untuk mencuri kredensial pengguna, mengakses data sensitif, atau mendapatkan akses administratif sistem[12].

8. Software and Data Integrity Failures

Kesalahan ini dapat terjadi dalam berbagai bentuk, terutama seiring perkembangan website, penggunaan kode dan layanan pihak ketiga di website semakin umum. Contoh kegagalan diantaranya adalah Penggunaan kode yang tidak menjamin integritas sumber, dan menggunakan plugin pihak ketiga tanpa mengontrol sumbernya [12].

9. Security Logging and Monitoring Failures

Kerentanan ini terjadi pada aplikasi website ketika pengelola mengabaikan hasil yang terdapat dalam log, seperti kegagalan login, kegagalan transaksi, dan peringatan kesalahan yang tidak jelas. Hal ini memungkinkan penyerang memanfaatkannya untuk melakukan berbagai suntikan ke dalam sistem keamanan [13].

10. Server-side Request Forgery

Kerentanan SSRF muncul saat sebuah aplikasi website meminta remote resource tanpa melakukan validasi URL yang diberikan oleh pengguna. Ini memperbolehkan penyerang untuk memaksa aplikasi untuk mengirim crafted request ke destinasi yang tidak diharapkan, meskipun sudah dilindungi oleh firewall, VPN, atau tipe lain dari Access Control List [12].

2.6 Nmap (Network Mapper)

Nmap adalah alat sumber terbuka yang populer untuk melakukan pemindaian jaringan. Alat ini membantu pengguna untuk menemukan perangkat yang terhubung ke jaringan serta menganalisis layanan yang dijalankan pada perangkat tersebut.Pengguna Nmap dapat melakukan pemindaian jaringan secara menyeluruh, menemukan perangkat yang terhubung ke jaringan termasuk komputer, printer, router, dan perangkat lainnya. Alat ini juga memungkinkan pengguna untuk melakukan pemindaian port, yang membantu dalam mengetahui layanan yang berjalan pada perangkat, serta deteksi sistem operasi dari perangkat tersebut.

2.7 WhatWeb

WhatWeb adalah alat open source yang digunakan untuk mengidentifikasi teknologi dan platform yang digunakan dalam sebuah situs melalui analisis respons HTTP dari server website. Dengan kemampuan pemindaian otomatis, WhatWeb memungkinkan pengguna untuk dengan cepat mendapatkan informasi tentang teknologi backend sebuah situs, termasuk sistem manajemen konten (CMS) dan bahasa pemrograman yang digunakan.

2.8 Whois

Whois merupakan suatu prosedur untuk mendapatkan informasi mengenai sebuah domain, alamat, nomor telepon, alamat surel, tanggal domain di daftarkan dan tanggal domain akan kadaluarsa.

2.9 Nikto

Nikto adalah alat open source yang bertujuan untuk melakukan pemindaian keamanan pada aplikasi website[12] Dikembangkan oleh Sullo (Chris Solo), alat ini dirancang untuk mengidentifikasi dan mengeksploitasi kerentanan keamanan yang ada dalam aplikasi website. Nikto secara otomatis melakukan pemindaian terhadap server website dan menemukan berbagai jenis kerentanan keamanan, seperti injeksi SQL, Cross-Site Scripting (XSS), kerentanan perangkat lunak yang kedaluwarsa, dan lainnya.

Dengan fleksibilitas konfigurasinya, Nikto memungkinkan pengguna untuk menyesuaikan parameter pemindaian sesuai kebutuhan, seperti batas waktu dan tingkat kebisingan. Setelah selesai pemindaian, Nikto menghasilkan laporan yang merinci hasil pemindaian, termasuk daftar kerentanan yang ditemukan dan rekomendasi untuk memperbaikinya.

2.10 Owasp ZAP (Zed Attack Proxy)

Untuk mengidentifikasi adanya vulnerability dan mengukur tingkat keamanan sebuah sistem, diperlukan penetration testing yang terstruktur. Pengujian keamanan tidak hanya berupa serangan simulasi dari pihak eksternal, tetapi juga mencakup proses analisis mendalam terhadap arsitektur, desain, kode sumber, hingga konfigurasi sistem untuk menemukan celah-celah kerentanan yang ada.

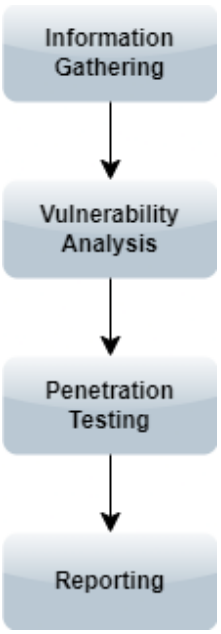
OWASP sendiri menyediakan metodologi pengujian keamanan aplikasi website yang meliputi tahapan pemodelan ancaman, pemetaan permukaan serangan, pengujian terpadu, review kode sumber, hingga analisis vulnerability. Dengan mengikuti panduan OWASP, proses pengujian keamanan website akan lebih komprehensif dan sistematis sehingga mampu mengungkap berbagai celah keamanan kritikal yang mungkin sebelumnya terlewat.

2.11 Burpsuite

Burpsuite adalah platform yang digunakan untuk melakukan pengujian keamanan aplikasi website. Dia bekerja secara simultan dengan mendukung keseluruhan proses pengujian dari tahap pemetaan awal hingga tahap pemetaan analisis kerentanan lengkap aplikasi website[14].

3. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah Penetration Testing, dengan alur penelitian yang sebenarnya terdiri dari lima tahapan yaitu Planning, Information Gathering, Vulnerability Analysis, Penetration Testing, dan Reporting [15], tetapi penulis tidak memasukkan tahap planning, dikarenakan planning seharusnya ada pada awal dari setiap tahap, berikut tahapan yang telah dirubah : Information Gathering, Vulnerability Analysis, Penetration Testing, dan Reporting. Berikut merupakan gambar alur dari rancangan penelitian yang dilakukan ditunjukkan pada Gambar berikut :



Gambar 1. *Alur Penelitian*

Metode OWASP Top 10 Tahun 2021 akan diterapkan dalam pengujian situs Pemerintah Desa Curug, mengikuti tahapan yang ditetapkan dalam metode tersebut. Alur penelitian akan diuraikan sebagai berikut:

3.1 Information Gathering

Pada tahap ini, semua informasi terkait website Pemerintah Desa Curug dikumpulkan. Dilakukan pemindaian untuk menghimpun data tentang domain, server, alamat IP, host, dan firewall.

3.2 Vulnerability Analysis

Pada tahap ini, pencarian celah keamanan dilakukan baik secara manual maupun otomatis, bergantung pada alat yang digunakan untuk prosesnya.

3.3 Penetration Testing

Setelah menemukan kerentanan, langkah berikutnya adalah mengeksploitasinya melalui serangan eksperimen. Ini melibatkan penentuan target dan pemilihan alat yang sesuai. Pada tahap ini, sering digunakan serangan website seperti cross-site scripting

(XSS) dan SQL Injection untuk mengeksplorasi kerentanan pada target.

3.4 Reporting

Reporting merupakan tahap yang cukup penting, agar institusi atau pihak yang di uji kerentanan situs web nya memahami hasil dari uji kerentanan tersebut. Pada tahap ini, akan disusun sebuah laporan yang mencakup langkah-langkah yang telah dilakukan serta kerentanan keamanan yang teridentifikasi, dengan mengacu pada parameter keamanan OWASP Top 10 Tahun 2021.

4. HASIL DAN PEMBAHASAN

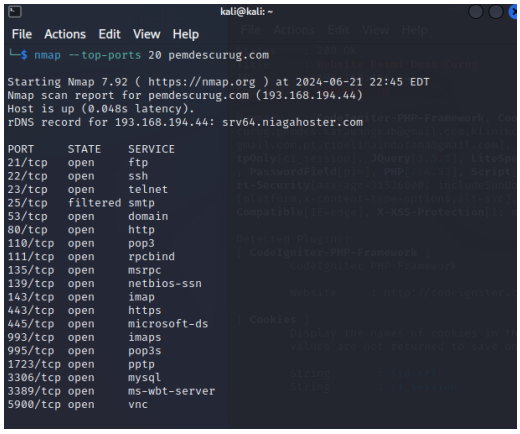
Proses yang dilakukan untuk menemukan celah keamanan pada website meliputi planning, information gathering, vulnerability analysis, penetration testing, dan reporting. Pada proses vulnerability analysis menggunakan tools OWASP ZAP (Zed Attack Proxy) dan Nikto dalam menemukan celah keamanan pada website pemdescurug.com. Pada proses penetration testing dilakukan berdasarkan kerentanan yang ditemukan pada tahap vulnerability analysis. Hasil dari penelitian ini adalah reporting atau hasil yang ditemukan pada saat pengujian penetration testing berdasarkan kerentanan yang termasuk dalam OWASP TOP 10 tahun 2021.

3.1 Information Gathering

Setelah menetukan rancangan penelitian, tahap pertama yang dilakukan adalah information gathering, pada tahap ini penulis akan mengumpulkan informasi mengenai website pemdescurug.com menggunakan beberapa tools yang telah ditentukan sebelumnya. Adapun informasi penting yang didapatkan dengan menggunakan tools nya adalah sebagai berikut:

a. Nmap

Dalam penelitian ini, Nmap digunakan untuk memindai port terbuka pada website pemdescurug.com guna mengidentifikasi layanan yang berjalan dan potensi kerentanan, Dengan menuliskan perintah “nmap --top-ports 20 pemdescurug.com”.



Gambar 2. *Pengujian dengan tools NMAP*

Informasi yang di dapatkan setelah melakukan pemindaian menggunakan nmap yang telah di ringkas yaitu sebagai berikut :

- Internet Protocol (IP) Address

IP yang digunakan oleh pemdescurug.com yaitu 193.168.194.44

- Port yang terbuka dan Service yang berjalan

Berikut merupakan ringkasan dari beberapa port yang terbuka pada website pemdescurug.com :

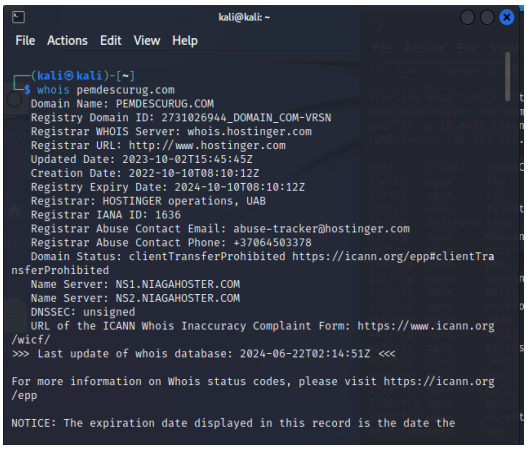
Tabel 1. *Port yang Terbuka*

No	*Port/Protocol*	*State*	*Service*
1.	21/tcp	*open*	ftp
2.	22/tcp	*open*	ssh
3.	23/tcp	*open*	telnet
4.	25/tcp	*filtered*	smtp
5.	53/tcp	*open*	*domain*
6.	80/tcp	*open*	http
7.	110/tcp	*open*	pop3
8.	135/tcp	*open*	msrpc
9.	139/tcp	*open*	netbios-ssn
10.	143/tcp	*open*	imap
11.	443/tcp	*open*	https
12.	445/tcp	*open*	microsoft-ds

No	*Port/Protocol*	*State*	*Service*
13.	993/tcp	*open*	imaps
14.	995/tcp	*open*	pop3s
15.	1723/tcp	*open*	pptp
16.	3306/tcp	*open*	mysql
17.	3389/tcp	*open*	ms-wbt-server
18.	5900/tcp	*open*	vnc

b. Whois

Setelah melakukan scanning port yang terbuka, kemudian menggunakan tools whois untuk menampilkan informasi terkait kepemilikan domain atau alamat IP. Dengan menuliskan perintah "whois pemdescurug.com", sebagai berikut :



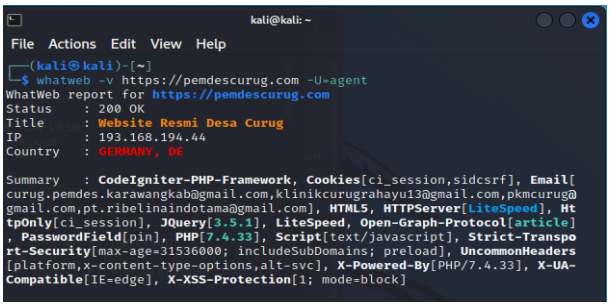
Gambar 3. *Pengujian dengan Tools Whois*

Adapun informasi penting yang didapat pada gambar diatas yaitu sebagai berikut :

- Domain Name: PEMDESCURUG.COM
- Registry Domain ID: 2731026944_DOMAIN_COM-VRSN
- Registrar WHOIS Server: whois.hostinger.com
- Registrar URL: http://www.hostinger.com
- Updated Date: 2023-10-02T15:45:45Z
- Creation Date: 2022-10-10T08:10:12Z
- Registry Expiry Date: 2024-10-10T08:10:12Z
- Registrar: HOSTINGER operations, UAB
- Registrar IANA ID: 1636
- Registrar Abuse Contact Email: abuse-tracker@hostinger.com
- Registrar Abuse Contact Phone: +37064503378
- Domain Status: clientTransferProhibited
- Name Server: NS1.NIAGAHOSTER.COM dan NS2.NIAGAHOSTER.CO
- DNSSEC: unsigned
- URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
- Last update of whois database: 2024-06-22T02:14:51

c. Whatweb

Tools information gathering lainnya yaitu Whatweb, yang berfungsi untuk melakukan pengenalan otomatis terhadap teknologi yang digunakan dalam sebuah situs. Dengan menggunakan perintah "whatweb –v https:pemdescurug.com –U=agent", seperti pada gambar berikut :



Gambar 4. *Pengujian dengan Tools Whatweb*

Hasil scanning pada gambar diatas menggunakan tools whatweb terhadap domain pemdescurug.com menunjukkan beberapa informasi penting tentang website tersebut. Website merespons dengan status 200 OK, yang berarti server berhasil merespons permintaan. Judul halaman adalah "Website Resmi Desa Curug", mengindikasikan bahwa ini adalah situs resmi dari sebuah desa Curug.

Website ini di-hosting pada IP 193.168.194.44 yang berlokasi di Jerman. Hal ini menarik mengingat situs ini adalah untuk desa di Indonesia, namun menggunakan hosting di luar negeri.

Situs ini dibangun menggunakan CodeIgniter PHP Framework, sebuah kerangka kerja populer untuk pengembangan website.

Beberapa teknologi lain yang terdeteksi meliputi HTML5, jQuery versi 3.5.1, dan PHP versi 7.4.33. Server website yang digunakan adalah LiteSpeed, untuk engine pada website pemdescurug.com menggunakan OpenSID versi 22.12.02.

Beberapa fitur keamanan juga terdeteksi, seperti Strict-Transport-Security yang memaksa koneksi HTTPS, dan X-XSS-Protection untuk melindungi dari serangan cross-site scripting. Website ini juga menggunakan beberapa cookie, termasuk untuk sesi pengguna.

Terdapat beberapa alamat email yang terkait dengan situs, yang mungkin digunakan untuk kontak atau administrasi. Penggunaan Open Graph Protocol menunjukkan bahwa situs ini dioptimalkan untuk berbagi di media sosial.

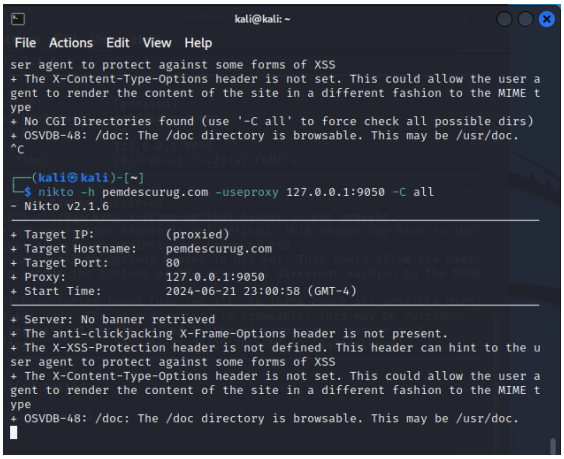
Secara keseluruhan, hasil scanning ini memberikan gambaran bahwa website Desa Curug dibangun dengan memperhatikan aspek performa dan keamanan, menggunakan teknologi web modern, meskipun lokasi hosting-nya mungkin tidak biasa untuk situs pemerintahan desa di Indonesia.

3.2 Vulnerability Analysis

Setelah melakukan tahap information gathering, tahap vulnerability analysis diperlukan untuk menemukan potensi atau adanya celah keamanan pada website pemdescurug.com, berikut hasil scanning dari tools vulnerability analysis yang digunakan yaitu Nikto dan Owasp ZAP sebagai berikut :

a. Nikto

Menggunakan Nikto yang berfungsi sebagai alat pemindaian keamanan sumber terbuka yang digunakan untuk mengidentifikasi potensi kerentanan di server website. Berikut merupakan gambar dari hasil scanning vulnerability analysis menggunakan Nikto dengan perintah “nikto –h pemdescurug.com –useproxy 127.0.0.1:9050 -C all” :



Gambar 5. *Pengujian dengan Tools Nikto*

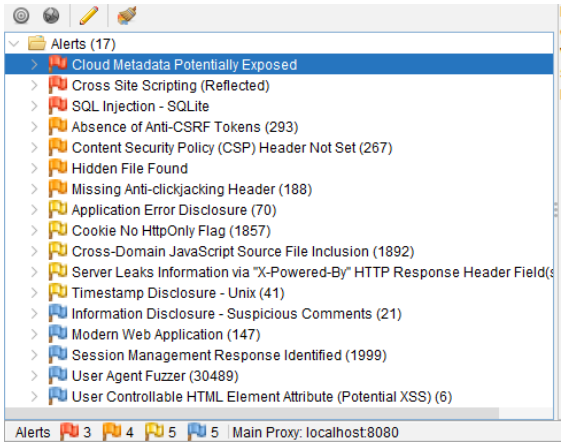
Pada Gambar tersebut menunjukan Nikto tidak berhasil mendapatkan informasi tentang banner server. Banner server biasanya memberikan informasi tentang jenis dan versi perangkat lunak server web yang sedang digunakan. Selain itu, header X-Frame-Options yang digunakan untuk melindungi terhadap serangan clickjacking dengan mengontrol apakah halaman dapat di-embed dalam frame dari domain lain tidak ada..

Lebih lanjut, header X-XSS-Protection tidak didefinisikan. Header ini menginstruksikan browser untuk mengaktifkan filter XSS.. Header X-Content-Type-Options juga tidak diatur. Header ini mencegah browser dari menafsirkan MIME type dari konten yang berbeda dengan apa yang didefinisikan oleh server, yang dapat mencegah beberapa serangan berbasis MIME.

Terakhir, ditemukan bahwa direktori /doc dapat di-browse, yang berarti direktori ini terbuka untuk diakses oleh siapa saja dan mungkin berisi dokumentasi atau file lain yang sensitif.

b. Owasp ZAP

Tools vulnerability selanjutnya yang digunakan yaitu Owasp ZAP, yang berfungsi untuk menemukan kelemahan dalam aplikasi web secara otomatis. Berikut merupakan hasil dari scanning website pemdescurug.com menggunakan OWASP ZAP :



Gambar 6. *Hasil Pengujian dengan Tools OWASP ZAP*

Pada Gambar tersebut menunjukan terdapat 17 potensi kerentanan yang ditemukan yang kemudian diurutkan berdasarkan tingkat risikonya yaitu sebagai berikut :

Tabel 2. *Potensi Kerentanan dari OWASP ZAP*

No	Nama Celah Keamanan	Risiko	Deskripsi	Dampak
1.	*Cloud Metadata Potentially Exposed*	*High*	Potensi terbukanya *metadata cloud* akibat konfigurasi NGINX yang salah.	Informasi sensitif *cloud* bisa diekspos, memungkinkan penyerang untuk mengambil alih sistem sepenuhnya.
2.	*Cross Site Scripting (Reflected)*	*High*	Kerentanan yang memungkinkan skrip berbahaya dijalankan di *browser* pengguna.	Penyerang bisa mencuri data pengguna atau sesi.
3.	*SQL Injection - SQLite*	*High*	Kerentanan yang memungkinkan perintah SQL berbahaya dijalankan pada *database* SQLite.	Pencurian, manipulasi data, atau pengambil alihan *server*.
4.	*Absence of Anti-CSRF Tokens*	*Medium*	Tidak adanya *token Anti-CSRF*, membuat aplikasi rentan terhadap serangan CSRF.	Pengguna bisa melakukan tindakan tidak diinginkan tanpa sepengetahuan mereka.
5.	*Content Security Policy (CSP) Header Not Set*	*Medium*	Tidak adanya *header CSP*, membuat aplikasi rentan terhadap serangan XSS dan injeksi data.	Membuka *website* rentan terhadap berbagai serangan injeksi skrip.
6.	*Hidden File Found*	*Medium*	Ditemukan *file* tersembunyi yang mungkin mengandung informasi sensitif atau	Penyerang bisa menggunakan *file* ini untuk mengumpulkan informasi dan

No	Nama Celah Keamanan	Risiko	Deskripsi	Dampak
			konfigurasi penting.	merencanakan serangan.
7.	*Missing Anti-clickjacking Header*	*Medium*	Tidak adanya *header Anti-clickjacking* yang melindungi aplikasi dari serangan *clickjacking*.	Pengguna bisa tanpa sadar mengklik elemen berbahaya.
8.	*Application Error Disclosure*	*Low*	Pengungkapan pesan kesalahan aplikasi yang memberikan informasi kepada penyerang.	Penyerang bisa menggunakan informasi ini untuk merencanakan serangan lebih lanjut.
9.	*Cookie No HttpOnly Flag*	*Low*	*Cookie* tidak dilindungi dengan *flag HttpOnly*.	*Cookie* bisa dicuri melalui skrip berbahaya.
10.	*Cross-Domain JavaScript Source File Inclusion*	*Low*	Penyertaan *file JavaScript* dari *domain* lain yang dapat membahayakan keamanan aplikasi.	*File* dari sumber tidak dipercaya bisa mengandung kode berbahaya.
11.	*Server Leaks Information via "X-Powered-By" HTTP Header Field(s)*	*Low*	Pengungkapan informasi *server* melalui *header X-Powered-By*.	Memberikan petunjuk kepada penyerang tentang teknologi yang digunakan, yang bisa dieksploitasi.
12.	*Timestamp Disclosure - Unix*	*Low*	Pengungkapan *timestamp Unix* yang bisa digunakan untuk mengumpulkan informasi tentang sistem.	Penyerang bisa menggunakan informasi ini untuk merencanakan serangan.
13.	*Information Disclosure - Suspicious Comments*	*Informational*	Ditemukan komentar mencurigakan dalam kode sumber yang mungkin mengandung informasi sensitif.	Memberikan informasi tambahan kepada penyerang tentang aplikasi atau *server*.
14.	*Modern Web Application*	*Informational*	Deteksi penggunaan teknologi *website* moderen	Risiko tergantung pada *update* dan

No	Nama Celah Keamanan	Risiko	Deskripsi	Dampak
			yang mungkin memiliki kerentanan tersendiri.	praktik keamanan yang diterapkan pada teknologi tersebut.
15.	*Session Management Response Identified*	*Informational*	Ditemukan manajemen sesi yang mungkin memiliki kelemahan.	Manajemen sesi yang buruk bisa menyebabkan pembajakan sesi.
16.	*User Agent Fuzzer*	*Infomational*	Aktivitas *fuzzer* yang mencoba memanipulasi *header User-Agent*.	Membantu menemukan *input* yang menyebabkan kesalahan atau kerentanan.
17.	*User Controllable HTML Element Attribute (Potential XSS)*	*Informational*	Kemungkinan adanya atribut *element HTML* yang dapat dikontrol oleh pengguna, membuka potensi serangan *XSS*.	Penyerang bisa menyuntikkan dan menjalankan skrip berbahaya di *browser* pengguna.

3.3 Penetration Testing

Pada tahap ini akan dilakukan pengujian celah kerentanan yang telah ditemukan pada tahap sebelumnya terhadap website pemdescurug.com, yang kemudian dilakukan uji validasi kerentanan yang ditemukan oleh vulnerability scanner pada tahap vulnerability analysis. Pengujiannya akan dilakukan dengan kerentanan – kerentanan yang lebih spesifik adalah sebagai berikut :

- Cloud Metadata Potentially Exposed
- Cross Site Scripting (Reflected)
- SQL Injection – SQLite
- Absence of Anti-CSRF Tokens
- Content Security Policy (CSP) Header Not Set
- Hidden File Found
- Missing Anti-clickjacking Header
- User Controllable HTML Element Attribute (Potential XSS)
- OSVDB – 48

Berikut merupakan hasil pengujian yang kerentanan lebih spesifik yang diperoleh pada tahap sebelumnya :

a. Cloud Metadata Potentially Exposed

Kerentanan ini muncul dari kesalahan konfigurasi NGINX server dalam percobaan untuk mengakses metadata yang disediakan penyedia layanan cloud seperti AWS, GCP, dan Azure. Semua penyedia layanan ini menyediakan metadata melalui alamat IP internal yang tidak dapat dirutekan. Kerentanan ini ditemukan oleh OWASP ZAP.

```
POST https://pemdescurug.com/latest/meta-data/ HTTP/1.1
host: 169.254.169.254
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:125.0) Gecko/20100101 Firefox/125.0
pragma: no-cache
cache-control: no-cache
content-type: application/x-www-form-urlencoded
referer: https://pemdescurug.com/artikel/2022/10/27/
program-kerja-pemerintahan-desa-curug
content-length: 74
Cookie:
ci_session=64248b4ba056a013331efc82b735fd65b550aa06;
sidcsrf=8e456f84cc70961e6717f3feeb6a82cd
```

Gambar 7. *IP Cloud Exposed pada Request*

Pada Gambar diatas request ke URL “https://pemdescurug.com/latest/meta-data”,

pada bagian host terdapat sebuah IP dari penyedia layanan cloud yang diekspos oleh server NGINX yang dikonfigurasi dengan salah dan diakses dengan menggunakan alamat IP ini di bidang header Host.

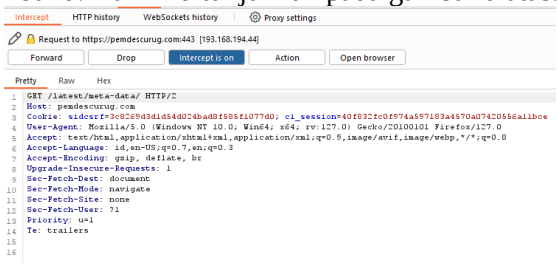
```
HTTP/1.1 302 Found
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100
content-type: text/html
content-length: 771
date: Mon, 24 Jun 2024 11:25:24 GMT
server: LiteSpeed
cache-control: no-cache, no-store, must-revalidate, max-age=0
location: https://ups-error.com
platform: hostinger
strict-transport-security: max-age=31536000; includeSubDomains; preload
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
vary: User-Agent
```

Gambar 8. *Respon setelah Request di Forward*

Berdasarkan kode status respon yang berhasil, informasi pada metadata cloud mungkin telah dikembalikan dalam respon. Metadata yang dikembalikan dapat mencakup informasi yang memungkinkan penyerang untuk sepenuhnya mendapatkan informasi penting soal sistem.

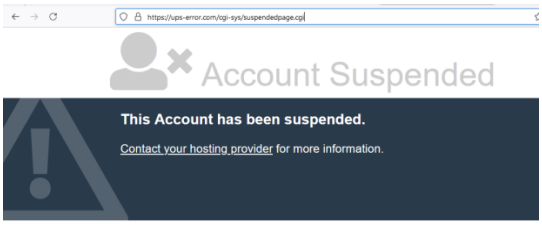
Gambar 9. *Akses Langsung pada Browser*

Kemudian, setelah dicek langsung pada browser, respon yang didapat adalah 404 Not Found. Hal ini ditunjukkan pada gambar diatas.



Gambar 10. *Request pada Browser Sebelumnya*

Pada gambar diatas merupakan request yang di cek pada aplikasi Burpsuite. dari tampilan browser sebelumnya. Hal ini ditunjukkan pada gambar sebelumnya yang menunjukan 404 not found.

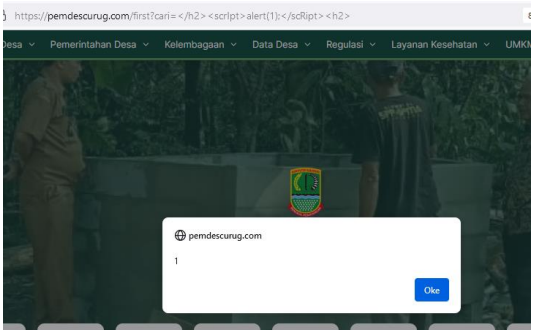


Gambar 11. *Tampilan Browser setelah mengedit Request*

Setelah mengganti request sesuai pada gambar sebelumnya untuk melihat tampilan situs webnya, dan didapat redirect URL. Hal tersebut ditunjukkan pada gambar diatas yang menunjukan bahwa tidak adanya informasi sensitif terkait meta-data cloud.

b. Cross Site Scripting (Reflected)

Cross Site Scripting (XSS) memanfaatkan celah keamanan dengan cara memasukkan script JavaScript berbahaya. Oleh karena itu, dalam pengujian penetration testing, script tersebut dimasukkan ke dalam form input di halaman website yang sedang diuji.



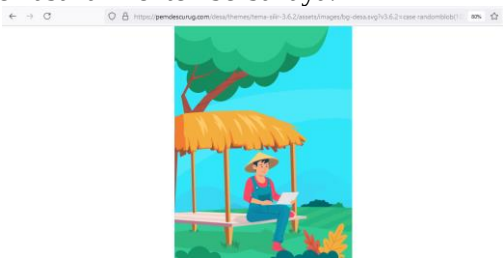
Gambar 12. *Pengujian XSS pada Menu Cari*

Pada Gambar diatas, URL tersebut digunakan untuk mencari mengenai arsip atau berita yang sesuai dengan yang diinputkan. Pengujian Cross-Site Scripting (XSS) pada URL "https://pemdescurug.com/first?cari=" dengan payload "</h2><scrIpt>alert(1);</scRipt><h2>" berdasarkan hasil scanning OWASP ZAP, dan hasil dari pengujian tersebut terdapat kerentanan XSS.

c. SQL Injection – SQLite

Injection merupakan jenis serangan yang memanfaatkan ketiadaan validasi input suatu aplikasi (baik di dalam method POST maupun GET). Teknik sql injection dapat mencuri informasi penting seperti username

dan password, merubah database, dan memasukan konten berbahaya.



Gambar 13. *Pengujian SQL Injection*

Pada gambar diatas, pengujian sql injection dengan URL "https://pemdescurug.com/desa/themes/tema-silir-3.6.2/assets/images/bg-desa.svg?v3.6.2" dengan payload "case randomblob(100000) when not null then 1 else 1 end" yang didapatkan pada hasil scanning OWASP ZAP, dan setelah dilakukan pengujian tidak adanya perbedaan waktu dalam load gambar, maka tidak adanya kerentanan sql injection.

d. Absence of Anti-CSRF Tokens

CSRF adalah jenis serangan di mana penyerang memaksa pengguna untuk melakukan tindakan yang tidak diinginkan di website. Serangan ini terjadi ketika pengguna sudah terautentikasi di website tersebut.

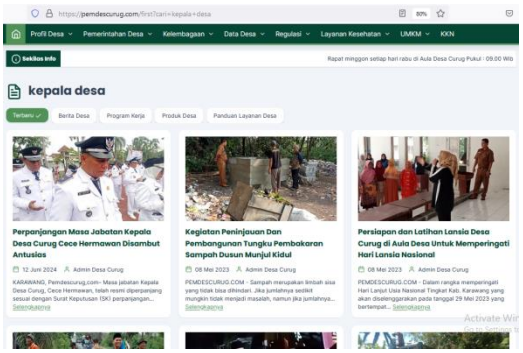
Evidence	<form action="https://pemdescurug.com/first" class="form lg:mt-0" method="get">

Gambar 14. *Source Code pada URL yang tidak ada CSRF Token*

Pada gambar diatas URL yang diperoleh pada hasil scanning owasp zap, menunjukan form input script html yang tidak ada nya script csrf token, tetapi sebenarnya tidak diperlukan, dikarenakan script csrf token biasanya hanya ada pada input yang memodifikasi data seperti method POST, PUT dan DELETE, sedangkan pada gambar diatas menunjukan bahwa method yang dipakai adalah method GET, Jadi walaupun tidak adanya csrf token tidak berpengaruh karena bukan pada request yang mengubah atau modifikasi data.

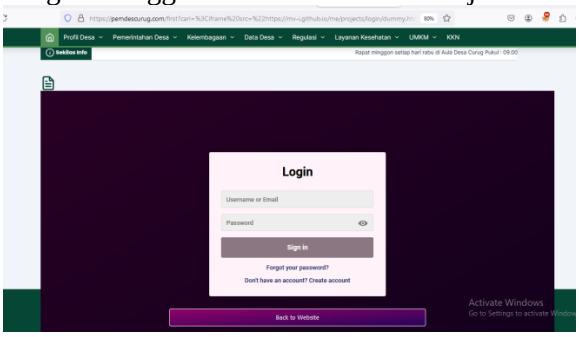
e. Content Security Policy (CSP) Header Not Set & Missing Anti clickjacking Header

CSP mencakup pengaturan keamanan untuk semua jenis sumber daya di halaman website, diantaranya skrip (termasuk iframe), gambar, CSS, font, dll. Sedangkan Header Anti-Clickjacking fokusnya lebih spesifik pada pengaturan untuk melindungi situs dari serangan clickjacking, khususnya dalam konteks pengaturan penggunaan iframe.



Gambar 15. *Percobaan Menu Cari*

Pada gambar diatas merupakan percobaan fitur search pada URL yang diperoleh pada hasil scanning owasp zap yang akan dicoba untuk menguji kerentanan Content Security Policy Header dan Missing Anti-clickjacking Header, yang kemudian akan di uji dengan menggunakan metode iframe injection.

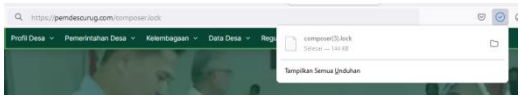


Gambar 16. *Pengujian Iframe Injection*

Pada gambar diatas telah dilakukan pengujian iframe injection pada fitur search sebelumnya dengan menggunakan payload : "<iframe src="https://mv-i.github.io/me/projects/login/dummy.html" width="80%" height="95%" style="position:absolute" frameBorder="0" allowFullScreen></iframe>", maka dari itu terbukti bahwa adanya kerentanan Content Security Policy Header Not Set dan Missing Anti-clickjacking Header.

f. Hidden File Found

Kerentanan ini ditemukan oleh OWASP ZAP, yang mana file sensitif diidentifikasi sebagai dapat diakses atau tersedia. Hal ini dapat membocorkan informasi administratif, konfigurasi, atau kredensial yang dapat dimanfaatkan oleh individu jahat untuk menyerang sistem lebih lanjut atau melakukan upaya rekayasa sosial.



Gambar 17. *Hidden File Found Terunduh*

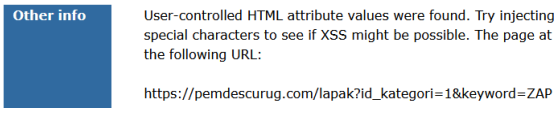
Pada gambar diatas menunjukan bahwa URL “https://pemdescurug.com/composer.lock” yang didapat pada proses scanning menggunakan OWASP ZAP, setelah mengakses URL tersebut terdapat file yang langsung terunduh bernama “composer.lock”.

Gambar 18. *Isi dari Hidden File*

Setelah terunduh dan file dibuka, pada gambar diatas menunjukan isi dari file “composer.lock” yang merupakan komponen penting dalam manajemen dependensi proyek PHP yang memastikan konsistensi versi paket di seluruh lingkungan pengembangan dan produksi, membantu mengurangi risiko masalah yang disebabkan oleh perbedaan versi. Menyimpan file “composer.lock” secara publik dapat menimbulkan risiko keamanan karena mengungkapkan versi dan informasi detail tentang paket yang digunakan, yang bisa dimanfaatkan oleh penyerang untuk mencari kerentanan.

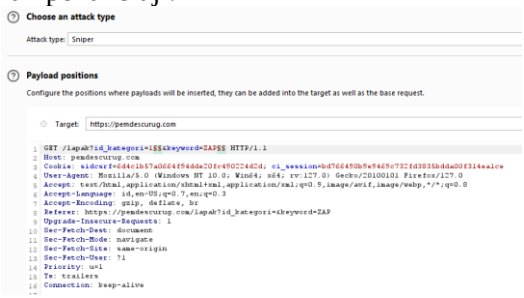
g. User Controllable HTML Element Attribute (Potential XSS)

Kerentanan ini merujuk pada situasi di mana atribut HTML dalam aplikasi website dapat dikendalikan oleh input pengguna. Jika input ini tidak divalidasi atau disanitasi dengan benar, hal ini dapat menyebabkan kerentanan Cross-Site Scripting (XSS).



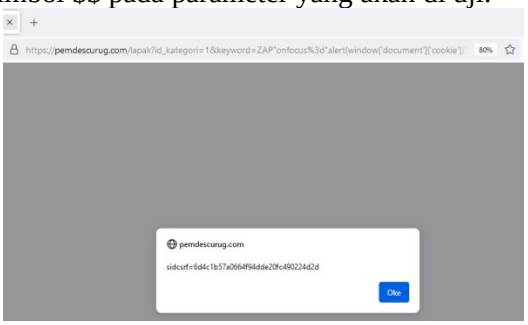
Gambar 19. *Hasil Scan pada URL yang Berpotensi Adanya XSS*

Pada gambar diatas merupakan hasil scanning dari OWASP ZAP yang melaporkan bahwa terdapat sebuah user input yang berada pada URL “https://pemdescurug.com/lapak?id_kategori=1&keyword=ZAP”, juga terdapat dua parameter yaitu “id_kategori=” dan “keyword=” yang berpotensi terdapat kerentanan Cross-Site Scripting (XSS), yang akan penulis uji.



Gambar 20. *Request URL Berpotensi XSS*

Pada gambar diatas merupakan request dari URL yang berpotensi adanya XSS menurut hasil scanning OWASP ZAP. Ditambahkannya simbol $$ pada parameter yang akan di uji.



Gambar 21. *Pengujian XSS*

Pada diatas merupakan langkah setelah menguji kedua parameter menggunakan payload yang dimasukkan menggunakan burpsuite. Hasil yang didapat hanya ada kerentanan XSS pada parameter “keyword=” dengan payload

“”onfocus="alert(window['document']['cookie']])" autofocus=””.

h. OSVDB - 48

Open Source Vulnerability Database (OSVDB) yang merujuk pada kerentanan umum yang dikenal sebagai "Path Disclosure" atau "Path Information Disclosure". Kerentanan ini memungkinkan seorang penyerang untuk mendapatkan informasi tentang struktur direktori dan jalur file pada server website, yang bisa digunakan untuk serangan lebih lanjut. Celah ini ditemukan pada saat scanning menggunakan Nikto.

Gambar 22. *URL 1 yang Berpotensi Adanya Kerentanan Path Disclosure*

Pada gambar diatas, ditunjukkan pengujian URL path "/doc" pada "pemdescurug.com". Hasil pengujian ini menunjukkan bahwa tidak ditemukan adanya kerentanan path disclosure.

Gambar 23. *URL 2 yang Berpotensi Adanya Kerentanan Path Disclosure*

Pada gambar diatas juga ditunjukkan bahwa tidak ditemukan adanya kerentanan path disclosure. Hal ini berlaku untuk URL path "/usr/doc" pada "pemdescurug.com".

5. KESIMPULAN

a. Keretanan yang ditemukan Cross Site Scripting (Reflected) dan User Controllable HTML Element Attribute (Potential XSS), yaitu dengan memasukan payload XSS pada halaman awal menu pencarian dengan halaman lapak menu pencarian yang masuk kedalam kategori Injection dan, Content Security Policy (CSP) Header Not Set & Missing Anti-clickjacking Header, yang mana dapat menyuntikan iframe halaman web lain pada halaman awal menu pencarian, dan yang masuk kedalam kategori Security Misconfiguration Hidden File Found pada URL https://pemdescurug.com/composer.lock halaman ini berisi file composer.lock yang akan langsung terunduh yang berisi paket-paket PHP yang diinstal dalam proyek.

b. Ditemukannya kerentanan Cross Site Scripting (Reflected) dan User Controllable HTML Element Attribute (Potential XSS), yang menurut OWASP Top 10 2021 masuk kedalam kategori Injection dengan keduanya pada tingkat Risiko High, kemudian untuk kategori Security Misconfiguration, terdapat kerentanan Content Security Policy (CSP) Header Not Set & Missing Anti-clickjacking Header dengan tingkat risiko High, selain itu ada kerentanan Hidden File Found dengan tingkat risiko Medium.

c. Rekomendasi perbaikan atau mitigasi untuk kerentanan injection seharusnya dilakukannya validasi dan escape input pengguna secara menyeluruh, gunakan context-aware encoding untuk semua data yang ditampilkan, serta terapkan Content Security Policy header (CSP) yang ketat untuk mencegah eksekusi skrip yang tidak diizinkan. Kemudian untuk kategori Security Misconfiguration tambahkan header berikut untuk mengatasi masalah: CSP: Content-Security-Policy: default-src 'self'; script-src 'self'; object-src 'none' Anti-clickjacking: X-Frame-Options: DENY. Untuk hidden file found dapat mengatur aturan pada file .htaccess atau konfigurasi server untuk mencegah akses langsung ke file sensitif.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang terkait, yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

[1] C. D. Berliana, T. A. Saputra, and I. Gunawan, "Analisis Serangan dan Keamanan pada Denial of Service (DOS): Sebuah Review Sistematik," *JIIFKOM (Jurnal Ilm. Inform. Komputer) STTR Cepu*, vol. 1, no. 2, pp. 33–38, 2022, [Online]. Available: https://www.sttrcepu.ac.id/jurnal/index.php/jiifkom/article/view/229/140

[2] A. Mutedi and B. Tjahjono, "Systematic Literature Review: Preventing SQL Injection Attacks Using Tools OWASP CSR Web Application Firewall," *J. Inform. Univ. Pamulang*, vol. 7, no. 1, pp. 151–156, 2022, [Online]. Available: http://openjournal.unpam.ac.id/index.php/informatika

[3] BSSN, "Laporan Tahunan Monitoring Keamanan Siber," *Direktorat Operasi Keamanan Siber Badan Siber Dan Sandi Negara*, pp. 1–236, 2022, [Online]. Available: https://cloud.bssn.go.id/s/Lyw8E4LxwNiJoNw

[4] I. O. Riandhanu, "Analisis Metode Open Web Application Security Project (OWASP) Menggunakan Penetration Testing pada Keamanan Website Absensi," *J. Inf. dan Teknol.*, vol. 4, no. 3, pp. 160–165, 2022, doi: 10.37034/jidt.v4i3.236.

[5] A. A. Arafat, *Penetration Testing Pada Website Registrar Pengelola Nama Domain Internet Indonesia (Pandi)*, vol. 20. 2020. [Online]. Available: http://repository.uinjkt.ac.id/dspace/handle/123456789/53637

[6] R. M. Wibowo and A. Sulaksono, "Web Vulnerability Through Cross Site Scripting (XSS) Detection with OWASP Security Shepherd," *Indones. J. Inf. Syst.*, vol. 3, no. 2, pp. 149–159, 2021, doi: 10.24002/ijis.v3i2.4192.

[7] M. Nurfathullah, "Pengujian Blackbox Pada Sistem Pemesanan Untuk Sales Order Di Pt Bukit Muria Jaya Berbasis Equivalence Partitions," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 1141–1147, 2024, doi: 10.23960/jitet.v12i2.4174.

[8] R. R. Daniswara, G. Made, A. Sasmita, P. Agus, and E. Pratama, "Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study : Udayana University SIMAK-NG Application)," *JITTER-Jurnal Ilm. Teknol. dan Komput.*, vol. 1, no. 1, 2020.

[9] Z. Faizi, Puwantoro, and A. A. Ridha, "Analisis Web Security Hole Menggunakan Metode Penetration Testing Execution and Standard (Studi Kasus: Universitas …," *J. Inf. dan Komput.*, no. 2, pp. 322–327, 2023, [Online]. Available: https://dcckotabumi.ac.id/ojs/index.php/jik/article/view/480%0Ahttps://dcckotabumi.ac.id/ojs/index.php/jik/article/download/480/324

[10] P. Sharma, "Securing Your Web Application A Deep Dive into OWASP Top 3 Security Risks," 2023.

[11] N. M. Farhan and B. Setiaji, "Indonesian Journal of Computer Science," *Indones. J. Comput. Sci.*, vol. 12, no. 2, pp. 284–301, 2023, [Online]. Available: http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135

[12] I. O. Riandhanu, "Analisis Metode Open Web Application Security Project (OWASP) Menggunakan Penetration Testing pada Keamanan Website Absensi," *J. Inf. dan Teknol.*, 2022, doi: 10.37034/jidt.v4i3.236.

[13] Y. Armando and R. Rosalina, "Penetration Testing Tangerang City Web Application With Implementing OWASP Top 10 Web Security Risks Framework," *JISA(Jurnal Inform. dan Sains)*, vol. 6, no. 2, pp. 105–109, 2023, doi: 10.31326/jisa.v6i2.1656.

[14] U. Ravindran and R. V. Potukuchi, "A Review on Web Application Vulnerability Assessment and Penetration Testing," *Rev. Comput. Eng. Stud.*, vol. 9, no. 1, pp. 1–22, 2022, doi: 10.18280/rces.090101.

[15] M. R. Ramdani, N. Heryana, and Y. S. A. Irawan, "Penetration Testing pada Website Universitas Singaperbangsa Karawang Menggunakan Open Web Application Security Project (OWASP)," *J. Pendidik. dan Konseling*, vol. 4, no. 3, pp. 5522–5529, 2022, [Online]. Available: http://journal.universitaspahlawan.ac.id/index.php/jpdk/article/view/6353