ANALISA PERANKINGAN KONTROL KEAMANAN ASPEK SISTEM PADA CMS WORDPRESS

Jacqueline Evangelista S.[1], Adityas Widjajarto[2], Avon Budiyono[3]

[123,]Program Studi Sistem Informasi, Universitas Telkom

Jl. Telekomunikasi No.1, Kabupaten Bandung, Provinsi Jawa Barat, 40257
[1] jekliinn@student.telkomuniversity.ac.id, [2] adtwjrt@telkomuniversity.ac.id,
[3] avonbudi@telkomuniversity.ac.id

Abstract

The widespread use of Content Management Systems (CMS) like WordPress makes them prime targets for cyberattacks. This research identifies and assesses vulnerabilities in WordPress, designs security controls based on OWASP Top Ten standards, and prioritizes security measures according to the severity of these vulnerabilities. Through experiments and simulations focusing on WordPress vulnerabilities, five major issues were found. First, in the Broken Access Control category, PHP 8.1.0-dev and Apache 2.4.49 exploits showed high vulnerability levels and alteration threats. Second, SQL Injection exploits were identified in the Injection category, presenting critical vulnerabilities and disclosure risks. Third, Path Traversal exploits in WordPress were detected under the Insecure Design category, with medium vulnerability levels and alteration threats. Fourth, the Social Warfare Plugin exploit, under Vulnerable and Outdated Components, exhibited moderate vulnerability levels and alteration risks. Proposed security controls include Web Application Firewall (WAF) use, regular updates, access restrictions, strict input validation, and parameterized queries to prevent further exploits. The focus is on addressing critical vulnerabilities, especially SQL Injection, to minimize exploitation risks. This research concludes that implementing security controls based on OWASP Top Ten standards is essential for mitigating risks in WordPress, prioritizing the most critical vulnerabilities.

Keywords : *WordPress, security, vulnerability, exploit, OWASP*

Abstrak

Penggunaan *Content Management System* (CMS) seperti WordPress sangat populer, namun juga rentan terhadap serangan siber. Laporan GoDaddy pada 2018 mencatat 18.302 insiden peretasan, di mana 90% menargetkan WordPress. Penelitian ini bertujuan untuk mengidentifikasi dan mengevaluasi kerentanan WordPress, serta menyusun desain kontrol keamanan berdasarkan standar OWASP *Top Ten*. Penelitian dilakukan melalui eksperimen dan simulasi yang menekankan eksploitasi kerentanan WordPress. Lima kerentanan utama ditemukan, yaitu *Broken Access Control* pada PHP 8.1.0-dev dan Apache 2.4.49 dengan tingkat "*High*", *Injection* melalui SQL *Injection* dengan tingkat "*Critical*", *Insecure Design* pada *Path Traversal* dengan tingkat "*Medium*", serta *Vulnerable and Outdated Components* pada *Plugin Social Warfare* dengan tingkat "*Medium*". Desain kontrol yang diusulkan meliputi penggunaan *Web Application Firewall* (WAF), pembaruan rutin, pembatasan akses, validasi input, dan penerapan *parameterized queries*. Prioritas diberikan pada kerentanan "*Critical*" seperti SQL *Injection*. Penerapan desain kontrol keamanan yang sesuai standar OWASP *Top Ten* terbukti penting dalam mengurangi risiko pada WordPress.

Kata kunci : *WordPress, Keamanan, Kerentanan, Eksploitasi, OWASP*

1. PENDAHULUAN

Keamanan merupakan salah satu aspek krusial dalam pengelolaan sistem informasi, terutama dalam konteks *Content Management System* (CMS). [1] CMS digunakan secara luas untuk mengelola konten di berbagai jenis situs *web*, mulai dari *blog* hingga situs *web* perusahaan besar. Namun, penggunaan CMS juga membawa risiko keamanan jika tidak diatur dengan baik. Banyak *developer* yang kurang memerhatikan aspek-aspek yang ada di dalam *website* sehingga tidak jarang adanya celah keamanan yang bisa dimanfaatkan oleh penyerang untuk masuk ke dalam sistem website tersebut. [2] Oleh karena itu, pentingnya mekanisme keamanan CMS berdasarkan aspek sistem untuk mengatasi tantangan keamanan yang dihadapi oleh pengguna CMS.[3][4]

WordPress merupakan salah satu platform CMS yang paling populer digunakan di seluruh dunia untuk membuat berbagai jenis situs *web*, mulai dari *blog* pribadi hingga situs *e-commerce* skala besar. Namun, popularitas WordPress juga menjadikannya sasaran utama bagi penyerang siber yang mencari celah keamanan. Berdasarkan laporan yang diterima oleh penyedia *web* hosting terkenal asal Amerika Serikat yaitu *GoDaddy*, data menunjukkan bahwa terdapat 18.302 laporan peretasan yang terjadi pada klien *GoDaddy* sepanjang tahun 2018. [5][6]

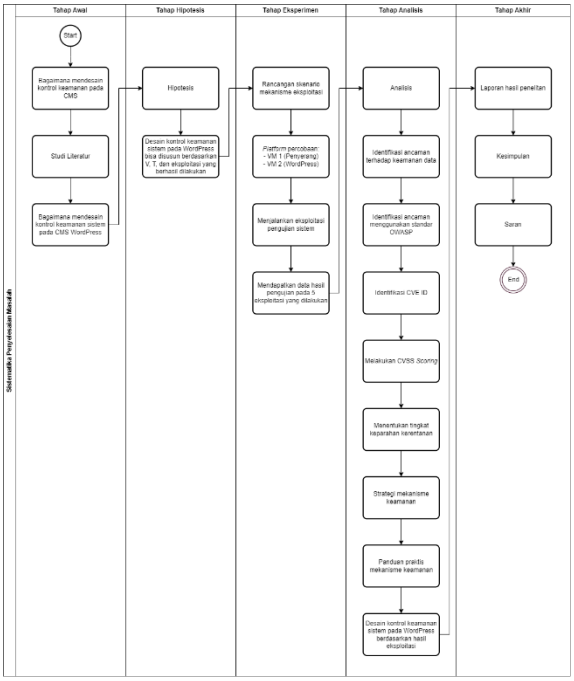
Penerapan desain kontrol merupakan langkah yang sangat penting untuk mengatasi berbagai risiko keamanan yang muncul pada *platform* CMS seperti WordPress. Desain kontrol merupakan pendekatan sistematis yang dirancang untuk mengidentifikasi, mencegah, dan memitigasi ancaman keamanan sebelum eksploitasi terjadi. Selain itu, desain kontrol yang efektif harus mengacu pada standar keamanan yang diakui secara global, seperti OWASP *Top Ten*, yang memberikan panduan tentang ancaman keamanan paling kritis dalam aplikasi *web*. Mengikuti standar ini membantu memastikan bahwa semua aspek keamanan, mulai dari pengelolaan akses hingga mitigasi kerentanan, diterapkan secara konsisten dan menyeluruh.[7][8]

2. METODOLOGI PENELITIAN

2.1. Sistematika Penyelesaian Masalah

Dalam melakukan penelitian, sistematika penyelesaian masalah diperlukan menjelaskan tahapan-tahapan yang dilakukan dalam penelitian yang digunakan untuk membantu dalam menggambarkan urutan langkah-langkah yang diambil dalam melakukan penelitian serta bagaimana menemukan solusi untuk mengatasi masalah yang ada pada penelitian tersebut.

Penelitian ini mencangkup 5 tahapan, yaitu: Tahap Awal, Tahap Hipotesis, Tahap Eksperimen, Tahap Analisis, dan Tahap Akhir.



Gambar 1. Sistematika Penyelesaian Masalah

2.2. Tahap Awal

Pada tahap ini, penelitian dimulai dengan mengidentifikasi pertanyaan penelitian yang berfokus pada bagaimana merancang kontrol keamanan pada CMS, khususnya WordPress. Proses ini diawali dengan Studi Literatur untuk memahami teori-teori yang sudah ada dan bagaimana hal tersebut dapat diaplikasikan dalam konteks penelitian ini.

2.3. Tahap Hipotesis

Pada tahap ini, hipotesis dikembangkan berdasarkan studi literatur yang telah dilakukan. Penelitian berusaha untuk menyusun desain kontrol keamanan sistem WordPress berdasarkan *Vulnerability* (V), *Threat* (T), dan eksploitasi yang berhasil dilakukan.

2.4. Tahap Eksperimen

VirtualBox berfungsi sebagai komputer virtual yang berjalan di dalam komputer fisik,

memungkinkan pengguna untuk menginstal dan menjalankan satu atau lebih sistem operasi di dalam sistem operasi utama tanpa mengubah atau merusak sistem operasi utama tersebut, karena seluruh prosesnya bersifat virtual. [9] Di tahap eksperimen, peneliti merancang skenario mekanisme eksploitasi yang akan diuji. Eksperimen dilakukan dengan menggunakan dua *platform virtual* (VM), satu sebagai penyerang dan satu lagi sebagai WordPress. Eksploitasi dilakukan untuk mendapatkan data hasil pengujian dari lima jenis eksploitasi yang dipilih.

2.5. Tahap Analisis

Setelah mendapatkan data dari eksperimen, tahap ini melibatkan analisis data untuk mengidentifikasi ancaman keamanan. Ancaman tersebut kemudian dinilai menggunakan standar OWASP. Peneliti juga akan mengidentifikasi *Common Vulnerabilities and Exposures* (CVE) ID dan melakukan CVSS Scoring untuk menilai tingkat keparahan kerentanan.

2.6. Tahap Akhir

Pada tahap akhir, peneliti menyusun laporan hasil penelitian, yang berisi kesimpulan dan saran untuk langkah-langkah keamanan yang harus diambil berdasarkan hasil eksperimen dan analisis. Desain kontrol keamanan sistem WordPress disusun berdasarkan eksploitasi yang telah dilakukan dan dinilai.

3. PERANCANGAN SKENARIO DAN IMPLEMENTASI PENGUJIAN

3.1. *Reconnaissance*

Reconnaissance adalah tindakan pengumpulan informasi atau data awal tentang target, baik itu individu, organisasi, atau lokasi, untuk tujuan strategis atau taktis. Dalam konteks keamanan siber, *reconnaissance* sering kali melibatkan pengumpulan informasi tentang sistem komputer atau jaringan untuk mengidentifikasi potensi kerentanan sebelum melancarkan serangan.

3.2. Spesifikasi *Hardware* dan *Software*

Spesifikasi *Hardware* dan *Software* yang digunakan untuk melakukan proses pengujian dan penelitian untuk melakukan eksploitasi pada sistem WordPress dapat dilihat pada Tabel 1 dan Tabel 2:

TABEL I SPESIFIKASI *HARDWARE*

Komponen	Informasi	
Spesifikasi *Server*	*Processor*	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz
	Memory	2GB RAM
	SSD	25GB SSD
	System Type	64-bit *operating system*, x64-*based processor*
	Operating System	Ubuntu 64-bit
Spesifikasi *Main OS*	*Processor*	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz
	Memory	16GB RAM
	SSD	512GB SSD
	System Type	64-bit *operating system*, x64-*based processor*
	Operating System	Windows 11 Home
Spesifikasi *Virtual Machine* (VM)	*Processor*	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz
	Memory	4GB RAM
	SSD	30 GB SSD
	System Type	64-bit *operating system*, x64-*based processor*
	Operating System	Kali Linux 64-Bit

TABEL II SPESIFIKASI *SOFTWARE*

Tipe	*Software*	*Versi*
Operating System	Kali Linux	2024.1 Kali-rolling

Content Management System (CMS)	WordPress	6.4.3
Attack Tools	Gobuster	3.6
	WPScan	3.8.25
	Metasploit	6.4.2
	Python	3.0

Tabel 2 dijabarkan beberapa spesifikasi yang digunakan pada penelitian dan pengujian. Pada bagian ini akan menjelaskan peran dari setiap perangkat lunak yang digunakan, sebagai berikut:

1. *Operating System*

Pengujian ini menggunakan sistem operasi Kali Linux versi 2024.1. Kali Linux dirancang untuk keamanan komputer dan pengujian penetrasi. Pada sistem operasi ini, terdapat berbagai alat dan aplikasi untuk melakukan tes keamanan, forensik digital, dan analisis kerentanan sistem.

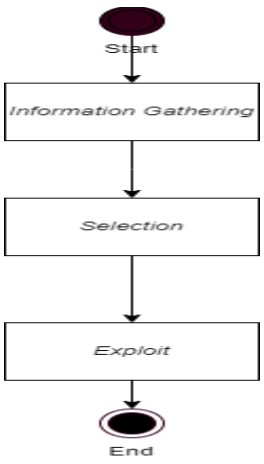
2. *Content Management System* (CMS)

WordPress merupakan *platform* manajemen konten *open source* yang sangat populer, memudahkan pembuatan dan pengelolaan situs *web* serta *blog*. Penelitian ini membahas tentang WordPress, sebuah *platform* populer untuk membuat *website*. WordPress ini menggunakan bahasa pemrograman PHP dan berjalan di komputer (*server*) yang menggunakan sistem operasi Ubuntu. Selain itu, WordPress juga menggunakan MySQL untuk menyimpan data dan Nginx untuk menampilkan *website*.

3. *Attack Tools*

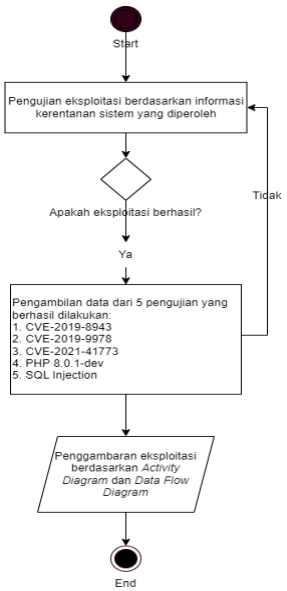
Penelitian ini menggunakan beberapa *attack tools* diantaranya Gobuster versi 3.6 untuk eksploitasi kerentanan *path traversal* pada WordPress, WPScan versi 3.8.25 untuk eksploitasi kerentanan pada *Plugin Social Warfare*, Metasploit versi 6.4.2 untuk SQL *Injection*, serta Python 3 untuk eksploitasi kerentanan pada sistem PHP 8.1.0-dev.

3.3. Skenario Pengujian



Gambar 2. Skenario Pengujian Eksploitasi

Gambar 2 menjelaskan mengenai proses pengujian eksperimen yang dilakukan. Proses pengujian dimulai dari penyerang yang mengumpulkan sejumlah informasi kerentanan pada WordPress. Informasi-informasi yang dikumpulkan berupa versi WordPress yang digunakan, IP *address*, *domain*, serta konfigurasi keamanan yang ada. Setelah informasi terkumpul, penyerang kemudian menyeleksi data tersebut untuk mengidentifikasi potensi celah keamanan. Selanjutnya, penyerang mencoba mengeksploitasi celah tersebut dengan berbagai metode serangan yang sesuai.



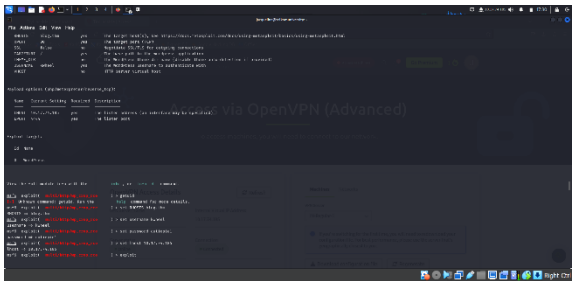
Gambar 3. Skenario Pengujian Serangan

Gambar 3 menggambarkan proses pengujian serangan menggunakan *flowchart*. Langkah awal dalam pengujian eksploitasi WordPress didasarkan pada informasi kerentanan sistem yang sudah ditemukan. Apabila eksploitasi berhasil dilakukan, maka data akan diambil dari lima pengujian yang telah dilakukan yaitu eksploitasi *Path Traversal* pada WordPress, eksploitasi *Plugin Social Warfare* pada WordPress, eksploitasi PHP 8.1.0-dev, eksploitasi Apache 2.4.49, serta SQL *Injection*.

3.4. *Eksperimen* Pengujian Eksploitasi

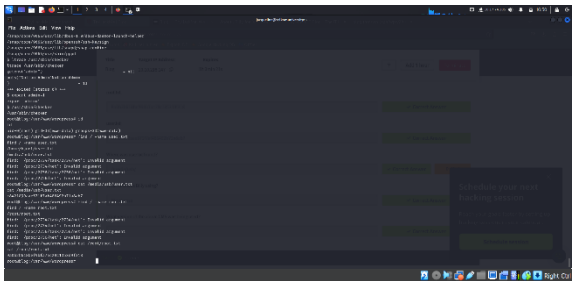
Dalam pelaksanaan uji coba eksploitasi kerentanan pada WordPress, diperlihatkan implementasi berupa *Proof of Concept* (POC) yang menggambarkan proses sebelum dan sesudah eksploitasi dilakukan.

1. Implementasi Pengujian Eksploitasi Kerentanan *Path Traversal* pada WordPress



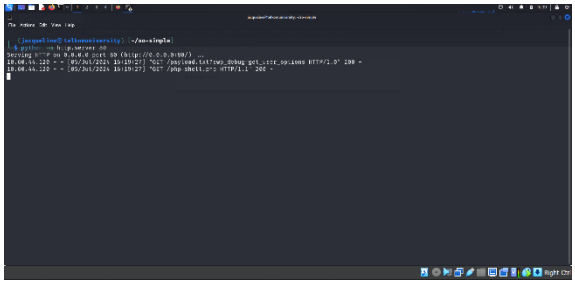
Gambar 4. Proses Eksploitasi *Path Traversal* pada WordPress

Pada gambar 4 menunjukkan proses konfigurasi pada saat dilakukan pengaturan informasi yang dibutuhkan untuk mengeksploitasi celah keamanan pada WordPress menggunakan Metasploit. Tampak beberapa perintah yang digunakan seperti "*use*", "*set*", dan "*exploit*" untuk menentukan target, mengatur *parameter* seperti "*RHOSTS*", "*LHOSTS*", dan lain-lain. Langkah ini bertujuan untuk mempersiapkan eksploitasi terhadap target yang ditentukan dengan menyiapkan *payload* dan *exploit* yang sesuai. Berikut merupakan hasil dari eksploitasi *path traversal* pada WordPress.



Gambar 5. Hasil Implementasi Eksploitasi *Path Traversal* pada WordPress

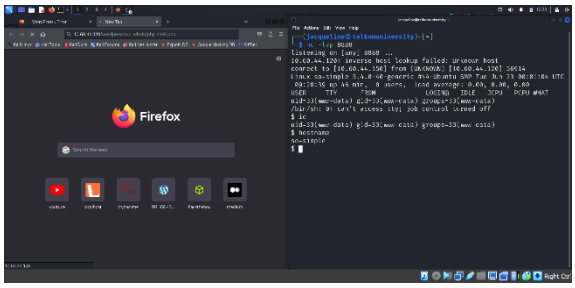
2. Implementasi Pengujian Eksploitasi *Plugin Social Warfare* pada WordPress



Gambar 6. Proses Eksploitasi *Plugin Social Warfare*

Plugin adalah cara untuk memperluas dan menambahkan fungsionalitas tambahan ke WordPress. Inti dari WordPress dirancang agar tetap ramping dan ringan, memberikan fleksibilitas maksimal dan mengurangi kode yang tidak perlu. Dengan menggunakan plugin, pengguna dapat menambahkan fitur dan fungsi khusus untuk menyesuaikan situs mereka sesuai dengan kebutuhan spesifik. [10]

Pada gambar 6 menunjukkan proses menjalankan *server* HTTP menggunakan perintah "*python -m http.server 80*". *Server* ini digunakan untuk melayani permintaan HTTP dari alamat IP 10.60.44.120. Tampak bahwa ada dua permintaan yang berhasil dilayani, yaitu permintaan untuk *file payload.txt* dan *php-shell.php*, yang keduanya mendapatkan respons status kode 200, menandakan bahwa permintaan berhasil diproses dan *file* berhasil diakses. Berikut merupakan hasil dari eksploitasi *plugin Social Warfare* pada WordPress.

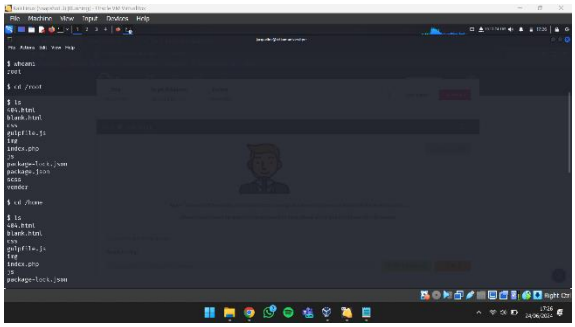


Gambar 7. Hasil Implementasi Eksploitasi *Plugin Social Warfare*

3. Implementasi Pengujian Eksploitasi PHP 8.1.0-dev

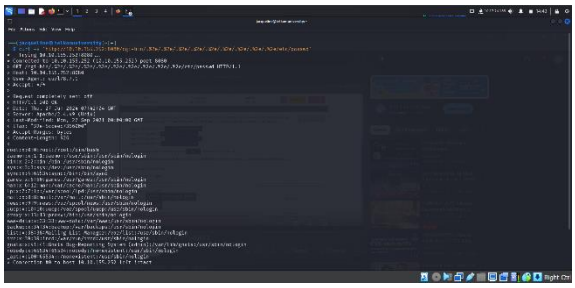
Gambar 8 Proses Eksploitasi PHP 8.1.0-dev

Pada gambar 8 menunjukkan proses pada saat membuat *file* baru dengan nama *agent.py* yang berisikan kode dari *Exploit* DB. *Script* ini berfungsi untuk membuka sesi interaktif dengan sebuah *host* melalui HTTP. Penyerang memasukkan *URL host*, kemudian *script* mengirim permintaan GET ke *URL* tersebut dan memeriksa responsnya. Jika respons berhasil (kode 200), *script* akan membuka *shell* interaktif yang memungkinkan pengguna untuk mengirim perintah ke *host*. Berikut merupakan hasil dari eksploitasi sistem PHP 8.1.0-dev.



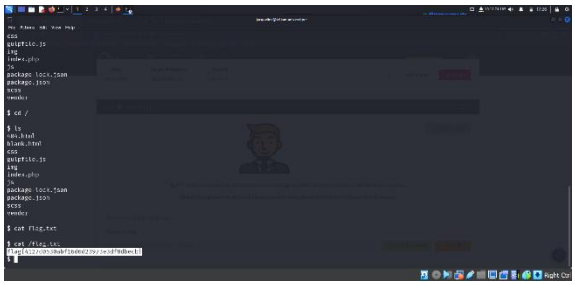
Gambar 9. Hasil Implementasi Eksploitasi PHP 8.1.0-dev

4. Implementasi Pengujian Eksploitasi Apache 2.4.49



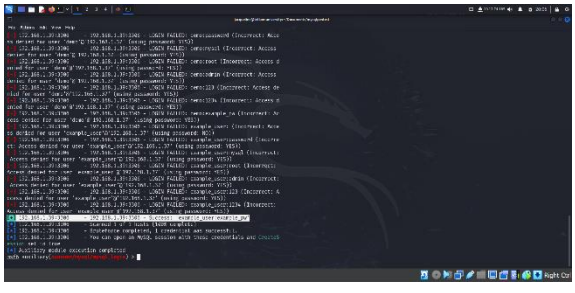
Gambar 10. Proses Eksploitasi Apache 2.4.49

Pada gambar 10 menunjukkan proses eksploitasi menggunakan perintah *curl*. Perintah yang digunakan mencoba untuk mengambil *file /etc/passwd* dari server yang berjalan di alamat IP 10.10.155.252 melalui *port* 8080. *File /etc/passwd* adalah *file* yang berisi informasi tentang akun pengguna pada sistem Kali Linux. Eksploitasi ini memanfaatkan celah keamanan untuk mengakses *file* sensitif. Hasil dari eksploitasi ini adalah tampilan isi *file /etc/passwd*, yang berisi daftar akun pengguna beserta *shell* yang digunakan oleh masing-masing akun. Pada akhirnya, eksploitasi ini berhasil dengan menunjukkan konten dari *file* tersebut.



Gambar 11. Hasil Eksploitasi Apache 2.4.49

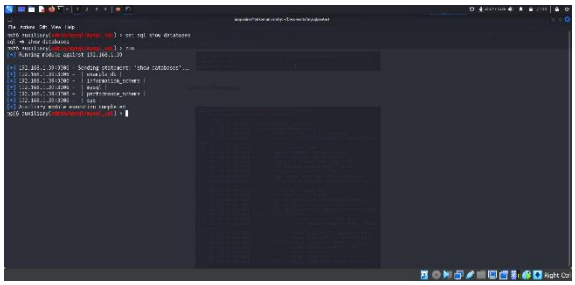
5. Implementasi Pengujian SQL *Injection*



Gambar 12. Proses Eksploitasi SQL *Injection*

SQL *Injection* merupakan serangkaian serangan yang berbentuk skrip tidak beraturan, baik berupa huruf, angka, tanda, lambang, dan lain sebagainya, yang bertujuan untuk mengganggu atau

menyerang *form* kolom *username* dan *password*. [11]. Pada umumnya serangan terhadap suatu komputer dalam suatu jaringan ialah melalui aplikasi yang masuk pada *port-port* terbuka dalam *server* [12]. Pada gambar 12 menunjukkan proses pada saat penyerang menggunakan modul "*mysql_login*" dari *Metasploit Framework* untuk melakukan serangan *brute force* terhadap *server* MySQL dengan alamat IP 192.168.1.39 pada *port* 3306. Hasil dari serangan ini menunjukkan banyak upaya *login* yang gagal dengan berbagai kombinasi *username* dan *password*, hingga akhirnya berhasil *login* menggunakan kredensial "*example_user:example_pw*". Pesan "*Success: 'example_user'*" menunjukkan bahwa kombinasi tersebut *valid* dan dapat digunakan untuk membuka sesi MySQL dengan kredensial tersebut. Berikut merupakan hasil pada saat berhasil melakukan proses *execute SQL queries*, yang mana penyerang berhasil melihat *database* yang ada di dalam MySQL.



Gambar 13. Hasil Eksploitasi SQL *Injection*

4. Hasil dan Pembahasan

4.1. Data Hasil Eksploitasi

Pada pengujian eksploitasi sistem WordPress yang telah dilakukan, didapatkan hasil analisis berupa data eksploitasi yang terdapat dalam tabel 3. Analisis ini menjelaskan secara rinci mengenai efek dari eksploitasi kerentanan pada sistem WordPress.

TABEL III DATA HASIL PENGUJIAN

Eksploitasi	*Vulnerability*	*Tools Attack*	*Data Attack*	Hasil Eksploitasi
Path Traversal pada WordPress	*Directory Traversal*	Metasploit (exploit/multi/http/wp_crop_rce)	• Kredensial • *Rhosts* dan *Lhosts*	Berhasil mendapatkan akses *shell* interaktif pada sistem target melalui sesi *meterpreter*.
Plugin Social Warfare	*Remote Code Execution* (RCE)	• Netcat • Python	• *payload.txt* • *php-shell.php*	Penyerang berhasil mendapatkan akses ke *shell* pada *server* target dengan hak akses pengguna "www-data", memungkinkan penyerang untuk menjalankan perintah pada *server* target.
PHP 8.1.0-dev	*Remote Code Execution* (RCE)	• Pyhton	• *php-reverse-shell.php*	Penyerang berhasil mendapatkan akses ke *shell* pada *server* target dengan hak akses pengguna "root", memungkinkan penyerang untuk melakukan kontrol penuh pada sistem target.
Apache 2.4.49	*Directory Traversal*	• Curl • Bash • Netcat	• Alamat URL target • *Payload data*	Penyerang berhasil mendapatkan akses *shell* interaktif pada *server* target. Hal ini memungkinkan penyerang untuk menjalankan perintah di sistem target secara *remote*, yang memberikan penyerang kontrol penuh atas sistem tersebut.
SQL *Injection*	SQL *Injection*	• Metasploit	• *Hash password*	Penyerang berhasil mengekstrak struktur *database* target, termasuk tabel dan kolom yang ada, memberikan penyerang informasi berharga tentang struktur dan konten *database* yang dapat digunakan untuk serangan lebih lanjut atau eksploitasi data.

Dari data tabel 3 di atas, menggambarkan hasil eksploitasi dari beberapa kerentanan yang ditemukan dalam sistem target. Pada kasus *Path Traversal* pada WordPress, eksploitasi dilakukan melalui kerentanan *Directory Traversal* menggunakan Metasploit dengan modul "*exploit/multi/http/wp_crop_rce*", yang memanfaatkan kredensial *Rhosts* dan *Lhosts*. Hasilnya, penyerang berhasil mendapatkan akses *shell* interaktif pada sistem target melalui sesi *meterpreter*.

Kerentanan *Remote Code Execution* (RCE) pada *Plugin Social Warfare* dieksploitasi dengan menggunakan Netcat dan Python, serta *file payload* seperti *payload.txt* dan *php-shell.php*, memungkinkan penyerang untuk mendapatkan akses ke *shell* pada *server* target dengan hak akses pengguna "*www-data*", yang memungkinkan penyerang untuk menjalankan perintah pada *server* target.

Pada PHP 8.1.0-dev, kerentanan *Remote Code Execution* (RCE) juga dieksploitasi melalui Python dengan memanfaatkan file "*php-reverse-shell.php*", sehingga penyerang berhasil mendapatkan akses ke *shell* pada server target dengan hak akses pengguna "*root*", yang memungkinkan penyerang untuk melakukan kontrol penuh terhadap sistem target.

Apache 2.4.49 juga mengalami kerentanan *Directory Traversal*, yang dieksploitasi menggunakan Curl, Bash, dan Netcat dengan memanfaatkan alamat *URL* target dan data *payload*. Hasilnya, penyerang berhasil mendapatkan akses *shell* interaktif pada *server* target secara *remote*, yang memberikan kontrol penuh atas sistem tersebut.

Terakhir, SQL *Injection* berhasil dieksploitasi menggunakan Metasploit dengan memanfaatkan *hash password*, yang memungkinkan penyerang mengekstraksi struktur *database* target, termasuk tabel dan kolom, sehingga memberikan informasi berharga yang dapat digunakan untuk serangan lebih lanjut atau eksploitasi data.

4.2. Analisis Ancaman

Berdasarkan perspektif pelaku serangan, ada tiga hal utama yang ingin dicapai ketika menargetkan data. Pertama, membocorkan data tersebut kepada pihak yang tidak berwenang (*Disclosure*). Kedua, berupaya merusak data sehingga tidak dapat digunakan lagi (*Alteration*). Serta yang ketiga, memblokir akses terhadap data sehingga tidak dapat diakses oleh pengguna yang sah (*Denial*). Ketiga tujuan ini merupakan kebalikan dari prinsip dasar keamanan informasi yang dikenal sebagai CIA *Triad* (*Confidentiality*, *Integrity*, *Availability*).[13] Dalam konteks ini, tujuan para penyerang sering disebut sebagai DAD Triad. Tabel 4 berikut ini mengklasifikasikan lima eksploitasi menjadi 2 kategori yaitu *disclosure* dan *alteration*.

TABEL IV ANALISIS ANCAMAN KEAMANAN DATA

No.	Eksploitasi	Jenis Ancaman	Alasan
1.	*Path traversal* pada WordPress	*Alteration*	Penyerang dapat mengakses *file* dan direktori di luar *root web server* yang ditentukan, memungkinkan pengungkapan informasi sensitif.
2.	*Plugin Social Warfare*	*Alteration*	Penyerang dapat melakukan *Remote Code Execution* (RCE), memungkinkan eksekusi perintah berbahaya pada *server* WordPress yang menjalankan *plugin* ini.
3.	PHP 8.1.0-dev	*Alteration*	Penyerang menjalankan perintah dari jarak jauh menggunakan *script* eksploitasi, membuka *shell* interaktif pada *server* target.
4.	Apache 2.4.49	*Alteration*	Penyerang dapat mengakses dan menjalankan *file* di luar *root web server*, membuka peluang untuk eksekusi kode berbahaya dari jarak jauh.
5.	SQL Injection	*Disclosure*	Penyerang mengeksploitasi kerentanan untuk mengakses dan mengekstrak informasi sensitif dari *database*, termasuk *hash password*.

Tabel 5 di bawah ini menyajikan klasifikasi ancaman keamanan data yang umum ditemukan dalam aplikasi web, dengan mengacu pada kerangka kerja OWASP. Ancaman-ancaman ini dikategorikan berdasarkan jenis serangan dan potensi dampaknya.

TABEL V ANALISIS ANCAMAN OWASP *TOP TEN*

No.	Eksploitasi	OWASP Categories	Alasan
1.	*Path traversal* pada WordPress	*Insecure Design* (A04:2021)	Desain sistem tidak mempertimbangkan dengan cukup aspek keamanan, memungkink

			an akses yang tidak sah ke file-file sistem.
2.	*Plugin Social Warfare*	*Vulnerable and Outdated Components* (A06:2021)	Memiliki kerentanan keamanan yang belum diperbaiki, sehingga membuka celah bagi eksploitasi oleh penyerang.
3.	PHP 8.1.0-dev	*Broken Access Control* (A01:2021)	Dapat menciptakan celah akses yang tidak terbatas atau tidak terkontrol dengan baik, memungkinkan pengguna tanpa izin untuk mengakses atau memodifikasi data sensitif.
4.	Apache 2.4.49	*Broken Access Control* (A01:2021)	Penyerang memanfaatkan kelemahan dalam kontrol akses untuk mengakses *file* sensitif dan menjalankan perintah secara tidak sah di *server*.
5.	SQL *Injection*	*Injection* (A03:2021)	Memungkinkan penyerang menyuntikkan *query* SQL berbahaya melalui input yang tidak aman, mengakibatkan akses tidak sah atau manipulasi data dalam database.

4.3. Analisis Kerentanan

Hasil analisis kerentanan dari pengujian ini sangat penting untuk merumuskan langkah-langkah pencegahan yang efektif. Tahapan yang perlu dilakukan adalah mengidentifikasi kode kerentanan (CVE), menilai tingkat keparahannya menggunakan skor CVSS, dan kemudian menyusun rencana perbaikan untuk mengatasi setiap kelemahan yang ditemukan.

4.3.1 Identifikasi CVE ID

Identifikasi CVE merupakan langkah yang sangat penting. CVE merupakan singkatan dari *Common Vulnerabilities and Exposures*, *database* yang berisi informasi tentang setiap kerentanan atau *vulnerability* yang dipublikasikan. CVE menyediakan referensi mengenai kerentanan yang ditemukan pada produk tertentu. Cara kerja CVE *checker* adalah dengan melakukan pemindaian (*scanning*) terhadap sistem operasi, lalu mencocokkan hasil pemindaian tersebut dengan *database* CVE yang tersedia. Aplikasi kemudian akan memberikan laporan hasil dari pencocokan tersebut menggunakan CVE *checker*. [14]

TABEL VI IDENTIFIKASI CVE DARI LIMA KERENTANAN YANG DIEKSPLOITASI

No.	Eksploitasi	*Vulnerability*	CVE
1.	*Path traversal* pada WordPress	*Directory Traversal*	CVE-2019-8943
2.	*Plugin Social Warfare*	*Remote Code Execution* (RCE)	CVE-2019-9978
3.	PHP 8.1.0-dev	*Remote Code Execution* (RCE)	CVE-2020-7067
4.	Apache 2.4.49	*Directory Traversal*	CVE-2021-41773
5.	SQL *Injection*	SQL *Injection Vulnerability*	CVE-2019-8457

4.3.1 Penentuan Skor CVE Menggunakan CVSS

Common Vulnerability Scoring System (CVSS) merupakan sebuah kerangka (*framework*) terbuka yang digunakan untuk mengkomunikasikan karakteristik dan dampak yang ditimbulkan oleh sebuah kerentanan aplikasi. CVSS terdiri dari tiga kelompok pengukuran yaitu *Base*, *Temporal*, dan *Environmental.* [15]
Untuk mengukur tingkat keparahan suatu kerentanan, diperlukan proses untuk menghitung skor CVSS. Proses ini dilakukan setelah kode CVE dari kerentanan tersebut telah diketahui. Skor CVSS dapat diperoleh melalui *database* NVD (*National Vulnerability Database*) atau dengan menggunakan alat hitung CVSS secara khusus.

Tingkat keparahan dibagi menjadi empat kategori, yaitu:

- *Low* : 0.1 – 3.9
- *Medium* : 4.0 – 6.9
- *High* : 7.0 – 8.9
- *Critical* : 9.0 – 10.0

Kerentanan dengan tingkat keparahan *critical* harus menjadi prioritas utama dalam mekanisme keamanan risiko. Berikut ini merupakan daftar tingkat keparahan dari lima eksplotasi yang telah dilakukan.

TABEL VII ANALISIS KERENTANAN

Eksploitasi	*Vulnerability*	CVE	CVSS *Vector*	Score	Level
Path traversal pada WordPress	*Directory Traversal*	CVE-2019-8943	AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N	6.5	*Critical*
Plugin Social Warfare	*Remote Code Execution* (RCE)	CVE-2019-9978	AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N	6.1	*High*
PHP 8.1.0-dev	*Remote Code Execution* (RCE)	CVE-2020-7067	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N	7.5	*High*
Apache 2.4.49	*Directory Traversal*	CVE-2021-41773	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N	7.5	*Medium*
SQL *Injection*	SQL *Injection Vulnerability*	CVE-2019-8457	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	9.8	*Medium*

Berdasarkan Tabel 7 diketahui bahwa skor tertinggi yaitu 9.8 yang berasal dari eksploitasi SQL *Injection*. Sementara skor kerentanan terendah berasal dari eksploitasi *Plugin Social Warfare* dengan skor 6.1.

4.4. Analisis Kontrol

Analisis kontrol dilakukan untuk menilai efektivitas langkah-langkah keamanan yang telah diterapkan pada WordPress. Dengan memeriksa secara menyeluruh kebijakan, prosedur, dan mekanisme pengendalian yang ada, dapat diidentifikasikan area yang masih rentan terhadap serangan siber. Proses ini memungkinkan untuk mengambil tindakan korektif guna memperkuat keamanan sistem dan melindungi data sensitif.

4.4.2 Strategi Mekanisme Keamanan

Standar OWASP menjadi acuan utama dalam menerapkan strategi mekanisme keamanan. Dengan mengikuti standar ini, dapat meminimalisir risiko keamanan dan mencegah serangan siber yang berpotensi merugikan.

TABEL VIII MEKANISME KEAMANAN

No.	Eksploitasi	*Level*	OWASP *Categories*	OWASP *Recommendation*	*Security Mechanism*
1.	SQL *Injection*	*Critical*	*Injection* (A03:2021)	Menggunakan *parameterized interface* untuk meningkatkan keamanan saat berinteraksi dengan *database*	Validasi input yang ketat, dan penggunaan *parameterized queries*.
2.	PHP 8.1.0-dev	*High*	*Broken Access Control* (A01:2021)	Menerapkan kontrol akses ketat, lindungi sesi, dan rutin uji serta perbarui keamanan.	Menggunakan *Web Application Firewall* (WAF).
3.	Apache 2.4.49	*High*	*Broken Access Control* (A01:2021)	Tolak akses secara *default* dan batasi akses direktori.	Membatasi akses direktori.
4.	*Path traversal* pada WordPress	*Medium*	*Insecure Design* (A04:2021)	Melakukan pemeriksaan validitas di setiap lapisan aplikasi.	Menggunakan *plugin* keamanan, seperti Wordfence, untuk memblokir akses ke *file* sensitif.
5.	*Plugin Social Warfare*	*Medium*	*Vulnerable and Outdated Components* (A06:2021)	Memperbarui *plugin* ke versi terbaru dan pastikan untuk mendapatkan *plugin* dari sumber resmi	Memperbarui *plugin* ke versi terbaru dan menggunakan *Web Application Firewall* (WAF).

Berdasarkan klasifikasi OWASP *Top Ten* 2021, Tabel 8 menjelaskan sejumlah kerentanan keamanan umum yang sering ditemukan pada aplikasi *web*. Kerentanan ini mencakup kelemahan dalam desain aplikasi, penggunaan komponen pihak ketiga yang tidak terbarui, kontrol akses yang tidak memadai, serta potensi penyuntikan kode berbahaya. Pada kerentanan dengan level "*critical*" seperti SQL *Injection* diatasi dengan menggunakan *parameterized queries*. Pada kategori "*Broken Access Control*" pada sistem PHP 8.1.0-dev diatasi dengan menggunakan *Web Application Firewall* (WAF) dan Apache 2.4.49 diatasi dengan membatasi akses direktori. Lalu untuk kategori "*Insecure Design*" dapat diatasi dengan menggunakan plugin keamanan, seperti Wordfence, untuk memblokir akses ke *file* sensitif. Sedangkan untuk kategori "*Vulnerable and Outdated Components*" dapat diatasi dengan Memperbarui *plugin* ke versi terbaru dan menggunakan *Web Application Firewall* (WAF).

4.4.3 Desain Kontrol Berdasarkan Kerentanan yang dieksploitasi Oleh *Threat*

Untuk memastikan keamanan WordPress secara menyeluruh, perlunya mendesain kontrol keamanan yang komprehensif. Kontrol keamanan yang baik akan melindungi situs *web* dari berbagai ancaman, mulai dari serangan *brute force* hingga injeksi SQL. Tabel 9 berikut merupakan desain

kontrol berdasarkan kerentanan yang di eksploitasi oleh *threat*.

TABEL IX DESAIN KONTROL KEAMANAN

No.	OWASP *Categories*	Eksploitasi	*Level*	Jenis Ancaman	*Security Mechanism*
1.	*Broken Access Control* (A01:2021)	PHP 8.1.0-dev	*High*	*Alteration*	Menggunakan *Web Application Firewall* (WAF).
		Apache 2.4.49	*High*		Membatasi akses direktori.
2.	*Injection* (A03:2021)	SQL *Injection*	*Critical*	*Disclosure*	Validasi input yang ketat, dan penggunaan *parameterized queries*.
4.	*Insecure Design* (A04:2021)	*Path traversal* pada WordPress	*Medium*	*Alteration*	Menggunakan *plugin* keamanan, seperti Wordfence, untuk memblokir akses ke *file* sensitif.
5.	*Vulnerable and Outdated Components* (A06:2021)	*Plugin Social Warfare*	*Medium*	*Alteration*	Memperbarui *plugin* ke versi terbaru dan menggunakan *Web Application Firewall* (WAF).

Analisis ini menggunakan OWASP *Top Ten* sebagai kerangka kerja utama. Daftar ini dipilih karena telah diakui sebagai standar industri yang merepresentasikan sepuluh risiko keamanan paling krusial dalam pengembangan aplikasi *web* saat ini. Adapun tingkat keparahan kerentanan dalam setiap kategori OWASP *Top Ten* mencerminkan urutan prioritas dalam penerapan mekanisme keamanan. Dengan demikian, penetapan tingkat keparahan ini dapat menjadi panduan dalam menentukan tindakan mekanisme keamanan yang harus diprioritaskan untuk meminimalkan risiko eksploitasi.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Keamanan merupakan aspek yang sangat penting dalam pengelolaan *Content Management System* (CMS), terutama dengan meningkatnya penggunaan CMS di berbagai jenis situs *web*, dari blog pribadi hingga situs *e-commerce* berskala besar. Meskipun CMS seperti WordPress menawarkan kemudahan dalam pengelolaan konten, kurangnya perhatian terhadap aspek keamanan oleh pengembang dapat menyebabkan celah yang dapat dimanfaatkan oleh penyerang siber. WordPress, sebagai salah satu *platform* CMS paling populer, sering menjadi target utama serangan karena popularitasnya yang luas. Data menunjukkan bahwa serangan terhadap *platform* seperti WordPress tidak jarang terjadi, menimbulkan risiko besar bagi para penggunanya.

Untuk mengatasi risiko tersebut, penerapan desain kontrol keamanan yang efektif menjadi sangat penting. Desain kontrol ini harus didasarkan pada pendekatan sistematis yang tidak hanya berfokus pada pencegahan dan mitigasi ancaman, tetapi juga mengikuti standar keamanan global seperti OWASP *Top Ten*. Standar ini memberikan panduan tentang ancaman keamanan paling kritis dalam aplikasi *web*, sehingga membantu memastikan bahwa semua aspek keamanan, dari pengelolaan akses hingga mitigasi kerentanan, diterapkan secara konsisten dan menyeluruh.

Desain kontrol yang disajikan memperlihatkan bagaimana berbagai mekanisme keamanan, seperti penggunaan *Web Application Firewall* (WAF), validasi input ketat, dan plugin keamanan, dapat diterapkan untuk mengatasi berbagai ancaman keamanan yang diidentifikasi oleh OWASP. Dengan demikian, desain kontrol yang tepat dapat membantu melindungi situs web berbasis CMS dari berbagai potensi eksploitasi, memastikan sistem tetap aman dan terlindungi dari serangan.

5.2. Saran

Berdasarkan penelitian yang telah dilakukan, berikut saran yang dapat dijadikan pedoman lanjutan dari penelitian ini yaitu penelitian selanjutnya dapat mengevaluasi seberapa efektif penerapan kontrol keamanan berdasarkan OWASP *Top Ten* dalam mencegah eksploitasi kerentanan di WordPress, serta mengkaji dampak dari penerapan langkah-langkah keamanan tersebut terhadap kinerja dan kecepatan situs. Selain itu, penelitian juga dapat membandingkan kerentanan dan strategi mekanisme keamanan pada WordPress dengan *platform* CMS lain, seperti Joomla atau Drupal, untuk menemukan pola keamanan yang lebih luas dan solusi yang lebih efektif.

6. UCAPAN TERIMA KASIH

Terima kasih penulis sampaikan kepada *Next Generation Laboratory of System and Networks, Sistem Informa*si, atas dukungan berupa fasilitas dan lingkungan percobaan yang telah disediakan untuk keperluan eksperimen dalam penelitian ini.

Daftar Pustaka:

[1] R. Dwi Putra and D. Dar, "ANCAMAN SIBER DALAM PERSFEKTIF PERTAHANAN NEGARA (STUDI KASUS SISTEM

PERTAHANAN SEMESTA) SIBER THREATS IN STATE DEFENSE PERSFECTIVES (TOTAL DEFENSE SYSTEM CASE STUDY)." [Online]. Available: https://ilmupengetahuan.org/sejarah-perkembangan-internet/,

[2] A. Febriandi, M. Lubis, and A. Widjajarto, "VULNERABILITY ANALYSIS OF WEBSITE SECURITY OF POPULATION AND CIVIL RECORDS DEPARTMENT OF XYZ USING THE ISSAF FRAMEWORK."

[3] Y. Kunang, M. Fatoni, and S. Sauda, "PENGUJIAN CELAH KEAMANAN PADA CMS (Content Management System)", doi: 10.13140/RG.2.1.3163.6080.

[4] M. Murahartawaty and B. Mahendratta, "IMPLEMENTASI WEB PORTAL PARIWISATA INDONESIA MENGGUNAKAN JOOMLA BERBASIS CONTENT MANAGEMENT SYSTEM (CMS)," *Jurnal Rekayasa Sistem & Industri (JRSI)*, vol. 1, no. 01, pp. 160–165, Apr. 2016, Accessed: Nov. 17, 2023. [Online]. Available: //jrsi.sie.telkomuniversity.ac.id/JRSI/article/view/124

[5] W. Nugroho, "Hati-hati, Situs dengan CMS Wordpress Paling Rentan Kena Retas," Info Komputer. Accessed: Aug. 15, 2024. [Online]. Available: https://infokomputer.grid.id/read/121659642/hati-hati-situs-dengan-cms-wordpress-paling-rentan-kena-retas

[6] B. Risdanto, "PENGEMBANGAN E-LEARNING BERBASIS WEB MENGGUNAKAN CMS (CONTENT MANAGEMENT SYSTEM) WORDPRESS DI SMA NEGERI 1 KOTA MAGELANG," 2014.

[7] R. R. Yusuf and T. N. Suharsono, "Prosiding Seminar Sosial Politik, Bisnis, Akuntansi dan Teknik (SoBAT) ke-5 Bandung, 28 Oktober 2023 PENGUJIAN KEAMANAN DENGAN METODE OWASP TOP 10 PADA WEBSITE EFORM HELPDESK".

[8] A. W. Kuncoro, J. Informatika, F. Rahma, and M. E. Jurusan Informatika, "Analisis Metode Open Web Application Security Project (OWASP) pada Pengujian Keamanan Website: Literature Review." [Online]. Available: https://www.sciencedirect.com

[9] T. Farida, "PENGEMBANGAN MEDIA PEMBELAJARAN VIRTUAL BOX UNTUK MENGUKUR KELAYAKAN MODUL PADA MATA PELAJARAN KOMPUTER DAN JARINGAN DASAR DI SMKN 7 SURABAYA," 2019. Accessed: Aug. 13, 2024. [Online]. Available: https://www-theknowledgeacademy-com.translate.goog/blog/what-is-kali-linux/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc

[10] WordPress, "Introduction to Plugins," wordpress.org. Accessed: Aug. 13, 2024. [Online]. Available: https://wordpress.org/documentation/article/introduction-to-plugins/

[11] D. Teguh Yuwono *et al.*, "DETEKSI SERANGAN VULNERABILITY PADA OPEN JURNAL SYSTEM MENGGUNAKAN METODE BLACK-BOX," 2021. [Online]. Available: http://e-journal.stmiklombok.ac.id/index.php/jire ISSN.2620-6900

[12] H. Muhammad, I. Wayan, A. Arimbawa, and A. H. Jatmika, "ANALISIS PERBANDINGAN SISTEM AUTENTIKASI PORT KNOCKING DAN SINGLE PACKET AUTHORIZATION PADA SERVER RASPBIAN," *Jurnal Informatika & Rekayasa Elektronika*, vol. 2, no. 1, 2019, [Online]. Available: http://e-journal.stmiklombok.ac.id/index.php/jire

[13] orangsiber.com, "Pojok SECURITY – CIA & DAD Triad," orangsiber.com. Accessed: Aug. 09, 2024. [Online]. Available: https://www.orangsiber.com/pojok-security-cia-dad-triad/

[14] A. Panuntun, "BAB II LANDASAN TEORI 2," 2016.

[15] "Common Vulnerability Scoring System v3.0: User Guide." [Online]. Available: https://www.first.org/cvss/examples