JACKA
Journal of Advanced Computer Knowledge and Algorithms
Vol. 1, No. 3, July, 2024, pp. 52-58
Journal homepage: http://ojs.unimal.ac.id/jacka

Web Application Firewall (WAF) Design to Detect and Anticipate Hacking in Web-Based Applications

Muhammad Annas[1*], Rizal Tjut Adek[2], Yesy Afrillia[3]

[1,2,3] Universitas Malikussaleh, Indonesia

[*]Corresponding Author Email: muhammad.170170080@mhs.unimal.ac.id

Received: 27 May 2024
Revised: 14 June 2024
Accepted: 15 June 2024
Available online: 1 July 2024

Kata Kunci:
Web Application Firewall (WAF), Cross Site Scripting (XSS), Website, SQL Injection

ABSTRAK

Kasus kebocoran data belakangan ini marak terjadi di Indonesia. Salah satu yang paling besar adalah kebocoran data pengguna dari BPJS Kesehatan pada tahun 2021 lalu, kebocoran data ini tentu sangat merugikan pengguna. Penelitian ini mengembangkan Web Application Firewall (WAF) menggunakan ModSecurity dan OWASP Core Rule Set untuk melindungi aplikasi web dari serangan SQL Injection dan XSS. Metodologi melibatkan analisis fungsionalitas sistem yang ada menggunakan UML, dengan DVWA dan WordPress sebagai objek uji. Hasil menunjukkan deteksi serangan SQL Injection 100% dan XSS 99,8%, dengan log yang mencatat serangan secara real-time. Temuan ini menekankan pentingnya integrasi WAF dengan keamanan bawaan aplikasi web, memberikan kontribusi signifikan dalam desain dan implementasi WAF yang tangguh, serta meningkatkan ketahanan terhadap ancaman siber yang berkembang.

Keywords:
Web Application Firewall (WAF), Cross Site Scripting (XSS), Website, SQL Injection

ABSTRACT

Data leakage cases have recently been rampant in Indonesia. One of the biggest is the leak of user data from BPJS Health in 2021, this data leak is certainly very detrimental to users. This research develops a Web Application Firewall (WAF) using ModSecurity and OWASP Core Rule Set to protect web applications from SQL Injection and XSS attacks. The methodology involves analyzing the functionality of the existing system using UML, with DVWA and WordPress as test objects. Results showed 100% SQL Injection and 99.8% XSS attack detection, with logs recording attacks in real-time. The findings emphasize the importance of WAF integration with web application built-in security, making significant contributions in the design and implementation of resilient WAFs, as well as improving resilience against evolving cyber threats.

1. INTRODUCTION

Pesatnya kemajuan pada bidang teknologi dan internet, membuat manusia semakin mudah dalam menjalankan aktifitas sehari-hari. Mulai dari berinteraksi sosial melalui media sosial, bekerja, belajar, mengakses hiburan, dan berbelanja. Dengan segala kemudahan dan kecepatan akses yang tersedia, saat ini internet telah menjadi kebutuhan penting dalam kehidupan manusia modern (Zulkarnain, 2021).

Kasus kebocoran data belakangan ini marak terjadi di Indonesia. Salah satu yang paling besar adalah kebocoran data pengguna dari BPJS Kesehatan pada tahun 2021 lalu, kebocoran data ini tentu sangat merugikan pengguna. Kasus ini hanya merupakan salah satu contoh dari 5000 kasus yang terjadi tiap bulann-ya, misalnya kebocoran data pengguna toko online, peretasan Website dan lain-lain (kusnandar, 2021).

Lalu lintas yang masuk ke aplikasi web akan dipantau dan disaring oleh sistem WAF. WAF akan mengidentifikasi lalu lintas dengan menganalisis permintaan GET dan POST yang dikirimkan melalui HTTP dan HTTPS, dan kemudian menerapkan aturan firewall yang telah ditetapkan. Jika WAF mengidentifikasi lalu lintas yang mencurigakan atau lalu lintas yang berpotensi menimbulkan risiko keamanan ke situs web, WAF akan memblokir dan mencegah akses tersebut (CNBC Indonesia, 2021).

Terlebih pada situasi ekonomi digital seperti sekarang, *web application* seringkali menjadi target penyerangan oleh penyerang. Penyerang menargetkan untuk menyerang sebuah *website* biasanya karena terdapat banyak data penting didalamnya. Mengacu pada isu keamanan yang terjadi, nyatanya kemudahan dan kecepatan akses saja tidaklah cukup, sektor keamanan juga menjadi hal yang patut diperhatikan agar pengguna semakin merasa nyaman dan aman menggunakan internet untuk berbagai keperluannya.

Salah satu cara untuk mengatasi isu ini adalah dengan membuat sebuah sistem keamanan website dengan menggunakan *Web Application Firewall*. *Web Application Firewall* (WAF) adalah jenis aplikasi *firewall* yang melindungi aplikasi HTTP. Intinya, WAF bertindak sebagai *gatekeeper* untuk situs web. Serangan siber seperti *cross-site-scripting* (XSS), *cross-site forgery*, *SQL Injection*, DDoS dan lain-nya dapat dicegah dengan WAF.

Lalu lintas yang masuk ke aplikasi web akan dipantau dan disaring oleh sistem WAF. WAF akan mengidentifikasi lalu lintas dengan menganalisis permintaan GET dan POST yang dikirimkan melalui HTTP dan HTTPS, dan kemudian menerapkan aturan *firewall* yang telah ditetapkan. Jika WAF mengidentifikasi lalu lintas yang mencurigakan atau lalu lintas yang berpotensi menimbulkan risiko keamanan ke situs web,

WAF akan memblokir dan mencegah akses tersebut (nkd, 2019).

Sebagai referensi untuk judul penelitian ini, telah dilakukan pencarian referensi terkait keunggulan dari WAF. Pada penelitian yang dilakukan (Robinson et al., 2018) yang berjudul *"SQL Injection and Cross Site Scripting Prevention Using OWASP Web Application Firewall",* menghasilkan kesimpulan bahwa *firewall* yang dirancang dapat 100% mendeteksi dan mengamankan *web application* dari *SQL Injection* setelah 15 kali percobaan serangan menggunakan 3 sistem operasi yang berbeda, namun gagal untuk mencegah serangan *Cross Site Scripting* kendati serangan-nya berhasil dideteksi. Selanjutnya pada penelitian yang dilakukan oleh (Rahmat et al., 2013) yang berjudul *"Sistem Pendeteksi dan Pencegahan Peretasan Terhadap Aplikasi Berbasis Web dengan Teknik Web Application Firewall (WAF)"*, memberikan hasil percobaan untuk pengujian dengan *tool* SQLMAP dengan 15 kali uji coba dengan parameter *risk* dan *level* yang berbeda menghasilkan akurasi deteksi rata-rata 99,49%. Sedangkan pada hasil percobaan untuk pengujian dengan *tool* XSSer dengan 98 injeksi pada *web application,* sistem mencatat 70 dari injeksi tersebut sebagai serangan, 12 injeksi gagal dideteksi dan 16 injeksi lainnya merupakan *request* biasa. Hasil tersebut menunjukkan bahwa 87.75% dari injeksi dapat dideteksi oleh WAF yang dirancang.

Berdasarkan latar belakang diatas, penulis memutuskan untuk membuat penelitian yang berjudul "**Perancangan *Web Application Firewall* (WAF) untuk mendeteksi dan mengantisipasi peretasan pada aplikasi berbasis web**". Pada penelitian yang akan dilakukan oleh penulis, pengujian performa WAF akan dilakukan dengan menggunakan *tools* yang berbeda dan *tools* yang sama dengan tujuan untuk mengetahui perbandingan performa WAF jika diuji menggunakan *tool* yang berbeda.

2. RESEARCH METHODS

2.1 Ruang Lingkup Penelitian

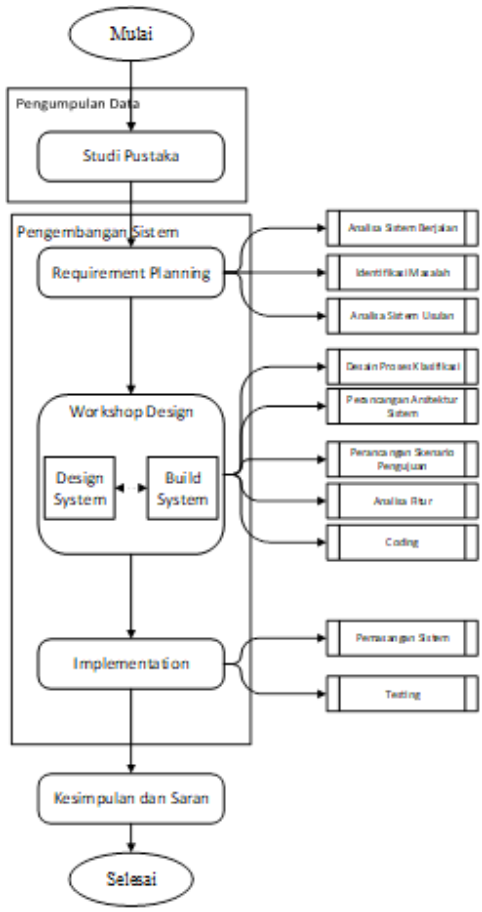
Ruang lingkup penelitian ini telah ditentukan agar penelitian lebih terarah, adapun ruang lingkup pada penelitian ini, yaitu perancangan *Web Application Firewall* untuk keamanan website. Agar ruang lingkupnya tidak terlalu melebar, pada penelitian ini juga ditetapkan batasan penelitian yaitu *Web Application Firewal* yang dibangun hanya akan diuji dengan jenis serangan *SQL Injection* dan *XSS.*

2.2 Pengumpulan Data

Teknik untuk mengumpulkan data diperlukan dalam sebuah penelitian, agar data dan teori yang terkandung di dalamnya valid dan juga sesuai dengan kenyataan, peneliti harus langsung ke sumbernya dan memahami teknik pengumpulan data. Dalam penelitian ini, penulis mengumpulkan data dengan menggunakan teknik dokumen.

2.3 Skema Penelitian

Adapun rincian dari tahapan yang ada pada gambar diatas adalah sebagai berikut:



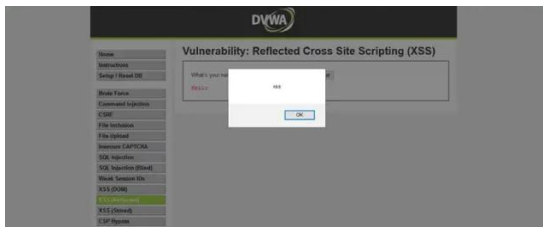
Gambar 1. Skema Penelitian

Studi pustaka dilakukan dengan cara menelusuri referensi yang relevan dengan penelitian yang dilakukan, referensi pada penelitian ini diperoleh dari jurnal, buku dan artikel yang bersumber dari internet.

Sistem yang berjalan dianalisis dengan menganalisis objek yang diperlukan untuk sistem yang akan dibuat, dengan tujuan berfokus pada fungsionalitas sistem yang sedang berjalan, tanpa menitik beratkan pada alur proses sistem. UML kemudian digunakan untuk visualisasi dan dokumentasi hasil analisis yang telah dilakukan.

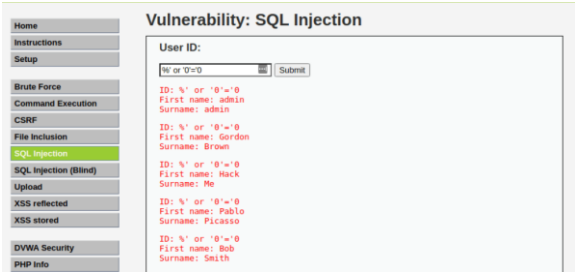
Pada penelitian ini sistem yang akan dijadikan sebagai bahan uji keamanan dari serangan berjenis *SQL Injection* dan *XSS* adalah situs web DVWA dan Wordpress. Sebagai awalan, dibawah ini merupakan kondisi yang dihasilkan setelah dilakukan kedua jenis serangan, sebelum *Web Application Firewal* diimpelementasikan pada server Apache dimana DVWA diinstall.

2.3.1 Hasil Serangan XSS pada DVWA dengan Payload



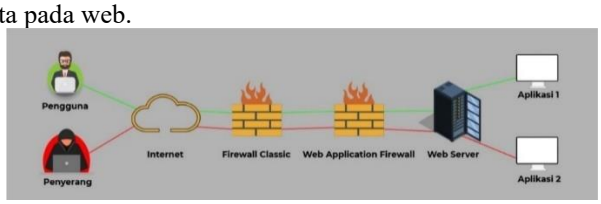
Gambar 2. Serangan XS

2.3.2 Hasil Serangan *SQL Injection* pada DVWA dengan Payload



Gambar 3. Serangan *SQL Injection* pada DVWA

Masalah yang ditemukan pada sistem berjalan adalah rentannya keamanan web DVWA untuk serangan *SQL Injection* dan *XSS.* Tujuan penggunaan DVWA sebagai contoh adalah untuk medeteksi kerentanan serangan terhadap server, karena DVWA memiliki tingkat keamanan yang dapat disesuaikan untuk kepentingan pengujian. Pada pengujian awal ini dibuktikan bahwa server yang belum diberikan fitur pengamanan rentan terhadap serangan, walaupun serangan seperti ini dapat diatasi dengan kode tertentu pada website yang dibangun, namun keamanan dari sisi server masih diperlukan untuk meminimalisir celah dan menjaga keamanan data pada web.



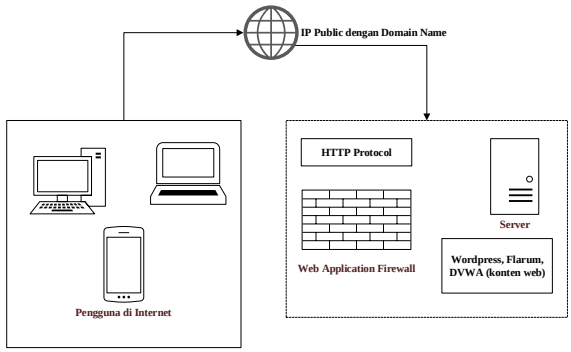
Gambar 4. Skema Sistem

Secara sederhana, gambar diatas menunjukkan skema dari sistem yang dibangun, dimana:

1. Pengguna, merupakan user yang mengakses web secara legal.
2. Penyerang, merupakan user yang terdeteksi sebagai akses yang berpotensi sebagai ancaman untuk sistem.
3. Firewall Classic, merupakan firewall yang ada pada masing-masing provider internet.
4. Web Application Firewall, merupakan Firewall yang digunakan pada website.
5. Aplikasi 1, menampilkan konten web yang semestinya. Sementara Aplikasi 2 menampilkan web halaman error
6. Garis merah, merupakan akses yang ditolak.
7. Garis hijau, merupakan akses yang diterima.

3. RESULT AND DISCUSSION

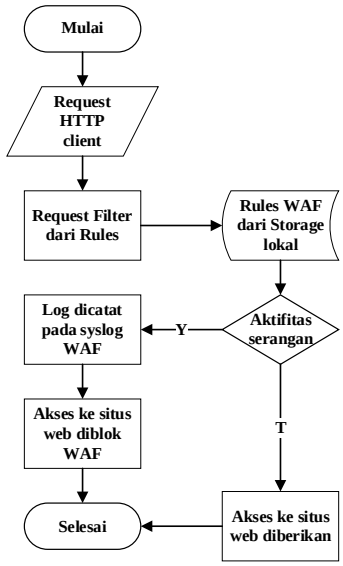
Pada penelitian ini, dengan menggunakan metode Naïve Fokus utama dalam penelitian ini adalah WAF yang akan diuji kemampuannya dalam mengamankan web server dan situs web yang ada pada web server tersebut. Situs web yang akan diamankan menggunakan beberapa aplikasi web seperti WordPress dan DVWA (*Damn Vulnerable Web App*) untuk sarana pengujian yang kemudian hasil dari pengujian WAF tersebut akan dianalisis. Gambar dibawah merupakan diagram blok sistem.



Gambar 5. Diagram Blok Sistem

3.1 Cara Kerja Sistem

WAF pada sistem ini secara umum memiliki tujuan melakukan fungsinya untuk menyeleksi request yang masuk, kemudian membedakan antara *client user* dan *attacker* yang mengakses situs web pada web server.

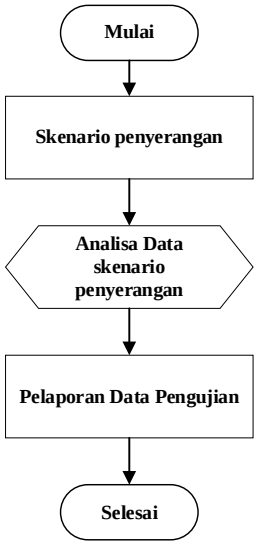


Gambar 6. Alur Kerja WAF (*Web Application Firewall*)

Pada penelitian ini alur kerja firewall akan digambarkan dengan flowchart untuk menyederhanakan penjelasan pada proses yang ada. Flowchart merupakan jenis diagram yang menggambarkan algoritma sekuensial sistem atau langkah-langkah instruksional. Flowchart biasanya digunakan dalam dokumentasi untuk menjelaskan gambaran logis dari sistem yang akan dibangun. Flowchart pada dasarnya digambarkan menggunakan simbol-simbol, setiap simbol mewakili proses tertentu, dan menghubungkan satu proses ke proses berikutnya digambarkan menggunakan garis penghubung.

3.2 Prosedur

Pada penelitian ini serangan akan dilakukan menggunakan *tools* yang telah tersedia. Pada jenis serangan *SQL Injection* digunakan *tool* ScanQLi dan SQLmap untuk menghasilkan serangan, sedangkan untuk jenis serangan XSS digunakan *tool* XSStrike dan XSSer untuk menghasilkan serangan.



Gambar 7. Alur Skenario Pengujian

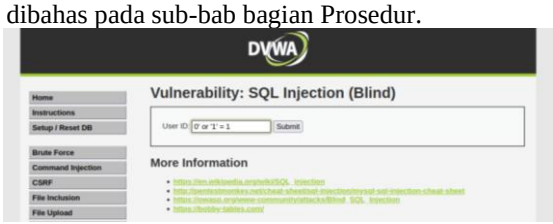
Pendeteksian dan antisipasi serangan *Cross Site Scripting* dan *SQL Injection* pada WAF yang dibangun dapat dilakukan dengan memanfaatkan sekumpulan *Rule* yang telah tersedia pada *OWASP Core Rule Set.*

3.3 Pengujian Sistem

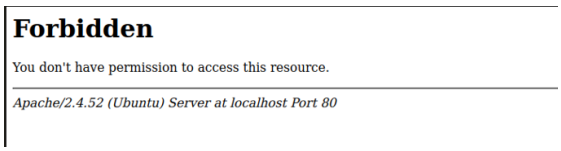
Tahapan dalam perancangan sebuah sistem keamanan tidak akan lengkap tanpa adanya bagian pengujian, pengujian sistem setelah pembuatan sistem merupakan bagian dimana sistem akan divalidasi untuk dapat berfungsi dan sesuai dengan target dari perancangan sistem.

3.3.1 Pengujian *SQL Injection*

Dalam penelitian ini WAF yang telah dibuat akan diuji kehandalan-nya dengan mengikuti prosedur pengujian yang telah dibahas pada sub-bab bagian Prosedur.



Gambar 8. Input *SQL Injection* DVWA



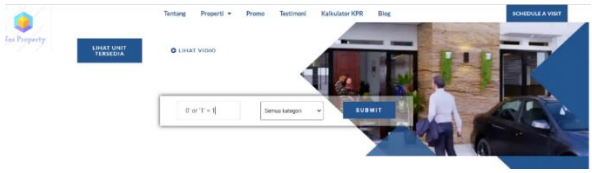
Gambar 9. Output *SQL Injection* DVWA

3.3.2 Pengujian menggunakan *Tools*

Gambar 10. Deteksi *SQL Injection* ModSecurity DVWA

Gambar 11. Pnegujian *SQL Injection* ScanQLi DVWA

3.3.3 Pengujian manual pada WordPress



Gambar 12. Input *SQL Injection* WordPress

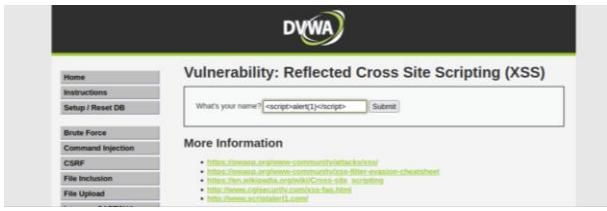
Gambar 13. Output *SQL Injection* WordPress

3.3.4 Pengujian menggunakan *Tools*

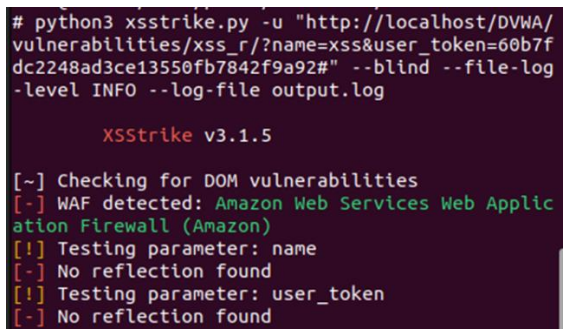
Gambar 14. Deteksi *SQL Injection* dengan ModSecurity

Gambar 15. Pengujian *SQL Injection* dengan ScanQLi

3.3.5 Pengujian XSS (*Cross Site Scripting*)



Gambar 16. Input XSS DVWA



Gambar 17. Pnegujian XSS dengan XSSer DVWA

3.3.6 Pengujian manual pada WordPress

Gambar 18. Input XSS

Gambar 19. Deteksi XSS ModSecurity

3.3.7 Pengujian pada WordPress menggunakan *Tools*

```
python3 xsstrike.py -u "http://localhost/desports/
?s=xss" --blind --file-log-level INFO --log-file o
utput.log

        XSStrike v3.1.5

[~] Checking for DOM vulnerabilities
[-] WAF detected: Amazon Web Services Web Applicat
ion Firewall (Amazon)
[!] Testing parameter: s
[!] Reflections found: 7
[~] Analysing reflections
[~] Generating payloads
[!] Payloads generated: 4608
 ~] Progress: 4608/4608
```

Gambar 20. Pengujian XSS dengan XSSer WordPress

3.4 Black box Testing

Tabel 1. Black Box Testing *SQL Injection* dan XSS

No.	Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	SQL injection (DVWA)	Input " 0' or '1' = 1" lalu proses	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid
2	SQL injection (WordPress)	Input" 0' or '1' = 1" lalu proses	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid
3	Deteksi XSS (Cross Site Scripting) (DVWA)	Input" <script>alert(1)</script>	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid
4	Deteksi XSS (Cross Site Scripting) (Wordpress)	Input' <script>alert(1)</script>"	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid

Tabel 2. Black-Box Testing *SQL Injection* (SQL Map)

No.	Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	SQL injection (DVWA)	Run tool SQLmap untuk uji serangan	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid
2	SQL injection (WordPress)	Run tool SQLmap untuk uji serangan	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid

Tabel 3. Black-Box Testing *SQL Injection*

No.	Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	SQL injection (DVWA)	Run tool ScanQLi untuk uji serangan	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid
2	SQL injection (WordPress)	Run tool ScanQLi untuk uji serangan	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid

Tabel 4. Black-Box Testing XSS (XSSer)

No.	Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	XSS (Cross Site Scripting) (DVWA)	Run tool XSSer untuk uji serangan	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid
2	XSS (Cross Site Scripting) (WordPress)	Run tool XSSer untuk uji serangan	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid

Tabel 5. Black-Box Testing XSS (XSSer)

No.	Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	XSS (Cross Site Scripting) (DVWA)	Run tool XSStrike untuk uji serangan	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid
2	XSS (Cross Site Scripting) (WordPress)	Run tool XSStrike untuk uji serangan	Deteksi dan blok akses oleh WAF	Sesuai harapan	Valid

3.5 Hasil Pnegujian Sistem

Pengujian WAF telah dilakukan pada bagian sebelumnya menggunakan beberapa *tools*, selanjutnya pada bagian ini akan dijelaskan secara lebih rinci mengenai hasil pengujian tersebut.

3.5.1 *SQL Injection*

1. Menggunakan Browser (Manual)

Cara pengujian ini yaitu dengan menginputkan baris *script* secara manual melalui browser. Pengujian dilakukan terhadap WAF untuk melindungi ketiga aplikasi web yang digunakan sebagai target, menunjukkan bahwa WAF berhasil melindungi aplikasi web dari serangan *SQL Injection* dengan mendeteksi dan melakukan blok kepada akses yang berbahaya. Pada pengujian jenis ini WAF berhasil mendeteksi dan melakukan blok terhadap seluruh akses yang dianggap berbahaya.

Tabel 6. Script (*SQL Injection*)

No.	Script	Hasil
1	1 or '1' = '1	Deteksi dan blok akses oleh WAF
2	1 or '1' = '1 UNION SELECT * from password	Deteksi dan blok akses oleh WAF
3	UNION SELECT user, password FROM users#	Deteksi dan blok akses oleh WAF

2. Pengujian menggunakan Tools (SQL map, ScanQLi)

Cara pengujian ini yaitu dengan menjalankan *tools* untuk melakukan serangan terhadap aplikasi web yang dilindungi oleh WAF. Pada pengujian serangan yang dilakukan menggunakan *tool* SQLmap, menghasilkan data sebagai berikut.

Tabel 7. Hasil Pengujian Tools SQLmap

Aplikasi	Level	Risk	Not Found	403 (Forbidden)	Sukses	Persentase Proteksi
DVWA	3	3	1917	3917	0	100%
Wordpress	3	3	1917	3918	0	100%

Selanjutnya pada pengujian serangan yang dilakukan menggunakan *tool* ScanQLi memberikan hasil, WAF berhasil mendeteksi dan melakukan blok terhadap seluruh serangan yang dilakukan. Pengujian yang dilakukan menggunakan *tool* ini kurang beragam karena sedikitnya fitur yang terdapat pada *tool* ini jika dibandingkan dengan SQLmap.

Dari hasil pengujian yang dilakukan menggunakan dua *tools* tersebut dapat diketahui bahwa WAF yang dirancang dapat mendeteksi dan mengantisipasi mayoritas serangan.

3.5.2 XSS (*Cross Site Scripting*)

1. Menggunakan Browser (Manual)

Cara pengujian ini yaitu dengan menginputkan baris *script* secara manual melalui browser. Pengujian dilakukan terhadap WAF untuk melindungi ketiga aplikasi web yang digunakan sebagai target, menunjukkan bahwa WAF berhasil melindungi aplikasi web dari serangan XSS dengan mendeteksi dan melakukan blok kepada akses yang berbahaya. Pada pengujian

jenis ini WAF berhasil mendeteksi dan melakukan blok terhadap seluruh akses yang dianggap berbahaya.

Tabel 8. Script (XSS)

No.	Script	Hasil
1	<script>alert(1)</script>	Deteksi dan blok akses oleh WAF
2	<script>alert(console.log)</script>	Deteksi dan blok akses oleh WAF
3	<script>alert(document.domain)</script>	Deteksi dan blok akses oleh WAF
4	<script>alert(window.origin)</script>	Deteksi dan blok akses oleh WAF
5	<script>alert(window.document.cookie)</script>	Deteksi dan blok akses oleh WAF

2. Menggunakan Tools (*XSSer, XSStrike*)

Cara pengujian ini yaitu dengan menjalankan *tools* untuk melakukan serangan terhadap aplikasi web yang dilindungi oleh WAF. Pada pengujian serangan yang dilakukan menggunakan *tool* XSSer, menghasilkan data sebagai berikut:

Tabel 9. Hasil pengujian Tools XSSer

Aplikasi	Total Injeksi	Gagal	Sukses	Persentase Proteksi
DVWA	1291	1291	0	100%
Wordpress	1291	1282	9	99,30%

Selanjutnya pada pengujian serangan yang dilakukan menggunakan tool XSStrike memberikan data sebagai berikut:

Tabel 10. Hasil pengujian Tools XSSer

Aplikasi	Total Injeksi	Gagal	Sukses	Persentase Proteksi
DVWA	0	0	0	100%
Wordpress	4608	4608	0	100%

Dari hasil pengujian yang dilakukan menggunakan dua *tools* tersebut dapat diketahui bahwa WAF yang dirancang dapat mendeteksi dan mengantisipasi mayoritas serangan dengan persentase rata-rata 99,8% dan hanya pada pengujian menggunakan *tool* XSSer didapati injeksi yang sukses dilakukan.

4. CONCLUSION

WAF (Web Application Firewall) yang dirancang menggunakan ModSecurity dan OWASP Core Rule Set dapat berfungsi dengan baik, yaitu dengan mendeteksi dan melakukan blok terhadap serangan *SQL Injection* dan XSS (Cross Site Scripting), hasil deteksi serangan ditampilkan pada file log.

Dari pengujian yang dilakukan, WAF yang dirancang memiliki persentase antisipasi serangan sebesar 100% untuk jenis serangan *SQL Injection*, sedangkan untuk jenis serangan XSS sebesar 99,8%.

Dari percobaan yang dilakukan juga didapati bahwa keamanan bawaan yang terdapat pada aplikasi web yang digunakan dalam penelitian ini juga berpengaruh terhadap proteksi terhadap serangan *SQL Injection* dan XSS, hal ini ditunjukkan dengan adanya perbedaan total injeksi dan jumlah injeksi sukses oleh *tools*.

REFERENCES

[1] Zulkarnain, A. 2021. Optimalisasi Penanggulangan Kebocoran Data Melalui Regulatory Blockchain Guna Mewujudkan Keamanan Siber di Indonesia. Jurnal Keamanan Siber dan Informasi, Institut Teknologi Bandung.

[2] Suryadi, B. 2021. Tanggung Jawab PT Tokopedia Dalam Kasus Kebocoran Data Pribadi Pengguna. Jurnal Hukum dan Teknologi Informasi, Universitas Indonesia.

[3] Wijaya, C. & Putra, D. E. 2020. Peningkatan Keamanan Aplikasi Web Menggunakan Web Application Firewall (WAF) Pada Sistem Informasi Manajemen Kampus Terintegrasi. Jurnal Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember.

[4] Hartanto, D. 2023. Analisis dan Implementasi Web Application Firewall dalam Meningkatkan Keamanan Aplikasi Web. Jurnal Keamanan Siber, Universitas Teknologi Yogyakarta.

[5] Aslan, I., Bahtiar, H., & Sudianto, A. 2024. Pengembangan Website Fakultas Teknik Universitas Hamzanwadi Berbasis Progressive WEB APP (PWA). Jurnal Teknologi Informasi dan Komunikasi, Universitas Hamzanwadi.

[6] Isnaini, H. A., & Nafisah, S. 2023. Analisis Elemen Kunci Website Berdasar Konsep Shedroff pada Website Perpustakaan Universitas Islam Indonesia. Jurnal Ilmu Perpustakaan dan Informasi, Universitas Islam Indonesia.

[7] Trimarsiah, Y., & Arafat, M. 2021. Analisis dan Perancangan Website sebagai Sarana Informasi pada Lembaga Bahasa Kewirausahaan dan Komputer AKMI Baturaja. Jurnal Teknologi Informasi dan Komunikasi, AMIK AKMI Baturaja.

[8] Chandra, A. Y. 2019. Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request. Jurnal Sistem dan Informatika (JSI), 14(1), 48-56.

[9] Retno, S., Hasdyna, N., Mutasar, M., & Dinata, R. (2020). Algoritma Honey Encryption dalam Sistem Pendataan Sertifikat Tanah dan Bangunan di Universitas Malikussaleh. INFORMAL: Informatics Journal, 5(3), 87 - 95. doi:10.19184/isj.v5i3.20804.

[10] Yondra, A. S., Triyanto, D., & Bahri, S. 2021. Implementasi Web Scraping untuk Mengumpulkan Informasi Produk dari Situs E-Commerce dan Marketplace dengan Teknik Pemrosesan Paralel. Jurnal Rekayasa Sistem Komputer, Universitas Tanjungpura.

[11] Yuwinanto, H. P. 2021. Privasi Online dan Keamanan Data. Jurnal Ilmiah Teknologi Informasi Asia, Universitas Airlangga.

[12] Nugroho, V. S., & Christanto, F. W. 2023. Analisis Keamanan Website dengan Information System Security Assessment Framework (ISSAF) dan Open Web Application Security Project (OWASP). Jurnal Ilmiah NERO, 8(2), 145-156.

[13] Septiawan, G. A., Irawan, K. W. S., Mayasari, I., & Listartha, I. M. E. 2021. Analisis Kerentanan XSS dan Rate Limiting Menggunakan Framework OWASP ZAP pada Website SMAN 8 Denpasar. Jurnal Ilmiah Teknologi Informasi, Universitas Pendidikan Ganesha.

[14] Hany, M. I., Bhawiyuga, A., & Kusyanti, A. 2021. Implementasi Cross Site Scripting Vulnerability Assessment Tools berdasarkan OWASP Code Review.

Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
[15] Pratama, T. I. M., Songida, M. D. F., & Gunawan, I. 2022. Analisis Serangan dan Keamanan pada *SQL Injection*: Sebuah Review Sistematik. Jurnal Ilmiah Informatika & Komputer, Sekolah Tinggi Teknologi Ronggolawe.
[16] Sugiyono. 2021. Sistem Keamanan Jaringan Komputer Menggunakan Metode Watchguard Firebox pada PT Guna Karya Indonesia. Jurnal Teknik Informatika, STIKOM Cipta Karya Informatika.
[17] Riska, & Alamsyah, H. 2023. Penerapan Sistem Keamanan WEB Menggunakan Metode WEB Aplication Firewall. Jurnal Keamanan Siber, Universitas Dehasen Bengkulu.
[18] Kuncoro, A. W. 2021. Analisis Metode Open Web Application Security Project (OWASP) pada Pengujian Keamanan Website: Literature Review. Jurnal Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
[19] Aji, S., Pratmanto, D., Ardiansyah, A., & Saifudin. 2021. Implementasi Framework Laravel dalam Perancangan Sistem Informasi Desa. Indonesian Journal on Software Engineering (IJSE), 7(2), 237-246.
[20] Tabrani, M., Suhardi, & Priyandaru, H. 2021. Sistem Informasi Manajemen Berbasis Website dengan Menggunakan Framework CodeIgniter. Jurnal Ilmiah M-Progress, Universitas Bina Sarana Informatika.
[21] Suliyanti, W. N. 2021. Studi Literatur Basis Data SQL dan NoSQL. Jurnal Teknik Informatika, Sekolah Tinggi Teknik PLN.
[22] Wamiliana, Wisnu Wardhana, dan Fahmi Kharismaldie. 2013. Pembangunan Sistem Operasi Berbasis Linux Menggunakan Metode Linux From Scratch. Jurnal Komputasi, FMIPA Unila.