Analisis Kerentanan *Web* DVWA Yang Menggunakan *Web Application Firewall* Dengan Menggunakan Standar OWASP

1st Mochamad Rizal Sumpena
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
mochamadrizals@student.telkomuniversity.ac.id

2nd Nyoman Bogi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
nyomanbogi@telkomuniversity.ac.id

3rd Sofia Naning
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
sofiananing.hertiana@telkomuniversity.ac.id

***Abstrak* — Keamanan aplikasi web sangat penting untuk menghindari serangan dan kerentanan yang dapat digunakan oleh pihak yang tidak bertanggung jawab. Dalam rangka meningkatkan keamanan aplikasi *web*, penggunaan *Web Application Firewall* (WAF) telah menjadi praktik umum untuk mengidentifikasi, memblokir, dan melindungi aplikasi web dari serangan yang berpotensi merusak. Penelitian ini bertujuan untuk melakukan analisis kerentanan pada aplikasi *web* DVWA (*Damn Vulnerable Web Application*) dan mengimplementasikan WAF menggunakan standar OWASP (*Open Web Application Security Project*). DVWA adalah aplikasi *web* yang sengaja dibuat dengan kerentanan yang dapat dimanfaatkan untuk tujuan pengujian keamanan. Hasil dari penelitian ini menunjukkan bahwa DVWA memiliki berbagai kerentanan yang dapat dieksploitasi oleh serangan-serangan umum. Namun, setelah mengimplementasikan WAF OWASP, kerentanan yang ada dapat dikurangi atau dihilangkan sepenuhnya, sehingga meningkatkan keamanan aplikasi *web* DVWA. Penelitian ini memberikan kontribusi penting dalam pemahaman tentang pentingnya penggunaan WAF dalam melindungi aplikasi *web* dari serangan-serangan yang berpotensi merusak. Selain itu, penggunaan standar OWASP memberikan pedoman yang kuat dalam mengimplementasikan kebijakan keamanan yang efektif.**

***Kata kunci*— Web Application Firewall (WAF), DVWA, OWASP, Kerentanan**

I. PENDAHULUAN

Seiring dengan perkembangan teknologi informasi yang semakin pesat dan dinamis, tingkat kejahatan siber juga semakin meningkat dewasa ini. Kejahatan siber (*cyber crime*) yang dimaksud ialah percobaan serangan terhadap suatu keamanan sistem informasi. Menurut Ditjen Aptika Kominfo, aktivitas yang dilakukan oleh sekelompok orang yang ingin menembus suatu sistem keamanan bertujuan untuk mendapatkan, mengubah, mencari, dan bahkan menghapus informasi yang ada pada sistem tersebut jika memang benar-benar dibutuhkan [1].

Pelaku kejahatan siber biasanya menyerang suatu sistem keamanan informasi yang memuat data-data penting dan rahasia. Biasanya penyerangan yang sering terjadi dilakukan terhadap aplikasi *web*. Banyak pengembang aplikasi *web* yang kurang memperhatikan sisi keamanan aplikasi *web* sehingga banyak dieksploitasi oleh para *hacker*. Menurut hasil *Web Security Report* SiteLock pada tahun 2022 , didapatkan 172 serangan dalam sehari untuk satu *website* dan *website* diakses oleh *bot* sebanyak 2306 kali per minggu. *Bot* digunakan oleh hacker untuk mencari kelemahan situs web. Jumlah serangan di tahun 2022 meningkat sebanyak 210% dibandingkan tahun 2020. Terdapat 2 jenis kerentanan tertinggi yang tercatat yaitu *Cross Site Scripting* (XSS) sebanyak 1 juta halaman *website*, dan SQL *Injection* sebanyak 332 ribu halaman *website*. Untuk mengatasi berbagai permasalahan terkait keamanan aplikasi *web* diperlukan suatu sistem yang dapat mencegah serangan berbahaya.

WAF atau *Web Application Firewall* adalah perangkat keamanan yang digunakan untuk melindungi aplikasi *web* dari serangan jaringan yang tidak sah. WAF menganalisis lalu lintas jaringan yang masuk ke aplikasi web dan mengeliminasi lalu lintas yang tidak diinginkan atau merusak sebelum lalu lintas tersebut sampai ke aplikasi. WAF dapat digunakan sebagai sistem yang dapat mencegah serta mendeteksi serangan *SQL injection*, *Cross Site Scripting* (XSS), dan *Local File Inclusion* (LFI) dengan menggunakan *detection rules* yang telah ditetapkan untuk dapat memblokir akses bagi penyerang ke dalam *website* [2].

DVWA, juga dikenal sebagai *Damn Vulnerable Web Application*, dirancang untuk membantu para profesional keamanan mengevaluasi kemampuan mereka. Selain itu, DVWA cocok bagi mereka yang ingin mempelajari teknik *web-hacking* terhadap aplikasi PHP/MySQL, seperti SQL *Injection* dan *Remote Command Execution* [3].

OWASP atau *Open Web Application Security Project* adalah organisasi nirlaba yang bertujuan untuk membantu para pengembang perangkat lunak dalam membangun aplikasi *web* yang aman. OWASP menyediakan berbagai sumber daya keamanan *web* yang berguna, termasuk dokumentasi, *tools*, dan proyek *open source* yang dapat digunakan oleh para pengembang untuk membangun aplikasi *web* yang aman. OWASP *Risk Rating Methodology* adalah

jenis penilaian resiko kerentanan keamanan aplikasi berbasis *web*. Mengidentifikasi efek negatif yang dihasilkan dari analisis kerentanan merupakan langkah penting dalam menilai resiko [4].

II. KAJIAN TEORI

Terdapat beberapa penelitian yang membahas mengenai penggunaan *Web Application Firewall* (WAF) dengan standar OWASP pada sebuah aplikasi *web*.

A. Implementasi dan Analisis *Open Source ModSecurity* WAF pada Aplikasi Berbasis *Web* dengan Standar OWASP (Kirana Dhiatama Ayunda, Adityas Widjajarto, Avon Budiyono*)*

Menurut penelitian ini, penggunaan *web application firewall* adalah kebutuhan untuk melindungi aplikasi berbasis *web* dari serangan. Penerapan *web application firewall* pada aplikasi berbasis *web* dapat mengurangi serangan. Dari perspektif keamanan, penting untuk mengetahui seberapa efektif penggunaan *firewall* aplikasi *web* untuk mencegah serangan berbahaya dari aplikasi berbasis *web*. Dengan menggunakan aplikasi berbasis *web* yang dilindungi atau tidak dilindungi oleh *web application firewall*, penelitian ini menggunakan metode ilmiah yang sesuai dengan standar OWASP. *Web application firewall* berhasil melindungi aplikasi *web* yang rentan sebesar 83% dari enam percobaan dan dapat melindungi lima kerentanan dengan risiko tinggi. Upaya pencegahan dan penilaian tingkat keamanan berdasarkan *Common Vulnerability and Exposures* (CVE) dapat digunakan untuk melindungi kerentanan ini [5].

B. Implementasi Keamanan Aplikasi *Web* Dengan *Web Application Firewall* (Risma Yanti Jamain, Periyadi, S.T., M.T., Setia Juli Irzal Ismail, S.T., M.T.)

Sebagai bagian dari sistem, nginx berfungsi sebagai *web server* dan naxsi berfungsi sebagai *web application firewall*. *Firewall* ini bertanggung jawab untuk memfilter konten yang melaluinya dan meblokir konten yang dianggap berbahaya sesuai dengan peraturan yang telah ditetapkan. Dengan menggunakan *web application firewall*, naxsi dapat digunakan untuk mencegah serangan SQL *Injection*, *Cross-Site Scripting*, dan *Command Execution* [3].

III. METODE

Terdapat beberapa tahapan dalam penelitian yang dilakukan.

A. Studi Literatur

Tahap awal adalah tahap sebelum penelitian dilakukan dengan mengidentifikasi masalah beberapa masalah yang timbul. Permasalahan yang timbul dari fakta-fakta yang ada adalah seiring dengan meningkatnya pengguna internet di Indonesia, maka semakin meningkat pula serangan siber yang ditujukan serangannya kepada website yang dimiliki oleh perusahaan/organisasi yang berada di Indonesia. Banyak *website* digunakan oleh perusahaan/organisasi untuk melakukan transaksi maupun pelayanan untuk para klien. Serangan yang terjadi dapat menyebabkan bocornya informasi data diri dari klien maupun perusahaan/organisasi yang tersimpan didalam *server*.

B. Tahap perancangan sistem

Pada tahap ini, membuat rancangan sesuai dengan solusi yang ditemukan pada tahap sebelumnya untuk menjawab dari permasalahan yang timbul. Permasalahan yang timbul dari fakta-fakta yang ditemukan adalah peningkatan pengguna internet membuat semakin banyaknya serangan siber yang terjadi di Indonesia terutama serangan yang menuju *website*. Oleh karena itu, solusi yang dapat diimplementasikan oleh peneliti adalah membuat *web application firewall* berdasarkan *rule proxy*. Sehingga dapat mencegah serangan yang akan dilakukan ke *server* dan akan melakukan pencatatan IP *address* yang melakukan serangan ke *database* jika terdapat serangan.

C. Tahap Pengujian Sistem

Pada Pada tahap ini, terdapat proses eksploitasi dan akan memiliki dua kondisi untuk melihat kondisi kerentanan pada *web* DVWA. Kondisi pertama adalah aplikasi *web* DVWA tidak akan menggunakan WAF dan akan diserang dengan menggunakan tiga serangan yaitu SQL *Injection*, XSS, serta *Local File Inclusion* . Kondisi kedua adalah aplikasi *web* DVWA akan menggunakan WAF dan akan diserang dengan menggunakan tiga serangan yaitu SQL *Injection*, XSS, serta *Local File Inclusion*.

D. Tahap Analisis

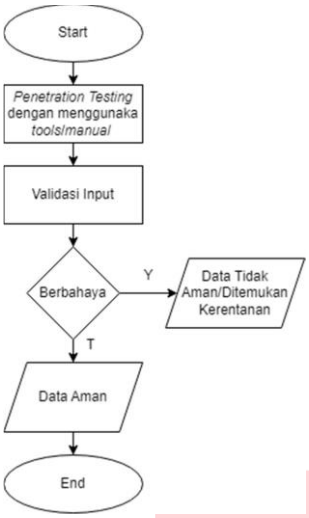
Pada tahap ini, proses analisis dilakukan terhadap hasil percobaan yang telah dilakukan pada tahap pengujian. Analisis kerentanan menggunakan CVSS *Calculator* 3.0 untuk menilai tingkat keparahan kerentanan. Akan dilakukan perbandingan hasil dari skenario pengujian kondisi pertama dan kondisi kedua.

E. Tahap Kesimpulan

Pada Hasil akhir dari penelitian berupa laporan penelitian, dengan langkah terakhir penarikan kesimpulan yang diperoleh dari hasil analisis penelitian yang dilakukan untuk mengetahui seberapa pentingnya implementasi WAF dalam sebuah aplikasi *web.*

IV. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan cara kerja *Damn Vulnerable Web Application* (DVWA). Dalam sistem ini, DVWA digunakan sebagai objek untuk melakukan *penetration testing* sebagai langkah untuk mengevaluasi sejauh mana WAF dapat melindungi sebuah aplikasi *web*. *Penetration testing* pada *web* adalah proses pengujian keamanan yang bertujuan untuk mengidentifikasi, mengeksplorasi, dan mengevaluasi kerentanan yang ada pada aplikasi *web*. Tujuan utama dari *penetration testing* adalah untuk menguji sejauh mana sistem dapat bertahan terhadap serangan dan mendapatkan pemahaman yang mendalam tentang kerentanan yang ada. Cara kerja DVWA akan dijelaskan melalui diagram alur atau *flowchart*.



GAMBAR 4.1
Cara Kerja DVWA

Berdasarkan Gambar 4.1, DVWA akan melakukan validasi *input* yang berasal dari injeksi kode berbahaya pada proses *penetration testing*. Apabila pada hasil validasi didapatkan bahwa *input* merupakan kode berbahaya maka *web* akan menampilkan data-data rahasia sehingga dapat dikatakan bahwa data dalam kondisi tidak aman dan terdapat kerentanan dalam aplikasi *web*. Jika hasil validasi menyatakan tidak terdapat kode berbahaya pada input maka data dalam kondisi aman dan aplikasi *web* hanya akan menampilkan data yang dapat diakses oleh publik tanpa membuka data rahasia.

Langkah pengujian dilakukan dengan skenario sistem tidak menggunakan *Web Application Firewall* (WAF) agar ketika DVWA diserang akan menghasilkan kerentanan yang dapat dieksploitasi oleh *hacker*. Sebelumnya atur *Security level* pada DVWA di *level low*. Penyerang akan memasukkan berbagai macam *payload* serangan seperti SQL *Injection*, *Cross Site Scripting* (XSS), dan *Local File Inclusion*. Setelah memasukkan *payload* serangan, akan dilakukan proses asesmen terhadap kerentanan dengan menggunakan acuan *Common Vulnerability Scoring System* (CVSS) *Calculator* 3.0 untuk menilai tingkat keparahan kerentanan. *Common Vulnerability Scoring System* (CVSS) *Calculator* 3.0 adalah alat yang digunakan untuk menghitung skor kerentanan berdasarkan standar CVSS versi 3.0. CVSS adalah metode yang secara objektif dan konsisten digunakan untuk mengukur tingkat kerentanan suatu sistem atau perangkat lunak. CVSS *Calculator* 3.0 mempertimbangkan beberapa metrik yang berhubungan dengan kerentanan, seperti tingkat kompleksitas serangan, dampak yang mungkin terjadi, dan tingkat keamanan yang ada.

Berikut adalah hasil dari asesmen DVWA dengan menggunakan CVSS *Calculator* 3.0:

TABEL 4.1
Asesmen DVWA

1	*Finding*	SQL *Injection*
	Risk Rating	7.3 (HIGH)
	CVSS *Vector*	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L
	Description	SQL *Injection* adalah serangan keamanan yang dilakukan dengan memanipulasi atau menyisipkan kode SQL berbahaya ke dalam *input* yang diterima oleh aplikasi yang berinteraksi dengan *database*. Serangan ini memanfaatkan kelemahan dalam validasi *input* pada aplikasi *web* yang menggunakan bahasa SQL untuk berinteraksi dengan *database*.
	PoC	Pada salah satu fitur DVWA dengan url http://localhost/DVWA/vulnerabilities/sqli/ dapat dilakukan SQL *Injection* dengan menggunakan berbagai *payload* serangan SQL *Injection* seperti *payload* ' or '1'='1 yang digunakan di bawah ini. 

TABEL 4.1
Asesmen DVWA (Lanjutan)

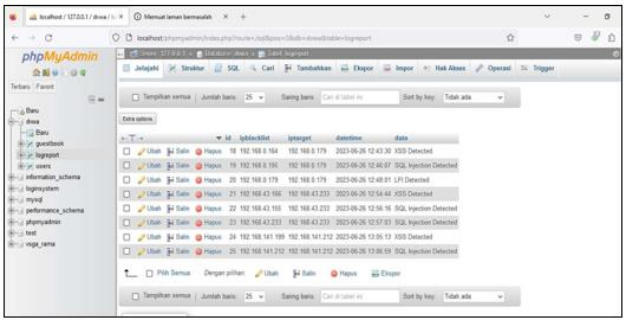
2	*Finding*	*Cross-Site Scripting* (XSS)
	Risk Rating	6.1 (MEDIUM)
	CVSS *Vector*	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
	Description	*Cross Site Scripting* (XSS) adalah serangan keamanan yang memanfaatkan celah dalam aplikasi *web* untuk menyisipkan skrip berbahaya (*script*) ke dalam halaman *web* yang dilihat oleh pengguna lain. Serangan ini memungkinkan penyerang untuk menjalankan skrip yang tidak sah pada *browser* pengguna lain, yang dapat mengakibatkan pencurian informasi, pengambilalihan akun, atau pengrusakan tampilan halaman.
	PoC	Pada salah satu fitur DVWA dengan url http://localhost/DVWA/vulnerabilities/xss_r/ dapat dilakukan *Cross Site Scripting* dengan menggunakan berbagai *payload* serangan XSS seperti *payload* <script>alert(document.cookie)</script> yang digunakan di bawah ini. 

TABEL 4.1
Asesmen DVWA (Lanjutan)

3	*Finding*	*Local File Inclusion*
	Risk Rating	7.3 (HIGH)
	CVSS *Vector*	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L
	Description	*Local File Inclusion* (LFI) adalah serangan keamanan yang memanfaatkan celah dalam aplikasi *web* untuk memasukkan dan mengakses file lokal yang ada di *server web*. Serangan ini terjadi ketika aplikasi *web* memperbolehkan pengguna untuk menyertakan *file* lokal ke dalam halaman *web* secara tidak aman.
	PoC	Pada salah satu fitur DVWA dengan url http://localhost/DVWA/vulnerabilities/fi/?page=file1.php dapat dilakukan LFI dengan menggunakan berbagai *payload* serangan LFI seperti *payload* ../../../../../../../../../etc/passwd yang digunakan di bawah ini. 

Berdasarkan Tabel 4.1, kerentanan SQL *Injection* menghasilkan tingkat resiko yang tergolong tinggi, XSS menghasilkan tingkat resiko yang tergolong menengah, dan LFI menghasilkan tingkat resiko yang juga tergolong tinggi. Semua perhitungan tingkat resiko didasarkan pada CVSS *Calculator* 3.0. Ketiga serangan diuji termasuk ke dalam OWASP TOP 10 2021 dimana SQL *Injection* termasuk ke dalam daftar A03:2021 – *Injection*, XSS termasuk ke dalam daftar A03:2021 – *Injection*, dan LFI termasuk ke dalam daftar A08:2021 – *Software and Data Integrity Failures*.

Dilakukan pengujian pada kondisi kedua yaitu aplikasi *web* DVWA menggunakan WAF. Pengujian menghasilkan daftar pemblokiran percobaan serangan pada *database* WAF yang terdapat dalam web DVWA.



GAMBAR 4.2
Database Pemblokiran IP

Berdasarkan gambar 4.2, WAF berhasil memblokir percobaan serangan terhadap web DVWA dengan menggunakan *payload* serangan yang berbahaya. Pemblokiran dilakukan dengan mencatat IP *address* yang digunakan oleh penyerang untuk mengakses *web* DVWA.

GAMBAR 4.3
Pemutusan Sambungan Penyerang

Berdasarkan Gambar 4.3, ketika dilakukan pemblokiran oleh WAF, sambungan penyerang terhadap *web* DVWA akan diputus sehingga penyerang tidak dapat mengakses *web* DVWA kembali.

V. KESIMPULAN

Berdasarkan hasil pengujian yang telah diperoleh dengan mencoba dua skenario pengujian, Tiga kerentanan yang diuji dapat diantisipasi dengan menerapkan *Web Application Firewall* (WAF) pada sistem. Apabila WAF tidak digunakan pada sistem maka ketiga kerentanan tersebut dapat mengancam keamanan sistem.

REFERENSI

[1] https://aptika.kominfo.go.id/2017/06/keamanan jaringan-internet-dan-firewall/ diakses pada tanggal 24 oktober 2022.

[2] Robinson, M. Akbar, and M. A. F. Ridha, “SQL Injection and Cross Site Scripting Prevention Using OWASP Web Application Firewall,” *Int. J. Informatics Vis., vol. 2, pp. 286-292, 2018.*

[3] R. N. Jamain, Periyadi, S. J. I. Ismail, “Implementasi Keamanan Aplikasi Web Dengan Web Application Firewall”, e-Proceeding of Applied Science, vol.1, no. 3, 2015.

[4] R.H. Hutagalung, L.E Nugroho, R. Hidayat, “Menentukan Dampak Resiko Keamanan Berbasis Pendekatan OWASP”, Prosiding SNATI F Ke-4 , Kudus, 2017.

[5] K. D. Ayunda, A. Widjajarto, A. Budiyono, “Implementasi dan Analisis Open Source ModSecurity WAF pada Aplikasi Berbasis Web dengan Standar OWASP”, Bandung: Universitas Telkom, 2021.