DESAIN DAN IMPLEMENTASI *WEB APPLICATION FIREWALL* DAN *RATE LIMITING* UNTUK *CYBER DEFENSE*

DESIGN AND IMPLEMENTATION OF WEB APPLICATION FIREWALL AND RATE LIMITING FOR CYBER DEFENSE

I Gusti Ngurah Bagus Dimas Wiradyaksa[1], Dwi Haura Putri[2], Roihan Muhammad Iqbal[3] Novia Helni Astari[4]

[1]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[2]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[3]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[4]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[1]**ignbagusdimasw@student.telkomuniversity.ac.id,** [2]**hauura@student.telkomuniversity.co.id,**
[3]**roihanmiqbal@student.telkomuniversity.ac.id,**
[4]**noviahelniastari@student.telkomuniversity.ac.id**

Abstrak

Aplikasi web sebagai salah satu media digital yang sering menjadi target serangan siber, mengalami sekitar 75% dari total serangan siber. Tahun 2023, sebanyak 30.000 situs web diretas setiap hari, menggarisbawahi kerentanan aplikasi web karena ketersediaannya yang harus konstan bagi pengguna. Salah satu jenis ancaman keamanan aplikasi web adalah adanya *request* berisi *payload* berbahaya seperti SQLi dan XSS ke aplikasi web. Jenis ancaman keamanan aplikasi web lainnya adalah DDoS. Serangan ini membuat situs web tidak berfungsi atau tidak dapat diakses. Untuk mengatasi masalah ini, dirancang sebuah solusi berupa desain dan implementasi WAF dan *rate limiting* pada keamanan aplikasi web. Berdasarkan hasil pengujian, WAF mendapatkani nilai *security quality* sebesar 100% dan *detection quality* sebesar 100%. Rata-rata *throughput* dengan *rate limiting* sebesar 8.092,972, sedangkan rata-rata *throughput* tanpa *rate limiting* sebesar 7.516,592. Rata-rata *packet loss* dengan *rate limiting* sebesar 0.419% sedangkan rata-rata *packet loss* tanpa *rate limiting* sebesar 0.408%. Rata-rata *delay* dengan *rate limiting* sebesar 0,000284814 sedangkan rata-rata *delay* tanpa *rate limiting* sebesar 0,000714118. Rata-rata jitter dengan *rate limiting* sebesar 0,000285191 sedangkan rata-rata jitter tanpa *rate limiting* sebesar 0,000714252. 3. Pada IDS, snort dapat mengirimkan pesan peringatan sesuai jenis serangan yang terjadi seperti SQLi, XSS, DDoS.

Kata kunci : WAF, SQLi, XSS, DDoS dan IDS

Abstract

Web applications, as one of the digital media that are often targeted by cyberattacks, experience around 75% of total cyberattacks. In 2023, as many as 30,000 websites are hacked every day, underscoring the vulnerability of web applications due to their availability that must be constant for users. One type of web application security threat is the existence of requests containing malicious payloads such as SQLi and XSS to web applications. Another type of web application security threat is DDoS. These attacks make the website dysfunctional or inaccessible. To overcome this problem, a solution was designed in the form of WAF design and implementation and rate limiting on web application security. Based on the test results, WAF obtained a security quality score of 100% and detection quality of 100%. The average throughput with rate limiting was 8,092,972, while the average throughput without rate limiting was 7,516,592. The average packet loss with rate limiting is 0.419% while the average packet loss without rate limiting is 0.408%. The average delay with rate limiting is 0.000284814 while the average delay without rate limiting is 0.000714118. The average jitter with rate limiting is 0.000285191 while the average jitter without rate limiting is 0.000714252. 3. In IDS, snort can send warning messages according to the type of attack that occurs such as SQLi, XSS, DDoS.

Keywords: WAF, SQLi, XSS, DdoS dan IDS

1. Pendahuluan

Berdasarkan data dari Statista tentang Number of Cyber Attack Cases in Indonesia from 2019 to 2022, pada tahun 2022 tercatat sekitar 976 juta kasus serangan siber di Indonesia yang

mengakibatkan penurunan lebih dari 600 juta kasus dibandingkan tahun sebelumnya. Menurut National Cyber Security Index Ranking, skor Indonesia terkait kebijakan keamanan siber masih relatif rendah. Hal ini menunjukkan adanya ruang untuk dilakukan perbaikan pada keamanan siber negara secara keseluruhan [1].

Salah satu media digital yang sering menjadi target adalah aplikasi web dengan angka mencapai sekitar 75% serangan siber [2]. Pada tahun 2023 sebanyak 30.000 situs web diretas setiap hari [3]. Salah satu jenis ancaman keamanan aplikasi web adalah adanya *request* berisi *payload* berbahaya seperti *SQL Injection* (SQLi) dan *Cross-Site Scripting* (XSS) ke aplikasi web. SQLi memungkinkan penyerang untuk memalsukan identitas user atau administrator sistem, merusak data yang ada, atau membuatnya tidak tersedia [4]. Seperti SQLi, XSS merupakan jenis serangan injeksi. Dampak dari serangan ini adalah penyerang dapat menyamar sebagai pengguna dan mengambil alih akun. Jika pengguna memiliki hak administratif, maka penyerang dapat melakukan *code execution* di server [5].

Jenis ancaman keamanan aplikasi web lainnya adalah *Denial of Service* (DoS) dan *Distributed Denial of Service* (DDoS). Berbeda dengan SQLi dan XSS, DoS dan DDoS umumnya mengirimkan request tanpa *payload* berbahaya dalam jumlah besar karena tujuan utamanya adalah memenuhi *bandwidth* pada web aplikasi. Pada tahun 2022, terjadi peningkatan serangan ini hingga 1,6 juta serangan. Serangan DDoS mengakibatkan fungsionalitas situs web menjadi buruk atau membuatnya tidak dapat diakses (*offline*) [6]. Untuk itu, dalam mengatasi serangan SQLi, XSS, DoS dan DDoS ditawarkan sebuah solusi berupa desain dan implementasi *Web Application Firewall* (WAF) dan *rate limiting* pada keamanan aplikasi web.

WAF merupakan perangkat keamanan untuk melindungi aplikasi web dengan cara memfilter, memantau dan memblokir permintaan jaringan yang tidak sah. WAF dapat mendeteksi serangan berupa XSS, SQLi, *Cross-Site Forgery*, *File Inclusion*, dan sebagainya. WAF bekerja berdasarkan sekumpulan *rules* yang bertujuan untuk mengatasi kerentanan dan memfilter trafik berbahaya [7]. WAF diintegrasikan dengan *rate limiting* untuk membatasi jumlah *request* yang dapat digunakan oleh seluruh IP client atau IP yang dicurigai sebagai anomali. Dalam memastikan kinerja dari WAF dan *rate limiting*, dilakukan pengujian berupa pengukuran akurasi dan *Quality of Service* (QoS). Akurasi diukur menggunakan persamaan *true-positive/false-positive*. Kemudian, QoS dari web akan dibandingkan pada saat *rate limiting* belum diimplementasikan dan sedang diimplementasikan. WAF dan *rate limiting* diharapkan dapat menjadi solusi untuk *cyber defense*.

2. Dasar Teori

2.1 *Web Application Firewall*

Keamanan siber selalu menjadi perhatian utama untuk aplikasi internet, dan permintaan perlindungan situs web terus meningkat. Saat ini *Web Application Firewalls* (WAF) umum digunakan oleh pemilik situs web, karena memberikan perlindungan terhadap berbagai jenis serangan dengan menyaring permintaan jaringan yang masuk. Adapun penelitian tentang *Web Application Firewall Using Machine Learning and Features Engineering* tahun 2022, yang merupakan sebuah penelitian untuk menganalisis permintaan yang masuk ke web server dan memfilter permintaan yang berupa URL, *payload* dan *header*. Dari hasil penelitian ini akurasi klasifikasi sebesar 99,6% dengan dataset yang digunakan pada penelitian ini dan akurasi klasifikasi 98,8% dengan dataset *real web server* [8]. Penelitian lain tentang *Improving Web Application Firewalls with Automatic Language Detection* pada tahun 2022, penelitian ini memberikan hasil yang baik dengan hampir tidak ada notifikasi palsu dan tingkat deteksi yang dapat diterima [9].

Berdasarkan penelitian-penelitian ini disimpulkan bahwa WAF berfungsi untuk melindungi aplikasi web dengan menyaring, memantau, dan memblokir trafik HTTP atau HTTPS berbahaya yang mengarah ke aplikasi web serta mencegah terjadinya kebocoran privasi data. Fungsi tersebut membuat WAF dapat mengidentifikasi kemungkinan terjadinya sebuah serangan. WAF bertindak berdasarkan *rule* atau kebijakan yang diberikan secara manual.

2.2 *Rate Limiting*

Rate limiting mengontrol jumlah *request* yang dapat diterima server selama periode tertentu. *Rate Limiting* bekerja dengan cara menetapkan *threshold* untuk sejumlah *request* yang masuk ke server, lalu *traffic* yang melebihi *threshold* dapat secara otomatis akan diperlambat atau mengantre sementara sehingga menjaga kestabilan server [10]. Berdasarkan penelitian tentang *RateGuard: A Robust Distributed Denial of Service (DDoS) Defense System,* penelitian ini fokus pada tiga jenis serangan DDoS yaitu *Fast Adaptive Attacks* (FAAs), *Adaptive Attacks with statistical filtering rules*

Scanning (AAS), dan *Low-Rate TCP Attack* (LRA). Solusi untuk menangani permasalahan ini dinamai dengan *RateGuard*, yaitu sistem pertahanan terhadap DDoS berbasis *Leaky Bucket* (LB). *RateGuard* mampu melawan tiga jenis serangan DDoS tersebut secara *real time* dengan membatasi *traffic* berlebihan berdasarkan *traffic* nominal korban [11].

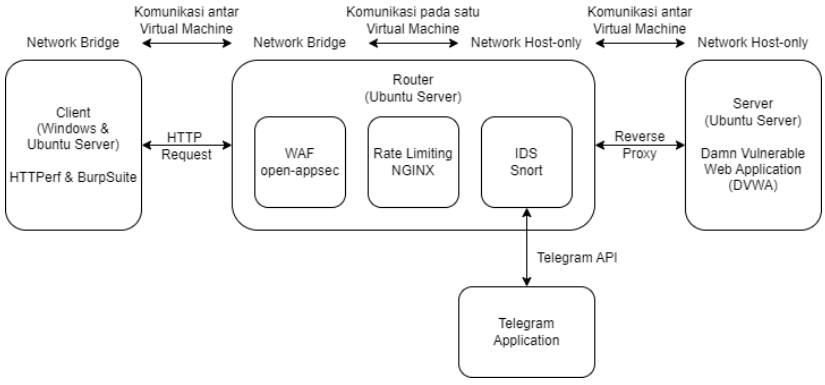
Dari penelitian tersebut dapat disimpulkan bahwa jumlah *request* yang dapat diterima oleh server dapat menjadi objek serangan yang sangat rentan. *Rate limiting* dapat membatasi *request* dan mengidentifikasi pola lalu lintas yang tidak normal atau mencurigakan. Dengan membatasi jumlah *request* per detik dari satu atau beberapa sumber, kapasitas *traffic* server atau jaringan dapat tetap stabil dan terjaga.

2.3 *Intrusion Detection System* (IDS)

Berdasarkan penelitian tentang *Intrusion detection system for cyberattacks in the Internet of Vehicles environment* tahun 2023, membahas tentang *framework* baru IDS yang dirancang khusus untuk serangan siber seperti DoS, DDoS, *Brute Force*, *Botnets* dan *Sniffing*. Penelitian tersebut membuat sistem IDS berbasis *machine learning* yang mampu mendeteksi perilaku tidak normal dengan memeriksa lalu lintas jaringan untuk menemukan aliran data yang tidak biasa. Dari hasil penelitian ini, didapatkan hasil akurasi tinggi sebesar 99,8% dalam waktu eksekusi 46,9 detik dan waktu deteksi 0,24 detik untuk ketiga dataset IDS yang digabungkan [12]. Penelitian lain tentang *Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for Live Analysis based on Machine Learning* tahun 2023. Penelitian ini membuat sebuah sistem yang dapat menganalisis, mendeteksi dan pemantauan serangan siber menggunakan *Security Information & Event Management* (SIEM) yang dianalisis langsung menggunakan *machine learning* pada *Intrusion Detection System* (IDS) [13]. Dari kedua penelitian tersebut dapat disimpulkan bahwa *Intrusion Detection System* (IDS) merupakan sistem keamanan yang efektif untuk digunakan. IDS dapat mendeteksi, memantau dan menganalisis serangan siber yang terjadi. IDS dapat memberikan peringatan kepada *user*, sehingga *user* dapat menentukan langkah yang tepat untuk mengatasi masalah yang ada.

3. Pembahasan

3.1. Blok Diagram

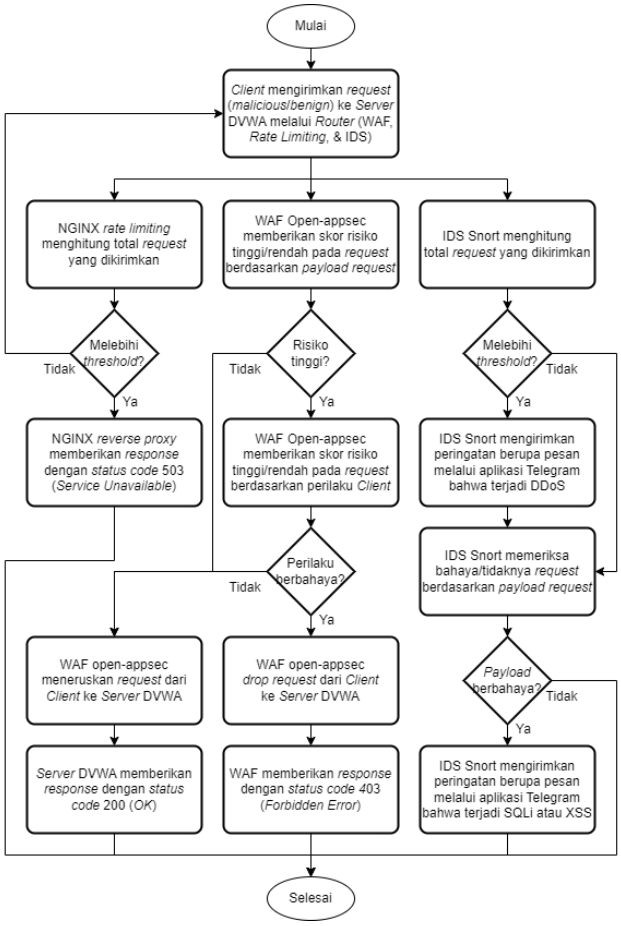


Gambar 3.1 Blok Diagram

Arsitektur sistem *cyber defense* dapat dilihat pada Gambar 3.1. Secara umum, *client*, *router*, dan *server* menggunakan *virtual machine* (VM) dengan *operating system* (OS) yang berbeda-beda sesuai dengan kebutuhan.

3.2. Diagram Alir

Berdasarkan diagram alir atau *flowchart* pada Gambar 3.2, sistem dimulai dengan *client* yang mengirimkan *request* ke *server* DVWA hingga *response* yang diberikan oleh *router* maupun *server*. Request yang dikirimkan oleh *client* akan diproses secara bersamaan oleh WAF, *rate limiting*, dan IDS pada *router*. Payload pada *request* dan jumlah *request* akan menentukan *response* yang diberikan oleh *router* dan *server*.



Gambar 3.2 Diagram Alir

4. Hasil dan Analisis

4.1. WAF Open-appsec

Hasil pengujian dari WAF berupa *file* Json yang diimport pada *dashboard* Open-appsec. *File* Json dianalisa untuk menentukan dua hal sebagai berikut.

1. *Security Quality*, yaitu menilai kemampuan WAF dalam mendeteksi serangan berbahaya dan melakukan *drop request (true positive rate)*.

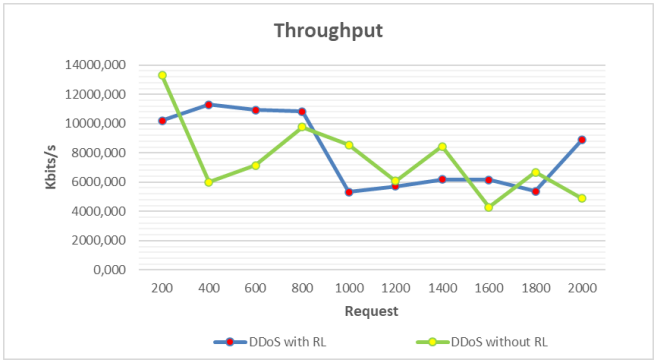
$$TPR = \frac{TP}{TP + FN} = \frac{1958}{1958 + 0} = 1$$

2. *Detection Quality*, yaitu menilai kemampuan WAF dalam mendeteksi request tidak berbahaya dan memberikan akses *request* ke server *(true negative rate)*.

$$TNR = \frac{TN}{TN + FP} = \frac{24}{24 + 0} = 1$$

Berdasarkan hasil pengujian, nilai dari *security quality* sebesar 1 atau 100% dan *detection quality* sebesar 1 atau 100%.

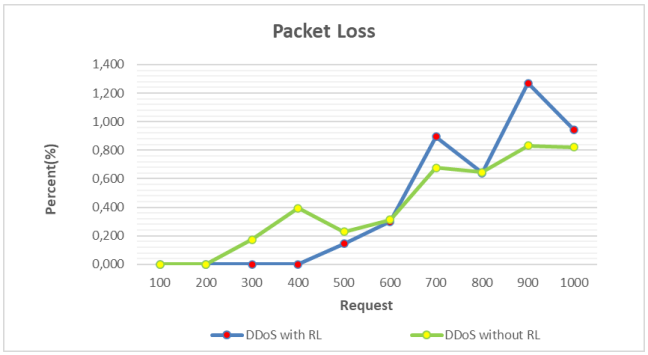
4.2. Rate Limiting



Gambar 4.1 Grafik *Throughput*

Analisis pada Grafik 4.1 *Throughput* adalah sebagai berikut.

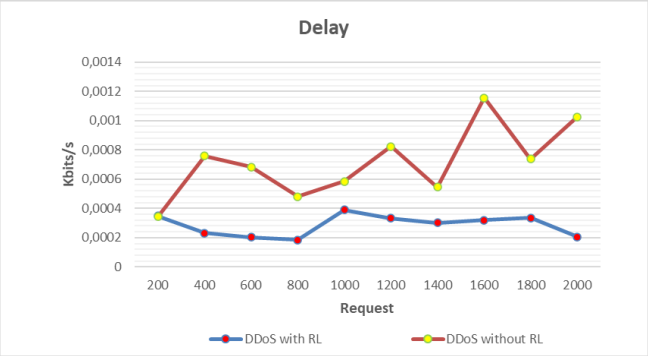
1. Dengan menggunakan *rate limiting*, *throughput* yang didapatkan pada awal *request* cukup tinggi, kemudian mulai menurun dan cukup stabil sebelum akhirnya terjadi lonjakan kembali. Hal ini dapat terjadi karena paket *bandwidth* masih dapat menampung jumlah *request* yang masuk sebelum akhirnya penuh. Perubahan *throughput* saat menggunakan *rate limiting* terjadi berdasarkan faktor penuhnya *bandwidth* dan jumlah *request* yang dibatasi.
2. Tanpa menggunakan *rate limiting*, *throughput* yang didapatkan pada awal hingga akhir tidak stabil. Hal ini dapat terjadi karena jumlah paket yang dikirimkan sangat besar dan tidak dibatasi. Perubahan *throughput* saat tidak menggunakan *rate limiting* terjadi berdasarkan faktor penuhnya *bandwidth* dan jumlah *request* yang tidak dibatasi.



Gambar 4.2 Grafik *Packet Loss*

Analisis pada Gambar 4.2 Grafik *Packet Loss* adalah sebagai berikut.

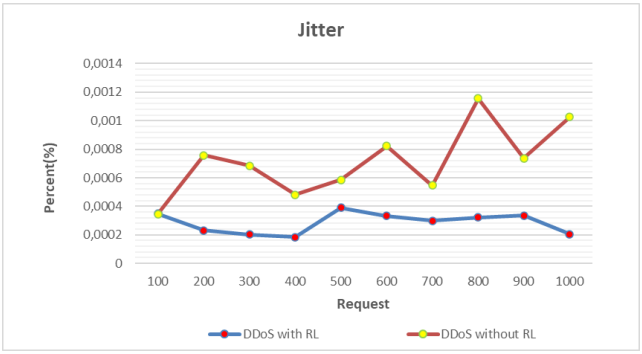
1. Dengan menggunakan *rate limiting*, *packet loss* yang didapatkan lebih tinggi. Hal ini dapat terjadi karena paket yang dikirimkan lebih sedikit, namun paket data yang diterima cenderung hampir sama banyak dengan sistem tanpa menggunakan *rate limiting*.
2. Tanpa menggunakan *rate limiting*, *packet loss* yang didapatkan lebih rendah. Hal ini dapat terjadi karena paket yang dikirimkan lebih banyak dan paket yang diterima cenderung hampir sama banyak dengan sistem yang menggunakan *rate limiting*.



Gambar 4.3 Grafik *Delay*

Analisis pada Gambar 4.3 Grafik *Delay* adalah sebagai berikut.

1. Dengan menggunakan *rate limiting*, *delay* yang terjadi lebih kecil karena paket yang dikirimkan lebih sedikit. Sehingga selisih waktu antara paket yang satu dengan yang lain tidak jauh.
2. Tanpa menggunakan *rate limiting*, *delay* yang terjadi lebih besar karena paket yang dikirimkan lebih banyak. Sehingga selisih waktu antara paket yang satu dengan yang lain jauh.



Gambar 4.4 Grafik Jitter

Analisis pada Gambar 4.4 Grafik Jitter adalah sebagai berikut.

1. Dengan menggunakan *rate limiting*, jitter yang terjadi lebih kecil karena *delay* yang yang terjadi lebih kecil.
2. Tanpa menggunakan *rate limiting*, jitter yang terjadi lebih besar karena *delay* yang yang terjadi lebih besar.

4.2. IDS Snort

Berdasarkan hasil pengujian pada Snort dapat dilakukan analisa seperti pada tabel berikut.

Tabel 4.1 Hasil Pengujian IDS Snort

Pengujian	Hasil Pengujian
Dapat mendeteksi serangan SQLi	Berhasil dideteksi
Dapat mendeteksi serangan XSS	Berhasil dideteksi
Dapat mendeteksi serangan DDoS	Berhasil dideteksi

5. Kesimpulan

Berdasarkan hasil penelitian dapat diambil kesimpulan sebagai berikut.

1. Pada WAF, nilai *security quality* sebesar 100% dan *detection quality* sebesar 100% yang berarti open-appsec adalah WAF yang sangat akurat.
2. Pada *rate limiting*, didapatkan beberapa *point* sebagai berikut.

a. Rata-rata *throughput* dengan *rate limiting* sebesar 8.092,972, sedangkan rata-rata *throughput* tanpa *rate limiting* sebesar 7.516,592. Sehingga *throughput* tanpa *rate limiting* lebih besar daripada dengan *rate limiting*.
b. Rata-rata *packet loss* dengan *rate limiting* sebesar 0.419% sedangkan rata-rata *packet loss* tanpa *rate limiting* sebesar 0.408%. Sehingga *packet loss* tanpa *rate limiting* lebih besar daripada dengan *rate limiting*.
c. Rata-rata *delay* dengan *rate limiting* sebesar 0,000284814 sedangkan rata-rata *delay* tanpa *rate limiting* sebesar 0,000714118. Sehingga *delay* tanpa *rate limiting* lebih besar daripada dengan *rate limiting*.
d. Rata-rata jitter dengan *rate limiting* sebesar 0,000285191 sedangkan rata-rata jitter tanpa *rate limiting* sebesar 0,000714252. Sehingga jitter tanpa *rate limiting* lebih besar daripada dengan *rate limiting*.

3. Pada IDS, snort dapat mengirimkan pesan peringatan sesuai jenis serangan yang terjadi seperti SQLi, XSS, DDoS.

Daftar Pustaka:

Referensi yang digunakan adalah sebagai berikut.

[1] Statista, "Number of cyber attack cases in Indonesia from 2019 to 2022," *Statista*, 2022. https://www.statista.com/statistics/1412527/indonesia-number-of-cyber-attacks/ (accessed Oct. 11, 2023).

[2] Acunetix, "What Is a Web Application Attack and how to Defend Against It," *Acunetix*, 2023. https://www.acunetix.com/websitesecurity/web-application-attack/ (accessed Oct. 11, 2023).

[3] R. Vardhman, "How Many Cyber Attacks Happen Per Day in 2023?," *Techjury*, 2023. https://techjury.net/blog/how-many-cyber-attacks-per-day/ (accessed Oct. 11, 2023).

[4] Kingthorin, "SQL Injection," *OWASP*. https://owasp.org/www-community/attacks/SQL_Injection (accessed Jun. 05, 2024).

[5] KirstenS, "Cross Site Scripting (XSS)," *OWASP*. https://owasp.org/www-community/attacks/xss/# (accessed Jun. 05, 2024).

[6] Microsoft, "Apa itu serangan DDoS?," *Microsoft*, 2023. https://www.microsoft.com/id-id/security/business/security-101/what-is-a-ddos-attack#:~:text=Serangan DDoS menargetkan situs web,atau membuatnya offline sama sekali. (accessed Oct. 11, 2023).

[7] CloudFlare, "What is a WAF? Web Application Firewall explained," *CloudFlare*. https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/#:~:text=A WAF or web application,and SQL injection%2C among others. (accessed Jun. 05, 2024).

[8] A. Shaheed and M. H. D. B. Kurdy, "Web Application Firewall Using Machine Learning and Features Engineering," *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/5280158.

[9] T. C. H. Nguyen, M. K. Le-Nguyen, D. T. Le, V. H. Nguyen, L. P. Tôn, and K. Nguyen-An, "Improving Web Application Firewalls with Automatic Language Detection," *SN Comput. Sci.*, vol. 3, no. 6, pp. 1–14, 2022, doi: 10.1007/s42979-022-01327-2.

[10] M. Medet, "Overview of Distributed Denial of Service (DDoS) Attack Types and Mitigation Methods," pp. 494–508, 2024, doi: 10.51582/interconf.19-20.03.2024.048.

[11] H. Sun, W. Ngan, and H. J. Chao, "RateGuard: A robust Distributed Denial of Service (DDoS) Defense System," *GLOBECOM - IEEE Glob. Telecommun. Conf.*, no. 3, 2009, doi: 10.1109/GLOCOM.2009.5425941.

[12] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1251–1260, 2020, doi: 10.1016/j.procs.2020.04.133.

[13] M. S. Korium, M. Saber, A. Beattie, A. Narayanan, S. Sahoo, and P. H. J. Nardelli, "Intrusion Detection System for Cyberattacks in the Internet of Vehicles environment," *Ad Hoc Networks*, vol. 153, no. November 2023, 2024, doi: 10.1016/j.adhoc.2023.103330.