Jurnal Ilmiah Wahana Pendidikan, Maret 2024, 10 (5), 403-412
DOI: https://doi.org/10.5281/zenodo.10526246
p-ISSN: 2622-8327 e-ISSN: 2089-5364
Accredited by Directorate General of Strengthening for Research and Development
Available online at https://jurnal.peneliti.net/index.php/JIWP

Analisis Ketahanan Web Application Firewall Terhadap Serangan SQL Injection

Hana Nazla Humaira[1], Asep Id Hadiana[2], Herdi Ashaury[3]

[1,2,3] Program Studi Informatika, Fakultas Sains Dan Informatika Universitas Jenderal Achmad Yani

Received: 15 November 2023
Revised: 13 Desember 2023
Accepted: 15 Januari 2024

Abstract

The ever-advancing technological transformation has brought benefits to our daily lives. Thanks to these technological advances, it is very easy to get access to information, communicate with platforms, and conduct transactions in an increasingly sophisticated digital environment. Web application services are one of the positive impacts of the development of the digital world. However, behind the ease of access offered by web applications, it is often targeted by cyber criminals to obtain sensitive data within it. The application of Web Application Firewall (WAF) as a web application security from SQL injection attacks can be a solution to security issues. This research involves several different WAF solution selections. The results show that the effectiveness of WAF in protecting applications from SQL injection attacks varies depending on the type of attack. From the attacks performed, Naxsi was able to filter out 99% of the attacks and ModSecurity 100%.

Keywords: *web application firewall, SQL injection, web attack*

(*) Corresponding Author: hana.nazla@student.unjani.ac.id

How to Cite: Humaira, H. N., Hadiana, A. I., & Ashaury, H. (2024). Analisis Ketahanan Web Application Firewall Terhadap Serangan SQL Injection. https://doi.org/10.5281/zenodo.10526246.

PENDAHULUAN

Kehidupan sehari-hari kita telah mendapatkan banyak manfaat dari pertumbuhan teknologi yang selalu berkembang. Perkembangan ini telah menyederhanakan akses informasi, komunikasi lintas platform, dan proses transaksional dalam lingkungan digital yang semakin kompleks. Salah satu manfaat dari kemajuan di dunia digital ini adalah aplikasi web. Aplikasi web digunakan oleh banyak organisasi, termasuk bank dan pemerintah, untuk menawarkan layanan yang lebih luas kepada masyarakat (Muzaki et al., 2020). Aplikasi web dibangun dengan basis data sebagai sistem penyimpanan dan manajemen data yang mendasarinya. Namun, penjahat siber dapat menyalahgunakan informasi pribadi yang disimpan dalam basis data ini untuk melakukan kejahatan termasuk penipuan keuangan, pencurian identitas, dan pelanggaran yang berpotensi merusak lainnya.

Injeksi SQL tetap menjadi risiko keamanan web yang signifikan, secara konsisten menduduki peringkat di antara 10 besar. Meskipun pindah ke posisi ketiga pada tahun 2021 dari posisi pertama pada tahun 2017 (OWASP, 2022), kepentingan historisnya tidak dapat disangkal. Jenis serangan ini melibatkan penyisipan kode SQL berbahaya sebagai input, yang bertujuan untuk mengeksploitasi basis data aplikasi web (Kingthorin, 2020).

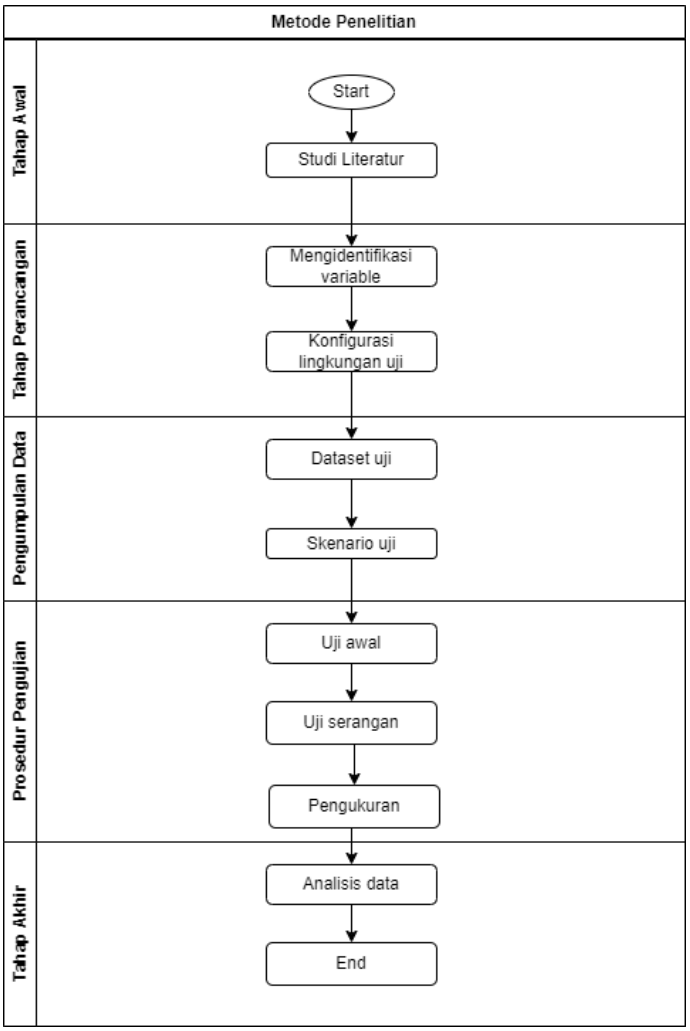
Kasus aktual serangan SQL injection pernah terjadi di Surabaya, Indonesia pada tahun 2018, setidaknya 44 situs web dari berbagai negara disusupi pada kasus ini. Menurut laporan, pelakunya menggunakan alat online gratis untuk masuk sebagai admin dan mengendalikan seluruh sistem (Librianty, 2023). Insiden lain terjadi di

perusahaan TalkTalk, di mana peretasan menyebabkan pengungkapan 150.000 catatan pribadi pelanggan, termasuk lebih dari 15.000 informasi keuangan sensitif pelanggan (Hern, 2016).

Berdasarkan hal tersebut, dapat disimpulkan bahwa serangan injeksi SQL adalah salah satu jenis serangan yang paling umum dan berbahaya. Oleh karena itu, pencegahan diperlukan yang dalam situasi ini, penulis memanfaatkan Naxsi dan ModSecurity, dua teknologi Firewall Aplikasi Web. Penelitian ini menguji ketahanan Web Application Firewall terhadap serangan injeksi SQL. Dalam penelitian ini, berbagai pilihan WAF dipilih dan diuji untuk melihat seberapa baik mereka dapat menghentikan dan mendeteksi serangan injeksi SQL.

METODOLOGI PENELITIAN

Terdapat beberapa fase penting dalam penelitian ini. Untuk memahami gagasan dan teori yang relevan dengan penerapan penelitian, langkah awal adalah menganalisis literatur. Tahap perancangan yang akan dilakukan merupakan tahap kedua. Langkah ketiga berkonsentrasi pada pengumpulan data yang diperlukan. Eksekusi prosedur pengujian dilakukan berikutnya pada langkah keempat. Penelitian kemudian bergerak menuju tahap akhir, ketika temuan-temuan dari rangkaian serangan yang telah dilakukan.



Gambar 1 Metode Penelitian

1. Tahap awal

Sebelum melakukan penelitian sebenarnya, tinjauan literatur dilakukan dengan tujuan mengumpulkan, mengevaluasi, dan mensintesis literatur yang berkaitan dengan penelitian.

2. Tahap perancangan

Tahap perancangan menjelaskan desain penelitian yang digunakan untuk membandingkan efektivitas Naxsi dan Modsecurity sebagai pertahanan terhadap serangan injeksi SQL.

Penggunaan perangkat lunak yang digunakan dijelaskan pada table 1.

Table 1 Spesifikasi perangkat lunak

Perangkat Lunak	Deskripsi
Naxsi	Web Application Firewall
ModSecurity	Web Application Firewall
Kali Linux	Sistem operasi yang digunakan untuk melakukan uji coba ModSecurity
Ubuntu	Sistem operasi yang digunakan untuk melakukan uji coba Naxsi

Nginx	Web server yang digunakan untuk melakukan uji coba Naxsi
Apahe2	Web server yang digunakan untuk melakukan uji coba ModSecurity
DVWA	Target uji coba serangan
Burp Suite	Tools untuk melakukan serangan

3. Pengumpulan data

Tahap pengumpulan data berisi mengenai implementasi Web Application Firewall (WAF) Naxsi dan ModSecurity, serta kumpulan data uji yang terdiri dari skrip serangan SQL injection.

4. Prosedur pengujian

Prosedur pengujian dilakukan tanpa menggunakan WAF pada aplikasi web, dilanjutkan dengan pengujian serangan dengan melibatkan WAF pada aplikasi web.

5. Tahap terakhir

Pada tahap akhir, dilakukan analisis dan penyimpulan terhadap serangkaian serangan yang telah dilakukan.

HASIL DAN PEMBAHASAN

HASIL

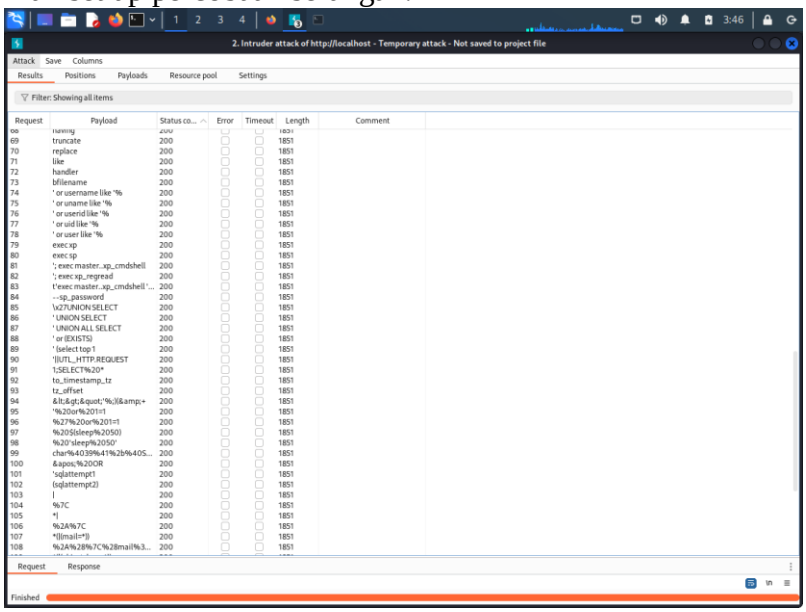
Penelitian ini menggunakan perangkat lunak yang diantaranya ModSecurity, Naxsi, DVWA, Nginx, Apache, Burp suite, Foxy proxy, Kali Linux dan Virtual Box. ModSecurity dan Naxsi dijalankan pada Kali Linux yang berada pada mesin virtual. Uji serangan SQL injection menggunakan Burp suite, kemudian kinerja web dilihat menggunakan network monitor pada Firefox.

Pengujian

Menguji apakah injeksi SQL secara efektif dihentikan atau tidak, digunakan penetration tools yaitu Burp Suite. skenario pertama dilakukan tanpa firewall, sementara skenario kedua menggunakan web application firewall Naxsi, dan scenario ketiga menggunaka ModSecurity sebagai web application firewall.

Pengujian SQL Injection Tanpa WAF

Burp Suite digunakan untuk menguji serangan secara otomatis terhadap DVWA, dan penyerang cukup menunggu alat tersebut melakukan pekerjaannya sebelum menganalisis hasilnya. Ketika 125 skrip serangan injeksi SQL dijalankan terhadap server web, hasilnya adalah terdapat kode status 200 untuk setiap upaya serangan. Hal ini mengarah pada kesimpulan bahwa server tidak dapat menggagalkan setiap percobaan serangan.



Gambar 2 Pengujian SQL injection

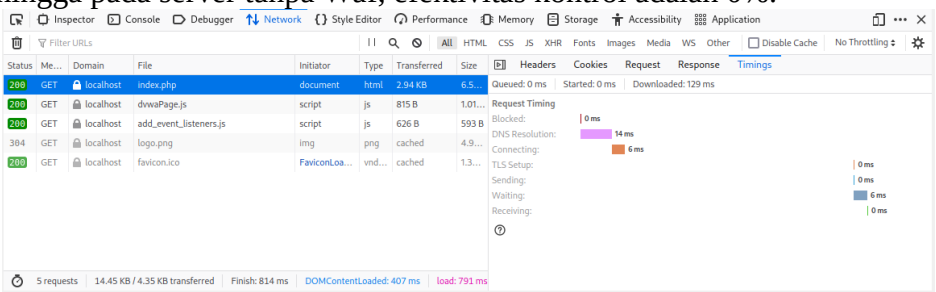
Pengujian efektivitas control terhadap server yang diserang, digunakan seperti pada persamaan 1.

$$ER = \frac{Jumlah\ serangan\ yang\ dihentikan}{total\ serangan}\ x\ 100\%$$

$$= \frac{0}{125}\ x\ 100 \qquad \ldots(1)$$

$$= 0\%$$

Sehingga pada server tanpa Waf, efektivitas kontrol adalah 0%.



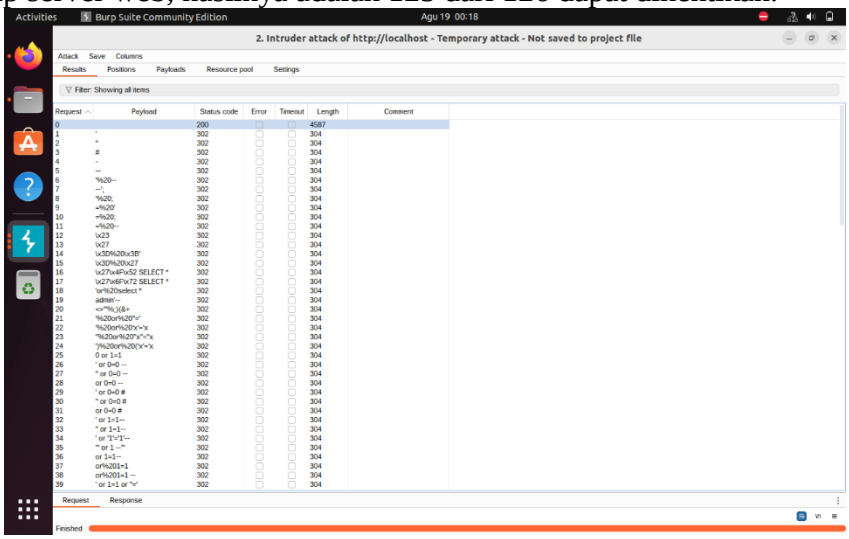
Gambar 3 Kinerja web tanpa waf

Waktu muat halaman rata-rata pada pengujian kinerja web tanpa WAF adalah 814 milidetik, dengan total 5 permintaan, ukuran halaman secara keseluruhan mencapai 14,45 KB. Jumlah waktu yang dihabiskan untuk setiap tahap proses juga cukup bervariasi; waktu pemblokiran rata-rata adalah 0 milidetik, sedangkan pencarian DNS rata-rata membutuhkan waktu 14 milidetik, koneksi awal membutuhkan waktu 6 milidetik, waktu menunggu respon adalah 6 milidetik,

waktu menerima respon adalah 0 milidetik, dan penyiapan TLS membutuhkan waktu 0 milidetik, dan transfer data membutuhkan waktu 0 milidetik.

Pengujian SQL injection dengan Naxsi

Burp Suite digunakan untuk menguji serangan secara otomatis terhadap DVWA, dan penyerang cukup menunggu alat tersebut melakukan pekerjaannya sebelum menganalisis hasilnya. Ketika 126 skrip serangan injeksi SQL dijalankan terhadap server web, hasilnya adalah 125 dari 126 dapat dihentikan.

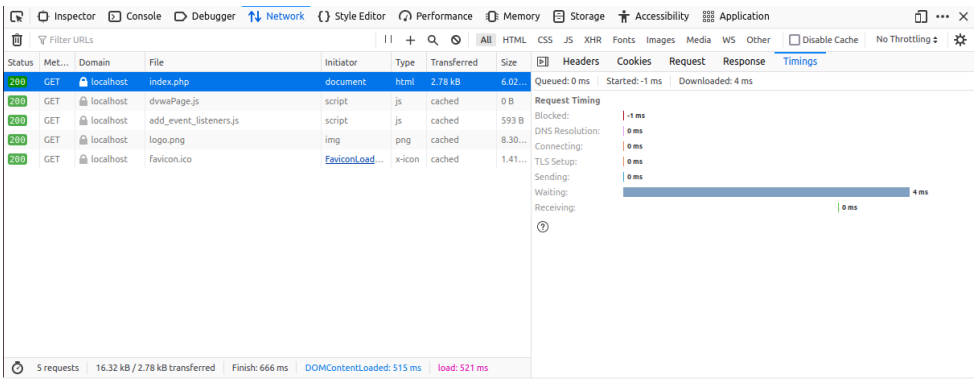


Gambar 4 Pengujian dengan naxsi

Pengujian efektivitas kontrol terhadap server yang diserang, digunakan seperti pada persamaan 2.

$$ER = \frac{125}{126}\ x\ 100\% = 99\% \qquad \ldots (2)$$

Sehingga pada server yang menggunakan Naxsi sebagai WAF, efektivitas control adalah 99%.

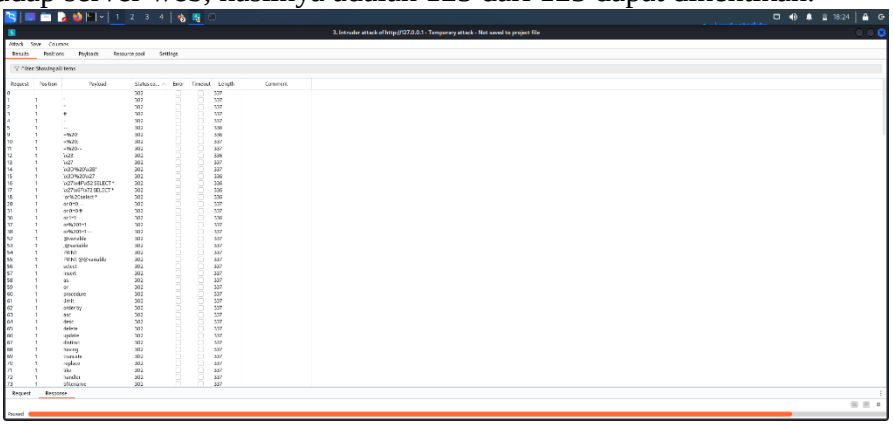


Gambar 5 Kinerja web dengan Naxsi

Waktu muat halaman rata-rata pada pengujian kinerja web dengan Naxsi sebagai WAF adalah 666 milidetik, dengan total 5 permintaan, ukuran halaman secara keseluruhan mencapai 16,32 KB. Jumlah waktu yang dihabiskan untuk setiap tahap proses juga cukup bervariasi; waktu pemblokiran rata-rata adalah -1 milidetik, sedangkan pencarian DNS rata-rata membutuhkan waktu 0 milidetik, koneksi awal membutuhkan waktu 0 milidetik, waktu menunggu respon adalah 4 milidetik, waktu menerima respon adalah 0 milidetik, setup TLS membutuhkan waktu 0 milidetik, dan transfer data membutuhkan waktu 0 milidetik.

Pengujian SQL injection dengan ModSecurity

Burp Suite digunakan untuk menguji serangan secara otomatis terhadap DVWA, dan penyerang cukup menunggu alat tersebut melakukan pekerjaannya sebelum menganalisis hasilnya. Ketika 125 skrip serangan injeksi SQL dijalankan terhadap server web, hasilnya adalah 125 dari 125 dapat dihentikan.

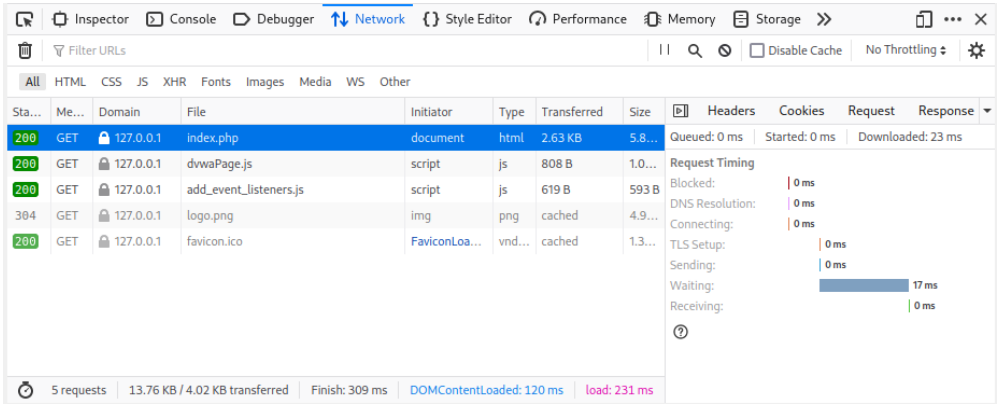


Gambar 6 Pengujian dengan modsecurity

Pengujian efektivitas control terhadap server digunakan seperti pada persamaan

$$ER = \frac{125}{125}\ x\ 100\% = 100\% \qquad \ldots (3)$$

Sehingga pada server yang menggunakan ModSecurity sebagai WAF, efektivitas control adalah 99%.



Gambar 7 Kinerja web modsecurity

Waktu muat halaman rata-rata pada pengujian kinerja web dengan Naxsi sebagai WAF adalah 309 milidetik, dengan total 5 permintaan, ukuran halaman secara keseluruhan mencapai 13,76 KB. Jumlah waktu yang dihabiskan untuk setiap tahap proses juga cukup bervariasi; waktu pemblokiran rata-rata adalah 0 milidetik, sedangkan pencarian DNS rata-rata membutuhkan waktu 0 milidetik, koneksi awal membutuhkan waktu 0 milidetik, waktu menunggu respon adalah 17 milidetik, waktu menerima respon adalah 0 milidetik, setup TLS membutuhkan waktu 0 milidetik, dan transfer data membutuhkan waktu 0 milidetik.

Hasil Pengujian dan hasil coba

PEMBAHASAN

Hasil dari dua tahap serangan yang dilakukan dalam uji coba, dapat dilihat bahwa kedua web application firewall dapat secara efektif mempertahankan diri

dari serangan SQL injection ketika diimplementasikan. Secara khusus, pada tahap pertama, ketika situs web diserang tanpa implementasi firewall aplikasi web, server tidak dapat menyaring serangan. Pada tahap kedua, setelah diimplementasikan dengan Naxsi dan diserang menggunakan Burp Suite, Naxsi dapat memblokir sebanyak 125 dari 126 serangan. Ketika Burp Suite dan ModSecurity digunakan untuk mengimplementasikan situs web, 125 dari 125 skrip serangan dapat diblokir oleh ModSecurity pada tahap ketiga. Hasil pengujian dirangkum sebagai pada table 1 dan 2:

Table 1. Kinerja Web

WAF	Waktu Rata-rata Pemuatan Halaman	Ukuran Total Halaman	Jumlah Permintaan
ModSecurity	309 ms	13.76 KB	5 permintaan
Naxsi	666 ms	16.32 KB	5 permintaan
Tanpa WAF	814 ms	14.45 KB	5 permintaan

Tabel 2. Hasil Uji Coba

WAF	Skrip Serangan Tertahan	Waktu rata-rata pemuatan halaman
Tanpa WAF	0	814 ms
Naxsi	125 dari 126	666 ms
ModSecurity	125 dari 125	309 ms

KESIMPULAN

Berdasarkan hasil pengujian dan implementasi dari penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa:

1. Implementasi kedua WAF, Naxsi dan ModSecurity, berhasil melindungi situs web dari serangan SQL injection dengan tingkat perlindungan yang tinggi.
2. Naxsi mampu menghalangi 125 dari 126 skrip serangan, namun memiliki dampak yang lebih besar terhadap kinerja situs web.
3. ModSecurity menghadang seluruh 125 skrip serangan dengan performa pemuatan halaman yang sangat baik, menunjukkan perlindungan yang lebih optimal.

ModSecurity menunjukkan kinerja terbaik dengan waktu pemuatan halaman yang cepat dan ukuran total halaman yang kecil, serta dampak minimanl terhadap performa. Naxsi mempengaruhi waktu pemuatan halaman lebih signifikan dibandingkan ModSecurity. Tanpa WAF, performa halaman lebih lambat dengan variasi waktu dalam tahapan proses yang lebih tinggi. Dengan pertimbangan perlindungan dan performa, ModSecurity menjadi pilihan yang optimal

DAFTAR PUSTAKA

R. A. Muzaki and A. Background, “Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall,” pp. 85–90, 2020.

OWASP, “OWASP Top Ten,” *OWASP Foundation, Inc*, 2022. https://owasp.org/www-project-top-ten/ (accessed Nov. 30, 2022).

Kingthorin, “SQL Injection,” *OWASP Foundation, Inc*, 2023.

https://owasp.org/www-community/attacks/SQL_Injection (accessed Jun. 08, 2023).

Andina Librianty. (2018, March 14). *Modus Peretasan Hacker Surabaya Pakai SQL Injection, Apa Itu?* Liputan6. https://www.liputan6.com/tekno/read/3373117/modus-peretasan-hacker-surabaya-pakai-sql-injection-apa-itu

J. P. Singh, "Analysis of SQL Injection Detection Techniques," vol. 28, no. 1, pp. 37–55, 2016, doi: 10.20904/281-2037.

A. Librianty, "Modus Peretasan Hacker Surabaya Pakai SQL Injection, Apa Itu?," *Liputan 6*, 2018. https://www.liputan6.com/tekno/read/3373117/modus-peretasan-hacker-surabaya-pakai-sql-injection-apa-itu (accessed Jun. 08, 2023).

Al. Hern, "TalkTalk hit with record £400k fine over cyber-attack," *The Guardian*, 2016. https://www.theguardian.com/business/2016/oct/05/talktalk-hit-with-record-400k-fine-over-cyber-attack.

M. Akbar, M. Arif, F. Ridha, and A. C. S. Scripting, "SQL Injection and Cross Site Scripting Prevention Using OWASP Web Application Firewall," *Joiv Int. J. Informatics Vis.*, vol. 2, pp. 286–292, 2018.

"Naxsi," *Github*. https://github.com/nbs-system/naxsi (accessed Jun. 09, 2023).

"ModSecurity." https://github.com/SpiderLabs/ModSecurity (accessed Jun. 09, 2023).

H. Alamsyah, "Penerapan Sistem Keamanan WEB Menggunakan Metode WEB Aplication Firewall," vol. 11, no. 1, 2021.

K. D. Ayunda *et al.*, "Implementation and Analysis ModSecurity on Web-Based Application with OWASP Standards," vol. 8, no. 3, 2021.

R. Yanti Jamain, Periyadi, and S. Juli Irzal Ismail, "IMPLEMENTASI KEAMANAN APLIKASI WEB DENGAN WEB APPLICATION FIREWALL," vol. 1, no. 3, pp. 2191–2195, 2015.

B. I. Mukhtar and M. A. Azer, "Evaluating the Modsecurity Web Application Firewall Against SQL Injection Attacks," 2020.

M. Chatham, *Structured Query Language By Example - Volume I: Data Query Language*. Lulu.com, 2012.

H. Zhang, "SQL Injection Attack Principles and Preventive Techniques for PHP Site," 2018.

F. Q. Kareem, S. Y. Ameen, A. Ahmed, and A. A. Salih, "SQL Injection Attacks Prevention System Technology : Review SQL Injection Attacks Prevention System Technology : Review," no. July, 2021, doi: 10.9734/AJRCOS/2021/v10i330242.

L. Zhang, D. Zhang, C. Wang, J. Zhao, and Z. Zhang, "ART4SQLi : The ART of SQL Injection," *IEEE Trans. Reliab.*, vol. PP, pp. 1–20, 2019, doi: 10.1109/TR.2019.2910285.

O. Ojagbule, H. Wimmer, and C. D. Q. Sqli, "Vulnerability Analysis of Content Management Systems to SQL Injection Using SQLMAP," *SoutheastCon 2018*, pp. 1–7, 2018.

A. Razzaq, A. Hur, S. Shahbaz, M. Masood, and H. F. Ahmad, "Critical Analysis on Web Application Firewall Solutions," 2013.

M. H. Amouei, M. Rezvani, and M. Fateh, "RAT : Reinforcement-Learning-Driven and Adaptive Testing for Vulnerability Discovery in Web," vol. 11, no. 4, pp. 10–20, 2021, doi: 10.1109/TDSC.2021.3095417.

B. Garn, D. S. Lang, M. Leithner, and D. R. Kuhn, "Combinatorially XSSing Web Application Firewalls," pp. 85–94, 2021, doi: 10.1109/ICSTW52544.2021.00026.

M. F. R. K. Raharjo, "Evaluasi Kinerja Web Server Apache menggunakan Protokol HTTP2," *J. Eng. Technol. Appl. Sci.*, pp. 19–31, 2020, doi: 10.36079/lamintang.jetas-0201.92.

"Usage statistics of Nginx," *w3tech*. https://w3techs.com/technologies/details/ws-nginx (accessed Dec. 06, 2023).

"VirtualBox," *VirtualBox*. https://www.virtualbox.org/ (accessed Jun. 12, 2023).

"No Title." https://github.com/sqlmapproject/sqlmap (accessed Jun. 12, 2023).

[V. K. Gudipati, T. Venna, S. Subburaj, and O. Abuzaghleh, "Advanced Automated SQL Injection Attacks and Defensive Mechanisms," 2016.

S. Lika, R. Dwi, P. Halim, and I. Verdian, "ANALISA SERANGAN SQL INJEKSI MENGGUNAKAN SQLMAP," vol. 4, no. 2, pp. 88–94, 2018.

"What is PHP?," *PHP*. https://www.php.net/manual/en/intro-whatis.php.

"What is SQL?" https://aws.amazon.com/what-is/sql/ (accessed Jul. 02, 2023).

"Apache" https://httpd.apache.org/docs/ (accessed Aug. 1, 2023).