JATIKOM: Jurnal Aplikasi dan Teori Ilmu Komputer
ISSN: 2615-577X (Online)

Analisis Keamanan Website Berbasis Black-box Fuzzing: Studi Kasus Kerentanan XSS Dan SQL Injection Dalam Website X

Bagus Ariwibowo[#1]

Teknik Informatika,
Universitas Muhammadiyah
Malang, Indonesia

[1] bagusari229@gmail.com

Abstract*—Website security has become a critical concern in the digital era, particularly in the face of threats such as Cross-Site Scripting (XSS) and SQL Injection. This study applies a Black-box Fuzzing approach to analyze vulnerabilities in Website X. Three testing tools are utilized, namely OWASP ZAP, Acunetix, and Arachni. The testing process involves identifying endpoints, injecting payloads into input parameters, and analyzing server responses to detect potential exploitations. The results reveal that Website X contains several XSS and SQL Injection. OWASP ZAP proved to be the most effective in detecting and exploiting vulnerabilities, Arachni demonstrated good detection performance, while Acunetix excelled in scanning speed but was less accurate in identifying XSS. These findings indicate that the Black-box Fuzzing approach is effective in uncovering web application security flaws.

Key Words*— Website Security, Black-box Fuzzing, XSS, SQL Injection

Abstrak—**Keamanan website menjadi isu krusial di era digital, khususnya terhadap ancaman Cross-Site Scripting (XSS) dan SQL Injection. Penelitian ini menggunakan pendekatan Black-box Fuzzing untuk menganalisis kerentanan pada Website X. Tiga alat pengujian digunakan, yaitu OWASP ZAP, Acunetix, dan Arachni. Proses pengujian meliputi identifikasi endpoint, penyisipan payload pada parameter input, serta analisis terhadap respon server guna mendeteksi potensi eksploitasi. Hasil pengujian menunjukkan bahwa Website X mengandung beberapa kerentanan XSS dan SQL Injection. OWASP ZAP terbukti paling efektif dalam mendeteksi dan mengeksploitasi kerentanan, sementara Arachni menunjukkan performa deteksi yang baik, dan Acunetix unggul dalam kecepatan analisis meskipun kurang akurat dalam mendeteksi XSS. Temuan ini menunjukkan bahwa pendekatan Black-box Fuzzing efektif untuk mengidentifikasi celah keamanan pada aplikasi web.**

***Kata kunci*— Keamanan Website, Black-box Fuzzing, XSS (Cross-Site Scripting), SQL Injection**

I. PENDAHULUAN

Kemajuan teknologi informasi (TI) telah meningkatkan efisiensi dan efektivitas dalam berbagai aspek kehidupan, termasuk di lingkungan organisasi. Website menjadi salah satu sarana utama dalam menyampaikan informasi dan layanan digital. Dengan adanya website, informasi dapat diakses dengan mudah dan cepat oleh masyarakat [1], [2]. Namun, di balik manfaat tersebut, website juga rentan terhadap berbagai ancaman keamanan, seperti Cross-Site Scripting (XSS) dan SQL Injection, yang dapat mengancam integritas data serta kepercayaan publik. Privasi dan reputasi organisasi memainkan peran penting dalam membangun kepercayaan masyarakat.[3]. Oleh karena itu, penting untuk menerapkan sistem deteksi dan mitigasi guna menjaga keamanan website [4].

Serangan siber dapat mengganggu operasional sistem, membahayakan data pengguna, menghambat pelayanan publik, dan bahkan merusak keamanan nasional [5]. Ancaman XSS memungkinkan penyusupan skrip berbahaya ke dalam situs web, sedangkan SQL Injection memanfaatkan celah keamanan untuk mengakses database tanpa izin. Berdasarkan laporan OWASP, ancaman XSS meningkat dari peringkat ke-7 (2017) ke-3 (2021), menunjukkan risiko yang semakin tinggi [6]. Kasus serangan terhadap website pemerintah menunjukkan bagaimana peretas dapat mengeksploitasi celah keamanan untuk mengubah tampilan situs [7]. Serangan ini dapat menimbulkan efek luas, mulai dari merusak reputasi hingga melemahkan kepercayaan masyarakat, sehingga menegaskan pentingnya penerapan mitigasi risiko untuk mencegah dampak yang lebih besar.

Selain itu, sistem informasi yang terintegrasi mampu meningkatkan efisiensi pengelolaan data, mendukung pengambilan keputusan berbasis data, serta meningkatkan transparansi dan akuntabilitas dalam organisasi [8]. Namun, setiap perusahaan maupun organisasi harus selalu melindungi kerahasiaan, integritas, dan ketersediaan data pada suatu web server yang mengacu pada standar keamanan Nasional. Hal ini karena sistem jaringan telah mengalami perkembangan, sehingga membutuhkan peningkatan keamanan secara keseluruhan [9]. Website juga memberikan akses informasi di mana dan kapan saja asalkan akses internet tersedia. Namun, dengan semakin populernya internet, jumlah ancaman siber juga meningkat, sehingga keamanan website menjadi aspek yang sangat krusial [10].

Salah satu metode mitigasi yang efektif adalah Black-box Fuzzing, yaitu teknik uji keamanan yang menyuntikkan input

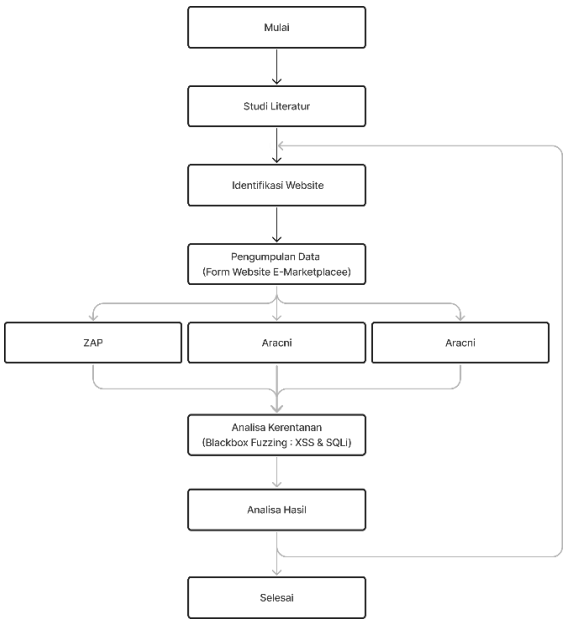
acak untuk mendeteksi celah tanpa mengetahui kode sumber aplikasi [11]. Beberapa alat yang mendukung pendekatan ini meliputi Wireshark untuk analisis jaringan [12], serta OWASP ZAP, Acunetix, dan Arachni untuk mendeteksi serta mengatasi kerentanan [13], [14], [15] Selain itu, alat otomatis juga dapat mengidentifikasi kerentanan dengan mengirimkan masukan khusus ke aplikasi web, yang dapat membantu dalam menemukan potensi eksploitasi sebelum digunakan oleh penyerang [16].

Informasi milik perusahaan merupakan aset penting yang harus dijaga dalam sistem web. Mengabaikan keamanan dapat menyebabkan pencurian data atau perubahan pada desain situs web yang berakibat fatal bagi organisasi [17]. Kerentanan pada situs web dapat menimbulkan konsekuensi serius, seperti rusaknya reputasi dan kredibilitas perusahaan atau organisasi, yang pada akhirnya dapat mengganggu operasional serta kepercayaan publik [18]. Oleh karena itu, pendekatan proaktif dalam pengujian keamanan, seperti penetration testing menjadi sangat penting untuk memastikan keamanan website tetap terjaga [10].

Penelitian ini bertujuan untuk menganalisis keamanan website X dengan fokus pada mitigasi serangan XSS dan SQL Injection. Hasil yang diperoleh diharapkan dapat memberikan wawasan bermanfaat mengenai keamanan website dan membantu meningkatkan tindakan pencegahan yang tepat [19].

II. METODE

Penelitian ini menggunakan metodologi untuk menganalisis keamanan situs web X yang menyediakan layanan informasi dan berbagai fitur interaktif bagi pengguna. Metodologi yang diterapkan mencakup studi literatur, identifikasi situs web, pengujian keamanan dengan berbagai alat, pengumpulan data, serta analisis kerentanan menggunakan pendekatan Black-box Fuzzing sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Alur penelitian

A. Studi Literatur

Beberapa studi telah dilakukan untuk mengevaluasi pemindai kerentanan web. Oualid Zaaza et al. [20] meneliti teknik analisis dinamis seperti eksekusi simbolik dan analisis hibrid, namun tidak menjelaskan secara rinci sumber data perbandingan alat yang digunakan. Aseel Alsaedi et al. [21] mengembangkan sistem otomatis untuk mendeteksi kerentanan web, namun masih terbatas dalam menangani eksploitasi kompleks seperti zero-day.

Danyang Zhao et al. [22] mengkaji implementasi fuzzer dengan modul bypass dan fuzz untuk keamanan aplikasi web dan IoT, meskipun tidak disertai dengan rincian hasil pengujian yang spesifik. Muzun Althunayyan et al. [23] mengevaluasi efektivitas pemindai black-box dalam mendeteksi bus[13]kerentanan Injection, dengan hasil terbaik yang diperoleh oleh ZAP dan Burp Suite Professional, tetapi penelitian ini terbatas pada satu jenis kerentanan.

Busra Zukran et al. membandingkan OWASP ZAP dan Skipfish dalam deteksi SQL Injection dan XSS, menunjukkan keunggulan OWASP ZAP, meskipun penelitian ini dilakukan pada aplikasi yang secara sengaja diberi kerentanan. Farismana, R et al. [24] mengevaluasi OWASP ZAP dan Acunetix terhadap keamanan sistem repositori web, menyoroti pentingnya penilaian berkala, namun tidak membahas secara mendalam dampak dari risiko yang ditandai sebagai peringatan informasional.

B. Identifikasi Website

Analisis awal dilakukan untuk memahami struktur situs web X, fitur interaktif, teknologi yang digunakan, serta potensi celah keamanan. Situs web diuji untuk mengidentifikasi komponen penting seperti sistem autentikasi dan formulir input pengguna. Wireshark digunakan untuk memonitor lalu lintas jaringan dan mendeteksi aktivitas mencurigakan [10].

C. Skenario pengujian

Beberapa tahap pengujian keamanan dilakukan dengan OWASP ZAP, Acunetix, dan Arachni. Pengujian mencakup:

a. OWASP ZAP: Menggunakan fitur Spider untuk *crawling* halaman dan melakukan fuzzing pada form input seperti login dan transaksi. Payload SQL Injection seperti ' OR 1=1 -- dan XSS <script>alert('XSS')</script> digunakan untuk menguji keamanan input aplikasi.
b. Acunetix: Melakukan pemindaian otomatis untuk mendeteksi kerentanan seperti SQL Injection dan XSS.
c. Arachni: Melakukan pemindaian otomatis dalam url yang sudah teridentifikasi.
d. Alat ini juga menyediakan rekomendasi mitigasi terhadap kerentanan dengan tingkat risiko tinggi.

D. Pengumpulan data

Proses pengumpulan data dilakukan untuk mendapatkan pemahaman menyeluruh terkait struktur, perilaku, dan potensi kerentanan pada situs web yang diteliti. Langkah pertama adalah *crawling* website, yaitu proses otomatis untuk menelusuri seluruh halaman di dalam situs untuk mengidentifikasi struktur serta elemen-elemen yang rentan terhadap serangan. Dengan teknik ini, informasi penting seperti

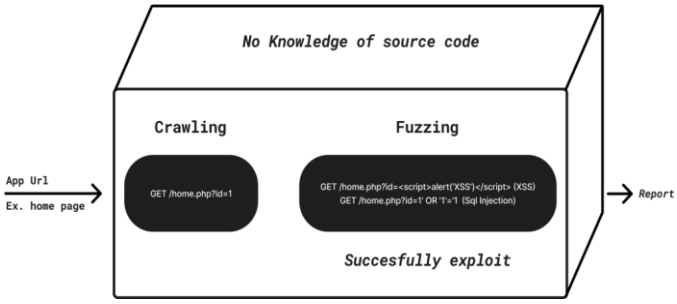
URL endpoint, form input, serta komponen lainnya dapat dikumpulkan secara sistematis.

Selanjutnya, dilakukan analisis struktur website untuk memahami bagaimana halaman-halaman dalam situs saling terhubung serta bagaimana data disimpan dan diproses. Analisis ini membantu dalam mengidentifikasi mekanisme keamanan yang diterapkan, seperti autentikasi pengguna dan enkripsi data. Setelah itu, penelitian berlanjut dengan identifikasi parameter input, yaitu pencarian terhadap elemen-elemen dalam situs yang menerima input dari pengguna, seperti form login, parameter dalam URL atau elemen interaktif lainnya. Parameter ini menjadi titik masuk potensial bagi serangan siber, sehingga penting untuk dipetakan dengan baik.

Langkah terakhir dalam proses pengumpulan data adalah evaluasi kerentanan, di mana hasil eksplorasi sebelumnya digunakan untuk menentukan titik kelemahan yang berpotensi dieksploitasi. Dengan memahami area yang memiliki risiko tinggi, pengujian keamanan dapat dilakukan secara lebih terfokus dan efektif.

E. Analisis Kerentanan (Black-box Fuzzing: XSS & SQL Injection)

Metode Black-box Fuzzing diterapkan dalam penelitian ini untuk mengidentifikasi kerentanan keamanan situs web, terutama terhadap serangan Cross-Site Scripting (XSS) dan SQL Injection. Langkah pertama dalam proses ini adalah identifikasi parameter rentan, yang dilakukan dengan melakukan *crawling* terhadap situs web untuk menemukan form input atau endpoint yang dapat dimanipulasi. Parameter ini kemudian menjadi target uji untuk mengevaluasi seberapa besar risiko keamanannya seperti pada *Gambar 2.5.1*.



Gambar 2.5.1 Black-box Fuzzing [21]

Setelah parameter rentan teridentifikasi, langkah berikutnya adalah pengiriman input berbahaya menggunakan teknik fuzzing. Dalam tahap ini, alat uji otomatis dikonfigurasi untuk mengirimkan berbagai payload berbahaya ke dalam form input atau parameter URL guna menguji reaksi sistem. Contohnya, untuk mendeteksi XSS, skrip seperti <script>alert('XSS')</script> dapat digunakan untuk melihat apakah sistem memvalidasi input dengan benar. Sementara itu, untuk SQL Injection, payload seperti ' OR 1=1 -- dikirimkan untuk menguji apakah sistem memiliki kelemahan dalam menangani kueri basis data [13].

Langkah terakhir dalam pengujian ini adalah observasi respon server, yaitu analisis terhadap bagaimana aplikasi web merespon input yang dikirimkan. Jika sistem menampilkan pesan kesalahan yang tidak diharapkan, mengalami crash, atau menampilkan perubahan tampilan yang mencurigakan, maka dapat disimpulkan bahwa situs tersebut memiliki kerentanan. Hasil dari berbagai alat uji, seperti OWASP ZAP, Acunetix, dan Arachni, kemudian diverifikasi secara silang untuk mengonfirmasi temuan dan memprioritaskan perbaikan pada celah keamanan yang paling kritis [13].

F. Evaluasi dan Analisis Hasil

Evaluasi dan analisis hasil dilakukan dengan tujuan untuk memahami tingkat keamanan situs web berdasarkan hasil uji yang telah dilakukan. Langkah pertama adalah mengumpulkan laporan dari berbagai alat uji, seperti OWASP ZAP, Acunetix, dan Arachni. Data yang diperoleh dari masing-masing alat kemudian dibandingkan untuk memperoleh gambaran menyeluruh terkait jenis-jenis kerentanan yang ditemukan serta frekuensi nya.

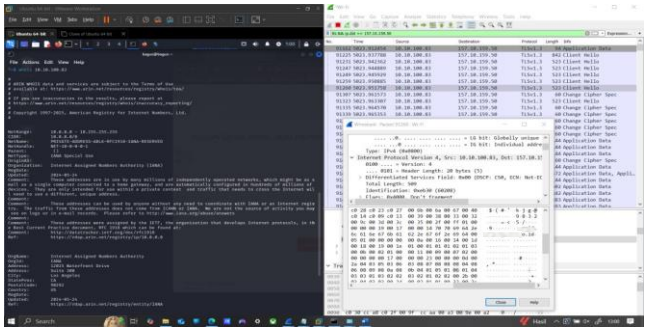
Setelah data dikumpulkan, dilakukan analisis dampak kerentanan untuk menentukan sejauh mana eksploitasi terhadap situs web dapat memengaruhi integritas, kerahasiaan, dan ketersediaan data. Beberapa faktor yang dianalisis mencakup kemungkinan pencurian data pengguna, manipulasi konten situs, serta potensi gangguan layanan akibat eksploitasi celah keamanan.

Berdasarkan hasil analisis, tahap akhir adalah menyusun rekomendasi mitigasi, yang bertujuan untuk memberikan solusi konkret dalam mengatasi dan mencegah eksploitasi serangan serupa di masa mendatang. Rekomendasi ini disusun berdasarkan referensi dari penelitian sebelumnya serta praktik terbaik dalam bidang keamanan siber. Dengan pendekatan ini, penelitian ini diharapkan dapat memberikan wawasan yang berguna dalam meningkatkan perlindungan situs web terhadap ancaman siber yang semakin kompleks

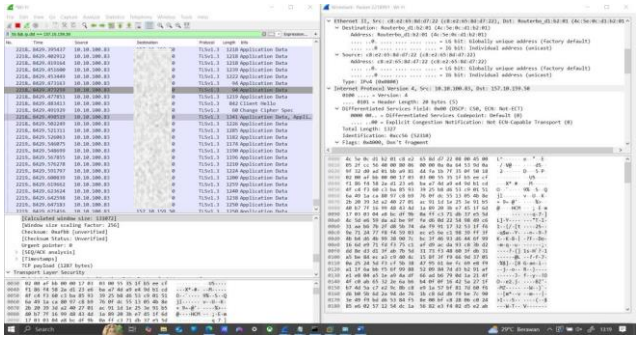
III. HASIL DAN PEMBAHASAN

A. Analisis Data Menggunakan Wireshark

Untuk mengidentifikasi perangkat yang mengakses situs web X, dilakukan pencarian alamat IP situs melalui DNS lookup untuk memastikan alamat server yang digunakan. Berdasarkan hasil nslookup, alamat IP yang terkait dengan situs X adalah 157.10.159.50. Selanjutnya, dilakukan proses pemantauan lalu lintas jaringan dilakukan menggunakan Wireshark dengan menerapkan filter ip.dst untuk menangkap paket yang menuju IP tersebut. Hasil dari filter ini menunjukkan beberapa perangkat dengan IP tujuan ke situs X, yaitu 10.10.100.48, 192.168.50.7, dan 10.10.100.83, seperti ditampilkan pada Gambar 3.1.1 dan Gambar 3.1.2. Alamat IP tersebut merupakan IP privat yang hanya dapat diakses dalam jaringan lokal, sehingga memerlukan akses ke perangkat jaringan internal seperti router atau server untuk identifikasinya.

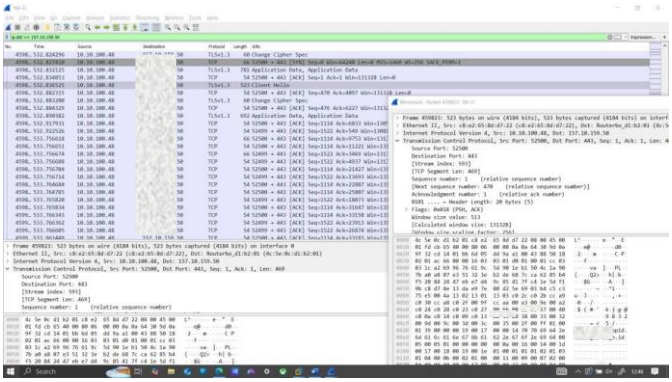


Gambar 3.1.1 Informasi IP 10.10.100.83



Gambar 3.1.2 Hasil Filter ip.dst Wireshark

Dari hasil analisis paket, diketahui bahwa situs X menggunakan protokol HTTPS dengan TLSv1.3. Hal ini menunjukkan bahwa komunikasi antara klien dan server dienkripsi, sebagaimana ditampilkan pada Gambar 3.1.3. Walaupun paket dapat ditangkap melalui Wireshark, isi komunikasinya tidak dapat dibaca karena dienkripsi. Informasi yang tersedia dalam tangkapan paket hanya meliputi alamat sumber dan tujuan, ukuran paket, serta jenis protokol yang digunakan.



Gambar 3.1.3 Hasil Apalication Data 10.10.100.83

Selain itu, dalam pengujian menggunakan perangkat seluler pribadi, tidak ditemukan lalu lintas data dari perangkat tersebut di Wireshark. Hal ini kemungkinan disebabkan oleh perbedaan protokol jaringan, di mana perangkat seluler menggunakan IPv6, sementara situs X hanya mendukung IPv4.

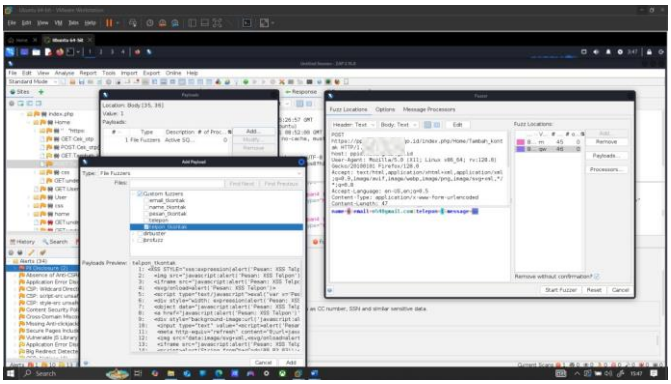
B. Pengujian dengan OWASP ZAP

OWASP ZAP digunakan untuk mengidentifikasi parameter input pada beberapa endpoint yang kemudian digunakan untuk pengujian lebih lanjut dengan metode fuzzing guna menguji potensi kerentanan terhadap XSS dan SQL Injection. Temuan endpoint dan parameter ini juga dikonfirmasi oleh Arachni dan Acunetix, sebagaimana dirangkum dalam Tabel 3.2.1.

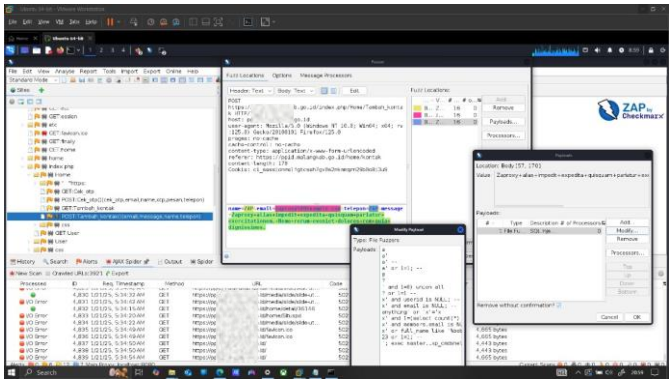
TABEL 3.2.1 ENDPOINT DAN PARAMETER YANG DIIDENTIFIKASI DENGAN TOOLS

NO	Endpoint target	Parameter masukan	Metode HTTP
1	https://x.go.id/index.php/User/Login	username, password	POST
2	https://x.go.id/index.php/User/Kirim	rincian_informasi, penggunaan_Informasi	POST
3	ttps://x.go.id/index.php/User/Save	nik, nama_lengkap, alamat, no_telepon, pekerjaan, email, username, password_hash	POST
4	https://x.go.id/index.php/Home/kontak	name, email, telepon, message	POST

Endpoint yang sudah diidentifikasi diuji dengan teknik fuzzing untuk menyisipkan skrip berbahaya ke dalam parameter input guna menguji kerentanan website terhadap serangan XSS dan SQL Injection. Berdasarkan hasil pengujian dengan OWASP ZAP, ditemukan celah keamanan pada halaman kontak, seperti yang ditunjukkan pada Gambar 3.2.1 dan Gambar 3.2.2.

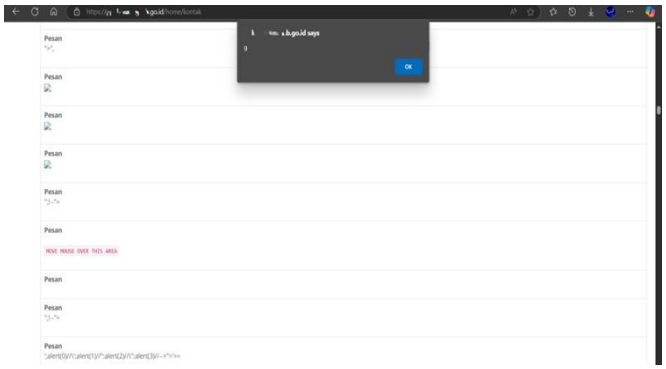


Gambar 3.2.1 Proses Fuzzing XSS dengan Payload

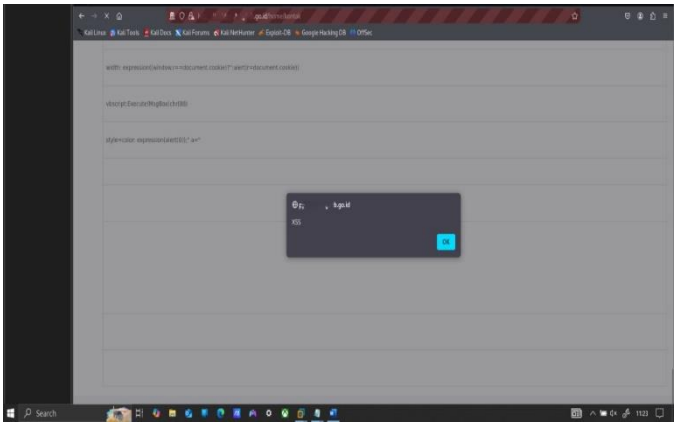


Gambar 3.2.2 Proses Fuzzing SQL Injection dengan Payload

Serangan ini dapat dieksekusi melalui parameter input seperti "nama", "telepon", "email", dan "pesan". Penyebab utama kerentanan ini adalah tidak adanya validasi atau pengecekan input yang memadai. Akibatnya, penyerang dapat menyisipkan karakter tertentu untuk menjalankan skrip berbahaya, memungkinkan terjadinya eksploitasi melalui XSS dan memiliki dampak pada website seperti pada Gambar 3.2.3 dan Gambar 3.2.4.



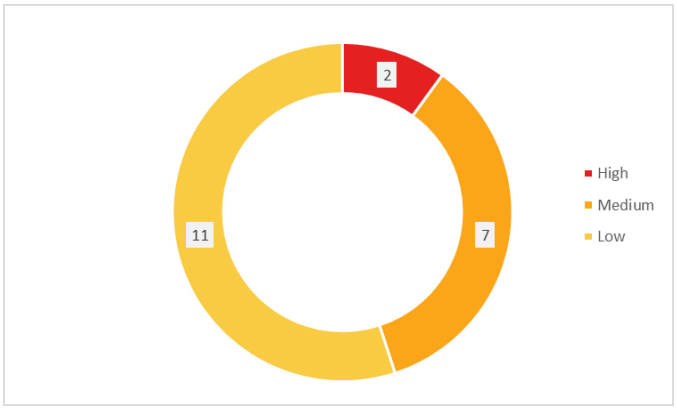
Gambar 3.2.3 Dampak Eksekusi Payload XSS Pertama



Gambar 3.2.4 Dampak Eksekusi Payload XSS

Sebagian besar kerentanan yang terdeteksi dalam pengujian ini tergolong dalam kategori medium dan low, sedangkan jumlah kerentanan dengan tingkat high lebih sedikit. Hal ini menunjukkan bahwa tingkat risiko bervariasi meskipun ancaman serangan tetap ada.

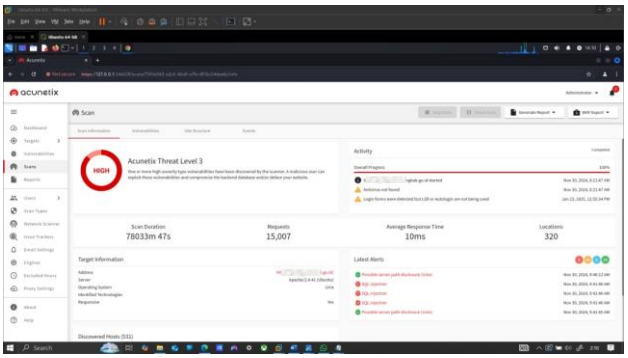
Jumlah kerentanan yang ditemukan oleh OWASP ZAP dapat dilihat pada Gambar 3.2.5.



Gambar 3.2.5 Jumlah Kerentanan yang teridentifikasi ZAP

C. Hasil Uji dengan Acunetix

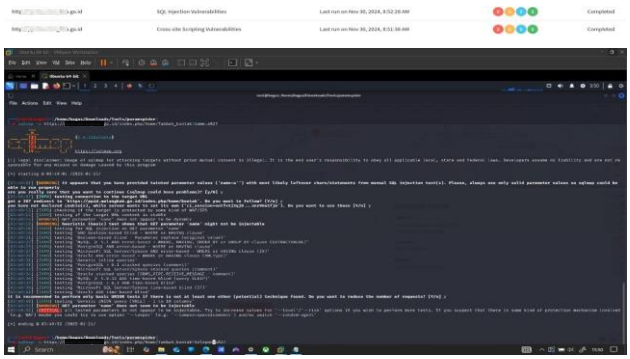
Pengujian dengan Acunetix tidak dapat melakukan fuzzing parameter input secara langsung. Namun, alat ini tetap mengidentifikasi beberapa potensi kerentanan, sebagaimana ditampilkan dalam Gambar 3.3.1.



Gambar 3.3.1 Hasil Analisa Acunetix

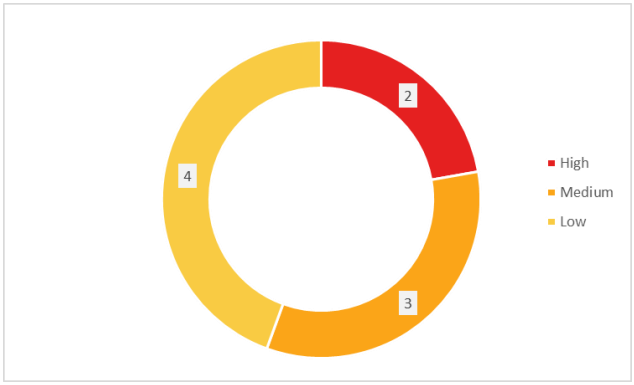
Hasil pengujian mengungkap beberapa kerentanan, termasuk SQL Injection, yang menjadi fokus utama penelitian ini. Untuk validasi, pengujian tambahan dilakukan dengan berbagai alat. Sebelumnya, pengujian menggunakan ZAP menunjukkan eksekusi skrip tanpa validasi input, namun tidak berdampak langsung terhadap basis data. Pengujian dengan sqlmap juga tidak menghasilkan reaksi berbahaya.

Saat dianalisis ulang menggunakan Acunetix dengan fokus pada XSS dan SQL Injection, tidak ditemukan kerentanan, sebagaimana yang ditunjukkan pada Gambar 3.3.2. Namun, Acunetix mendeteksi potensi eksploitasi karena kolom input masih memungkinkan penggunaan karakter khusus, yang berisiko digunakan dalam serangan SQL Injection.



Gambar 3.3.2. Pengujian Acunetix dan SQLmap untuk Validasi Kerentanan

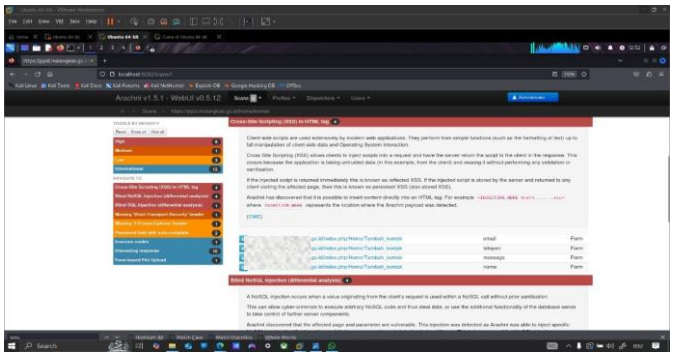
Acunetix cenderung menghasilkan laporan dengan jumlah temuan yang lebih sedikit. Mayoritas kerentanan yang terdeteksi masuk dalam kategori low, sementara kategori medium dan high hanya ditemukan dalam jumlah yang lebih terbatas. Hasil deteksi dari Acunetix divisualisasikan pada Gambar 3.3.3.



Gambar 3.3.3 Jumlah Kerentanan yang teridentifikasi Acunetix

D. Pengujian dengan Arachni

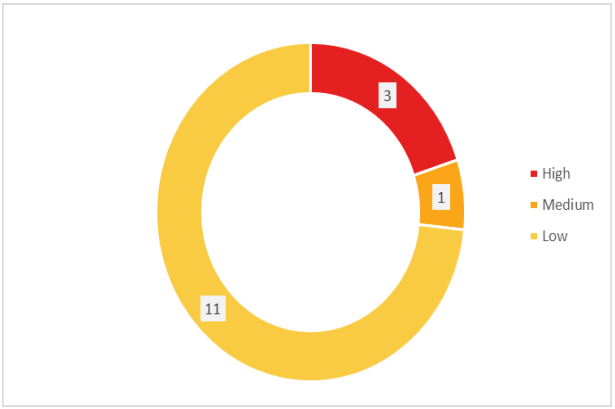
Arachni menemukan beberapa kerentanan XSS, seperti ditampilkan dalam Gambar 3.4.1. Hasil ini sejalan dengan temuan yang diperoleh menggunakan OWASP ZAP.



Gambar 3.4.1 Analisa Menggunakan Arachni

Arachni juga digunakan untuk mengidentifikasi kelemahan keamanan pada aplikasi web yang diuji. Berdasarkan hasil pengujian, Arachni mendeteksi tiga kerentanan high, termasuk serangan XSS, serta beberapa varian SQL Injection. Temuan ini menunjukkan adanya potensi eksploitasi yang lebih berbahaya. Visualisasi hasil deteksi oleh Arachni dapat dilihat pada Gambar 3.4.2

Gambar 3.4.2 Jumlah Kerentanan yang teridentifikasi Arachni



Dalam penggunaan alat Arachni, pengujian fuzzing dengan payload khusus melalui antarmuka baris perintah (CLI) tidak dapat dilakukan melalui CLI. Hal ini dikarenakan keterbatasan dokumentasi yang tersedia. Penulis tidak menemukan petunjuk atau informasi yang jelas mengenai cara melakukan fuzzing melalui CLI. Dokumentasi yang ada hanya menyebutkan bahwa Arachni memiliki fitur fuzzing, tetapi tidak memberikan rincian teknis atau panduan penggunaan fitur tersebut.

E. Evaluasi Hasil dan Mitigasi

Evaluasi Hasil dan Mitigasi Berdasarkan hasil pengujian, ditemukan beberapa kerentanan utama seperti XSS dan SQL Injection pada website x. Untuk mengurangi risiko ini, disarankan untuk menerapkan validasi input yang lebih ketat, menggunakan mekanisme keamanan seperti Content Security Policy (CSP), serta menerapkan parameterized queries guna mencegah SQL Injection. Selain itu, dianjurkan untuk menerapkan token CSRF untuk melindungi aplikasi dari serangan CSRF.

Berdasarkan hasil pengujian, OWASP ZAP menunjukkan performa terbaik dalam mendeteksi XSS dan SQL Injection dengan dukungan custom. Acunetix mampu menemukan beberapa kerentanan lebih cepat, namun akurasinya masih kurang optimal, terutama untuk XSS tidak teridentifikasi. Sementara itu, Arachni juga berhasil mendeteksi kedua kerentanan tersebut, tetapi keterbatasan dokumentasi membuat pengujian fuzzing dengan custom payload lebih sulit dilakukan dibandingkan ZAP. Secara keseluruhan, OWASP ZAP unggul dalam fuzzing, Acunetix lebih cepat dalam analisis tetapi kurang mendukung fuzzing, sedangkan Arachni cukup andal dalam deteksi namun terbatas dalam eksploitasi lebih lanjut.

IV. KESIMPULAN

Berdasarkan hasil penelitian, ditemukan bahwa website X memiliki beberapa kerentanan utama, terutama terhadap serangan Cross-Site Scripting (XSS) dan SQL Injection. Pengujian menggunakan OWASP ZAP, Acunetix, dan Arachni mengonfirmasi adanya celah keamanan pada beberapa

parameter input. OWASP ZAP terbukti paling efektif dalam mendeteksi dan mengeksploitasi kerentanan, karena mampu mengidentifikasi lebih banyak XSS dan SQL Injection dengan akurasi tinggi menggunakan dukungan custom payload. Sementara itu, Acunetix lebih unggul dalam kecepatan analisis namun kurang efektif dalam mendeteksi XSS. Arachni juga menunjukkan kemampuan deteksi yang baik, namun keterbatasan dokumentasi membatasi eksploitasi lebih lanjut.

Secara keseluruhan, OWASP ZAP menjadi pilihan terbaik untuk pengujian fuzzing, Acunetix unggul dalam kecepatan pemindaian tetapi tidak mendukung fuzzing secara langsung, dan Arachni cukup efektif dalam deteksi namun memiliki keterbatasan dalam melakukan eksploitasi lebih lanjut. Temuan ini menegaskan pentingnya penerapan mitigasi keamanan untuk mengurangi risiko eksploitasi.

V. REFERENSI

[1] V. R. S. Nastiti and D. R. Akbi, "Pengembangan sistem informasi dan majalah digital di pimpinan cabang Muhammadiyah Lawang," *Community Dev. J. J. Pengabdi. Masy.*, vol. 5, no. 1, pp. 1369–1373, 2024.

[2] M. Bin jafar Al Hamid, I. Nuryasin, and Z. Sari, "Penerapan progressive web application pada website Online Public Access Catalog (OPAC) UMM," *J. Repos.*, vol. 4, no. 2, 2022.

[3] A. Frik and L. Mittone, "Factors influencing the perception of website privacy trustworthiness and users' purchasing intentions: the behavioral economics perspective," *J. Theor. Appl. Electron. Commer. Res.*, vol. 14, no. 3, pp. 89–125, 2019.

[4] M. Q. Syahputra, D. R. Akbi, and D. Risqiwati, "Deteksi dan mitigasi serangan DDoS pada software defined network menggunakan algoritma decision tree," *J. Repos.*, vol. 2, no. 11, 2020.

[5] M. Zaidan, F. Noeraini, Z. Sari, and D. R. Akbi, "Website vulnerability analysis of AB and XY office in East Java," *JITEKI J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 2, pp. 455–492, 2023.

[6] X. Song, R. Zhang, Q. Dong, and B. Cui, "Grey-box fuzzing based on reinforcement learning for xss vulnerabilities," *Appl. Sci.*, vol. 13, no. 4, p. 2482, 2023.

[7] M. S. Rizal and others, "Perbandingan perlindungan data pribadi Indonesia dan Malaysia," *J. Cakrawala Huk.*, vol. 10, no. 2, pp. 218–227, 2019.

[8] Y. Azhar, Z. Sari, and A. S. Kholimi, "Optimasi pengelolaan data anggota melalui sistem informasi di pimpinan daerah Muhammadiyah kota Batu," *J. Abdimas BSI J. Pengabdi. Kpd. Masy.*, vol. 8, no. 1, pp. 74–83, 2025.

[9] Y. Ardiansah, "Analisis vulnerability Sistem Manajemen Tugas Akhir (Simanta) Universitas Muhammadiyah Malang," Universitas Muhammadiyah Malang, 2024.

[10] A. Alkatiri, "Analisis celah keamanan dan monitoring website menggunakan Owasp Zed Attack Proxy (ZAP) & Wazuh (Studi kasus: Website DUKCAPIL Kab. Nganjuk)," Universitas Muhammadiyah Malang, 2024.

[11] D. A. Arifah, "Kasus cybercrime di indonesia," *J. Bisnis dan Ekon.*, vol. 18, no. 2, 2011.

[12] A. Hidayat, "Analisis trafik abnormal menggunakan Wireshark (Studi kasus: Sister. ummetro. ac. id)," *Bull. Netw. Eng. Informatics*, vol. 1, no. 1, pp. 7–11.

[13] B. Zukran and M. M. Siraj, "Performance comparison on sql injection and xss detection using open source vulnerability scanners," in *2021 International Conference on Data Science and Its Applications (ICoDSA)*, 2021, pp. 61–65.

[14] A. Rajan and E. Erturk, "Web vulnerability scanners: A case study." [Online]. Available: http://testasp.vulnweb.com/

[15] K. Abdulghaffar, N. Elmrabit, and M. Yousefi, "Enhancing web application security through automated penetration testing with multiple vulnerability scanners," *Computers*, vol. 12, no. 11, p. 235, 2023.

[16] D. A. P. Sasongko, "Analisis metode vulnerability scanning dan perbandingan penetration testing dengan intrusion detection system terhadap vulnerable website," Universitas Muhammadiyah Malang, 2024.

[17] A. I. Fandy, "Pengujian celah keamanan website menggunakan teknik penetration testing dengan metode Owasp (Studi kasus: website rapor online SMP Muhammadiyah 1 Malang)," Universitas Muhammadiyah Malang, 2024.

[18] M. Zaidan, "Eksplorasi keamanan website Dinas XY di Jawa Timur dalam pendekatan pengujian brute force untuk mendeteksi kerentanan," Universitas Muhammadiyah Malang, 2024.

[19] A. Alsaedi, A. Alhuzali, and O. Bamasag, "Black-box fuzzing approaches to secure web applications: Survey", [Online]. Available: www.ijacsa.thesai.org

[20] O. Zaazaa and H. El Bakkali, "Dynamic vulnerability detection approaches and tools: State of the Art," in *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, 2020, pp. 1–6.

[21] A. Alsaedi, A. Alhuzali, and O. Bamasag, "Effective and scalable black-box fuzzing approach for modern web applications," *J. King Saud Univ. Inf. Sci.*, vol. 34, no. 10, pp. 10068–10078, 2022.

[22] D. Zhao, "Fuzzing technique in web applications and beyond," in *Journal of Physics: Conference Series*, 2020, p. 12109.

[23] M. Althunayyan, N. Saxena, S. Li, and P. Gope, "Evaluation of black-box web application security scanners in detecting injection vulnerabilities," *Electronics*, vol. 11, no. 13, p. 2049, 2022.

[24] R. Farismana and D. Pramadhana, "Perbandingan vulnerability assesment menggunakan owasp zap dan acunetix pada sistem informasi repositori politeknik negeri indramayu," *J. Tek*, 2023.