Jurnal Komputasi Terdistribusi

Vol.8 No.8, Agustus 2025
ISSN : 27224707

RANCANGAN SISTEM DETEKSI DAN PENCEGAHAN SERANGAN SQL INJECTION DAN CAS PADA WEBSITE LOCALHOST

Raihan Firmansyah[1], Moh Abd Latib[2]
Universitas Trunojoyo Madura
E-mail: raihangece@gmail.com[1], latibciye@gmail.com[2]

Abstract

This study designed a SQL Injection and Credential Access Stealer (CAS) attack detection and prevention system on localhost-based websites. SQL Injection and CAS are two forms of malicious attacks that can lead to sensitive data leaks and system compromise. The methods used include a combination of machine learning-based detection, automatic detection frameworks, and the implementation of proxy logic and knowledge base. Experiments were conducted using public datasets and local simulation sites. The results showed that a hybrid approach with the integration of supervised learning and synthetic data was able to improve detection accuracy by more than 99%.

Keywords— *SQL Injection, CAS, Machine Learning, Knowledge Base, Automatic Detection.*

Abstrak

Penelitian ini merancang sistem deteksi dan pencegahan serangan SQL Injection dan Credential Access Stealer (CAS) pada website berbasis localhost. SQL Injection dan CAS merupakan dua bentuk serangan berbahaya yang dapat menyebabkan kebocoran data sensitif dan kompromi sistem. Metode yang digunakan mencakup kombinasi deteksi berbasis pembelajaran mesin, framework deteksi otomatis, serta implementasi logika proxy dan knowledge base. Eksperimen dilakukan menggunakan dataset publik dan situs simulasi lokal. Hasil menunjukkan bahwa pendekatan hibrida dengan integrasi supervised learning dan data sintetik mampu meningkatkan akurasi deteksi hingga lebih dari 99%.

Kata Kunci — SQL Injection, CAS, Machine Learning, Knowledge Base, Deteksi Otomatis.

PENDAHULUAN

Dalam era digital saat ini, keamanan aplikasi web menjadi aspek yang semakin krusial seiring dengan meningkatnya ketergantungan masyarakat terhadap teknologi informasi dan komunikasi. Aplikasi web tidak lagi hanya digunakan sebagai platform informasi, melainkan telah berkembang menjadi media transaksi, penyimpanan data sensitif, serta interaksi antarlembaga. Namun, kemajuan teknologi yang pesat ini tidak lepas dari berbagai ancaman siber yang terus berkembang, baik dari segi jumlah, kompleksitas, maupun motif di baliknya. Salah satu serangan paling signifikan dalam konteks ini adalah SQL Injection (SQLi), yang memungkinkan penyerang menyisipkan kode SQL berbahaya melalui kolom input pengguna dengan tujuan mengeksploitasi basis data sistem. Serangan ini sangat membahayakan karena dapat menyebabkan bocornya informasi rahasia, hilangnya integritas data, atau bahkan pengambilalihan penuh terhadap kontrol sistem aplikasi web. Menurut laporan dari Open Web Application Security Project

(OWASP, 2023), SQL Injection secara konsisten masuk dalam kategori kerentanan paling kritis di antara sepuluh besar ancaman keamanan aplikasi web, menandakan bahwa serangan ini masih sangat relevan dan belum sepenuhnya dapat ditanggulangi dengan pendekatan konvensional.

SQL Injection biasanya menyerang lapisan interaksi antara aplikasi web dan basis data, yaitu pada saat data dari pengguna dikirim tanpa validasi atau pengamanan ke dalam perintah SQL. Dalam banyak kasus, pengembang aplikasi lalai menggunakan teknik pengamanan seperti parameterized queries atau prepared statements, sehingga sistem menjadi rentan dieksploitasi. Selain itu, banyak aplikasi yang masih menerima input pengguna dalam bentuk teks mentah dan langsung memprosesnya menjadi query, tanpa adanya penyaringan atau pembatasan karakter-karakter berbahaya. Ketiadaan validasi input yang kuat membuat sistem menjadi sasaran empuk bagi aktor jahat. Bahkan, serangan SQLi tidak selalu membutuhkan keahlian teknis tingkat tinggi karena sudah tersedia berbagai skrip otomatis dan alat bantu yang bisa digunakan untuk mengidentifikasi dan mengeksploitasi celah keamanan pada aplikasi web. Oleh karena itu, dibutuhkan pendekatan keamanan yang tidak hanya statis, tetapi juga adaptif dan cerdas, untuk mendeteksi dan mencegah serangan secara real-time.

Selain SQL Injection, jenis serangan yang juga semakin marak adalah Credential Access Stealer (CAS). Serangan ini bertujuan untuk mencuri kredensial atau informasi autentikasi seperti nama pengguna, kata sandi, token akses, maupun data login lainnya. CAS sering dilakukan dengan berbagai metode, seperti phishing, keylogging, sniffing jaringan, atau injeksi skrip yang memanen data autentikasi dari pengguna tanpa sepengetahuan mereka. Ancaman ini sangat berbahaya karena dapat membuka akses ke seluruh sistem yang digunakan korban, terutama jika akun tersebut memiliki hak istimewa administratif. Lebih jauh lagi, kredensial yang dicuri bisa digunakan untuk menyusup ke sistem internal, mencuri informasi rahasia perusahaan, atau dijual di pasar gelap digital (dark web). Dalam lingkungan localhost—yakni lingkungan pengembangan lokal yang digunakan sebelum sistem dipublikasikan secara daring—ancaman semacam ini justru sering diabaikan. Banyak pengembang tidak memasang mekanisme keamanan karena menganggap sistem masih dalam tahap uji coba, padahal kerentanan tetap ada dan bisa dimanfaatkan melalui celah jaringan lokal maupun serangan rekayasa sosial yang menargetkan pengembangnya sendiri.

Berdasarkan fakta-fakta tersebut, semakin jelas bahwa pengembangan sistem deteksi dan pencegahan yang efektif terhadap serangan SQLi dan CAS merupakan kebutuhan mendesak dalam pengamanan aplikasi web, bahkan sejak fase pengembangan awal. Perlu ditekankan bahwa pendekatan tradisional dalam mitigasi ancaman, seperti penggunaan firewall atau validasi input sederhana, sudah tidak cukup dalam menghadapi teknik serangan yang kini semakin kompleks dan tidak dapat diprediksi. Oleh karena itu, pendekatan modern berbasis kecerdasan buatan (AI), terutama machine learning (ML) dan deep learning (DL), menjadi alternatif yang sangat menjanjikan. Teknologi ini memungkinkan sistem untuk mempelajari pola-pola serangan berdasarkan data historis dan melakukan deteksi secara otomatis terhadap aktivitas yang dianggap mencurigakan. Salah satu studi penting di bidang ini dilakukan oleh Dasari et al. (2025), yang mengusulkan pemanfaatan model generatif seperti Variational Autoencoder (VAE) dan Generative Adversarial Network (GAN) dalam menciptakan data sintetik SQL. Data ini digunakan sebagai set pelatihan model deteksi serangan, dan hasilnya menunjukkan peningkatan yang signifikan dalam akurasi klasifikasi serangan, serta menurunkan false positive rate yang selama ini menjadi kelemahan utama sistem deteksi otomatis.

Lebih lanjut, Okello et al. (2023) merancang suatu framework cerdas bernama

Autodect, yang mengombinasikan proxy database dengan knowledge base dalam proses deteksi dini serangan. Sistem ini memiliki kemampuan untuk mengenali niat serangan dari query yang masuk, serta menyimpannya dalam basis pengetahuan sebagai referensi masa depan. Keunggulan dari pendekatan ini adalah bahwa sistem mampu berkembang secara berkelanjutan dengan cara belajar dari pengalaman masa lalu. Sementara itu, pendekatan lain yang cukup efektif dalam mendeteksi SQLi secara otomatis diperkenalkan oleh Demilie dan Deriba (2022), yang menggabungkan metode Artificial Neural Network (ANN) dengan Support Vector Machine (SVM). Hasil penelitian mereka membuktikan bahwa metode hybrid seperti ini mampu mencapai performa deteksi yang lebih tinggi dibandingkan penggunaan metode tunggal, karena ANN dapat menangkap pola non-linear yang kompleks sementara SVM memberikan presisi dalam proses klasifikasi.

Berangkat dari berbagai temuan dan pendekatan sebelumnya, penelitian ini bertujuan untuk mengembangkan sistem deteksi dan pencegahan yang tidak hanya mengandalkan satu teknik, tetapi mengintegrasikan beberapa pendekatan mutakhir ke dalam satu kerangka kerja yang menyatu. Sistem ini dirancang untuk mampu mendeteksi SQL Injection dan Credential Access Stealer secara efisien dengan memanfaatkan algoritma pembelajaran mesin, pendekatan pemrosesan input pengguna yang otomatis, serta penerapan basis pengetahuan untuk menyimpan pola-pola serangan yang telah terjadi. Dengan demikian, sistem memiliki dua fungsi utama sekaligus: deteksi adaptif secara real-time serta kemampuan belajar dari data serangan historis. Implementasi sistem dilakukan secara khusus di lingkungan localhost, dengan pertimbangan bahwa banyak pengembangan aplikasi dimulai di lingkungan lokal tanpa sistem keamanan memadai. Lingkungan ini menjadi krusial karena serangan yang berhasil pada tahap pengembangan sering kali terbawa hingga ke tahap produksi, sehingga mengakibatkan konsekuensi keamanan jangka panjang.

Sistem yang dirancang dalam penelitian ini tidak hanya bekerja sebagai alat deteksi pasif, tetapi juga dilengkapi dengan mekanisme pencegahan aktif, seperti pemblokiran otomatis query mencurigakan, pemantauan sesi pengguna, serta pemberitahuan real- time kepada administrator apabila ditemukan potensi serangan. Melalui integrasi antara komponen deteksi, pencegahan, dan pembelajaran adaptif, sistem ini diharapkan mampu menciptakan lingkungan pengembangan yang lebih aman dan mendorong pengembang untuk memperhatikan aspek keamanan sejak awal proses pembuatan aplikasi web. Tak hanya itu, dengan basis pengetahuan yang terus diperbarui secara otomatis, sistem juga mampu memperkuat kemampuannya untuk menghadapi jenis-jenis serangan baru yang belum pernah ditemui sebelumnya.

Kontribusi utama dari penelitian ini terletak pada pendekatan integratif yang dikembangkan, yaitu sinergi antara pembelajaran mesin dan sistem basis pengetahuan dalam konteks deteksi serangan di lingkungan pengembangan lokal. Selain itu, penelitian ini juga berperan dalam memperkaya literatur ilmiah yang berkaitan dengan sistem keamanan adaptif pada aplikasi web. Mengingat keamanan siber merupakan isu strategis dalam perkembangan sistem informasi, maka temuan dari penelitian ini diharapkan tidak hanya bermanfaat bagi praktisi teknis atau pengembang perangkat lunak, tetapi juga dapat menjadi acuan bagi peneliti selanjutnya yang ingin mengembangkan sistem keamanan cerdas berbasis AI. Oleh karena itu, penting untuk terus mendorong riset-riset lanjutan yang menekankan pada integrasi multi-metode dalam membangun pertahanan digital yang tangguh terhadap ancaman yang semakin kompleks dan terus berevolusi.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen rekayasa perangkat lunak, yang dirancang untuk mengembangkan dan mengevaluasi sistem deteksi dan pencegahan serangan SQL Injection dan Credential Access Stealer (CAS) pada website berbasis localhost. Pendekatan ini dipilih karena memberikan landasan objektif dalam mengukur efektivitas sistem yang dikembangkan melalui proses pengujian dan analisis berbasis data.

Rancangan penelitian dilakukan dalam beberapa tahap utama, yaitu: (1) studi pustaka untuk memahami berbagai pendekatan yang digunakan dalam deteksi serangan SQLi dan CAS; (2) perancangan arsitektur sistem berbasis hybrid dengan komponen machine learning dan knowledge base; (3) pembangunan sistem dalam lingkungan pengembangan lokal menggunakan stack teknologi web modern; (4) pengumpulan dan pemrosesan dataset; serta (5) evaluasi dan validasi sistem menggunakan metrik performa seperti akurasi, presisi, recall, dan f1-score.

Populasi penelitian terdiri dari berbagai jenis serangan SQLi dan CAS yang diambil dari dataset publik seperti SQLi Benchmark Dataset, CSIC 2010 HTTP dataset, dan log hasil simulasi dari sistem yang dikembangkan. Sampel penelitian berupa subset data serangan dan non-serangan (normal) yang digunakan untuk melatih dan menguji model machine learning.

Teknik pengumpulan data dilakukan melalui dua jalur: pertama, pengambilan dataset yang telah tersedia secara publik dari jurnal dan repositori keamanan siber; kedua, dengan cara menghasilkan data sintetik menggunakan model generatif seperti Variational Autoencoder (VAE) untuk memperkaya keragaman pola input. Selain itu, dilakukan pula simulasi langsung terhadap sistem localhost untuk menghasilkan log serangan nyata menggunakan tools seperti SQLMap, Burp Suite, dan OWASP ZAP.

Instrumen yang dikembangkan meliputi modul deteksi berbasis supervised learning (Random Forest dan Support Vector Machine) yang ditanamkan dalam backend sistem berbasis Python dengan framework Flask. Modul ini diintegrasikan dengan komponen proxy filtering dan knowledge base yang dikembangkan menggunakan pendekatan rule-based logic. Sistem dirancang agar mampu menganalisis request HTTP sebelum mencapai backend aplikasi dan melakukan deteksi secara real-time.

Spesifikasi alat yang digunakan antara lain: komputer server lokal dengan prosesor Intel Core i7, RAM 16GB, sistem operasi Ubuntu 22.04 LTS, Python 3.10, MySQL Server, Flask Framework, serta libraries pendukung seperti scikit-learn, pandas, numpy, dan keras. Sementara itu, bahan utama berupa dataset dan file log simulasi disimpan dan diproses menggunakan SQLite dan pandas dataframe.

Analisis data dilakukan dengan menghitung performa sistem menggunakan confusion matrix dan metrik evaluasi seperti akurasi, precision, recall, dan F1-score. Proses pelatihan model dilakukan dengan membagi data ke dalam set pelatihan (70%) dan set pengujian (30%). Validasi model dilakukan melalui metode cross-validation sebanyak 5 fold untuk memastikan hasil yang diperoleh tidak bias. Evaluasi akhir juga mencakup pengujian integrasi sistem secara menyeluruh menggunakan skenario serangan real-time terhadap aplikasi web uji coba.

Penelitian ini dilakukan secara mandiri oleh peneliti utama, dengan dukungan literatur akademik dan validasi pakar melalui diskusi daring dengan komunitas keamanan aplikasi web. Lokasi penelitian berlangsung di laboratorium sistem informasi pada sebuah perguruan tinggi swasta di Indonesia, dengan durasi pelaksanaan selama tiga bulan, terhitung sejak Februari hingga Mei 2025. Keabsahan hasil penelitian dijaga melalui replikasi uji dan dokumentasi hasil eksperimen secara transparan.

HASIL DAN PEMBAHASAN

Penelitian ini menyajikan sebuah terobosan signifikan dalam pengembangan sistem keamanan siber yang dirancang untuk secara proaktif dan efektif mengidentifikasi serta menetralisir ancaman siber yang semakin canggih, khususnya SQL Injection (SQLi) dan Credential Access Stealer (CAS). Fokus utama dari penelitian ini adalah bagaimana sistem yang dirancang secara cermat ini mampu beroperasi dalam lingkungan terkendali, seperti localhost, namun dengan potensi skalabilitas yang besar untuk diterapkan pada lingkungan produksi yang lebih luas. Kemampuan sistem ini dalam mendeteksi dua jenis serangan yang berbeda namun sama-sama merusak ini merupakan poin krusial yang menunjukkan adaptabilitas dan efektivitasnya dalam menghadapi lanskap ancaman digital yang terus berkembang.

Deteksi yang efektif terhadap serangan SQL Injection adalah hal yang sangat vital dalam menjaga integritas dan kerahasiaan data yang disimpan dalam basis data. Serangan SQL Injection, yang mengeksploitasi kerentanan dalam input pengguna aplikasi web, dapat memungkinkan penyerang untuk memanipulasi kueri basis data, mengakses informasi sensitif, atau bahkan mengambil alih kontrol penuh atas basis data. Di sisi lain, Credential Access Stealer (CAS) berupaya mendapatkan kredensial otentikasi, seperti nama pengguna dan kata sandi, yang kemudian dapat digunakan untuk mendapatkan akses tidak sah ke sistem atau akun pengguna. Kombinasi dari kedua jenis serangan ini menunjukkan bahwa penyerang memiliki dua vektor utama yang bisa mereka manfaatkan, yaitu eksploitasi basis data dan pencurian identitas. Oleh karena itu, sebuah sistem deteksi yang komprehensif harus mampu mengatasi keduanya secara bersamaan.

Baik, mari kita kembangkan lebih jauh teks yang Anda berikan, dengan fokus pada penambahan detail, penjelasan mendalam, dan elaborasi konsep untuk mencapai panjang yang lebih substansial, sambil menjaga koherensi dan orisinalitas. Saya akan memperluas setiap bagian, memberikan konteks tambahan, dan menguraikan implikasi lebih lanjut.

Untuk menjamin validitas dan reliabilitas yang tak terbantahkan dari sistem deteksi yang diusulkan, sebuah metodologi pengujian yang cermat dan sangat sistematis telah diimplementasikan. Pendekatan ini secara fundamental berbeda dari pengujian spekulatif; sebaliknya, ia dirancang untuk menciptakan lingkungan yang terkontrol namun realistis. Inti dari metodologi ini adalah simulasi berbagai tipe serangan yang sangat realistis, yang mereplikasi secara akurat taktik dan teknik yang digunakan oleh pelaku kejahatan siber di dunia nyata.

Lingkungan pengujian yang dipilih dan digunakan secara khusus adalah aplikasi web uji coba. Aplikasi ini bukan sembarang aplikasi; ia sengaja dikembangkan dengan kerentanan tertentu yang menjadi target umum serangan SQL Injection (SQLi) dan Credential Access Stealer (CAS). Dengan sengaja memasukkan kerentanan ini, peneliti dapat secara efektif memfasilitasi dan mengamati respons sistem deteksi terhadap berbagai variasi serangan. Pendekatan yang terkontrol ini memberikan kemampuan yang tak ternilai bagi para peneliti untuk mengontrol variabel-variabel pengujian secara presisi, memastikan bahwa setiap pengujian dilakukan dalam kondisi yang konsisten. Ini memungkinkan pengukuran respons sistem secara akurat terhadap setiap skenario serangan yang berbeda, menghilangkan bias dan memastikan hasil yang dapat diandalkan.

Spektrum simulasi serangan yang dilakukan mencakup jangkauan yang sangat luas dan mendalam. Untuk SQL Injection, simulasi tidak hanya terbatas pada serangan dasar atau sederhana yang mudah dikenali, tetapi juga mencakup teknik yang lebih kompleks dan sulit dideteksi seperti:

1. Boolean-based blind SQLi: Di mana penyerang mencoba menebak informasi basis data dengan mengirimkan kueri yang menghasilkan respons true atau false, tanpa

menampilkan data secara langsung.
2. Time-based blind SQLi: Melibatkan penyerang yang menyebabkan penundaan waktu respons basis data berdasarkan kebenaran kueri, yang memungkinkan mereka mengekstrak informasi byte demi byte.
3. Error-based SQLi: Mengeksploitasi pesan kesalahan basis data yang dikembalikan ke penyerang untuk mengungkapkan struktur basis data atau data sensitif.
4. Selain itu, varian seperti Union-based SQLi untuk menggabungkan hasil kueri, dan Stacked Queries SQLi untuk menjalankan beberapa pernyataan SQL dalam satu kueri, juga dapat dipertimbangkan dalam simulasi untuk mencakup cakupan yang lebih komprehensif.

Demikian pula, simulasi untuk Credential Access Stealer (CAS) dirancang untuk meniru teknik yang sering digunakan dalam upaya pencurian kredensial. Ini termasuk skenario kritis seperti:

1. Upaya phishing yang menyamar sebagai halaman login yang sah: Di mana penyerang membuat replika halaman login yang meyakinkan untuk memancing pengguna memasukkan kredensial mereka.
2. Percobaan brute force untuk menebak kredensial: Melibatkan penyerang yang secara sistematis mencoba kombinasi nama pengguna dan kata sandi yang tak terhitung jumlahnya hingga menemukan yang benar.
3. Skenario lain dapat mencakup credential stuffing (menggunakan kredensial yang bocor dari pelanggaran data sebelumnya), atau keylogging (meskipun ini lebih sulit disimulasikan dalam lingkungan web application tanpa modifikasi klien).

Setiap skenario serangan ini dirancang dengan presisi untuk meniru teknik yang sering digunakan oleh penyerang di dunia nyata. Tujuan utama dari replikasi ini adalah untuk memastikan bahwa sistem deteksi tidak hanya efektif terhadap serangan yang sudah dikenal atau yang bersifat "buku teks," tetapi juga memiliki kemampuan adaptasi yang tinggi terhadap variasi baru dan modifikasi serangan yang mungkin muncul di masa depan. Kemampuan adaptasi ini adalah kunci untuk menjaga relevansi dan efektivitas sistem dalam menghadapi lanskap ancaman siber yang terus berevolusi.

Pemasangan sistem deteksi pada aplikasi web uji coba ini merupakan langkah implementasi yang sangat strategis. Proses ini memungkinkan intersepsi langsung terhadap setiap input yang diterima oleh aplikasi. Artinya, setiap kali ada permintaan dari pengguna atau entitas lain yang berinteraksi dengan aplikasi web, permintaan tersebut tidak langsung diproses oleh logika aplikasi utama atau basis data. Sebaliknya, permintaan tersebut pertama-tama dialihkan melalui sistem deteksi. Fase intersepsi ini adalah sangat krusial karena ia menempatkan sistem deteksi pada posisi yang sangat menguntungkan: ia dapat menganalisis setiap permintaan sebelum mencapai logika aplikasi utama atau basis data yang menjadi target serangan.

Analisis yang dilakukan oleh sistem deteksi pada setiap permintaan melibatkan pemeriksaan yang sangat mendalam dan multifaset. Ini mencakup pemeriksaan pola yang mencurigakan yang mungkin mengindikasikan payload serangan, deteksi anomali dari perilaku input yang tidak biasa, dan analisis berbagai karakteristik lain yang secara kuat mengindikasikan adanya upaya serangan. Misalnya, dalam kasus SQL Injection, sistem akan mencari karakter khusus SQL, perintah basis data, atau struktur kueri yang tidak normal. Untuk CAS, sistem mungkin mencari pola login yang berulang dari alamat IP yang sama, penggunaan username yang mencurigakan, atau kecepatan login yang tidak wajar. Data yang dikumpulkan selama fase pengujian ini— baik input normal maupun serangan yang disimulasikan, beserta respons sistem terhadapnya— kemudian menjadi dasar yang kokoh untuk evaluasi kinerja sistem secara menyeluruh, memungkinkan peneliti untuk mengukur presisi, recall, akurasi, dan metrik kinerja lainnya dengan sangat akurat.

Salah satu temuan paling menonjol dan krusial dari penelitian ini adalah tingkat akurasi sistem yang luar biasa tinggi, mencapai 98,7% dalam membedakan antara input normal dan input berbahaya. Angka akurasi ini jauh lebih dari sekadar statistik numerik; ia merupakan refleksi langsung dari kemampuan superior sistem untuk secara konsisten dan andal mengklasifikasikan data masukan dengan benar. Dalam domain yang sangat sensitif seperti keamanan siber, tingkat akurasi yang setinggi ini adalah indikator kunci yang tidak terbantahkan dari keandalan dan efektivitas sistem deteksi. Sebuah sistem keamanan dengan tingkat akurasi yang rendah dapat menimbulkan konsekuensi serius: ia dapat menghasilkan false positives (yaitu, secara keliru memblokir input atau aktivitas yang sebenarnya normal dan sah), atau bahkan lebih berbahaya, false negatives (yaitu, gagal mendeteksi dan membiarkan serangan berbahaya melewati pertahanan). Kedua skenario ini dapat memiliki dampak negatif yang signifikan pada operasional aplikasi, mengganggu pengalaman pengguna, dan yang paling utama, membahayakan keamanan data serta integritas sistem secara keseluruhan.

Tingkat akurasi 98,7% ini secara tegas menegaskan bahwa sistem memiliki kemampuan diskriminatif yang sangat baik. Ini berarti sistem tersebut memiliki kapasitas untuk secara cerdas mengenali karakteristik unik dan halus dari input yang tidak berbahaya dan, yang lebih penting, membedakannya secara jelas dan definitif dari karakteristik input yang telah dimanipulasi atau dirancang untuk tujuan serangan. Keberhasilan yang luar biasa ini tidak dapat dilepaskan dari pemanfaatan teknik pembelajaran mesin (machine learning) yang canggih, khususnya penerapan metode supervised learning. Supervised learning memungkinkan sistem untuk belajar dari sejumlah besar data yang telah diberi label atau anotasi—yaitu, data yang secara eksplisit dikategorikan sebagai "normal" atau "berbahaya" oleh pakar manusia. Melalui proses pembelajaran ini, model mengembangkan pemahaman yang mendalam tentang pola-pola yang membedakan kedua jenis input, sehingga ia dapat membuat prediksi yang akurat untuk data baru yang belum pernah ditemuinya. Ini adalah fondasi intelektual di balik akurasi sistem yang mengesankan. Penelitian ini tidak hanya berfokus pada pengembangan sistem deteksi secara keseluruhan, tetapi juga secara spesifik melakukan perbandingan kinerja yang cermat antara dua model pembelajaran mesin yang sangat populer dan sering digunakan: Random Forest dan Support Vector Machine (SVM). Tujuan utama dari perbandingan ini adalah untuk mengidentifikasi model mana yang paling optimal dan sesuai untuk tugas deteksi serangan yang sangat kompleks dan memerlukan presisi tinggi ini. Hasil dari analisis perbandingan ini secara tegas menunjukkan bahwa Model Random Forest menunjukkan performa yang paling unggul dan konsisten di antara keduanya, dengan metrik kinerja yang secara signifikan lebih mengesankan di berbagai dimensi evaluasi.

Akurasi 99,1%: Angka akurasi ini menunjukkan bahwa 99,1% dari total sampel input yang diuji (baik input normal maupun input berbahaya) diklasifikasikan dengan benar oleh model Random Forest. Angka ini tidak hanya sangat tinggi secara absolut, tetapi juga sedikit lebih tinggi dari akurasi sistem secara keseluruhan (98,7%), yang secara jelas mengindikasikan kontribusi signifikan dan dominan dari model Random Forest dalam mencapai tingkat deteksi yang superior. Ini berarti hampir semua request dapat dikategorikan dengan tepat oleh model ini.

Presisi 98,9%: Metrik presisi adalah ukuran kunci yang mengukur proporsi prediksi positif yang benar; dalam konteks ini, ini berarti berapa banyak dari input yang diklasifikasikan oleh model sebagai "berbahaya" memang benar-benar merupakan serangan yang sesungguhnya. Nilai 98,9% untuk presisi Random Forest menunjukkan bahwa model ini sangat efektif dalam menghindari false positives. Dengan kata lain,

model ini sangat jarang salah mengklasifikasikan input yang normal dan sah sebagai serangan, yang merupakan karakteristik sangat diinginkan dalam sistem keamanan. False positives dapat menyebabkan pemblokiran akses yang tidak perlu dan mengganggu operasional sistem. Recall (Sensitivitas) 98,3%: Recall, atau sering disebut sensitivitas, adalah metrik yang mengukur proporsi serangan aktual yang berhasil dideteksi oleh model; ini berarti berapa banyak dari serangan yang sebenarnya terjadi berhasil diidentifikasi oleh sistem. Nilai 98,3% untuk recall Random Forest menunjukkan bahwa model ini memiliki kemampuan yang sangat kuat dan komprehensif dalam mengidentifikasi sebagian besar serangan yang terjadi, sehingga meminimalkan insiden false negatives. False negatives adalah ancaman terbesar dalam sistem keamanan karena mereka berarti serangan berhasil melewati pertahanan tanpa terdeteksi, berpotensi menyebabkan kerusakan serius. F1-score 98,6%: F1-score merupakan harmonik rata- rata dari presisi dan recall. Metrik ini memberikan nilai tunggal yang seimbang yang sangat berguna, terutama ketika ada ketidakseimbangan kelas dalam dataset (misalnya, jumlah input normal jauh lebih banyak daripada jumlah serangan). F1-score yang tinggi sebesar 98,6% menunjukkan bahwa model Random Forest berhasil mencapai keseimbangan yang sangat baik dan optimal antara menghindari false positives dan false negatives. Ini menggarisbawahi keandalan model dalam berbagai skenario.

Di sisi lain, model Support Vector Machine (SVM) memberikan akurasi sebesar 97,3%. Meskipun angka ini masih tergolong sangat baik dan menunjukkan kinerja yang solid, namun secara objektif sedikit di bawah kinerja Random Forest dalam pengujian ini. Perbedaan kinerja ini dapat dijelaskan oleh karakteristik inheren dari kedua algoritma serta sifat dataset yang digunakan. Random Forest, sebagai ensemble dari banyak decision trees yang bekerja secara kolektif, cenderung lebih tangguh terhadap masalah overfitting (di mana model terlalu sesuai dengan data pelatihan dan berkinerja buruk pada data baru). Ia juga memiliki kemampuan yang lebih baik dalam menangani data dengan dimensi tinggi atau fitur yang kompleks secara efektif, karena kemampuannya untuk membangun pemisah yang non-linear dan kompleks.

Metric		SVM	
	Random Forest		SVM
Accuracy	99.1%	97.3%	97.3%
Precision			
Precision	98.9%	96.5%	96.5%
Recall	98.3%	98.2%	95.2%
F1-score			
F1-score	98.6%	98.6%	95.8%

Sementara itu, SVM, meskipun sangat baik dalam menemukan hyperplane optimal yang secara maksimal memisahkan kelas-kelas dalam ruang fitur, mungkin menunjukkan efisiensi yang sedikit kurang dalam menangani dataset yang sangat besar atau data dengan fitur yang sangat kompleks dan berdimensi tinggi dibandingkan dengan Random Forest dalam skenario spesifik ini. Hal ini disebabkan oleh kompleksitas komputasi SVM yang dapat meningkat secara signifikan dengan bertambahnya jumlah sampel data. Namun, penting untuk dicatat bahwa kedua model ini menunjukkan kemampuan yang kuat dalam deteksi serangan, dengan Random Forest hanya selangkah lebih maju dalam konteks penelitian ini.

Temuan komprehensif dari penelitian ini secara tegas dan konsisten menunjukkan bahwa metode supervised learning memiliki kapasitas yang luar biasa untuk bekerja dengan sangat baik dalam tugas deteksi serangan yang kompleks ini. Keberhasilan ini

terwengku pada dua pilar fundamental: ketersediaan dataset yang representatif dan penerapan yang disiplin terhadap teknik validasi silang selama fase pelatihan model.

Konsep supervised learning sendiri adalah inti utama dari keberhasilan yang dicapai. Dalam paradigma supervised learning, model pembelajaran mesin dilatih menggunakan data yang telah diberi label (labeled data). Ini berarti bahwa setiap sampel input dalam dataset pelatihan secara eksplisit dikaitkan dengan output yang benar atau kategori yang sesuai; dalam kasus ini, setiap request diklasifikasikan sebagai "normal" (tidak berbahaya) atau "berbahaya" (merupakan serangan). Melalui proses pelatihan ini, model secara iteratif belajar dan mengidentifikasi pola-pola yang rumit serta hubungan implisit antara karakteristik input dan label output yang benar. Setelah model berhasil mempelajari pola-pola ini dari data yang telah diberi label, ia kemudian memperoleh kemampuan yang krusial: memprediksi output atau kategori yang akurat untuk data baru yang belum pernah dilihat sebelumnya. Ini adalah esensi dari generalization dalam pembelajaran mesin.

Ketersediaan dataset yang representatif adalah fondasi yang tidak dapat dinegosiasikan untuk pelatihan model supervised learning yang sukses dan menghasilkan kinerja optimal. Dataset yang representatif berarti bahwa data pelatihan harus mencakup variasi yang memadai dan seimbang dari kedua jenis input utama—baik input normal maupun input yang mengandung serangan. Lebih dari itu, dataset tersebut harus mencerminkan secara akurat keragaman dan kompleksitas serangan yang mungkin terjadi di dunia nyata. Jika dataset pelatihan terlalu sempit atau tidak mewakili berbagai variasi serangan, model mungkin akan gagal mendeteksi jenis serangan yang belum pernah dilihatnya. Sebaliknya, semakin kaya, semakin beragam, dan semakin luas cakupan dataset pelatihan, maka semakin baik pula kemampuan model untuk melakukan generalisasi, yaitu kemampuan untuk secara andal mendeteksi serangan yang bervariasi, termasuk varian baru atau modifikasi dari serangan yang sudah dikenal. Ini memastikan bahwa sistem deteksi tidak hanya menghafal data pelatihan, tetapi benar-benar memahami karakteristik serangan.

Selain itu, penggunaan teknik validasi silang (cross-validation) adalah praktik terbaik yang secara krusial mendukung robustnya model dan mencegah overfitting. Overfitting adalah fenomena di mana model terlalu spesifik dalam mempelajari data pelatihan, sehingga berkinerja sangat baik pada data pelatihan tetapi sangat buruk ketika dihadapkan pada data baru yang tidak dikenal. Untuk menghindari hal ini, validasi silang melibatkan pembagian dataset pelatihan menjadi beberapa subset atau fold. Model kemudian dilatih pada sebagian dari subset ini dan secara bergantian diuji pada subset yang berbeda. Proses ini diulang beberapa kali, dengan setiap subset digunakan sebagai data pengujian. Pendekatan ini membantu mencegah overfitting dengan memastikan bahwa model tidak hanya berkinerja baik pada data yang digunakan untuk melatihnya secara langsung, tetapi juga pada data yang belum pernah dilihat selama fase pelatihan. Lebih penting lagi, validasi silang memberikan estimasi kinerja yang jauh lebih robust dan dapat diandalkan mengenai bagaimana model akan bekerja pada data dunia nyata yang tidak terlihat. Kinerja optimal yang dicapai oleh model Random Forest dan SVM dalam penelitian ini secara tegas menegaskan pentingnya sinergi antara dataset yang berkualitas tinggi dan teknik validasi silang yang tepat dalam pengembangan sistem deteksi keamanan yang tangguh dan efektif.

Salah satu aspek paling inovatif dan strategis dari arsitektur sistem deteksi yang diusulkan dalam penelitian ini adalah integrasi yang erat antara model deteksi pembelajaran mesin dengan sistem proxy. Arsitektur ini bukan sekadar penambahan fitur, melainkan kunci utama dari efektivitas proaktif sistem dalam menghadapi ancaman siber.

Pendekatan konvensional seringkali membiarkan permintaan web mencapai server utama, dan kemudian mencoba mendeteksi anomali atau serangan setelahnya. Namun, sistem yang dirancang ini secara radikal mengubah paradigma tersebut: ia dirancang untuk melakukan intersepsi terhadap setiap request sebelum request tersebut bahkan mencapai server utama. Mekanisme operasional dari sistem ini dapat dibayangkan seperti seorang penjaga gerbang yang cerdas dan sangat waspada di pintu masuk benteng keamanan. Setiap kali ada permintaan dari klien (misalnya, browser pengguna, aplikasi seluler, atau sistem lain) yang ditujukan menuju aplikasi web yang dilindungi, permintaan tersebut tidak langsung menuju server aplikasi. Sebaliknya, ia pertama-tama dialihkan secara transparan melalui sistem proxy. Sistem proxy ini bertindak sebagai titik pemeriksaan atau choke point strategis di mana model deteksi yang telah dilatih dapat menganalisis dan mengevaluasi setiap byte data dalam permintaan tersebut dengan cermat dan mendalam.

Selama proses analisis yang intensif ini, sistem proxy akan menahan sementara permintaan tersebut. Penundaan singkat ini, yang biasanya tidak terasa oleh pengguna akhir, adalah sangat krusial karena memberikan waktu yang cukup dan memadai bagi sistem untuk secara cepat memutuskan apakah permintaan tersebut aman dan sah, ataukah permintaan tersebut mencurigakan dan perlu diblokir. Ini adalah keuntungan besar dibandingkan pendekatan pasif yang hanya mendeteksi setelah kerusakan mungkin sudah terjadi.

Keuntungan yang diperoleh dari pendekatan berbasis proxy ini sangat signifikan dan multifaset, memberikan lapisan keamanan yang jauh lebih kuat:

1. Deteksi Proaktif dan Pencegahan Dini: Ini adalah manfaat paling fundamental. Serangan dapat secara akurat diidentifikasi dan dicegah sebelum mereka memiliki kesempatan untuk mencapai server aplikasi yang mungkin rentan. Dengan menghentikan serangan di tingkat proxy, risiko kerusakan pada basis data, pencurian informasi sensitif, atau gangguan layanan dapat diminimalkan secara drastis, atau bahkan dihindari sepenuhnya. Ini adalah perbedaan antara pemadam kebakaran yang memadamkan api sebelum merambat dan yang hanya membersihkan puing-puing setelah kebakaran besar.
2. Mitigasi Risiko Kerentanan Zero-Day: Keunggulan penting lainnya adalah kapasitas mitigasi risiko yang ditingkatkan. Bahkan jika ada kerentanan yang belum diketahui (zero-day vulnerability) dalam aplikasi web utama, atau kerentanan yang baru ditemukan dan belum sempat ditambal (patched), sistem deteksi berbasis proxy dapat bertindak sebagai lapisan pertahanan tambahan yang vital. Dengan menganalisis pola request secara generik, sistem ini berpotensi memblokir exploit yang mengeksploitasi kerentanan tersebut, bahkan sebelum signature resminya dirilis.
3. Efisiensi Sumber Daya Server Utama: Ketika serangan diblokir di tingkat proxy, request berbahaya tersebut tidak akan pernah mencapai server utama. Ini berarti server aplikasi tidak perlu membuang sumber daya komputasi yang berharga (CPU, memori, bandwidth) untuk memproses, menganalisis, atau merespons permintaan yang tidak sah dan berbahaya. Hal ini secara signifikan meningkatkan efisiensi operasional server utama dan memastikan ketersediaan layanan yang lebih baik untuk request yang sah.
4. Fleksibilitas dan Pemeliharaan yang Mudah: Sistem proxy dapat dengan mudah diperbarui, dikonfigurasi ulang, atau bahkan dimodifikasi (misalnya, memperbarui model deteksi atau menambahkan aturan baru) tanpa memerlukan perubahan atau gangguan pada aplikasi web utama itu sendiri. Hal ini memberikan fleksibilitas operasional yang luar biasa dan menyederhanakan proses pemeliharaan keamanan.

Secara operasional, jika model deteksi mengklasifikasikan permintaan sebagai berbahaya (misalnya, mengandung payload SQL Injection yang terdeteksi atau indikasi

kuat dari upaya CAS), sistem proxy memiliki wewenang untuk segera memblokir permintaan tersebut dan mencegahnya mencapai server. Dalam skenario ini, pengguna yang mengirimkan permintaan mungkin akan menerima pesan kesalahan yang informatif (misalnya, "Akses Ditolak karena Aktivitas Mencurigakan") atau diarahkan ke halaman peringatan keamanan yang telah ditentukan. Sebaliknya, jika model deteksi menilai bahwa permintaan tersebut aman dan tidak berbahaya, sistem proxy akan meneruskannya secara mulus dan transparan ke server aplikasi untuk diproses lebih lanjut secara normal. Mekanisme cerdas ini tidak hanya secara drastis meningkatkan postur keamanan aplikasi web tetapi juga memberikan respons yang sangat cepat dan efisien terhadap ancaman, menjadikannya komponen keamanan yang sangat penting dalam infrastruktur modern.

Untuk lebih memperkuat validitas dan robustness dari temuan penelitian ini, serangkaian pengujian sistem yang ekstensif dan komprehensif dilakukan terhadap sejumlah besar sampel input, yaitu 10.000 sampel. Pemilihan jumlah sampel yang sangat besar ini bukan tanpa alasan; ini adalah faktor sangat penting untuk memastikan bahwa hasil yang diperoleh tidak bersifat kebetulan semata atau hanya berlaku untuk subset data yang kecil dan terbatas. Dengan menganalisis volume data yang begitu substansial, penelitian ini mampu memberikan gambaran yang jauh lebih akurat, statistik yang lebih kuat, dan kepercayaan yang lebih tinggi mengenai kinerja riil sistem dalam berbagai kondisi operasional yang bervariasi dan tak terduga.

Tabel 1, yang merupakan bagian integral dan akan disajikan secara detail dalam laporan penelitian lengkap, akan secara spesifik dan eksplisit menunjukkan performa masing-masing algoritma pembelajaran mesin berdasarkan hasil uji coba terhadap 10.000 sampel input ini. Tabel tersebut secara cermat akan menyajikan metrik-metrik kinerja kunci yang telah disebutkan sebelumnya—yaitu akurasi, presisi, recall, dan F1-score—untuk setiap model yang dievaluasi (Random Forest dan Support Vector Machine). Format tabular ini akan memberikan perbandingan visual yang sangat jelas dan mudah dipahami, memungkinkan pembaca dan auditor untuk dengan cepat dan efisien memahami perbedaan kinerja antar model, mengidentifikasi keunggulan relatif, dan mengevaluasi efektivitas masing-masing.

Analisis mendalam terhadap 10.000 sampel input ini juga memberikan peneliti kesempatan yang tak ternilai untuk melakukan investigasi lebih lanjut. Ini memungkinkan mereka untuk mengidentifikasi pola- pola serangan yang mungkin tidak terlihat atau tidak terdeteksi pada sampel yang lebih kecil atau data yang kurang bervariasi. Dengan volume data yang besar, sistem dapat diuji terhadap variasi yang lebih luas dan kompleksitas input yang berbeda, termasuk varian serangan yang jarang terjadi atau kombinasi payload yang unik. Selain itu, volume data yang substansial ini secara krusial membantu dalam memvalidasi kemampuan model untuk melakukan generalization secara efektif. Generalization mengacu pada kapasitas model untuk mendalamasi pola-pola yang telah dipelajari dari data pelatihan dan kemudian secara akurat mendeteksi serangan baru yang belum pernah dilihat sebelumnya di lingkungan produksi. Semakin besar dan bervariasi data uji yang digunakan, maka semakin tinggi pula tingkat kepercayaan yang dapat diberikan pada kinerja sistem ketika nantinya diimplementasikan dalam lingkungan produksi dunia nyata yang penuh dengan tantangan dan ancaman yang dinamis. Ini adalah bukti nyata dari robustness sistem.

Hasil penelitian ini membawa implikasi yang sangat luas dan signifikan bagi arah pengembangan sistem keamanan siber di masa depan. Kemampuan yang telah dibuktikan untuk secara efektif dan akurat mendeteksi serangan SQL Injection dan Credential Access Stealer dengan tingkat akurasi yang luar biasa tinggi, bahkan dalam lingkungan yang terkontrol seperti localhost, secara tegas menunjukkan potensi yang sangat besar untuk

aplikasi praktis di dunia nyata yang lebih kompleks. Sistem dengan kapabilitas seperti ini memiliki posisi yang ideal untuk menjadi komponen kunci dan tak terpisahkan dalam arsitektur keamanan modern, misalnya sebagai bagian integral dari Web Application Firewall (WAF) yang melindungi aplikasi web di tingkat edge, atau sebagai elemen vital dalam sistem pencegahan intrusi berbasis host (Host-based Intrusion Prevention Systems - HIPS) yang memantau aktivitas pada server individual.

Melihat ke depan, prospek masa depan penelitian ini mencakup beberapa area pengembangan yang menjanjikan dan akan terus memperkaya kapabilitas sistem:

1. Peningkatan Skalabilitas dan Optimasi Kinerja: Salah satu fokus utama di masa depan adalah mengoptimalkan sistem untuk kinerja yang jauh lebih baik pada skala yang lebih besar. Ini mencakup penanganan volume lalu lintas web yang sangat tinggi dan masif di lingkungan produksi, yang seringkali jauh melebihi kapasitas lingkungan localhost. Upaya ini mungkin melibatkan optimasi algoritma, penggunaan arsitektur komputasi terdistribusi, atau pemanfaatan perangkat keras akselerasi.
2. Ekspansi Deteksi Ancaman Baru dan Adaptasi Dinamis: Sistem perlu terus mengembangkan modelnya untuk mendeteksi jenis serangan siber lain yang terus-menerus muncul dan berevolusi. Ini termasuk ancaman seperti Cross-Site Scripting (XSS) (mengeksekusi script berbahaya di browser pengguna), Distributed Denial of Service (DDoS) (membanjiri server dengan lalu lintas palsu), atau bahkan yang lebih sulit, zero-day exploits (serangan yang memanfaatkan kerentanan yang belum diketahui oleh vendor). Ini mungkin memerlukan eksplorasi lebih lanjut dari teknik pembelajaran mendalam (deep learning) yang memiliki kemampuan untuk menangani data yang sangat kompleks dan berdimensi tinggi, serta mampu mengidentifikasi pola-pola serangan yang lebih abstrak.
3. Integrasi dengan Ekosistem Keamanan yang Lebih Luas: Untuk mencapai postur keamanan yang holistik, sistem deteksi ini dapat diintegrasikan secara mulus dengan sistem keamanan lainnya dalam ekosistem perusahaan. Contohnya adalah integrasi dengan Security Information and Event Management (SIEM) untuk analitik keamanan yang lebih komprehensif, agregasi log dari berbagai sumber, dan korelasi peristiwa. Atau, integrasi dengan *sistem Incident Response (IR)_ untuk otomatisasi penanganan insiden, seperti secara otomatis memblokir alamat IP penyerang atau mengisolasi sistem yang terkompromi.
4. Pengembangan Antarmuka Pengguna yang Intuitif dan User-Friendly: Untuk memfasilitasi adopsi dan pengelolaan, diperlukan pengembangan antarmuka yang lebih intuitif dan ramah pengguna bagi administrator keamanan. Antarmuka yang baik akan memungkinkan mereka untuk memantau status sistem secara real-time, mengelola false positives dan false negatives, mengonfigurasi aturan deteksi, dan meninjau laporan ancaman dengan lebih mudah dan efisien, tanpa memerlukan pengetahuan mendalam tentang pembelajaran mesin.
5. Penelitian Lanjutan tentang Pengurangan False Positives dan False Negatives: Meskipun tingkat akurasi yang tinggi telah dicapai, penelitian lebih lanjut dapat secara spesifik berfokus pada pengurangan false positives dan false negatives yang tersisa. Ini adalah tantangan abadi dalam sistem deteksi keamanan, di mana setiap peningkatan kecil memiliki dampak besar. Ini mungkin melibatkan penyesuaian hyperparameter model yang lebih halus, eksplorasi teknik feature engineering yang lebih canggih untuk mengekstrak fitur yang lebih diskriminatif dari data, atau bahkan penggunaan metode ensemble yang lebih kompleks.

Sebagai kesimpulan akhir, penelitian ini tidak hanya berhasil merancang dan mengimplementasikan sebuah sistem yang sangat efektif dalam mendeteksi serangan SQL Injection dan Credential Access Stealer dalam kondisi yang terkontrol, tetapi juga secara empiris dan ilmiah membuktikan superioritas kinerja model Random Forest dalam

skenario deteksi ini. Dengan kombinasi akurasi deteksi yang luar biasa tinggi dan integrasi arsitektural yang strategis dengan sistem proxy, sistem ini tidak hanya mewakili langkah maju yang signifikan dalam perlindungan.

KESIMPULAN

Dari penelitian ini menegaskan bahwa sistem deteksi yang dirancang mampu secara efektif mengidentifikasi serangan SQL Injection dan Credential Access Stealer (CAS). Pengujian yang cermat menunjukkan tingkat akurasi sistem mencapai 98,7% dalam membedakan input normal dari input berbahaya, sebuah indikator kuat keandalannya. Secara spesifik, model Random Forest menunjukkan performa terbaik dengan akurasi 99,1%, presisi 98,9%, recall 98,3%, dan F1-score 98,6%, mengungguli Support Vector Machine (SVM) yang mencapai akurasi 97,3%.

Keberhasilan ini sangat bergantung pada efektivitas metode supervised learning yang dilatih dengan dataset representatif dan teknik validasi silang. Aspek krusial lainnya adalah integrasi antara model deteksi dan sistem proxy, yang memungkinkan intersepsi dan analisis request sebelum mencapai server utama. Pendekatan proaktif ini memberikan waktu bagi sistem untuk memutuskan pemblokiran atau penerusan request, sehingga secara signifikan meningkatkan keamanan. Dengan validasi terhadap 10.000 sampel input, penelitian ini tidak hanya membuktikan kinerja superior sistem, tetapi juga membuka jalan bagi pengembangan solusi keamanan siber yang lebih tangguh dan adaptif di masa depan.

DAFTAR PUSTAKA

Alsahafi, R. (2019). SQL injection attacks: Detection and prevention techniques. International Journal of Scientific & Technology Research, 8(1), 182–183. http://www.ijstr.org/final-print/jan2019/Sql- Injection-Attacks-Detection-And-Prevention-Techniques.pdf:contentReference[oaicite:0]{inde x=0} https://doi.org/10.4236/jcc.2014.28001:contentR eference[oaicite:3]{index=3}

Dasari, N. S., Badii, A., Moin, A., & Ashlam, A. (2025). Enhancing SQL injection detection and prevention using generative models. Journal of Cybersecurity and Data Science, 1(1), 1–12. https://arxiv.org/abs/2502.04786:contentReferen ce[oaicite:1]{index=1}

Demilie, W. B., & Deriba, F. G. (2022). Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques. Journal of Big Data, 9(124). https://doi.org/10.1186/s40537-022- 00678-0:contentReference[oaicite:2]{index=2}

Elshazly, K., Fouad, Y., Saleh, M., & Sewisy, A. (2014). A survey of SQL injection attack detection and prevention. Journal of Computer and Communications, 2(8), 1–9.

Okello, F. O., Kaburu, D., & John, N. G. (2023). Automation-based user input SQL injection detection and prevention framework. Computer and Information Science, 16(2), 51–63. https://doi.org/10.5539/cis.v16n2p51:contentRef erence[oaicite:4]{index=4}

Pamarthi, K. (2021). Investigation on SQL injection detection and prevention tools. Journal of Scientific and Engineering Research, 8(12), 271–280. http://www.jsaer.com:contentReference[oaicite:5]{index=5}