Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

INTEGRASI ODOO DAN WAZUH UNTUK DETEKSI DAN VISUALISASI SERANGAN SIBER DENGAN GRAFANA DI PT. XYZ

Muhammad Nazir [1]*, Antoni Haikal [2]**
* Rekayasa Keamanan Siber, Politeknik Negeri Batam
** Politeknik Negeri Batam
nazirkatury@gmail.com [1], antoni@polibatam.ac.id [2]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword:

Wazuh,
Odoo,
SIEM,
Log Interpretation,
Grafana

ABSTRACT

PT. XYZ relies on the Odoo Enterprise Resource Planning (ERP) platform to manage its inventory and invoicing processes, making the system a critical repository of sensitive operational data. However, Odoo logs cannot be interpreted directly by Security Information and Event Management (SIEM) platforms, because decoder not suited for odoo log format. Consequently, application-layer attacks leave traces in raw logs without ever being elevated to correlatable security events, creating a blind spot. This study addresses that log-interpretation gap by designing and implementing an integrated monitoring architecture comprising Odoo, Wazuh SIEM, and Grafana. The main contribution is the mapping of Odoo's log structure, covering werkzeug HTTP logs and res_users authentication logs, into Wazuh's SIEM architecture through custom decoders and detection rules. The monitoring scope focuses on the login form and URL input, covering common web attack patterns such as repeated authentication attempts and malicious URL manipulation. Evaluated in a controlled environment, the system achieved a 100% detection rate, a mean time to detect (MTTD) of 0.277 seconds, and a false positive rate of 3.00%, with real-time visualization through a Grafana dashboard.

I. PENDAHULUAN

PT. XYZ merupakan perusahaan yang mengandalkan Odoo, salah satu *platform Enterprise Resource Planning* (ERP) yang banyak digunakan, sebagai sistem inti untuk mengelola proses *inventory* dan *invoice*. Dengan peran tersebut, Odoo menyimpan data operasional yang sensitif, mulai dari catatan persediaan hingga transaksi penagihan pelanggan, sehingga menjadikannya aset kritis bagi keberlangsungan bisnis perusahaan[1]. Saat ini Odoo di PT. XYZ hanya dapat diakses melalui jaringan internal. Namun, perusahaan berencana membuka akses melalui internet untuk mendukung fleksibilitas operasional. Rencana ini secara langsung memperluas permukaan serangan sistem, sehingga kesiapan pengamanan ERP menjadi kebutuhan strategis yang harus dipenuhi sebelum perluasan akses dilakukan.

Sebagai aplikasi ERP berbasis web, Odoo memiliki riwayat kerentanan yang terdokumentasi resmi. Berdasarkan data *Common Vulnerabilities and Exposures* (*CVE*), berbagai kerentanan telah dilaporkan sejak tahun 2017 hingga 2023, meliputi *SQL injection*, *Cross-Site Scripting* (*XSS*), autentikasi, hingga directory traversal[2], yang kategorinya selaras dengan OWASP Top 10:2021, yaitu *Broken Access Control* (A01), *Injection* (A03), dan *Identification and Authentication Failures* (A07)[3]. Sebagian besar kerentanan Odoo baru dapat dieksploitasi setelah penyerang memperoleh akses atau sesi yang sah seperti *CVE*-2024-12368, ketika penyerang yang telah terautentikasi dapat mencuri token *OAuth* pengguna lain[4]. Kondisi ini menjadikan proses autentikasi sebagai gerbang kritis sehingga serangan *Brute Force* pada formulir *login* juga menjadi salah satu fokus deteksi. Sementara itu, serangan injeksi seperti *SQL injection*, *XSS*, dan *Path Traversal* tetap relevan bahwa meskipun inti Odoo terlindungi mekanisme *Object-Relational Mapping* (ORM)[5], modul pihak ketiga tetap membuka celah, sebagaimana dibuktikan *SQL injection* pada modul absensi biometrik pihak ketiga dalam *CVE*-2023-48050[6]. Berdasarkan pemetaan tersebut, penelitian ini fokus pada deteksi pada empat jenis serangan, yaitu *Brute Force* melalui formulir *login* serta *SQL injection*, *XSS*, dan *Path Traversal*

e-ISSN: 2548-6861

melalui *input* URL. Persoalannya, seluruh aktivitas tersebut hanya terekam sebagai log mentah di sisi server. Tanpa mekanisme yang mampu membaca dan menafsirkan log secara terpusat, indikasi serangan akan tertimbun di antara ribuan baris catatan aktivitas normal[7][8].

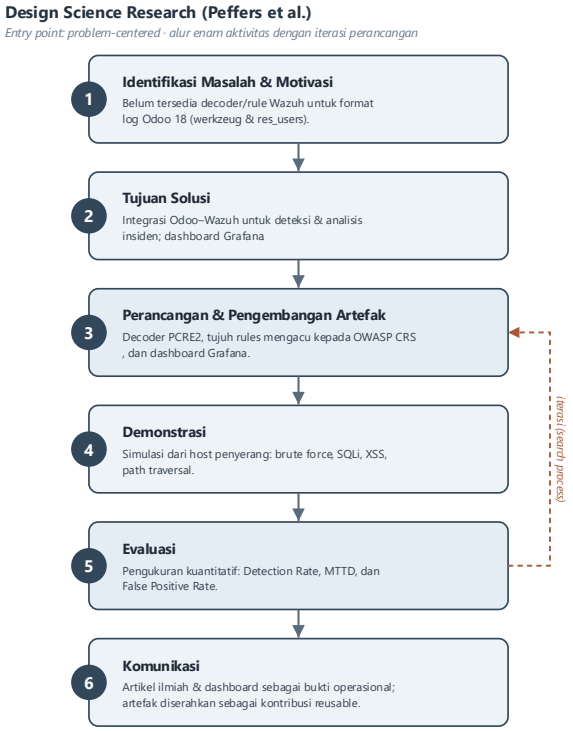
Kondisi ini menimbulkan kesenjangan interpretasi log antara Odoo dan *platform* pemantauan keamanan. Jejak serangan pada lapisan aplikasi memang tersimpan di dalam log Odoo, tetapi tidak terangkat menjadi insiden keamanan yang dapat dikorelasikan dan dianalisis, karena belum tersedia mekanisme yang memahami format log aplikasi tersebut. Akibatnya, meskipun perusahaan memasang sistem pemantauan, serangan yang menyasar Odoo berpotensi tetap tidak terdeteksi pada lapisan aplikasi.

Menjawab kesenjangan tersebut, penelitian ini mengusulkan integrasi *Security Information and Event Management* (SIEM) ke dalam lingkungan Odoo PT. XYZ. SIEM berbasis perangkat lunak *opensource* yang dinilai optimal dari sisi biaya dan keandalan[9]. Wazuh juga telah terbukti efektif mendeteksi serta memantau ancaman secara *real-time*[10]. Kemampuan deteksi tersebut perlu dilengkapi dengan visualisasi agar hasilnya dapat dianalisis dengan cepat. Pemilihan Grafana tidak didasarkan pada anggapan bahwa visualisasi bawaan Wazuh buruk, melainkan fleksibilitas sumber data karena Wazuh Dashboard terikat pada indeks OpenSearch sebagai satu-satunya sumber data, sedangkan Grafana mendukung banyak sumber data sehingga memungkinkan korelasi insiden keamanan dan pemantauan kesehatan server dalam satu tampilan dan kombinasi efektivitasnya telah dievaluasi pada penelitian sebelumnya [11][12][13].

Berbagai penelitian terdahulu telah membuktikan efektivitas implementasi Wazuh untuk pemantauan keamanan server dan jaringan[9][10]. Akan tetapi, penelitian-penelitian tersebut umumnya bekerja pada log sistem operasi atau perangkat jaringan yang formatnya telah dikenali Wazuh secara bawaan, dan hingga saat ini, belum ditemukan literatur yang secara spesifik menjembatani log aplikasi Odoo ke dalam arsitektur deteksi *SIEM*. Dari celah tersebut, penelitian ini berfokus pada implementasi dan integrasi Odoo dengan Wazuh serta Grafana di lingkungan PT. XYZ, dengan lingkup pemantauan tahap awal pada formulir *login* dan *input* URL sebagai titik interaksi yang paling awal terpapar. Kontribusi utama penelitian ini adalah memetakan struktur log Odoo ke dalam standar arsitektur *SIEM* Wazuh melalui *decoder* dan *rule* yang dirancang khusus sehingga memungkinkan korelasi insiden keamanan yang sebelumnya tidak bisa dilakukan.

II. METODE

Penelitian ini dirancang dengan paradigma *Design Science Research (DSR)* yang berorientasi pada pembangunan dan evaluasi artefak untuk menyelesaikan masalah, mengikuti pedoman alur prosedural enam aktivitas, yaitu identifikasi masalah dan motivasi, penetapan tujuan solusi, perancangan dan pengembangan artefak, demonstrasi, evaluasi, dan komunikasi[14]. Subbagian A sampai F berikut menjabarkan keenam aktivitas tersebut secara berurutan, dengan hasil evaluasi dipaparkan pada Bagian III. Keseluruhan tahapan ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian DSR

A. *Identifikasi masalah dan motivasi*

Permasalahan yang diangkat adalah Wazuh belum menyediakan *decoder* dan *rule* siap pakai yang memahami format log Odoo, baik log HTTP dari komponen werkzeug maupun log autentikasi dari modul res_users. Akibatnya, jejak serangan pada lapisan aplikasi tetap tersimpan sebagai log mentah dan tidak terangkat menjadi insiden keamanan yang dapat dikorelasikan serta dianalisis.

Permasalahan tersebut semakin kompleks karena Odoo mencatat *query string* dalam bentuk *decoded* atau literal, berbeda dengan asumsi yang umum digunakan pada *signature* deteksi yang berbasis *percent-encoded*. Perbedaan format ini menyebabkan *rule* deteksi generik tidak bekerja secara optimal apabila tidak disesuaikan dengan karakteristik log Odoo. Oleh karena itu, diperlukan *decoder* dan *rule* yang dirancang agar proses deteksi dapat berjalan secara efektif.

Sebagai dasar dalam menentukan jenis serangan yang menjadi fokus deteksi, penelitian ini melakukan penilaian

resiko secara kualitatif terhadap sistem Odoo yang dilakukan pada lingkungan pengujian menggunakan kerangka ISO/IEC 27005 sebagaimana diterapkan dalam penelitian sebelumnya[15]. Tingkat kemungkinan (*likelihood*) ditentukan berdasarkan data publik yang meliputi riwayat kerentanan Odoo pada National Vulnerability Database (NVD), statistik insiden OWASP Top 10:2021, serta laporan Verizon 2025. Tingkat dampak (*impact*) ditentukan berdasarkan aset yang dikelola Odoo di PT. XYZ, yaitu data *inventory* dan *invoice*. Ringkasan penilaian resiko disajikan pada Tabel 1.

Ancaman	Kemungkinan	Dampak	Tingkat Resiko
Brute Force	Tinggi — sebagian besar serangan aplikasi web melibatkan penyalahgunaan kredensial yang dicuri, sehingga B*rute Force* menjadi ancaman yang relevan[16]	Tinggi — akses terhadap data ERP	Tinggi
SQL Injection	Sedang — kerentanan masih dapat muncul pada modul pihak ketiga maupun implementasi yang tidak mengikuti praktik yang aman[17]	Tinggi — manipulasi atau kebocoran data *inventory* dan *invoice*	Tinggi
XSS	Sedang — masih terdapat riwayat kerentanan XSS pada Odoo[18]	Sedang — pembajakan sesi dan pencurian data	Sedang–Tinggi
Path Traversal	Sedang — tercatat pada riwayat kerentanan Odoo[19]	Tinggi — pembacaan berkas konfigurasi dan kredensial server	Tinggi

Tabel 1. Penilaian Resiko Kualitatif terhadap Sistem Odoo

Odoo telah menerapkan berbagai kontrol preventif, seperti ORM, token CSRF, dan pembaruan perangkat lunak secara berkala. Meskipun demikian, kontrol tersebut hanya mengurangi kemungkinan eksploitasi dan tidak dapat menghilangkan resiko sepenuhnya. Dalam ISO/IEC 27005, resiko yang masih tersisa setelah penerapan kontrol disebut sebagai *residual risk*. Resiko tersebut perlu dikelola melalui pemantauan berkelanjutan menggunakan kontrol detektif, sehingga setiap upaya eksploitasi dapat segera diidentifikasi dan divisualisasikan. Berdasarkan hasil penilaian resiko pada Tabel 1, empat ancaman dengan tingkat resiko tertinggi, yaitu *Brute Force*, *SQL Injection*, XSS, dan *Path Traversal*, dipilih sebagai fokus pengembangan *rule* deteksi pada penelitian ini.

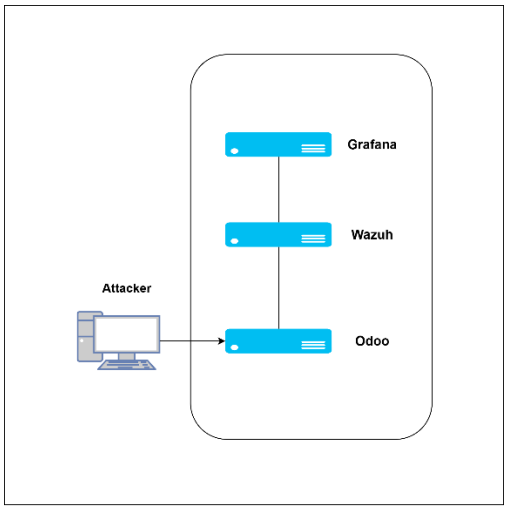
B. Tujuan Solusi

Penelitian ini menetapkan dua tujuan utama, yaitu mengembangkan integrasi Odoo dengan Wazuh sehingga log dan insiden keamanan pada server Odoo dapat dikirim, dikenali, dan dianalisis oleh Wazuh serta membangun *dashboard* Grafana yang menampilkan visualisasi *real-time* hasil pemantauan Wazuh terhadap Odoo. Ruang lingkup deteksi dibatasi pada *Brute Force* berbasis *login* dan injeksi berbasis URL, sedangkan vektor *body* JSON-RPC dan *logging* basis data berada di luar cakupan.

C. Perancangan dan Pengembangan Artefak

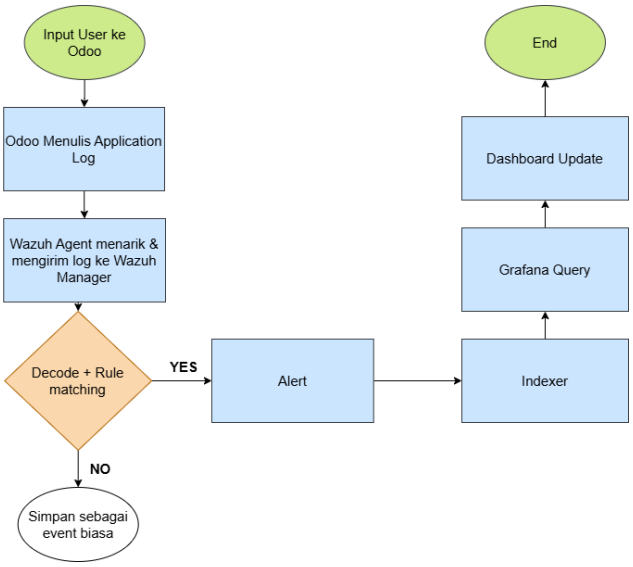
1) Lingkungan Uji

Lingkungan pengujian dibangun dalam satu segmen jaringan yang terdiri atas dua server dan satu komputer penguji. Server pertama menjalankan aplikasi Odoo, sedangkan server kedua menjalankan Wazuh sebagai SIEM dan Grafana sebagai media visualisasi. Seluruh skenario serangan dilakukan menggunakan komputer penguji yang berada pada jaringan yang sama. Topologi lingkungan pengujian dan arsitektur deteksi ditunjukkan pada Gambar 2 dan Gambar 3.



Gambar 2. Topologi Sistem

e-ISSN: 2548-6861



Gambar 3. Arsitektur Deteksi

Pada server monitoring, Wazuh dan Grafana ditempatkan pada host yang sama sehingga komunikasi antara Grafana dan Wazuh Indexer berlangsung secara lokal. Grafana dikonfigurasi dengan autentikasi aktif, akses anonim dinonaktifkan, serta menggunakan akun sumber data yang memiliki hak akses *read-only* terhadap indeks *alert* pada Wazuh Indexer. Akses ke layanan Grafana dibatasi hanya dari jaringan internal, sementara komunikasi dengan Wazuh Indexer dilakukan melalui protokol HTTPS yang diamankan menggunakan TLS. Konfigurasi tersebut memastikan Grafana berfungsi semata sebagai media visualisasi tanpa memiliki hak untuk memodifikasi data.

2) Decoder

Komponen utama artefak yang dikembangkan adalah *decoder* kustomisasi berbasis PCRE2. Struktur *decoder* yang dikembangkan pada penelitian ini ditunjukkan pada Gambar 4. *Decoder* untuk log werkzeug mengekstrak request_uri, srcip, http_method, dan status_code, dengan request_uri memuat *path* beserta *query string* dalam satu *field* sehingga *payload* pada *query* maupun pada segmen *path* dapat tertangkap oleh mekanisme yang sama. *Decoder* untuk log res_users mengekstrak odoo_*login* dan srcip sebagai sumber sinyal *Brute Force*.

```
<decoder name="odoo-users">
  <prematch type="pcre2">odoo\.addons\.base\.models\.res_users:</prematch>
</decoder>

<decoder name="odoo-login-failed">
  <parent>odoo-users</parent>
  <prematch>Login failed</prematch>
  <regex type="pcre2">login:(.*?) from (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})</regex>
  <order>odoo_login, srcip</order>
</decoder>

<decoder name="odoo-werkzeug">
  <prematch type="pcre2">werkzeug: \d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3} - -</prematch>
</decoder>

<decoder name="odoo-http-request">
  <parent>odoo-werkzeug</parent>
  <prematch type="pcre2">"[A-Z]+ [^\s]+ HTTP/</prematch>
  <regex type="pcre2">werkzeug: (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}) - - \[[^\]]+\] "([A-Z]+) ((([A-Za-z0-9/_\-]*)([^\?\s]*))(?:\?[^\s]*)?) HTTP/[0-9.]+" (\d+)</regex>
  <order>srcip, http_method, request_uri, url_path, status_code</order>
</decoder>
```

Gambar 4. *Decoder* kostumisasi

3) Rule Deteksi

Contoh implementasi *rule* deteksi pada Wazuh ditunjukkan pada Gambar 5. Di atas *decoder* dibangun tujuh *rule* deteksi yang menerapkan mekanisme deteksi dua tingkat. *Rule* pada tingkat *Low* digunakan untuk mengidentifikasi indikasi awal berupa kemunculan metakarakter, sedangkan *rule* pada tingkat *High* digunakan untuk mengenali pola serangan yang telah memenuhi *signature* tertentu.

```
<!-- LEVEL 10 (Multiple user generated errors): >=10 login failed / 60s dari IP sama -->
<rule id="100102" level="10" frequency="10" timeframe="60">
  <if_matched_sid>100100</if_matched_sid>
  <same_source_ip />
  <description>Odoo: Brute Force Attempt (>=10 login failed / 60s)</description>
  <group>brute_force,authentication_failures,</group>
</rule>

<!-- ===== SQL INJECTION via URL | CRS REQUEST-942 ===== -->

<!-- LOW / level 6 — analog CRS PL2+ metacharacter rules: -->
<rule id="100110" level="6">
  <decoded_as>odoo-werkzeug</decoded_as>
  <field name="request_uri" type="pcre2">(?i)('|%27|%2527|"|%22|--|%2d%2d|;|%3b|\/\*|%2f%2a)</field>
  <description>Odoo: SQLi probe via URL Attempt </description>
  <group>recon,sql_injection,crs_942,</group>
</rule>
```

Gambar 5. Rule Kostumisasi

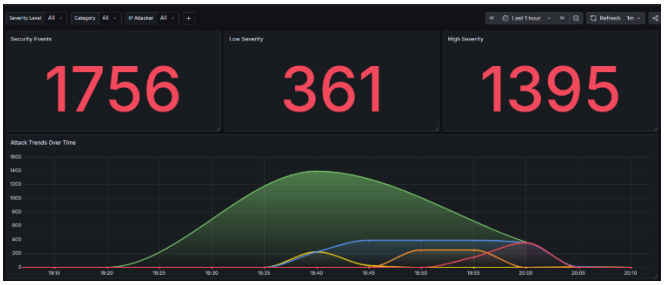
Untuk serangan *SQL Injection*, XSS, dan *Path Traversal*, pola deteksi disusun dengan mengacu pada OWASP Core Rule Set (CRS), yaitu kelas 942 untuk *SQL Injection*, kelas 941 untuk XSS, dan kelas 930 untuk *Path Traversal*[20]. Selanjutnya, pola dengan tingkat keyakinan tinggi dipetakan ke *rule* Wazuh level 12, sedangkan pola yang lebih sensitif dan berpotensi menghasilkan *false positive* dipetakan ke level 6. Berbeda dengan ketiga jenis serangan tersebut, deteksi *Brute Force* memanfaatkan mekanisme bawaan Wazuh, yaitu level 5 untuk setiap kegagalan autentikasi dan level 10 apabila terdeteksi sepuluh kali kegagalan login dalam interval 60 detik dari alamat IP yang sama. Ringkasan ketujuh *rule* deteksi disajikan pada Tabel I.

ID	Level	Kelas	Pemicu	CRS
100100	5	*BF Low*	Satu kali *login* gagal (res_users)	–
100101	10	*BF High*	≥10 gagal/60s dari IP sama	–
100110	6	*SQLi Low*	Metakarakter *SQL*	942
100111	12	*SQLi High*	UNION, tautologi, time/error-based	942
100120	6	*XSS Low*	Metakarakter	941
100121	12	*XSS High*	*Tag, event handler*, dan sink	941
100130	12	*Traversal*	/, /etc/passwd, wrapper file/php	930

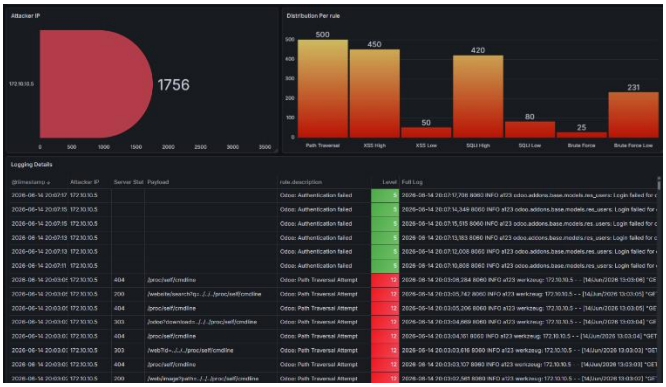
Tabel 2. Ringkasan *Custom Rule* Deteksi Odoo

4) Visualisasi pada Dashboard

Hasil deteksi disajikan pada *dashboard* Grafana yang mengambil sumber data dari Wazuh Indexer. *Dashboard* menyediakan filter tingkat keparahan, kategori, dan IP penyerang, serta panel ringkasan jumlah *event*, tren serangan dalam kurun waktu, distribusi *alert* per *rule*, IP penyerang teratas, dan tabel rincian log. Tampilan *dashboard* ditunjukkan pada Gambar 4 dan Gambar 5.



Gambar 6. Panel Ringkasan Security *Events*



Gambar 7. Panel Distribusi Alert dan Rincian Log

D. Demonstrasi

Pengujian dilakukan melalui simulasi serangan berbasis *script* Python yang dijalankan dari *host* penyerang. Skenario *Brute Force* mengirimkan 250 percobaan *login* melalui *form* /web/login dengan token *CSRF*, sedangkan skenario *SQLi*, *XSS*, dan *Path Traversal* masing-masing mengirimkan 500 *request* GET berisi *wordlist* yang memuat *probe* maupun *payload* penuh. Untuk mengukur *false positive*, satu skenario traffic sah berisi 200 *request* dijalankan secara terpisah, mencakup *login* sah, penelusuran JSON-RPC, *request* GET *query* bersih, serta 6 percobaan *login* salah ketik sebagai representasi kekeliruan manusiawi. Seluruh *alert* kemudian ditarik dari Wazuh Indexer, dengan total 1.756 *alert* sepanjang pengujian.

E. Evaluasi

Evaluasi dilakukan dengan mengukur kinerja deteksi melalui tiga metrik, dan hasil pengukurannya dipaparkan pada Bagian III.

1) Detection Rate

Detection Rate (DR) menyatakan serangan yang benar-benar berhasil diidentifikasi dengan benar.

$$DR = \frac{TP}{TP + FN}$$

Dalam konteks ini, istilah-istilah tersebut didefinisikan sebagai berikut:

- *True Positive* (TP): Jumlah serangan yang benar-benar terjadi dan berhasil dideteksi oleh sistem.
- *False Negative* (FN): Jumlah serangan yang terjadi tetapi tidak berhasil dideteksi oleh sistem.

2) Mean Time To Detect

Mean Time To Detect (MTTD) menyatakan rata-rata selisih waktu antara saat serangan dikirim dan saat *alert* terbentuk di Wazuh. MTTD menunjukkan durasi rata-rata yang diperlukan oleh sistem untuk mengidentifikasi suatu insiden keamanan sejak momen insiden tersebut terjadi.

$$MTTD = \frac{Total\ waktu\ deteksi}{Jumlah\ total\ insiden}$$

3) False Positive Rate

False Positive Rate (FPR) menyatakan aktivitas normal salah diidentifikasi sebagai sebuah serangan.

$$FPR = \frac{FP}{FP + TN}$$

Dalam konteks ini, istilah-istilah tersebut didefinisikan sebagai berikut:

- *False Positive* (FP): Jumlah insiden di mana sistem mendeteksi adanya serangan, padahal sebenarnya tidak ada serangan (alarm palsu).
- *True negative* (TN): Jumlah insiden di mana sistem tidak mendeteksi adanya serangan karena memang tidak ada serangan yang terjadi.

F. Komunikasi

Hasil penelitian dikomunikasikan melalui artikel ilmiah ini serta *dashboard* Grafana sebagai bukti operasional, sementara artefak berupa *decoder*, *rule*, skrip simulasi, dan *dashboard* diserahkan sebagai kontribusi yang dapat digunakan kembali.

III. Hasil dan Pembahasan

A. Detection Rate

Seluruh tujuh *rule* aktif selama pengujian, dan keempat jenis serangan terdeteksi seluruhnya dengan *Detection Rate* 100%. Rincian per jenis serangan beserta pembagian tingkat *Low* dan *High* disajikan pada Tabel II. Munculnya *alert* pada

e-ISSN: 2548-6861

kedua tingkat membuktikan strategi deteksi bertingkat bekerja sebagaimana dirancang: *probe* metakarakter ditandai sebagai *Low*, sedangkan *payload* terkonfirmasi ditandai sebagai *High*. *Path Traversal* hanya menghasilkan *alert High* karena diwakili satu *rule* tunggal.

Jenis Serangan	Dikirim	*Low*	*High*	Total	DR(%)
Brute Force	250	225	25	250	100,0
SQL injection	500	80	420	500	100,0
XSS	500	50	450	500	100,0
Path Traversal/LFI	500	0	500	500	100,0
Total	**1750**	**355**	**1395**	**1750**	**100,0**

Tabel 3. *Detection Rate*

B. Mean Time To Detect

Waktu deteksi rata-rata keseluruhan 0,277 detik dan median 0,267 detik, sebagaimana disajikan pada Tabel III. Nilai ini menunjukkan deteksi berlangsung mendekati waktu nyata, jauh lebih cepat dibanding penelusuran log secara manual.

Jenis Serangan	Sampel	Rata-rata(s)	Median	Maks
Brute Force	250	0,333	0,303	1,426
SQL injection	500	0,256	0,245	1,118
XSS	500	0,273	0,277	1,232
Path Traversal/LFI	500	0,273	0,226	1,239
Keseluruhan	1750	0,277	0,267	1,426

Tabel 4. *Mean Time To Detect*

C. False Positive Rate

Dari 200 *request* sah pada pengetesan, terdapat 6 *alert false positive* sehingga FPR sebesar 3,00%. *false positive* seluruhnya berasal dari *rule* 100100, yaitu 6 percobaan *login* salah ketik, sedangkan *rule* tingkat tinggi (level sepuluh ke atas) tidak menghasilkan satu pun *false positive*. 194 *request* sisanya terklasifikasi sebagai *true negative*, dan *reques*t GET ber-query bersih tidak memicu *rule*.

D. Pembahasan

Hasil pengujian menunjukkan bahwa nilai *Detection Rate* sebesar 100%, *Mean Time to Detect* (MTTD) rata-rata 0,277 detik, dan *False Positive Rate* (FPR) sebesar 3,00% mengindikasikan bahwa integrasi Odoo, Wazuh, dan Grafana mampu mendeteksi serangan pada lapisan aplikasi secara akurat dengan waktu respons yang mendekati *real-time*. Distribusi *alert* pada kategori *Low* (355) dan *High* (1.395) mengonfirmasi efektivitas strategi deteksi bertingkat yang diterapkan, di mana *probe* metakarakter berhasil dikenali sebagai indikator awal adanya aktivitas serangan. Temuan tersebut menunjukkan bahwa *decoder* dan *custom rule* yang dikembangkan mampu mengatasi keterbatasan interpretasi log yang menjadi permasalahan pada tahap awal penelitian.

Dari aspek keandalan operasional, keenam kejadian *false positive* hanya ditemukan pada *rule Brute Force* tingkat rendah (*rule* 100100) yang dipicu oleh kesalahan *login* pengguna, sedangkan seluruh *rule* kritis dengan level sepuluh ke atas tidak menghasilkan *false positive*. Kondisi ini menunjukkan bahwa setiap *alert* pada kategori tingkat tinggi memiliki tingkat kepercayaan yang tinggi sehingga dapat dijadikan dasar yang andal bagi analis untuk melakukan tindak lanjut.

keunggulan MTTD sub-detik bersumber dari arsitektur deteksi yang memproses log pada saat terbentuk, berbeda dengan saat penelusuran log pasif yang baru diselidiki setelah insiden terjadi. Kontribusi utama penelitian ini adalah pengembangan *decoder* PCRE2 beserta tujuh *rules* yang dirancang khusus untuk menginterpretasikan format log Odoo. Dengan pendekatan tersebut, serangan pada lapisan aplikasi ERP yang sebelumnya tidak dapat dikenali kini dapat dideteksi oleh sistem. Implementasi hasil penelitian ini memberikan manfaat bagi PT. XYZ, sebagaimana ditunjukkan melalui hasil pengujian yang memperlihatkan bahwa sistem mampu menyediakan visibilitas terhadap empat jenis serangan yang berpotensi menargetkan formulir *login* dan parameter URL. Selain itu, dashboard Grafana memudahkan proses pemantauan tanpa memerlukan analisis log mentah secara langsung, sehingga proses deteksi tidak lagi bergantung pada kemampuan pengguna dalam membaca log server.

Penelitian ini masih memiliki beberapa keterbatasan yang perlu diperhatikan. Mekanisme deteksi hanya difokuskan pada serangan yang memanfaatkan parameter URL, sehingga injeksi yang dilakukan melalui *body* JSON-RPC maupun lapisan basis data belum termasuk dalam cakupan penelitian. Selain itu, *rule* yang dikembangkan ditujukan untuk mengidentifikasi adanya upaya serangan, bukan untuk memastikan keberhasilan serangan terhadap sistem. Proses evaluasi juga dilakukan pada lingkungan laboratorium dengan hasil simulasi, sehingga belum sepenuhnya mencerminkan karakteristik lalu lintas pada lingkungan produksi. Untuk menjaga transparansi hasil evaluasi, pipeline metrik dirancang agar dapat menampilkan payload yang tidak terdeteksi apabila ditemukan, sehingga nilai *Detection Rate* yang dilaporkan tetap dapat diverifikasi dan ditelusuri kembali.

IV. KESIMPULAN

Penelitian ini memiliki dua tujuan utama, yaitu mengintegrasikan Odoo dengan SIEM Wazuh sehingga log maupun insiden keamanan dapat dipantau secara terpusat, serta menyajikan hasil deteksi melalui *dashboard* Grafana. Kedua tujuan tersebut berhasil diwujudkan dengan mengembangkan *decoder* berbasis PCRE2 dan tujuh *rule* deteksi yang disusun mengacu pada OWASP *Core Rule Set*,

sehingga log mentah yang dihasilkan Odoo dapat diterjemahkan menjadi informasi kejadian keamanan yang lebih bermakna.

Pengujian yang dilakukan pada lingkungan laboratorium menunjukkan bahwa sistem mampu mendeteksi seluruh skenario serangan *Brute Force*, *SQL Injection*, XSS dan *Path Traversal* dengan *Detection Rate* sebesar 100%, MTTD rata-rata 0,277 detik, serta FPR sebesar 3,00%. Selain itu, seluruh *rule* pada kategori kritis tidak menghasilkan *false positive*. Temuan ini mengindikasikan bahwa penerapan *decoder* dan *rule* yang dirancang secara khusus untuk Odoo mampu memberikan kemampuan deteksi yang akurat sekaligus memiliki keandalan dalam operasional. *Decoder, rule*, skrip simulasi, dan dashboard yang dihasilkan juga dapat dimanfaatkan kembali sebagai artefak pendukung oleh organisasi yang menggunakan Odoo.

Penelitian ini masih memiliki beberapa keterbatasan, yaitu cakupan deteksi yang hanya difokuskan pada serangan melalui parameter URL serta proses evaluasi yang dilakukan pada lingkungan laboratorium. Oleh karena itu, penerapannya pada lingkungan produksi masih memerlukan pengujian lebih lanjut. Sebagai pengembangan di masa mendatang, penelitian dapat diarahkan pada perluasan cakupan deteksi hingga mencakup *body* JSON-RPC dan log basis data, penambahan variasi jenis serangan yang dapat dikenali, serta penerapan mekanisme *active response* sehingga sistem tidak hanya mampu mendeteksi, tetapi juga memberikan respons otomatis terhadap serangan yang terjadi.

DAFTAR PUSTAKA

[1] L.-E. Anica-Popa, M. Vrîncianu, I.-B. Pugna and D.-M. Boldeanu, "Addressing Cybersecurity Issues," *sciendo,* p. 108, 2024.

[2] CVE, "CVE Details," [Online]. Available: https://www.cvedetails.com/vendor/16543/odoo.html. [Accessed 16 02 2026].

[3] OWASP, "OWASP Top 10:2021," 2021. [Online]. Available: https://owasp.org/Top10/2021/id/.

[4] NIST, "CVE-2024-12368 Detail," [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2024-12368. [Accessed 02 16 2026].

[5] Odoo, "Security in Odoo," [Online]. Available: https://www.odoo.com/documentation/17.0/developer/reference/backend/security.html. [Accessed 16 02 2026].

[6] NIST, "CVE-2023-48050," [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2023-48050. [Accessed 16 02 2026].

[7] W. P. Putra, M. A. al hilmi, M. A. Al Hilmi and A. Sumarudin, "Implementasi Sistem Manajemen Log untuk Penanggulangan Serangan Server dengan SIEM," *IKRAITH-INFORMATIKA,* vol. 3, 2024.

[8] A. Aldabjan, S. Furnell, X. Carpent and M. Papadaki, "Cybersecurity Incident Response Readiness in Organisations," *International Conference on Information Systems Security and Privacy,* pp. 262-269, 2024.

[9] J. Manzoor, A. Waleed, A. F. Jamali and A. Masood, "Cybersecurity on a budget: Evaluating security and performance of open-source SIEM solutions for SMEs," *PLOS ONE,* 2024.

[10] W. S. Ahmed and Z. T. Mustafa, "Analysis of Wazuh SIEM's Effectiveness in Cloud Security," *American Scientific Publishing Group,* vol. 15, pp. 244-252, 2025.

[11] Wazuh, "Wazuh Documentation," [Online]. Available: https://documentation.wazuh.com/current/getting-started/components/wazuh-dashboard.html. [Accessed 16 02 2026].

[12] Grafana, "Data Sources," [Online]. Available: https://grafana.com/docs/grafana/latest/datasources/. [Accessed 16 02 2026].

[13] A. Sutanto and A. Rakhman, "Experimental Evaluation of Wazuh-Grafana Integration for Real-Time Cyber Threat Detection in Resource-Constrained Environments," *Journal of Applied Informatics and Computing (JAIC),* vol. 9, 2025.

[14] K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems,* vol. 24, pp. 45-77, 2007.

[15] J. A. Ambarwati and C. Darujati, "Penilaian Risiko Data Sistem Informasi Manajemen," *SISTEMASI: Jurnal Sistem Informasi,* vol. 10, pp. 13-25, 2021.

[16] Verizon, ”2025 Data Breach Investigation Report,” 2025.

[17] NIST National vulnerability Database, "CVE-2023-48050," 2023. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2023-48050. [Accessed 16 02 2026].

[18] NIST National Vulnerability Database, ”CVE-2021-45071,” 2021. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2021-45071. [Använd 16 02 2026].

[19] NIST National Vulnerability Database, "CVE-2017-9416," 2017. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2017-9416. [Accessed 16 02 2026].

[20] OWASP, "OWASP CRS," [Online]. Available: https://owasp.org/www-project-modsecurity-core-rule-set/.

e-ISSN: 2548-6861