

IMPLEMENTASI MONITORING DAN PENCEGAHAN SERANGAN DEFACEMENT JUDI ONLINE MENGGUNAKAN WAZUH

Fahrizal Arman, Azmuri Wahyu Azinar*, Arif Senja Fitrani, Ade Eviyanti

Teknik Informatika, Universitas Muhammadiyah Sidoarjo

Jl. Mojopahit No.666 B Celep Sidoarjo, Indonesia

azmuri@umsida.ac.id

ABSTRAK

Akselerasi transformasi digital yang masif di berbagai sektor kehidupan saat ini secara signifikan berbanding lurus dengan meningkatnya kerentanan sistem serta frekuensi serangan siber. salah satunya adalah serangan *defacement* yang menyisipkan konten judi online pada situs web, khususnya situs pemerintahan. Berdasarkan laporan BSSN tahun 2024, kasus *defacement* dengan muatan perjudian online masih banyak ditemukan dan berdampak negatif terhadap reputasi serta kepercayaan publik. Penelitian ini bertujuan untuk mengimplementasikan *Wazuh* sebagai sistem monitoring dan pencegahan serangan *defacement* judi online. Metode penelitian yang digunakan adalah *Action Research* dengan pendekatan *SANS 504-B Incident Response Cycle*. Implementasi sistem memanfaatkan fitur *File Integrity Monitoring*, *Active Response*, integrasi *VirusTotal* untuk deteksi *malware*. Hasil pengujian menunjukkan bahwa *Wazuh* berhasil mendeteksi dan mencegah serangan *brute force SSH*, *upload webshell*, *connection reverse shell*, serta perubahan file yang mengandung konten judi online dengan tingkat keberhasilan 100%. Integrasi *VirusTotal* menghasilkan tingkat akurasi deteksi *malware* sebesar 80%. Meskipun ditemukan kendala berupa keterlambatan visualisasi *alert* pada dashboard dan keterbatasan pemrosesan data berukuran besar, sistem tetap efektif dalam menjaga integritas dan keamanan website. Penelitian ini membuktikan bahwa *Wazuh* dengan kustomisasi *rules* dan integrasi layanan pendukung dapat menjadi solusi yang efektif untuk monitoring dan pencegahan serangan *defacement* judi online.

Kata kunci: *Defacement, Judi Online, Keamanan Siber, SIEM, Wazuh*

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong percepatan proses digitalisasi di berbagai bidang, seperti pemerintahan, layanan publik, dan organisasi swasta. Perubahan ini memberikan kemudahan dalam penyelenggaraan layanan dan pengelolaan data, sekaligus membuka peluang baru yang perlu dijaga keamanannya[1]. Dengan semakin diterapkannya teknologi digital, aktivitas kejahatan siber juga berkembang dalam jumlah tingkat kompleksitas, serta metode penyerangannya[2]. Semakin luasnya ruang digital membuat sistem informasi rentan terhadap berbagai ancaman, terutama bila tidak didukung oleh mekanisme keamanan yang memadai.

Berdasarkan laporan Lanskap Keamanan Siber Indonesia 2024 yang diterbitkan oleh BSSN, pada tahun 2024 terdapat 330.527.636 trafik anomali yang terdeteksi di Indonesia. Dari jumlah tersebut, terdapat 2.487.041 aktivitas *Advanced Persistent Threat* (APT), 514.508 serangan *ransomware*, serta 26.771.610 kasus aktivitas *phishing*. Selain itu, terdapat 5.780 kasus *defacement* yang teridentifikasi, dari jumlah tersebut sebanyak 4.071 kasus terkait dengan penyebaran konten judi online yang menargetkan situs pemerintahan[3].

Defacement adalah tindakan yang dilakukan oleh pelaku peretasan dengan tujuan mengubah atau merusak tampilan serta isi sebuah website. Tindakan ini dapat merugikan reputasi organisasi atau individu tertentu, karena sering digunakan untuk menyampaikan pesan atau tujuan tertentu[4].

Dalam upaya menjaga keamanan data dan infrastruktur digital, Dinas Komunikasi dan Informatika Kabupaten Sidoarjo telah menerapkan mekanisme pengamanan seperti *firewall* dan *Web Application Firewall (WAF)*. Namun, dalam implementasinya masih ditemukan serangan *defacement* dengan modus penyisipan konten judi online pada aplikasi dan website yang dikelola, yang menunjukkan bahwa mekanisme pengamanan tersebut belum sepenuhnya efektif dalam mendeteksi maupun mencegah perubahan konten secara tidak sah. Selain itu, proses analisis insiden keamanan masih dilakukan secara manual melalui penelusuran *access log* dan *log* sistem, sehingga memerlukan waktu yang relatif lama dan sangat bergantung pada ketelitian dan pengalaman petugas. Kondisi ini berpotensi menyebabkan keterlambatan dalam deteksi dan penanganan insiden, serta menurunkan efektivitas pengelolaan keamanan sistem secara keseluruhan.

Wazuh salah satu perangkat lunak *open source* yang berfungsi sebagai sistem *Host-based Intrusion Detection System (HIDS)* sekaligus *Security Information and Event Management (SIEM)*[5]. *Wazuh* memiliki kemampuan pemantauan dan analisis *log* secara *real-time* untuk mendeteksi berbagai ancaman keamanan siber[6]. Berdasarkan beberapa penelitian yang telah dipublikasikan, *Wazuh* memiliki kemampuan untuk mendeteksi beragam serangan, seperti *Distributed Denial of Service (DDoS)*, pemindaian sistem, pengunggahan *malware*, serta upaya eksploitasi kerentanan[7]. Salah satu fitur utama *Wazuh* adalah *File Integrity Monitoring (FIM)*, yang

memungkinkan pemantauan perubahan file secara *real-time*, termasuk modifikasi, penghapusan, dan penambahan file yang tidak sah[8]. Selain itu, Wazuh juga dilengkapi dengan fitur *Active Response* yang memungkinkan sistem melakukan tindakan otomatis, seperti pemblokiran akses ilegal dan penghapusan file berbahaya[9].

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem *Security Information and Event Management (SIEM)* menggunakan Wazuh dalam mendeteksi dan menangani serangan *defacement* yang berkaitan dengan penyisipan konten judi online. Penelitian ini memanfaatkan fitur *File Integrity Monitoring* yang diintegrasikan dengan *API VirusTotal* untuk mengidentifikasi penyisipan file dan perubahan file berbahaya secara *real-time*, serta melakukan kustomisasi *rule* deteksi guna meningkatkan presisi identifikasi serangan dan menurunkan tingkat *false positive*[9]. Selain itu, penelitian ini mengimplementasikan mekanisme *active response* dan notifikasi menggunakan *SMTP Server Relay* melalui layanan *Google Mail* sebagai sarana penyampaian peringatan insiden secara otomatis dan *real-time*, serta mengembangkan *dashboard* pemantauan yang terfilter untuk meningkatkan efektivitas monitoring dan pengambilan keputusan.

2. TINJAUAN PUSTAKA

2.1. Web Defacement

Web defacement adalah tindakan ilegal yang melibatkan perubahan tampilan atau konten situs web dengan tujuan merusak reputasi, menyampaikan pesan tertentu, atau menunjukkan kelemahan pada sistem keamanan target[10]. Dalam serangan ini, pelaku biasanya memanfaatkan celah keamanan pada aplikasi web atau server, seperti kerentanan *file upload*, serangan *SQL Injection*, *Cross-Site Scripting (XSS)*, atau kesalahan dalam pengaturan sistem[4]. Setelah berhasil memanfaatkan celah tersebut, pelaku mengunggah file untuk memperoleh akses ke sistem dan mengedit file halaman web, seperti *index.html* atau *index.php*, sehingga tampilan situs berubah menjadi pesan yang disusun oleh pelaku, yang sering kali berupa slogan, simbol, atau peringatan tertentu[11].

2.2. Wazuh SIEM

Wazuh adalah platform keamanan *open-source* yang berperan sebagai sistem *Security Information and Event Management (SIEM)* serta *Host-based Intrusion Detection System (HIDS)*[6]. Wazuh dirancang untuk melakukan pemantauan keamanan, mendeteksi ancaman, menganalisis *log*, dan merespons insiden secara terpusat. Platform ini kompatibel dengan berbagai sistem operasi seperti *Linux*, *Windows*, dan *macOS*, serta dapat diintegrasikan dengan berbagai layanan dan alat keamanan lainnya seperti *VirusTotal*, *Suricata*, *OSQuery*, dan *Elasticsearch*[12].

2.3. VirusTotal

VirusTotal adalah layanan analisis keamanan yang menyediakan *Application Programming Interface (API)* untuk mendeteksi file berbahaya dengan memanfaatkan berbagai mesin antivirus[12]. Dengan terintegrasi ke dalam Wazuh SIEM, *hash file* yang dihasilkan dari *File Integrity Monitoring (FIM)* dapat dianalisis oleh *VirusTotal*. Hasil analisis tersebut digunakan sebagai dasar untuk mengkorelasikan peristiwa dan menghasilkan *alert*, sehingga meningkatkan tingkat akurasi dalam mendeteksi *malware*, termasuk file yang berpotensi digunakan sebagai *backdoor*.

2.4. Penelitian Terdahulu

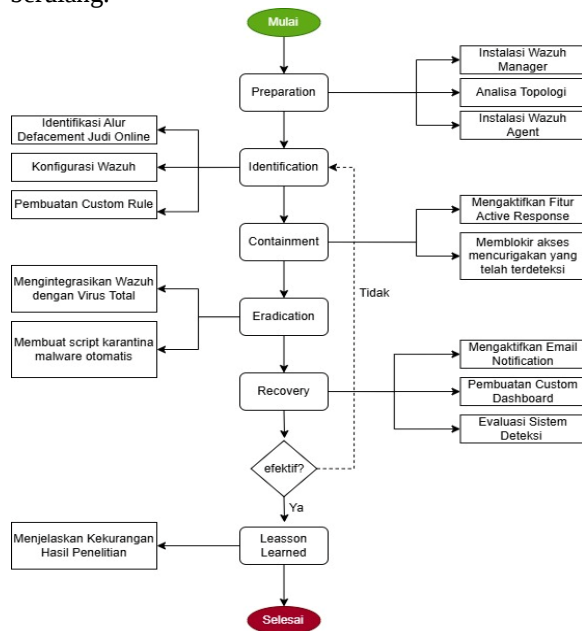
Penelitian sebelumnya dilakukan oleh M. Rizky Reza Pahlevi, Chaerul Umam, dan L. Budi Handoko pada tahun 2025 dengan judul "Deteksi dan Pencegahan Web Defacing Judi Online dengan Wazuh SIEM dan Snort IDS Berbasis Signature". Hasil penelitian menunjukkan bahwa Wazuh berhasil mendeteksi 100% perubahan file dalam direktori web, sedangkan Snort mampu mengidentifikasi sebanyak 76% dari serangan berbasis unggahan *backdoor*[13]. Namun, dalam penelitian tersebut, file *backdoor* yang terdeteksi hanya terbatas pada file dengan ekstensi *.php*. Selain itu, tidak ada tindakan otomatis untuk mengkarantina atau menghapus file *backdoor*, sehingga proses *defacement* masih bisa dilakukan oleh pelaku kejahatan.

Penelitian lainnya dilakukan oleh Rizki Nurul Fahmi, Rudi Harton, dan Dede Syahrul Anwar pada tahun 2025 dengan judul "Integrasi Wazuh SIEM Dengan Modsecurity Dan Virus Total Menggunakan Nist Framework Untuk Mendeteksi Serangan Website". Sistem dalam penelitian ini dibangun dengan mengintegrasikan Wazuh sebagai SIEM, ModSecurity sebagai WAF, dan VirusTotal sebagai alat analisis *malware* berbasis API. Deteksi dilakukan melalui pemasangan Wazuh Agent, integrasi *log ModSecurity*, penambahan *rules VirusTotal*, serta *script* otomatis untuk penghapusan file berbahaya. Pengujian dilakukan melalui *penetration testing* pada website lokal. Hasilnya, sistem berhasil mendeteksi dan memblokir 10 dari 17 jenis serangan, seperti *File Inclusion*, *SQL Injection*, *Command Injection*, *XSS*, *Brute Force*, *Open Redirect*, dan *File Upload*[12]. Keterbatasan dari penelitian ini adalah integrasi Wazuh dengan VirusTotal menggunakan *default rule* yang dapat membatasi kemampuan API VirusTotal. Selain itu, jika *malware* tidak dapat terdeteksi oleh VirusTotal, maka file tersebut tidak akan dihapus dan tetap berada di sistem.

3. METODE PENELITIAN

Metodologi yang di gunakan pada penelitian ini adalah *Action Research* dengan pendekatan *SANS 504-B Incident Response Cycle* yang meliputi empat tahapan utama yaitu Perencanaan, Tindakan, Pengamatan, dan Refleksi[14]. Pendekatan ini dipilih

karena bertujuan untuk menerapkan solusi langsung dalam lingkungan nyata, mengevaluasi dampaknya, dan menyempurnakan implementasi melalui siklus berulang.



Gambar 1. Metode Penelitian

3.1. Preparation

Langkah pertama yang dilakukan adalah mengidentifikasi kebutuhan perangkat dan lingkungan pengujian, yang mencakup instalasi *Wazuh Manager* pada server pusat serta *Wazuh Agent* pada web server yang dijadikan objek uji. Selain itu, dilakukan analisis terhadap topologi jaringan *web server* yang digunakan dalam ekosistem penelitian.

3.2. Identification

Tahapan selanjutnya mengidentifikasi alur serangan *defacement* judi online serta mengkonfigurasi *Wazuh Manager* dan *Wazuh Agent* untuk melakukan pemantauan aktifitas log server secara *real-time* dengan melakukan simulasi serangan *defacement* dengan mengunggah file berbahaya dan memodifikasi halaman web secara tidak sah pada server uji. Hasil pemantauan log serangan tersebut dijadikan sebagai acuan untuk membuat *rule* dan mengkonfigurasi pengaturan yang efektif untuk deteksi serangan *defacement*. Tahapan ini dapat memberikan gambaran kemampuan *Wazuh* dalam mendeteksi aktivitas anomali dan pola serangan *defacement* pada server.

3.3. Containment

Pada tahap Pengendalian, dilakukan upaya membatasi dan menghentikan aktivitas berbahaya yang sudah terdeteksi. Pada tahap ini, fitur *Active Response* pada *Wazuh* diaktifkan untuk secara otomatis memblokir akses mencurigakan atau berbahaya yang terdeteksi pada server. Mekanisme ini memungkinkan *Wazuh* untuk memblokir alamat IP penyerang secara *real-time*, sehingga mencegah

penyerang melakukan tahapan lanjutan dalam proses *defacement*.

3.4. Eradication

Selanjutnya dilakukan proses pembersihan terhadap file berbahaya, skrip, maupun *backdoor* yang berhasil disisipkan oleh penyerang ke dalam direktori web server. Kegiatan ini memanfaatkan fitur FIM pada *Wazuh* yang terintegrasi dengan layanan *VirusTotal*, sehingga setiap file yang diunggah ke sistem dapat diperiksa secara otomatis untuk mendeteksi indikasi keberadaan *malware*. Apabila file teridentifikasi mengandung kode berbahaya, *Wazuh* akan menghapus file tersebut secara otomatis serta mencatat aktivitasnya dalam log keamanan. Melalui mekanisme ini, ancaman lanjutan dapat dieliminasi sehingga integritas dan keamanan server tetap terjaga.

3.5. Recovery

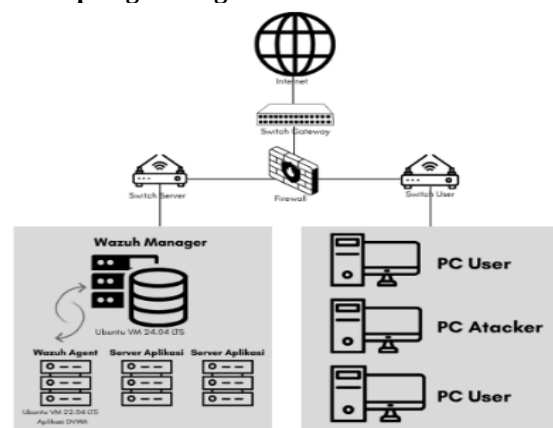
Kegiatan ini dilakukan dengan menganalisa hasil deteksi serta menyusun rekomendasi keamanan berdasarkan temuan hasil pengujian. Evaluasi dilakukan untuk menilai efektivitas sistem *Wazuh* dalam mendeteksi dan mengidentifikasi serangan *defacement* pada web server. Selain itu, dilakukan analisa terhadap serangan yang masih tidak dapat di deteksi serta melakukan perbaikan dan penyempurnaan untuk sistem yang dibangun dengan membuat dashboard pemantauan khusus dan notifikasi *email* ketika serangan *defacement* terdeteksi.

3.6. Lessons Learned

Pada tahap Lessons Learned, dilakukan evaluasi menyeluruh terhadap kapabilitas sistem deteksi *Wazuh* dalam menangani serangan *defacement*. Hasil evaluasi digunakan sebagai landasan untuk menyusun rekomendasi peningkatan sistem, khususnya terkait penyesuaian *rules* deteksi dan strategi monitoring yang lebih responsif. Sebagai luaran akhir, disusun dokumentasi pembelajaran yang dapat dijadikan acuan dalam pengembangan sistem keamanan server pada penelitian selanjutnya

4. HASIL DAN PEMBAHASAN

4.1. Topologi Jaringan Sistem



Gambar 2. Topologi Pengujian

Pada penelitian ini, sistem pengujian dibangun pada lingkungan *intranet* (lokal). Komponen utama pada penelitian ini terdiri dari tiga sistem yaitu *Wazuh Manager*, *Wazuh Agent* dan penyerang seperti yang diilustrasikan pada topologi pengujian Gambar 2.

Berikut spesifikasi sistem yang digunakan pada pengujian di jelaskan pada table 1, 2, dan 3

Tabel 1. Spesifikasi Mesin *Wazuh Manager*

Komponen	Spesifikasi
Prosesor	Intel Xeon 5220 (Virtualisasi)
Memori	8 GB
Storage	150 GB
OS	Ubuntu 24.04.1 LTS
Alamat URL	Wazuh-2025.sidoarjo.kab.go.id

Tabel 2. Spesifikasi Mesin Web Target

Komponen	Spesifikasi
Prosesor	Intel Xeon 6140 (Virtualisasi)
Memori	2 GB
Storage	150 GB
OS	Ubuntu 22.04.5 LTS
Alamat URL	dvwa-2025.sidoarjo.kab.go.id

Tabel 3. Spesifikasi Mesin Penyerang

Komponen	Spesifikasi
Prosesor	Intel Core i5-10400T
Memori	16 GB
Storage	932 GB
OS	Windows 11 Home
Alamat IP	10.99.1.112

Wazuh Manager ditempatkan pada satu lingkup server yang sama namun berada pada VPS terpisah dengan *Wazuh Agent*. proses pengujian serangan pada target (*Wazuh Agent*) akan melalui *firewall* (*Palo Alto*) yang menunjukkan kondisi *real case*, di mana meskipun server telah dilindungi oleh *firewall*, serangan terhadap sistem masih dapat dilakukan akibat adanya celah atau konfigurasi keamanan yang belum optimal.

4.2. Instalasi *Wazuh Manager*

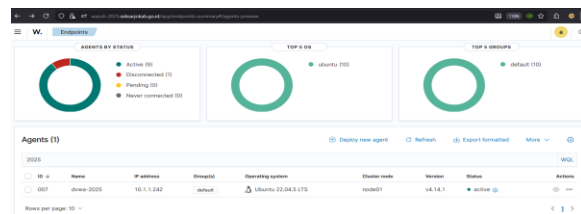
Proses instalasi *Wazuh Manager* dilakukan dengan mengikuti pedoman pada halaman dokumentasi *Wazuh* menggunakan metode *all-in-one installation*, di mana komponen *Wazuh Indexer*, *Wazuh Server*, dan *Wazuh Dashboard* dijalankan pada satu *node* dengan alamat IP yang sama. Metode ini dipilih untuk mempermudah proses konfigurasi dan pengelolaan sistem. Instalasi dilakukan dengan mengunduh *Wazuh installer* versi terbaru (4.14), yaitu berkas *Wazuh-install.sh*. Selanjutnya, proses instalasi dijalankan dengan mengeksekusi perintah sebagai administrator pada server yang telah disediakan menggunakan perintah *Wazuh-install.sh -a*. Setelah proses instalasi seluruh komponen *Wazuh* berhasil, sistem akan menampilkan *username* dan *password* yang digunakan untuk mengakses *Wazuh Dashboard*. Seperti yang ditunjukkan pada gambar 3.

```
08/12/2025 18:01:30 INFO: Initializing Wazuh dashboard web application.
08/12/2025 18:01:32 INFO: Wazuh dashboard web application not yet initialized. Waiting...
08/12/2025 18:01:49 INFO: Wazuh dashboard web application not yet initialized. Waiting...
08/12/2025 18:02:03 INFO: Wazuh dashboard web application initialized.
08/12/2025 18:02:03 INFO: --- Summary ---
08/12/2025 18:02:03 INFO: You can access the web interface https://wazuh-dashboard-ip:443
User: admin
Password: T39.rmGehy32B1F2wQp1Tf433*EDW1
08/12/2025 18:02:03 INFO: Installation finished.
root@wazuh-2025:/home/escanor# ls
```

Gambar 3. proses intalasi *Wazuh Manager*

4.3. Instalasi *Wazuh Agent*

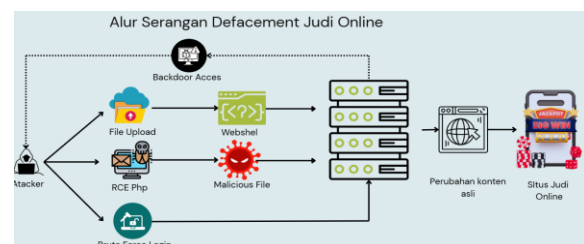
Instalasi *Wazuh Agent* dilakukan melalui *Wazuh Dashboard* dengan mengakses menu *Endpoint*, kemudian memilih opsi *Deploy new agent*. Pada tahap ini, sistem akan menampilkan menu konfigurasi untuk menghubungkan *Wazuh Agent* dengan *Wazuh Manager*. Pada menu konfigurasi tersebut dipilih paket *Linux RPM amd64*, kemudian diisikan alamat server pada kolom *server address* dengan alamat *Wazuh-2025.sidoarjo.kab.go.id*, serta diberikan nama *agent dvwa-2025*. Setelah seluruh parameter konfigurasi diisi, sistem akan menampilkan *command* instalasi yang selanjutnya dijalankan pada *endpoint* yang telah ditentukan untuk menyelesaikan proses instalasi *Wazuh Agent*.

Gambar 4. Tampilan *Wazuh Agent*

Halaman web pada Gambar 4 menampilkan seluruh informasi dan status *endpoint* yang terhubung dengan *Wazuh Manager*.

4.4. Identifikasi Alur Defacement Judi Online

Secara garis besar, proses serangan *defacement* judi online terdiri dari beberapa tahapan yang saling berkaitan, mulai dari upaya akses tidak sah hingga perubahan konten pada sistem target.

Gambar 5. Alur serangan *defacement* judi online

Dari alur pada Gambar 5 terdapat 4 langkah pencegahan utama untuk menangani serangan *defacement* judi online, yaitu:

- Memblokir percobaan *bruteforce login* pada *endpoint*.
- Menyaring dan menghapus file *malware* yang di tanamkan pada *endpoint*.
- Memutus koneksi *reverse shell* dari file yang mencurigakan.

- d. Memeriksa setiap perubahan file yang menggunakan *keyword* judi online.

4.5. Konfigurasi Wazuh

Konfigurasi dilakukan pada *Wazuh Manager* dengan mengaktifkan fitur *FIM* pada direktori */var/www*, yang merupakan *working directory* dari web server. Hasil dari konfigurasi tersebut menunjukkan bahwa setiap penambahan file pada direktori */var/www* dan subdirektornya akan memicu *rule 554*, sedangkan setiap perubahan file akan memicu *rule 550*. Selain itu, *rule 553* akan terpicu ketika terjadi penghapusan file pada direktori tersebut.

4.6. Pembuatan Custom Rule

Untuk mendeteksi percobaan *brute force* melalui layanan *Secure Shell (SSH)* yang umum digunakan untuk akses jarak jauh ke server, dilakukan penambahan *script local_rules.xml* seperti pada Gambar 6 di *Wazuh Manager*. Konfigurasi ini bertujuan untuk mengidentifikasi pola percobaan autentikasi berulang yang mengindikasikan upaya akses tidak sah terhadap server.

```
rule id="100011" level="15" frequency="10" timeframe="5"
<if_matched_sid=5760</if_matched_sid>
<same_source_ip />
<description>Percebahan Brute Force SSH (5x/60s) dari IP $(srcip)</description>
<group>attack, brute_force, ssh_attack</group>
</rule>

rule id="100012" level="15" frequency="10" timeframe="5"
<if_matched_sid=5710</if_matched_sid>
<same_source_ip />
<description>Percebahan Brute Force SSH (5x/60s) dari IP $(srcip)</description>
<group>attack, brute_force, ssh_attack</group>
</rule>
```

Gambar 6. Konfigurasi *Rule* deteksi serangan bruteforce

Konfigurasi tersebut terdiri dari *rule* berikut:

100011	Mendeteksi percobaan login melalui <i>ssh</i> dengan <i>user</i> yang valid, apabila percobaan login gagal 10 kali dalam 5 detik maka <i>rule</i> akan ter-trigger
100012	Mendeteksi percobaan login melalui <i>ssh</i> dengan <i>user</i> yang tidak valid, apabila percobaan login gagal 5 kali dalam 1 menit maka <i>rule</i> akan ter-trigger

Selanjutnya pada Gambar 7 dibuatkan *rule* untuk deteksi *malware* atau *webshell* yang di tanamkan oleh penyerang dengan menambahkan *custom rule* *10 webshell rule.xml*

[illegible]

Gambar 7. Konfigurasi *Rule* deteksi penyisipan *malware*

Konfigurasi tersebut terdiri dari *rule* berikut:

100500	Mendeteksi file <i>upload</i> /baru dengan ekstensi yang telah di tentukan
100501	Mendeteksi perubahan file dengan ekstensi yang telah ditentukan
100502	Mendeteksi perubahan file dengan <i>keyword script</i> yang umumnya digunakan pada file <i>backdoor</i>
700002	Menjalankan <i>Active Response</i> untuk mengkarantina file yang terdeteksi pada <i>rule</i> 100500

Selain itu tambahkan *rule* pada Gambar 8 untuk mengetahui laman yang digunakan penyerang saat mengunggah file *malware* pada sistem

```
<group name="web_post_request" >
  <rule id="106060" level="1">
    <if_sid>31108</if_sid>
    <match>POST</match>
    <description>POST request detected: $(url) from $(arcpip)</description>
  </rule>

  <rule id="106061" level="10" frequency="2" timeframe="1">
    <if_sid>106060</if_sid>
    <if_matched_sid>700892</if_matched_sid>
    <name>agent />
    <description>POST request detected: $(url) from $(arcpip)</description>
  </rule>
</group>
```

Gambar 8. Konfigurasi *rule* deteksi laman yang rentan

Konfigurasi tersebut terdiri dari *rule* berikut:

100600	Mengkoleksi setiap akses <i>post</i> pada sistem tanpa menampilkan <i>alert</i> ketika ter-trigger
100601	Menampilkan <i>alert</i> ketika terdapat akses <i>post</i> dan <i>rule</i> 700002 ter-trigger sehingga laman sistem yang dimanfaatkan untuk mengunggah <i>malware</i> dapat diketahui

Selain metode deteksi berdasarkan file, penguatan sistem keamanan dilakukan dengan menerapkan *rule* untuk mendeteksi aktifitas *webshell* yang menjalankan *reverse connection*. Langkah ini untuk memitigasi risiko *backdoor* yang berpotensi meloloskan diri dari inspeksi berbasis *signature file*. langkah ini *mengintegrasikan package auditd* yang dikonfigurasi secara spesifik untuk memantau setiap eksekusi perintah sistem serta upaya *reverse connection* yang diinisiasi oleh *webshell*. Data log yang dihasilkan oleh *auditd* selanjutnya didefinisikan ke dalam konfigurasi *Wazuh Agent* untuk dilakukan monitoring log yang dihasilkan dengan menggunakan *rule 20_webshell_connection_rule.xml*

```

#group name="Linux Shell Shell"
# This rule detects web shell command execution.
#rule id="80700" level="1"
#cf.id=80700-1 if $id=
#field name="audit.log" webshell_command_exec/$if.id=
#description[Command execution ($audit.exe)]: Possible web shell attack detected/$description=
#mitre=
#id=11585.000/$id=
#id=11590.000/$id=
#mitre=
#rule=
# This rule detects web shell network connections.
#rule id="80701" level="1"
#cf.id=80701-1 if $id=
#field name="audit.log" webshell_net_connect/$if.id=
#description[Network connection via $audit.exe]: Possible web shell attack detected/$description=
#mitre=
#id=11587/$id=
#id=11587.000/$id=
#mitre=
#rule=
#group=
# This rule detects network connections from scripts. --
#group name="Linux Webshell >"
#rule id="80701" level="1"
#cf.id=80701-1 if $id=
#script name="ossec" /ossec.log=
#match-ossid: output: webshell connections/$match=
#description[Network connection]: Script attempting network connection on source port: {local_port} and destination port: {
#(foreign_port)}/$description=
#mitre=
#id=11587/$id=
#id=11587.000/$id=
#mitre=
#rule=

```

Gambar 9. Konfigurasi *rule* deteksi *reverse connection*

Konfigurasi tersebut terdiri dari *rule* berikut:

100520	Mendeteksi eksekusi perintah oleh web shell melalui <i>auditd (webshell_command_exec)</i>
100521	Mendeteksi koneksi jaringan yang dilakukan oleh <i>webshell</i> melalui <i>auditd (webshell_net_connect)</i>
100510	Mendeteksi <i>script</i> berbahaya yang mencoba melakukan <i>reverse connection</i>

Rule pada Gambar 9 tidak dapat memblokir akses koneksi yang dilakukan penyerang karena *rule* tersebut tidak menangkap IP sehingga di perlukan tambahan *rule* seperti Gambar 10

[illegible]

Gambar 10. Konfigurasi *rule* deteksi IP penyerang

Konfigurasi tersebut terdiri dari *rule* berikut:

100602	Mendeteksi akses <i>http</i> ke web dengan ekstensi <i>php, asp, jsp</i> , dll dari <i>access log</i> .
100603	Mengorelasikan akses file web dengan aktivitas <i>webshell</i> dari <i>agent</i> yang sama.

Sebagai langkah preventif terakhir apabila mekanisme deteksi sebelumnya gagal mengantisipasi serangan *webshell*, diterapkan aturan tambahan untuk mendeteksi perubahan konten pada berkas asli web yang di modifikasi dengan *keyword* yang umumnya digunakan pada situs perjudian online. Implementasi ini dilakukan dengan membuat *custom rule* *09_judol_signature.xml* pada Gambar 11 untuk melakukan inspeksi terhadap perubahan konten secara aktual.

```
group name: syscheck, type: "rule"

rule id="SYNOPSIS_Level15"
  if id:559;/if id:559
  if name:"syscheck" content:"%pcr62-(71)SPACETOGEL[togel|toto|bender.|togel|tuto.|togel|gocor|maxwin|zeus|alor|/field|
  description:File [%path] emulating signature job online:description-
  group:syscheck_entry_modified, syscheck_file, job, file_group-
  meta:
    -title:
    -title:559:/id-
    -title:559.003:/id-
  /rule
  /group
```

Gambar 11. Konfigurasi *rule* deteksi defacement judol

Konfigurasi tersebut terdiri dari *rule* berikut:

500550	Mendeteksi perubahan file dengan menggunakan keyword judi online (togel, bandar, slot)
--------	--

Melalui implementasi seluruh custom *rules* yang telah disusun, setiap perilaku aktivitas ancaman dalam upaya melakukan *defacement* judi online dapat dipantau secara *real-time*. Hal ini memungkinkan mekanisme respons insiden dilakukan secara lebih proaktif, sehingga tindakan pencegahan dapat dieksekusi dengan tingkat akurasi yang lebih tinggi guna meminimalkan dampak serangan pada infrastruktur web.

4.7. Containment

Tahap ini dilakukan dengan membatasi dampak serangan agar tidak menyebar lebih luas ke sistem lain dengan memblokir akses dari serangan yang telah di deteksi seperti percobaan *brute force* pada layanan *ssh*, dan memutus koneksi *reverse shell* yang terdeteksi. Sehingga penyerang tidak dapat mengakses sistem dalam rentang waktu yang telah di tentukan.

```
<active-response>
  <command>firewall-drop</command>
  <location>local</location>
  <rules_id>100603,100011,100012</rules_id>
  <timeout>600</timeout>
</active-response>
```

Gambar 12. Konfigurasi *Firewall-drop active response*

Script pada Gambar 12 diimplementasikan pada *Wazuh Manager*. Dengan merestart manager setelah penambahan script dilakukan. dengan ini Ketika *rule* yang telah dideskripsikan ter-trigger, sistem secara otomatis akan memblokir akses penyerang selama 600 detik (10 menit) sehingga tindakan eksploitasi tidak dapat di jalankan.

4.8. Eradication

Ketika *Wazuh* mendeteksi file mencurigakan sebagai *malware* atau indikasi modifikasi konten perjudian, diperlukan mekanisme otomatis untuk mengarantina file tersebut untuk proses analisis lanjutan. Prosedur ini bertujuan untuk menghentikan penyebaran *malware* dan mencegah penyerang untuk aktivitas eksploitasi.

Implementasi aksi respons aktif tersebut dilakukan dengan menambahkan *script Python quarantine-file.py* pada direktori */var/ossec/active-response/bin* di sisi *Wazuh Agent*, serta melakukan sinkronisasi konfigurasi pada file *ossec.conf* di sisi *Wazuh Manager* seperti pada Gambar 13. File yang teridentifikasi berbahaya akan dikarantina ke direktori */var/ossec/quarantine*, yang hanya dapat diakses oleh administrator dan memiliki restriksi akses langsung dari antarmuka web.

```
<ossec_config>
  <command>
    <name>karantina</name>
    <executable>quarantine-file.py</executable>
    <timeout_allowed>no</timeout_allowed>
  </command>

  <active-response>
    <disabled>no</disabled>
    <command>karantina</command>
    <location>local</location>
    <rules_id>700002,500550</rules_id>
  </active-response>
</ossec_config>
```

Gambar 13. Konfigurasi *script* karantina file

Setelah proses karantina berhasil dilakukan, sistem akan menjalankan pemindaian file secara otomatis melalui integrasi dengan *VirusTotal*. Mekanisme ini diimplementasikan dengan menghubungkan *API VirusTotal* ke dalam konfigurasi *Wazuh Manager*, seperti pada Gambar 14.

```
<ossec_config>
<integration>
<name>virustotal</name>
<api_key>tuliskan api key disini</api_key>
<rule_id>700002</rule_id>
<alert_format>json</alert_format>
</integration>
</ossec_config>
```

Gambar 14. Integrasi API VirusTotal dan Wazuh Manager

4.9. Aktivasi Email Notifikasi

Notifikasi email alert pada Wazuh diimplementasikan menggunakan metode SMTP Server Relay melalui layanan Google Mail. Tahap awal integrasi dilakukan dengan instalasi beberapa paket dependensi pada sistem, meliputi *postfix*, *mailutils*, *libsasl2-2*, *ca-certificates*, dan *libsasl2-modules*. Selanjutnya, dilakukan konfigurasi pada berkas */etc/postfix/main.cf* serta autentikasi menggunakan *app password* dari akun *google mail* yang digunakan. Hal ini bertujuan agar Wazuh Manager dapat mengirimkan notifikasi keamanan secara otomatis kepada administrator melalui protokol SMTP yang telah terautentikasi.

Tabel 4. Konfigurasi notifikasi email

Parameter	Isi
Smtplib_server	localhost
Email_from	armanfahrizal98@gmail.com
Email_to	diskominfo@daWazuh@gmail.com
Email_alert_level	15
Rule_id	700002, 500550, 100603, 100011, 100012, 100502

Berdasarkan Tabel 4, untuk meminimalkan email spam akibat *alert false positive*, dilakukan pembatasan pada Wazuh Manager. Pembatasan ini diterapkan melalui dua parameter, yaitu *rule level* dan identifikasi *rule id*. Mekanisme ini memastikan hanya insiden dengan *severity* tinggi yang dikirimkan kepada administrator sebagai notifikasi keamanan.



Wazuh Notification.
2025 Dec 31 00:30:37

Received From: (dwa-2025) any->syscheck
Rule: 500550 fired (level 15) -> "File mengandung signature judi online"
Portion of the log(s):

File "/var/www/html/DVWA/surga11.html" modified
Mode: realtime

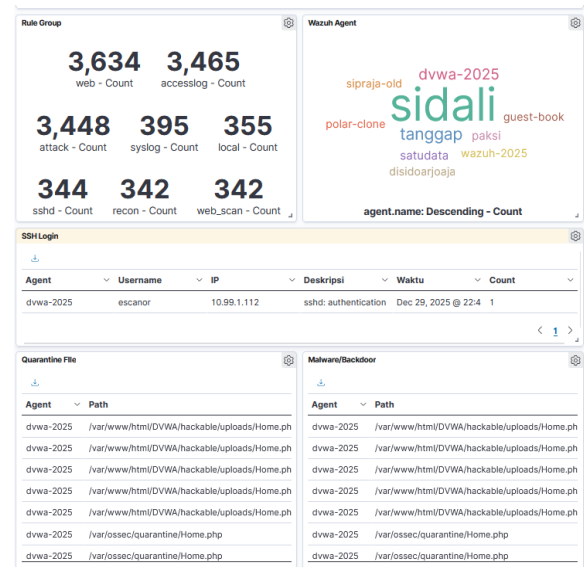
Gambar 15. Alert notifikasi email

Gambar 15. Menunjukkan notifikasi yang dikirimkan wazuh ketika terdapat indikasi percobaan *defacement* judi online pada *agent* yang di monitoring.

4.10. Wazuh Dashboard

Terdapat beberapa *custom dashboard* yang dibuat pada Wazuh diantaranya: menu Statistik Insiden, Lokasi Penyerang, Nama Agent, Rule Group,

SSH Login, Quarantine File, Malware/Backdoor, Blocked Attack, dan Indexer. Seperti yang ditampilkan pada Gambar 16.



Gambar 16. Custom Dashboard Wazuh

Seluruh dashboard tersebut berfungsi untuk mempermudah proses monitoring seluruh *agent* yang terhubung dengan Wazuh Manager sehingga dapat meningkatkan efektivitas pengawasan serta ketepatan analisis dalam membedakan ancaman yang valid dengan pelaporan *false positive*.

4.11. Evaluasi Sistem

Tahap pengujian pertama dilakukan dengan melakukan serangan *bruteforce* pada port SSH menggunakan aplikasi *Hydra*. Pengujian ini memanfaatkan berkas *password-wordlist.txt* yang memuat 2000 kata sandi populer untuk menguji ketahanan sistem autentikasi. Hasil dari pengujian tersebut diuraikan pada Tabel 5.

Tabel 5. Hasil pengujian serangan *brute force*

Aktivitas	Alert rule	Respon
Bruteforce dengan username acak	100012	Akses di blokir otomatis
Bruteforce dengan password acak	100011	Akses di blokir otomatis
Bruteforce dengan username dan password acak	100012	Akses di blokir otomatis

Selanjutnya, pengujian deteksi *webshell* dilakukan dengan dua skenario. Skenario pertama dengan mengunggah *webshell* melalui antarmuka laman web (*file upload*), sementara skenario kedua dilakukan dengan membuat berkas *webshell* secara langsung pada sistem server (*file creation*). Hasil dari kedua pengujian tersebut diuraikan pada Tabel 6.

Tabel 6. Hasil pengujian deteksi *webshell*

	Nama file	Respon	VirusTotal
File upload	Shell.php	Dikarantina	Terdeteksi
	Home.php	Dikarantina	Terdeteksi
	Index.phtml	Dikarantina	Terdeteksi
	Priv.php	Dikarantina	Tidak Terdeteksi
	TinyFileM.phar	Dikarantina	Terdeteksi
File creation	MWS-Shell.php	Dikarantina	Terdeteksi
	BypassServ.asp	Dikarantina	Tidak Terdeteksi
	Yavuzlar.php	Dikarantina	Terdeteksi
	Ani-Shell.php	Dikarantina	Terdeteksi
	DCSC.php5	Dikarantina	Terdeteksi

Tahap pengujian ketiga dilakukan untuk mengevaluasi respons *Wazuh* terhadap koneksi *reverse shell* yang diinisiasi oleh penyerang. Pengujian ini bertujuan untuk memvalidasi efektivitas sistem dalam mendeteksi upaya komunikasi ilegal antara *host* target dan server kendali *aggressor*. Hasil pengujian tersebut diuraikan pada Tabel 7.

Tabel 7. Hasil uji respon *Wazuh* koneksi *reverse shell*

Nama file	Port Akses	Respon
webshell-script.php	4444	Diblokir
reverseshell.php	1234	Diblokir
Naocat.php	2222	Diblokir
Tinyshell.phar	4444	Diblokir
Syscheck.php	9998	Diblokir

Tahap pengujian terakhir dilakukan deteksi modifikasi konten asli situs web di dalam direktori utama `/var/www/html/DVWA`. Pengujian ini mensimulasikan serangan *defacement* di mana konten asli diubah menjadi dengan *signature* judi online. Pengujian ini bertujuan untuk memvalidasi apakah *custom rule* yang dibuat dapat bekerja sesuai harapan. Hasil pengujian tersebut diuraikan pada Tabel 8.

Tabel 8. Hasil uji deteksi serangan *defacement*

File Script	Rule	Total Karakter	Respon
Toto.html	-	11.455	Dikarantina
Index.php	500550	58	Dikarantina
Spacetogel.html	500550	421	Dikarantina
Zeus.php	-	18.423	Dikarantina
Surga11.html	500550	1860	Dikarantina

4.12. Lesson Learned

Hasil pengujian menunjukkan bahwa implementasi *Wazuh* sangat efektif dengan tingkat keberhasilan 100% dalam memitigasi serangan *bruteforce SSH*, mengarantina *webshell*, memutus koneksi *reverse shell*, serta mendeteksi modifikasi file dengan *signatur judi online*. Meskipun integrasi *VirusTotal* memberikan akurasi deteksi *malware* sebesar 80%, sistem terbukti andal dalam menjaga integritas file melalui mekanisme *response active* yang akurat.

Namun, terdapat anomali berupa *propagation delay* pada visualisasi alert di dashboard yang disebabkan oleh beban antrean data. Selain itu, ditemukan limitasi pada komponen *Filebeat* yang gagal memproses dokumen *json* berskala besar pada *rule* 500550, sehingga *alert* tidak muncul ketika perubahan file melebihi ambang batas *buffer* transmisi. Namun proses karantina file dan peringatan melalui email tetap di jalankan pada kasus tersebut sehingga mekanisme pencegahan dan deteksi tetap berjalan.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil membuktikan bahwa integrasi *Wazuh* dengan customisasi *rules* serta integrasi *VirusTotal* efektif dalam mendeteksi dan mencegah serangan *defacement* bermuatan konten perjudian online. Berdasarkan hasil pengujian, sistem terbukti mampu meningkatkan akurasi deteksi dan menjalankan mekanisme pencegahan otomatis secara responsif. Sebagai langkah pengembangan sistem di masa mendatang, konfigurasi *rules* dan *main configuration file (ossec.conf)* dapat diperbarui secara berkala guna mengoptimalkan efektivitas deteksi terhadap celah keamanan yang ada. Selain itu, pemanfaatan *Wazuh* perlu diperluas untuk memitigasi berbagai jenis serangan siber lainnya yang termasuk dalam daftar *OWASP Top 10*, seperti *Broken Access Control*, *SQL Injection*, *Cross-Site Scripting (XSS)*, serta *Security Misconfiguration*.

DAFTAR PUSTAKA

- [1] I. Z. Satrya, "Serangan Siber Dalam Perkembangan Perbankan Digital di Indonesia," vol. 9, no. 10, 2024.
- [2] M. A. Djibu, "Transformasi Digital dan Keamanan Siber: Upaya Penanggulangan Kejahatan di Era Teknologi di Indonesia," *Judge J. Huk.*, vol. 6, no. 1, p. 346, 2025.
- [3] Badan Siber dan Sandi Negara, "Lanskap Keamanan Siber Indonesia 2024," *Id-SIRTII /CC*, no. 70, pp. 1–107, 2024, [Online]. Available: bit.ly/44bzpHM
- [4] J. Desmon, Y. Hidayatulloh, and S. Jumaryadi, "Systematic Literature Review: Serangan Deface," vol. 14, no. 2, pp. 106–112, 2024.
- [5] R. A. P. Azzah Shafiyyah, Gigih Forda, "Implementasi Wazuh Menggunakan Metode Ppdioo Di Sistem Keamanan Jaringan Psdku Universitas Lampung Waykanan Sebagai Deteksi Dan Respon Serangan Siber," vol. 12, no. 2, 2024.
- [6] A. Kamil, M. Tahir, S. Juliah, A. L. Rahmat, and Y. D. Mahendra, "Sistem Keamanan Berbasis Host-Based Intrusion Detection System (Hids) Menggunakan Wazuh," vol. 9, no. 3, pp. 5460–5466, 2025.
- [7] M. D. Pratama, F. Nova, and D. Prayama, "Wazuh sebagai Log Event Management dan Deteksi Celah Keamanan pada Server dari Serangan Dos," vol. 3, no. 1, pp. 1–7, 2022.

- [8] M. R. T. Hidayat, N. Widiyasono, and R. Gunawan, "Optimasi Deteksi Malware Pada Siem Wazuh Melalui Integrasi Cyber Threat Intelligence Dengan Misp Dan Dfir-Iris," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 1, 2025, doi: <http://dx.doi.org/10.23960/jitet.v13i1.5686>.
- [9] B. Haryanto and D. W. Chandra, "Implementasi Wazuh Integritas File untuk Perlindungan Keamanan Berdasarkan Aktivitas Log di BTSI UKSW," *J. Indones. Manaj. Inform. dan Komun.*, vol. 5, no. 1, pp. 183–192, 2024, doi: <https://doi.org/10.35870/jimik.v5i1.447>.
- [10] V. E. Pattiradjawane and D. Upuy, "Deteksi Serangan Web Defacement pada Infrastruktur Kritis Menggunakan Machine Learning," vol. 1, no. 1, pp. 37–42, 2025.
- [11] Badan Siber dan Sandi Negara, "Panduan Penanganan Insiden Web Defacement Judi Online," *Pandu. Penanganan Insid. Web Defacement Judi Online*, p. 26, 2023, [Online]. Available: <https://www.kemhan.go.id/bacadnas/wp-content/uploads/migrasi/admin/CyberDefence.pdf>
- [12] R. N. Fahmi, R. Hartono, and D. S. Anwar, "Integrasi Wazuh Siem Dengan Modsecurity Dan Virus Total Menggunakan Nist Framework Untuk Mendeteksi Serangan Website," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 9, no. 4, pp. 6578–6586, 2025.
- [13] M. R. Reza Pahlevi, C. Umam, and L. B. Handoko, "Deteksi dan Pencegahan Web Defacing Judi Online dengan Wazuh SIEM dan Snort IDS Berbasis Signature," *J. Algoritma.*, vol. 22, no. 1, pp. 197–208, 2025, doi: [10.33364/algoritma/v.22-1.2220](https://doi.org/10.33364/algoritma/v.22-1.2220).
- [14] A. Makmur, I. Jasman, and U. C. Palopo, "Optimalisasi manajemen bandwidth jaringan komputer menggunakan action research pada dinas komunikasi dan informatika kota palopo," vol. 6, 2023.