Jurnal Riset Rumpun Ilmu Teknik
Volume 4, Nomor 1, April 2025
p-ISSN: 2829-0178; e-ISSN: 2829-016X, Hal. 422-447
DOI: https://doi.org/10.55606/jurritek.v4i1.5004
Available Online at: https://prin.or.id/index.php/JURRITEK

Evaluasi Celah Keamanan dengan Metodologi Vulnerability Assessment Sebagai Penilaian Tingkat Kerentanan pada Domain Unud.Ac.Id

Dd Hassel Putra Q [1], Ilham Ammarul Aziz [2], Eginna Gresia Br Purba [3*], Dewa Made Wiharta [4], I Gusti Ayu Garnita Darmaputri [5]
[1-5] Universitas Udayana, Indonesia
Email: eginna.gresia002@student.unud.ac.id *

Abstract. *Website security is a crucial aspect, especially for educational institutions that manage sensitive data. Udayana University has over 500 subdomains, but not all have undergone security evaluation, potentially posing significant risks. This study aims to identify security vulnerabilities, assess risk levels, and provide mitigation recommendations. The subdomain ee.unud.ac.id was selected as a sample because it uses a template similar to many other university websites. The method employed is Vulnerability Assessment using white box testing, with tools such as OWASP ZAP, Nessus, RapidScan, and the Snort Intrusion Detection System (IDS). The analysis is based on the OWASP Top 10 (2021) and the CIA Triad principles. The results revealed 25 types of threats across three risk levels and 24 alerts from Snort, indicating potential internal and external threats. Recommended mitigations include strengthening security configurations, implementing firewalls, and regularly updating systems. This study emphasizes the importance of routine security testing and the use of IDS to safeguard systems against cyberattacks.*

Keywords: *CIA Triad, IDS, OWASP ZAP, OWASP Top 10, Website Security, Snort, Udayana University, Vulnerability Assessment*

Abstrak. Keamanan website merupakan aspek penting, terutama bagi institusi pendidikan yang mengelola data sensitif. Universitas Udayana memiliki lebih dari 500 subdomain, namun belum seluruhnya dievaluasi keamanannya, sehingga berpotensi menimbulkan risiko. Penelitian ini bertujuan mengidentifikasi celah keamanan, menilai tingkat risiko, dan memberikan rekomendasi mitigasi. Subdomain ee.unud.ac.id dipilih sebagai sampel karena menggunakan template serupa dengan banyak situs lain. Metode yang digunakan adalah *Vulnerability Assessment* dengan *white box testing*, menggunakan alat seperti OWASP ZAP, Nessus, RapidScan, dan IDS Snort. Analisis dilakukan berdasarkan OWASP Top 10 2021 dan prinsip CIA Triad. Hasil menunjukkan 25 jenis ancaman dari tiga level risiko serta 24 *alert* dari Snort, menunjukkan potensi ancaman internal dan eksternal. Mitigasi yang disarankan meliputi penguatan konfigurasi keamanan, firewall, dan pembaruan sistem. Studi ini menegaskan pentingnya pengujian keamanan berkala dan penggunaan IDS untuk melindungi sistem dari serangan siber.

Kata Kunci: CIA Triad, IDS, Keamanan Website, OWASP ZAP, OWASP TOP 10, Snort, Universitas Udayana, Vulnerability Assessment

1. PENDAHULUAN

Aspek keamanan *website* adalah hal yang penting dari sistem informasi. Namun, masalah keamanan ini sering kali tidak diperhatikan oleh pemilik atau *administrator* sistem informasi. Persyaratan keamanan sistem aplikasi untuk melindungi data meliputi aspek kerahasiaan (*confidentiality*), aspek integritas (*integrity*) dan aspek ketersediaan (*availability*) dari keamanan sistem aplikasi (Betty Yel, M., & M Nasution, M. K.2022). Untuk meminimalisir permasalahan tersebut, penting bagi *administrator* sistem untuk secara proaktif mengidentifikasi dan mengatasi potensi kerentanan pada *website* mereka.

Received Maret 12, 2025; Revised Maret 28, 2025; Accepted April 15, 2025; Published April 30, 2025

Dengan melakukan analisa terhadap sebuah *website* yang dirancang ataupun akan dirilis menjadikan administrator dapat dengan mudah melakukan sebuah evaluasi untuk mencegah terjadinya kejahatan. Mendeteksi kerentanan sebuah aplikasi berbasis *website* merupakan sebuah kebutuhan dalam menanggapi kemajuan teknologi saat ini, sebab dari berbagai macam kejahatan dalam dunia internet, *website* mudah dijadikan target dari orang-orang yang tidak bertanggung jawab. Termasuk soal pengambilan sebuah data adalah salah satu dari kejahatan yang cukup fatal untuk bisa mengakses sebuah *website*.

Penilaian kerentanan membantu mengidentifikasi kategori dan kekritisan kerentanan dalam suatu organisasi. Organisasi menilai kerentanan dan memprioritaskannya, serta merancang metode untuk memperbaiki situasi tersebut. Metode penilaian membantu mengukur efektivitas solusi tersebut. Tujuan dari penilaian kerentanan mencakup pemindaian, pemeriksaan, evaluasi, dan pelaporan kerentanan dalam jaringan untuk meminimalkan tingkat risiko terhadap suatu organisasi (D. Arnaldy and A. R. Perdana 2019).

Universitas Udayana adalah salah satu organisasi yang telah memanfaatkan sistem informasi berbasis *website* untuk operasional proses bisnisnya, seperti proses akademik dengan SIMAK (Sistem Informasi Akademik), pengelolaan data dosen dan pegawai melalui SIMDOS (Sistem Informasi Dosen) dan SIANITA (Sistem Informasi Tenaga Kependidikan), proses keuangan dengan SIAKU (Sistem Realisasi Keuangan) dan SILUNA (Sistem Perencanaan Keuangan) dan berbagai sistem informasi untuk proses lainnya. Semua sistem informasi ini terintegrasi dengan IMISSU sebagai SSO *(Single Sign On)* sehingga user cukup memiliki satu akun untuk bisa mengakses semua aplikasi sesuai dengan otoritas yang diberikan.

Saat ini, Universitas Udayana memiliki lebih dari 500 *sub-domain* yang terdiri dari sekitar 50 sistem informasi utama yang terintegrasi IMISSU dan sebagian besar sisanya berupa *website* untuk menyebarkan informasi yang dikelola oleh Program Studi, Fakultas, Unit, Organisasi Mahasiswa dan pihak internal udayana lainnya. Semua aplikasi dan *website* yang menggunakan *domain* unud.ac.id berada dibawah pengelolaan Unit Sumber Daya Informasi (USDI).

Unit Sumber Daya Informasi (USDI) adalah unit yang bertanggung jawab terhadap seluruh aset teknologi informasi yang dimiliki oleh Universitas Udayana mulai dari sistem informasi, infrastruktur jaringan internet dan data center serta data organisasi. USDI memiliki satu bidang yang secara khusus bertugas untuk mengembangkan dan mengintegrasikan sistem informasi, termasuk membuat format standar (*template*) untuk memenuhi permintaan *website* dari berbagai pihak internal Universitas Udayana.

Pihak yang memerlukan sebuah *website* untuk menyebarkan informasi, wajib mengirimkan permohonan pembuatan *sub-domain* ke USDI. USDI akan menindaklanjuti permintaan tersebut dengan meng-*hosting subdomain* tersebut ke salah satu *server* yang ada di *data center* Universitas Udayana dengan *template* yang sudah ditentukan. Kemudian, USDI akan memberikan hak akses sebagai pengelola *website* kepada pihak terkait, sehingga dapat mengunggah informasi yang diperlukan.

Dengan demikian, sekitar 400 *website* tersebut di-*hosting* menggunakan *template* yang sama. Hal ini dapat menjadi celah keamanan yang serius karena *template website* tersebut belum pernah diuji kerentanan-nya sehingga *website* dengan *domain* unud.ac.id berpotensi besar untuk di-*hack* oleh sekelompok tertentu untuk mencuri data, menghapus data atau merusak data. Sebagai contoh, pada rentang waktu Juni hingga Juli 2023, terjadi kasus serangan pada *sub-domain* yang menggunakan *template website* USDI, yaitu fp.unud.ac.id dan s2 kimia.unud.ac.id. Dalam serangan ini, *hacker* berhasil menyisipkan halaman promosi judi *online* ke dalam kedua *website* tersebut. Dalam kasus ini, *hacker* menyembunyikan *file* PHP dengan ekstensi foto seperti .jpg, .jpeg, dan lainnya. untuk mengelabui sistem. Setelah *file* foto berhasil diunggah, *hacker* akan mencoba mengakses dan mengeksekusinya untuk membuat celah yang mudah dimasuki atau disebut sebagai *backdoor*. Meskipun tampaknya sebagai *file* foto, sebenarnya *file* tersebut adalah *file* PHP yang dapat dieksekusi menggunakan perintah PHP. Melalui *backdoor* inilah, *hacker* dapat mengunggah *file* lain yang tidak semestinya seperti halaman promosi judi online. Kasus ini menunjukkan bahwa jika ditemukan satu celah pada *website* yang menggunakan format standar USDI tersebut, maka *website* lain juga pasti akan memiliki celah yang sama, maka perlu untuk dilakukan pengujian.

Oleh karena itu, penting untuk melakukan pengujian keamanan yang mendalam. Pengujian tersebut mengacu pada tingkatan keamanan *website* yang hasilnya bukan menggaransikan sistem bebas dari resiko serangan, tetapi dapat meminimalisir serangan yang dapat disalahgunakan, karena untuk menjelajahi semua aspek diperlukan pengujian tingkat lanjut, yang dapat dijadikan informasi dalam upaya meminimalisir kerentanan yang terdapat pada *website* dengan domain unud.ac.id dan dapat mengambil tindakan untuk meningkatkan sistem keamanan *website*.

Penelitian ini berfokus pada penerapan standar keamanan OWASP untuk mengevaluasi kerentanan sistem. OWASP merupakan sebuah organisasi yang berfokus untuk meningkatkan keamanan perangkat lunak dan aplikasi web yang dipilih sebagai metode utama dalam penelitian ini. Melalui metode dan alat OWASP, celah keamanan dapat diidentifikasi dan tingkat keamanan aplikasi web dapat ditingkatkan. Proses paenelitian ini berfokus pada analisis

aktif terhadap aplikasi web untuk menemukan kerentanan dan kelemahan sistem. Hasil pengujian akan mengacu pada OWASP Top 10 2021 sebagai standar untuk mengidentifikasi kerentanan. Penelitian ini akan menghasilkan laporan analisis yang akan digunakan untuk mengevaluasi risiko pada sistem informasi akademik.

Mengamankan sistem informasi Universitas Udayana, termasuk *website subdomain* unud.ac.id, memerlukan pendekatan berlapis. Beberapa metode diterapkan secara sistematis, yakni analisis kerentanan, pengujian penetrasi dan audit keamanan. Analisis kerentanan adalah metode untuk mengidentifikasi kelemahan dalam sistem menggunakan alat pemindai otomatis dan tinjauan manual. Pengujian penetrasi merupakan metode untuk mengetahui seberapa rentan suatu jaringan *website* terhadap serangan dari luar dimana seorang penguji menyimulasikan dirinya seperti pihak luar yang berusaha masuk kedalam jaringan. Audit keamanan bertujuan untuk memastikan kebijakan dan prosedur sesuai dengan standar industri, pemantauan berkelanjutan memonitor aktivitas jaringan dan sistem secara *real-time* (F. Tinambunan, A. Junaidi, AM Rizki 2024).

Selain melakukan evaluasi kerentanan dengan OWASP, pada *capstone* ini juga menambahkan *Intrusion Detection System* (IDS) untuk mendeteksi jika terjadi aktivitas mencurigakan yang berpotensi sebagai serangan. IDS mengumpulkan *log* untuk mengidentifikasi akses ilegal sesuai dengan aturan yang diterapkan pada IDS. IDS juga dapat mengirimkan peringatan dan mengotomatisasi respons terhadap ancaman. *Capstone project* ini menggunakan Snort sebagai IDS karena dapat mendeteksi serangan berbasis *signature rule* sehingga Universitas Udayana dapat mendeteksi dan merespons ancaman keamanan secara proaktif. Dengan langkah-langkah ini, Universitas Udayana dapat meningkatkan keamanan sibernya, mengurangi risiko serangan, dan melindungi data sensitif, memastikan kelancaran operasional dan menjaga kepercayaan civitas akademika.

2. METODE PENELITIAN

Capstone project ini dilaksanakan di Unit Sumber Daya Informasi (USDI) Universitas Udayana selama Maret hingga Desember 2024. Proyek ini berfokus pada pengujian keamanan situs web ee.unud.ac.id secara real-time menggunakan sistem operasi Kali Linux dan alat otomatisasi OWASP ZAP serta IDS Snort. Pengumpulan data dilakukan melalui observasi terhadap jaringan dan dokumentasi teknis dari berbagai literatur. Proses pengujian dilakukan melalui beberapa tahapan, mulai dari studi literatur, instalasi perangkat lunak, simulasi serangan, hingga analisis hasil output dari sistem deteksi intrusi. Seluruh aktivitas dilakukan

dengan konfigurasi jaringan yang mengintegrasikan akses lokal dan jarak jauh melalui metode remote.

Pengujian keamanan mengikuti metode OWASP dengan pendekatan kualitatif untuk mengidentifikasi celah keamanan berdasarkan standar OWASP Top 10 dan CIA Triad. Infrastruktur pengujian mencakup topologi jaringan yang memisahkan IP publik dan lokal, serta menggunakan server virtual untuk menjalankan alat pengujian. Sistem deteksi Snort digunakan untuk memantau dan mencatat aktivitas mencurigakan selama simulasi serangan. Hasilnya dianalisis untuk mengevaluasi efektivitas pertahanan sistem dan menghasilkan rekomendasi mitigasi yang akan dilaporkan kepada pihak pengelola sistem di USDI.

3. HASIL DAN PEMBAHASAN

Instalasi dan Konfigurasi Snort sebagai *Intrusion Detection System* (IDS) Instalasi dan konfigurasi Snort sebagai Intrusion Detection System (IDS) pada web server *ee.unud.ac.id* dilakukan guna meningkatkan keamanan jaringan dengan mendeteksi potensi ancaman secara real-time. Langkah awal dimulai dengan memperbarui sistem operasi Ubuntu untuk memastikan bahwa semua dependensi berada dalam versi terbaru. Setelah itu, dilakukan pemeriksaan terhadap antarmuka jaringan web server, yaitu *ens160* (IP lokal) dan *ens192* (IP publik). Snort difokuskan untuk memantau lalu lintas pada *ens192*, karena merupakan jalur akses eksternal yang berisiko terhadap serangan dari luar.

Proses instalasi Snort dilakukan setelah sistem diperbarui, dan dilanjutkan dengan konfigurasi awal seperti penentuan *HOME_NET* dan antarmuka jaringan utama. Dalam penelitian ini, Snort dikonfigurasi sebagai *Host-based Intrusion Detection System* (HIDS), di mana pemantauan hanya dilakukan terhadap lalu lintas masuk dan keluar dari server itu sendiri, dengan alamat IP publik 103.29.196.121. Konfigurasi address range dan interface disesuaikan agar IDS dapat beroperasi secara optimal sesuai tujuan monitoring pada lapisan host.

Pengaturan file konfigurasi utama *snort.conf* mencakup penyesuaian variabel ipvar HOME_NET dan RULE_PATH. Nilai HOME_NET disesuaikan dengan IP publik server agar fokus pemantauan tidak meluas ke jaringan lain yang tidak relevan. Selain itu, output log Snort dikonfigurasi agar dapat disimpan dalam format CSV dan PCAP guna memudahkan analisis lebih lanjut. Setelah konfigurasi disimpan, pengujian dilakukan dengan perintah snort -T -c /etc/snort/snort.conf untuk memastikan tidak ada kesalahan sintaks dan sistem dapat berjalan dengan baik.

Jika proses pengujian berhasil, sistem akan menampilkan pesan "Snort successfully validated the configuration!" sebagai tanda bahwa IDS siap digunakan. Keberhasilan ini

menunjukkan bahwa seluruh parameter penting, seperti lokasi file rule, rentang IP yang dipantau, dan plugin log, telah dikonfigurasi dengan benar. Validasi konfigurasi merupakan tahap krusial sebelum IDS aktif digunakan dalam lingkungan produksi.

Terakhir, dilakukan konfigurasi lanjutan terhadap *Snort rules* agar sistem dapat mengenali jenis-jenis lalu lintas mencurigakan berdasarkan layanan tertentu. Misalnya, variabel ipvar FTP_SERVERS diatur untuk memantau protokol FTP, dan portvar HTTP_PORTS disesuaikan dengan daftar port umum dan non-standar yang digunakan oleh layanan HTTP. Dengan demikian, Snort dapat memberikan cakupan deteksi yang lebih luas dan responsif terhadap potensi ancaman terhadap layanan web server yang bersangkutan.

Proses dan Tahapan *Penetration Testing* (Pentest)

Penelitian ini menggunakan metode *white box testing*, di mana peneliti memiliki akses penuh terhadap sistem atas izin dari pengelola USDI. Akses sebagai admin diberikan pada website *ee.unud.ac.id*, memungkinkan evaluasi fitur internal dan potensi celah keamanan dari perspektif pengguna internal. Selain itu, peneliti juga memperoleh akses ke server melalui SSH (PuTTY) ke IP lokal, yang memungkinkan pemantauan layanan, port terbuka, serta kondisi sistem secara menyeluruh sebagai dasar untuk *vulnerability assessment* dan *penetration testing*.

Tahapan *information gathering* diawali dengan *passive reconnaissance* menggunakan tools seperti WhatWeb, Whois, dan Wappalyzer. Hasilnya menunjukkan bahwa server menggunakan IP 103.29.196.121, sistem operasi Ubuntu 18.04.4 LTS, dan web server Nginx 1.14.0. Backend-nya berjalan pada PHP 7.2.24 dan MySQL 5.7.29 dengan framework Laravel dan Handlebars.js. Teknologi frontend yang digunakan meliputi HTML5, Bootstrap 3.3.7, serta berbagai pustaka JavaScript seperti jQuery dan Moment.js. Selain itu, mekanisme keamanan seperti *XSRF-TOKEN* dan *laravel_session* telah diterapkan untuk mencegah serangan CSRF.

Selanjutnya, dilakukan *network mapping* menggunakan Nmap dan Amass untuk mengidentifikasi layanan, port terbuka, serta potensi celah. Beberapa port penting yang ditemukan dalam keadaan terbuka adalah 80/tcp (HTTP), 443/tcp (HTTPS), 465/tcp (SMTPS), 3389/udp (Remote Desktop), dan 161/udp (SNMP). Hasil *subdomain enumeration* juga menunjukkan bahwa domain menggunakan layanan email dari Google. Analisis ini memberikan gambaran menyeluruh terhadap arsitektur dan keamanan jaringan target sebelum dilakukan pengujian lebih lanjut.

Tahapan *Scanning* dan *Penetration Testing*

Dalam upaya mengidentifikasi potensi kerentanan keamanan pada sistem web, dilakukan serangkaian proses pengujian menggunakan berbagai tools pemindaian otomatis yang dirancang untuk mendeteksi celah dan kelemahan umum. Tujuan dari proses ini tidak hanya untuk mengidentifikasi celah, tetapi juga untuk membuktikan keberadaan celah tersebut melalui dokumentasi bukti hasil uji, serta memahami sejauh mana sistem dapat bertahan terhadap serangan yang berpotensi dieksploitasi oleh penyerang. Tools yang digunakan dalam proses ini meliputi Nikto, OWASP ZAP, Nessus, Rapidscan, dan SQLMap. Hasil dari setiap pengujian dianalisis untuk memberikan rekomendasi perbaikan yang bertujuan meningkatkan keamanan sistem web secara menyeluruh.

Pengujian awal dilakukan menggunakan Nikto v2.5.0 pada domain *ee.unud.ac.id* dengan IP 103.29.196.121, yang mengungkap sejumlah celah serius. Di antaranya adalah ketiadaan header keamanan penting seperti X-Frame-Options, HSTS, dan X-Content-Type-Options, serta penggunaan versi nginx yang usang. Selain itu, beberapa cookie penting tidak dilindungi dengan atribut Secure dan HttpOnly, yang meningkatkan risiko terhadap serangan XSS dan pencurian data. Teridentifikasi pula direktori sensitif seperti /staff/ dan file web.config yang dapat diakses publik, serta penggunaan wildcard certificate dan encoding deflate yang dapat dimanfaatkan dalam serangan lanjutan seperti *BREACH*. Temuan ini menegaskan pentingnya peningkatan konfigurasi keamanan untuk meminimalkan risiko eksploitasi.

Selanjutnya, pengujian dilakukan menggunakan OWASP ZAP melalui metode manual explore, AJAX spidering, active scan, dan attack mode. Proses dimulai dengan eksplorasi manual melalui browser yang dikonfigurasi sebagai proxy untuk memantau lalu lintas HTTP, dilanjutkan dengan AJAX Spider untuk menjelajahi konten dinamis berbasis JavaScript. Active Scan kemudian mengirimkan payload pengujian keamanan ke berbagai parameter yang terdeteksi, dan mode Attack diaktifkan untuk melakukan analisis secara real-time saat eksplorasi berlangsung. Hasil dari pengujian ini mengungkapkan adanya 19 celah keamanan dengan klasifikasi tingkat risiko yang beragam, yakni 3 risiko tinggi (termasuk SQL Injection dan penggunaan library JavaScript yang rentan), 5 risiko sedang (seperti buffer overflow dan celah pada Content Security Policy), serta 11 lainnya berupa risiko rendah dan informasi tambahan, seperti kebocoran versi server dan cookie tanpa atribut keamanan.

Beberapa temuan penting meliputi kerentanan pada library JavaScript dan buffer overflow, di mana penggunaan library usang seperti *jQuery Validation v1.13.1* dan *Bootstrap v3.3.4* berpotensi membuka celah XSS serta manipulasi tampilan antarmuka pengguna. Celah buffer overflow yang ditemukan juga memiliki risiko sedang karena dapat menyebabkan crash

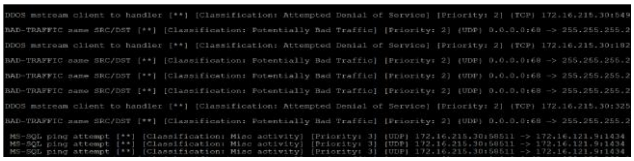
sistem hingga eksekusi kode ilegal. Selain itu, kelemahan konfigurasi keamanan web teridentifikasi dari tidak adanya Content Security Policy (CSP), yang memungkinkan serangan XSS dan injeksi skrip. Tidak diterapkannya anti-clickjacking header juga memungkinkan situs disisipkan dalam iframe berbahaya, yang berhasil dibuktikan melalui eksploitasi visual.

Masalah lainnya termasuk mixed content, di mana halaman HTTPS memuat skrip dari HTTP seperti html5.js, yang dapat membuka celah *man-in-the-middle*. Temuan dari alat **Nessus** menguatkan hasil sebelumnya, dengan mengidentifikasi tiga kerentanan utama: akses tidak sah ke file web.config, potensi clickjacking, dan tidak diterapkannya kebijakan HTTP Strict Transport Security (HSTS). Ketiganya tergolong risiko sedang, namun tetap signifikan dalam meningkatkan eksposur sistem terhadap kebocoran data, penurunan kepercayaan pengguna, serta kemungkinan serangan lanjutan. Keseluruhan hasil ini menjadi dasar penting bagi tindakan mitigasi dan peningkatan keamanan sistem secara menyeluruh melalui uji penetrasi yang lebih dalam dan penyesuaian konfigurasi keamanan.

Hasil Deteksi Snort Terhadap Serangan

Snort beroperasi sebagai *Host-based Intrusion Detection System* (HIDS) yang berfungsi untuk mendeteksi serangan yang menargetkan sistem secara *real-time*. Sistem ini memantau lalu lintas jaringan yang masuk dan keluar dari perangkat tempat Snort di instal menggunakan *library* libpcap. *Library* ini menangkap paket dalam berbagai format protokol, seperti TCP, UDP, ICMP, dan lainnya. Snort kemudian menganalisis paket yang melewati antarmuka jaringan perangkat tersebut dengan membandingkan pola lalu lintas terhadap aturan (*rules*) yang telah ditentukan. Proses ini memungkinkan identifikasi berbagai ancaman, seperti serangan berbasis *host*, akses ilegal, eksploitasi layanan, serta aktivitas mencurigakan lainnya.

Hasil analisis lalu lintas disimpan dalam *file log*, yang mencakup informasi detail mengenai paket yang terdeteksi, termasuk alamat IP sumber dan tujuan, protokol yang digunakan, serta klasifikasi ancaman berdasarkan aturan deteksi. Format penyimpanan *log* ini biasanya berupa *plaintext*, JSON, atau format lain yang dapat dianalisis lebih lanjut. Dalam penelitian ini, pengambilan data dilakukan secara manual dengan membaca dan menyeleksi isi *file log* berdasarkan bagian yang dianggap penting. Langkah ini bertujuan untuk menghindari informasi redundan dan memastikan bahwa data yang dikumpulkan relevan untuk analisis keamanan pada perangkat yang dipantau.



Gambar 1. Sampel Hasil deteksi pada Snort terhadap serangan Rapidscan

Pada hasil *log* di gambar 1 menunjukkan hasil pengujian yang dilakukan menggunakan *tools* Rapidscan, ditemukan beberapa indikasi serangan yang tercatat dalam *log* hasil deteksi snort. Snort berhasil mendeteksi aktivitas mencurigakan seperti serangan DDOS *mstread client to handler*, yang mengindikasikan adanya upaya *Denial of Service* (DoS) melalui komunikasi antara *handler* dan *client*.

Selain itu, terdapat deteksi "*BAD-TRAFFIC same* SRC/DST", yang menunjukkan adanya lalu lintas mencurigakan dengan alamat sumber dan tujuan yang sama, berpotensi menjadi indikasi *scanning* atau *loop* dalam jaringan.

Aktivitas lainnya adalah MS-SQL *ping attempt*, yang menunjukkan upaya ping mencurigakan terhadap *server* Microsoft SQL, kemungkinan sebagai langkah awal eksploitasi. Sebagian besar serangan melibatkan protokol UDP dengan tingkat prioritas yang bervariasi, menunjukkan celah keamanan dalam pengaturan jaringan yang dapat dimanfaatkan oleh penyerang.

Tabel 1. Hasil deteksi pada snort terhadap *tool* Rapidscan

Alert Type	**Classification**	**Priority**	**Protocol**	**Source IP & Port**	**Destination IP & Port**
ICMP PING	Misc activity	3	ICMP	65.2.161.86	103.29.196.121
ICMP traceroute	Attempted Information Leak	2	ICMP	16.50.246.37	103.29.196.121
ICMP PING	Misc activity	3	ICMP	16.50.246.37	103.29.196.121
ICMP traceroute	Attempted Information Leak	2	ICMP	195.123.209.115	103.29.196.121
EXPLOIT ntpdx	Attempted Administrator	1	UDP	159.223.81.166:35880	103.29.196.121:123

overflow attempt	Privilege Gain				
ICMP Destination Unreachable Port Unreachable	Misc activity	3	ICMP	103.29.196.121	103.29.196.113
DDOS Trin00 Master to	Attempted Denial of	2	UDP	103.29.196.113:61517	103.29.196.121:27444
Daemon default password	Service				
MISC AFS access	Misc activity	3	UDP	103.29.196.113:61517	103.29.196.121:7001
ICMP Destination Unreachable Port Unreachable	Misc activity	3	ICMP	103.29.196.121	103.29.196.113
SCAN UPnP service discover attempt	Detection of a Network Scan	3	UDP	103.29.196.113:61517	103.29.196.121:1900
DNS named version attempt	Attempted Information Leak	2	UDP	103.29.196.113:61517	103.29.196.121:53

SCAN Amanda client version request	Attempted Information Leak	2	UDP	103.29.196.113:61517	103.29.196.121:10080
POLICY PCAnywhere server response	Misc activity	3	UDP	103.29.196.113:61517	103.29.196.121:5632
ICMP PING	Misc activity	3	ICMP	47.236.4.122	103.29.196.121
MISC xdmcp info query	Attempted Information Leak	2	UDP	103.29.196.113:61517	103.29.196.121:117
MISC source port 53 to <1024	Potentially Bad Traffic	2	TCP	83.6.58.113:53	103.29.196.121:80
MISC source port 53 to <1024	Potentially Bad Traffic	2	TCP	83.6.58.113:53	103.29.196.121:443
ICMP PING Sun Solaris	Misc activity	3	ICMP	175.6.145.238	103.29.196.121
DDOS mstream client to handler	Attempted Denial of Service	2	TCP	103.29.196.113:34844	103.29.196.121:15104
MS-SQL ping attempt	Misc activity	3	UDP	146.88.241.55:52831	103.29.196.121:1434
SCAN UPnP service	Detection of a Network Scan	3	UDP	128.232.21.75:50249	103.29.196.121:1900

discover attempt					
EXPLOIT ntpdx overflow attempt	Attempted Administrator Privilege Gain	1	UDP	198.98.58.11:51165	103.29.196.121:123
ICMP PING	Misc activity	3	ICMP	210.210.172.80	103.29.196.121
ICMP traceroute	Attempted Information Leak	2	ICMP	5.188.108.90	103.29.196.121

Berdasarkan hasil dari tabel 1 pengujian yang dilakukan menggunakan alat pemindai RapidScan pada *server* Kali Linux dengan IP 103.29.196.113 terhadap web elektro dengan domain ee.unud.ac.id yang memiliki IP 103.29.196.121. *Output* deteksi Snort yang telah diklasifikasikan menunjukkan berbagai jenis trafik yang terdeteksi selama proses pengujian. Beberapa pola trafik yang berasal dari IP 103.29.196.113 dikategorikan sebagai aktivitas uji keamanan, di antaranya adalah trafik ICMP dan UDP yang menunjukkan pemindaian jaringan seperti ICMP PING, *traceroute*, dan UDP *scan* terhadap berbagai *port* seperti 1900 (UPnP), 7001, 27444, dan 15104. Selain itu, terdapat indikasi percobaan eksploitasi dan serangan DoS, termasuk percobaan *ntpdx overflow attempt, DDOS Trin00 Master,* dan *mstream client to handler*, yang menunjukkan uji kerentanan terhadap layanan tertentu. Upaya pemindaian dan pengumpulan informasi juga terdeteksi melalui aktivitas seperti *Amanda client version request* dan *DNS named version attempt*, yang umumnya dilakukan untuk mengetahui versi perangkat lunak yang digunakan di *server target*. Dengan demikian, *traffic* dari IP 103.29.196.113 memang sejalan dengan tujuan pengujian keamanan yang dilakukan terhadap IP 103.29.196.121.

Selain *traffic* dari IP pengujian, terdapat beberapa trafik dari IP yang bukan berasal dari peneliti. Beberapa di antaranya mencakup IP 195.123.209.115 dan 210.210.172.80 yang melakukan ICMP *traceroute*, yang berpotensi sebagai upaya pemetaan jaringan oleh pihak eksternal. Selain itu, IP 159.223.81.166:35880 dan 198.98.58.11:51165 terdeteksi mencoba melakukan *ntpdx overflow attempt*, yang mengindikasikan potensi eksploitasi terhadap

layanan NTP dari sumber luar. Terdapat pula IP 146.88.241.55:52831 yang melakukan MS-SQL *ping attempt*, yang menunjukkan eksplorasi terhadap layanan *database*, serta IP 128.232.21.75:50249 yang mencoba melakukan UPnP *service discover attempt*, yang dapat menjadi indikasi pemindaian layanan jaringan. Kehadiran *traffic* dari sumber eksternal ini menunjukkan adanya aktivitas eksplorasi dan potensi serangan dari pihak yang tidak terkait dengan pengujian yang dilakukan. Oleh karena itu, penting untuk melakukan analisis lebih lanjut untuk menentukan apakah ada ancaman nyata terhadap *server* yang diuji.

Hasil deteksi menunjukkan bahwa pengujian RapidScan menghasilkan banyak *traffic* yang masuk dalam kategori pemindaian dan eksploitasi keamanan, yang memang merupakan bagian dari pengujian yang dilakukan terhadap IP 103.29.196.121. Namun, selain *traffic* dari pengujian, terdapat juga *traffic* dari sumber eksternal yang mencurigakan dan berpotensi berbahaya.

Tabel 2. Hasil deteksi pada snort terhadap *tool* Nessus

Alert Type	Classification	Priority	Protocol	Source IP & Port	Destination IP & Port
ICMP PING	Misc activity	3	ICMP	128.9.29.128	103.29.196.121
ICMP PING NMAP	Attempted Information Leak	2	ICMP	103.29.196.113	103.29.196.121
ICMP PING	Misc activity	3	ICMP	103.29.196.113	103.29.196.121
ICMP Echo Reply	Misc activity	3	ICMP	103.29.196.121	103.29.196.113
ICMP traceroute	Attempted Information Leak	2	ICMP	103.68.60.105	103.29.196.121
ICMP PING	Misc activity	3	ICMP	103.68.60.105	103.29.196.121
DDOS mstream	Attempted Denial of	2	TCP	103.29.196.113:62886	103.29.196.121:15104

client to handler	Service				
MS-SQL ping attempt	Misc activity	3	UDP	103.29.196.113:56052	103.29.196.121:1434
ICMP PING NMAP	Attempted Information Leak	2	ICMP	103.29.196.113	103.29.196.121
ICMP PING undefined code	Misc activity	3	ICMP	103.29.196.113	103.29.196.121
ICMP Echo Reply undefined code	Misc activity	3	ICMP	103.29.196.121	103.29.196.113
ICMP Timestamp Request	Misc activity	3	ICMP	103.29.196.113	103.29.196.121
ICMP Address Mask Request	Misc activity	3	ICMP	103.29.196.113	103.29.196.121
SCAN UPnP service discover attempt	Detection of a Network Scan	3	UDP	103.29.196.113:40501	103.29.196.121:1900
ICMP Address Mask	Misc activity	3	ICMP	103.29.196.113	103.29.196.121

Request					
ICMP Information Request	Misc activity	3	ICMP	103.29.196.113	103.29.196.121
MISC source port 53 to <1024	Potentially Bad Traffic	2	TCP	37.230.137.187:53	103.29.196.121:80
ICMP PING Sun Solaris	Misc activity	3	ICMP	207.211.214.163	103.29.196.121

Pengujian deteksi dari tabel 2 dilakukan menggunakan *tool Nessus* pada *server* Kali Linux dengan IP 103.29.196.113 terhadap web elektro dengan domain ee.unud.ac.id yang memiliki IP 103.29.196.121. Hasil deteksi *Snort* menunjukkan berbagai jenis trafik yang terdeteksi selama pengujian. Dari tabel klasifikasi, terlihat bahwa sebagian besar trafik yang berasal dari IP 103.29.196.113 merupakan aktivitas yang sesuai dengan metode pemindaian keamanan, seperti ICMP PING, ICMP PING NMAP*, traceroute*, dan beberapa permintaan eksplorasi jaringan lainnya seperti ICMP *Timestamp Request,* ICMP *Address Mask Request*, serta pemindaian layanan UPnP. Selain itu, terdapat beberapa percobaan eksploitasi terhadap layanan yang berjalan di *server*, termasuk DDOS *mstream client to handler*, yang terdeteksi pada *port* 15104, serta upaya eksplorasi layanan *database* melalui MS-SQL *ping attempt* pada *port* 1434. Aktivitas ini sejalan dengan tujuan pengujian Nessus yang bertujuan untuk mengidentifikasi celah keamanan pada web elektro dengan IP 103.29.196.121.

Selain trafik dari pengujian yang dilakukan, terdapat juga beberapa trafik yang berasal dari IP lain yang bukan bagian dari pengujian peneliti. Misalnya, terdapat ICMP *PING* dari IP 128.9.29.128 dan 207.211.214.163, yang menunjukkan adanya aktivitas pemetaan jaringan oleh pihak eksternal. IP 103.68.60.105 juga terdeteksi melakukan ICMP *PING* dan *traceroute*, yang mengindikasikan eksplorasi terhadap *server target*. Selain itu, terdapat *traffic* dari IP 37.230.137.187, yang mencoba mengakses port 53 dan 443, diklasifikasikan sebagai *potentially bad traffic*, yang berpotensi menjadi aktivitas berbahaya. Keberadaan *traffic* dari sumber eksternal ini mengindikasikan adanya pihak lain yang mungkin melakukan

pemindaian atau eksploitasi terhadap web elektro, yang tidak terkait dengan pengujian yang sedang dilakukan.

Berdasarkan analisis hasil deteksi Snort, dapat disimpulkan bahwa pengujian menggunakan Nessus menghasilkan berbagai jenis trafik pemindaian dan eksploitasi keamanan, yang memang merupakan bagian dari proses identifikasi kerentanan terhadap IP 103.29.196.121. Namun, keberadaan trafik dari sumber eksternal yang tidak terkait dengan pengujian menunjukkan potensi ancaman dari pihak luar yang perlu diperhatikan.

Dalam sistem deteksi intrusi Snort, setiap peringatan (*alert*) memiliki tingkat prioritas (*Priority*) yang menunjukkan tingkat keparahan atau kepentingannya.

1. *Priority 1* menandakan ancaman tingkat tinggi yang berpotensi menyebabkan eksploitasi langsung atau kerusakan sistem. Biasanya, peringatan dengan prioritas ini berkaitan dengan serangan serius seperti eksekusi kode jarak jauh (*remote code execution*), eskalasi hak akses (*privilege escalation*), atau eksploitasi kerentanan kritis. Contoh serangan yang termasuk dalam kategori ini adalah *buffer overflow* dan injeksi kode berbahaya yang dapat menyebabkan sistem dikendalikan oleh pihak yang tidak berwenang.
2. *Priority 2* menunjukkan ancaman tingkat menengah yang mengindikasikan aktivitas mencurigakan yang dapat menjadi bagian dari serangan lebih besar tetapi belum dikategorikan sebagai ancaman langsung. Aktivitas seperti *port scanning*, pencarian kerentanan (*vulnerability scanning*), atau usaha eksploitasi yang gagal termasuk dalam kategori ini. Meskipun tidak langsung berbahaya, aktivitas ini dapat menjadi tanda awal dari upaya peretasan yang lebih serius.
3. *Priority 3* mencerminkan ancaman tingkat rendah yang lebih bersifat informatif dan biasanya berkaitan dengan pola lalu lintas jaringan yang tidak biasa. Contoh peringatan dengan prioritas ini meliputi akses ke halaman *error* yang sering, percobaan koneksi ke layanan yang tidak tersedia, atau perubahan pola lalu lintas yang mencurigakan. Meskipun tidak selalu menandakan serangan, analisis lebih lanjut tetap diperlukan untuk memastikan bahwa aktivitas tersebut tidak berkembang menjadi ancaman yang lebih besar.

Rekomendasi Perbaikan Kerentanan

Tabel 3. Rekomendasi Perbaikan Kerentanan

No	Vulnerability	Level	Tool	Solusi

1	*Vulnerable JS Library*	High	OWASP ZAP	Segera perbarui ke versi terbaru dari jQuery Validation untuk menutup celah keamanan yang ada.
2	*Buffer Overflow*	Medium	OWASP ZAP	Disarankan untuk menulis ulang bagian program yang rentan dengan menerapkan pemeriksaan panjang pengembalian yang tepat (return length checking) guna mencegah input melebihi kapasitas buffer.

3	*Missing Anti-clickjacking Header*	Medium	Nikto, Nessus, OWASP ZAP	Cegah browser memuat halaman lain dari website selain Universitas Udayana dengan menggunakan header HTTP "X-FramE-Options" atau "Content Security Policy". Terapkan kode Javascript untuk mencegah dimuatnya halaman lain dalam tampilan halaman yang sama ("Frame Busting").
4	*Content Security Policy (CSP) Header Not Set*	Medium	Nessus, OWASP ZAP	Pastikan server web, server aplikasi, load balancer dll. dikonfigurasi untuk menyetel header Content-Security-Policy.

5	*Secure Pages Include Mixed Content*	Medium	OWASP ZAP	Identifikasi semua sumber konten HHTP yang diakses dari halaman HTTPS. Halaman yang tersedia melalui SSL/TLS harus terdiri dari konten yang lengkap dikirim melalui SSL/TLS. Halaman tidak boleh berisi konten apa pun yang dikirimkan melalui HTTP yang tidak terenkripsi.
6	*Vulnerable JS Library*	Medium	OWASP ZAP	Perbedaan library javaScript yang digunakan ke versi terbaru yang aman dan tingkatkan ke versi terbaru jquery-migrate.
7	*Application Error Disclosure*	Medium	OWASP ZAP	Konfigurasi server untuk tidak menampilkan pesan error rinci pada pengguna umum.
8	*Big Redirect Detected*	Low	OWASP ZAP	Melakukan evaluasi kebutuhan redireksi pada aplikasi. Hindari penggunaan redireksi berlebihan untuk meningkatkan efisiensi dan keamanan.
9	*Cookie No HttpOnly Flag*	Low	OWASP ZAP	Tambahkan atribut HttpOnly pada semua cookie yang tidak memerlukan akses JavaScript. Hal ini dapat mencegah akses cookie oleh skrip yang berjalan di sisi klien.

10	*Cookie Without Secure Flag*	Low	OWASP ZAP	Rekomendasi yang dapat diberikan untuk celah ini adalah dengan menambahkan pengaturan pada cookies, dimana cookies hanya dapat diakses oleh admin saja. Konfigurasi ini dilakukan dengan menambahkan script tambahan pada file htaccess atau httpd.conf sebagai berikut. Set-Cookie: <PHPSEDDID>=<cookie-value>; Domain=<ee.unud.ac.id>; Secure; HttOnly
11	*Cross-Domain JavaScript Source File Inclusion*	Low	OWASP ZAP	Celah ini ditemukan karena web ee.unud.ac.id menggunakan beberapa file javascript dari sumber lain yang mungkin saja telah terinfeksi bug atau malware, sehingga rekomendasi yang dapat diberikan adalah menggunakan file javascript dari sumber lain yang terpercaya dan tidak dikendalikan oleh end users aplikasi.
12	*Information Disclosure*	Low	OWASP ZAP	Pastikan informasi sensitif seperti struktur direktori atau data pengguna tidak dapat diakses publik. Gunakan izin file yang tepat dan batasi akses pada server.

13	*Server Leaks Version Information Via "Server" HTTP Response Header Field*	Low	OWASP ZAP	Konfigurasikan server untuk menyembunyikan versi aplikasi atau server yang digunakan. Contoh, pada server Apache, gunakan direktif ServerTokens Prod dan ServerSignature Off
14	*Strict-TransportSecurity Header Not Set*	Low	OWASP ZAP, Nessus	Terapkan header HTTP Strict-Transport-Security (HSTS) untuk memastikan bahwa semua komunikasi dengan server hanya terjadi melalui HTTPS. Contoh: Strict-Transport-Security: max-age=31536000;
				includeSubDomains
15	*Timestamp Disclosure*	Low	OWASP ZAP	Pastikan timestamp yang digunakan dalam aplikasi tidak mengungkapkan informasi yang tidak perlu. Hilangkan atau obfuscate timestamp dari HTTP response headers jika tidak diperlukan.
16	*X-Content-Type-Options Header Missing*	Low	OWASP ZAP	Tambahkan header HTTP X-Content-Type-Options: nosniff untuk mencegah browser dari mencoba menebak tipe MIME dan memaksa browser untuk menggunakan tipe konten yang dikonfigurasi.

Berdasarkan hasil analisis kerentanan menggunakan Nikto, OWASP ZAP, dan Nessus, ditemukan berbagai celah keamanan pada sistem. Rekomendasi perbaikan ini dikaitkan

dengan kategori risiko yang diidentifikasi dalam OWASP Top 10 2021, yang mencerminkan ancaman keamanan aplikasi web yang paling umum dan berbahaya. Dengan memahami bagaimana setiap kerentanan yang ditemukan dalam sistem sesuai dengan kategori OWASP, mitigasi yang lebih spesifik dapat diterapkan untuk mengurangi risiko serangan secara efektif.

Beberapa kerentanan yang ditemukan, seperti *Vulnerable JavaScript Library* dan *Cross-Domain JavaScript Inclusion*, berkaitan erat dengan A06:2021 – *Vulnerable and Outdated Components*, di mana penggunaan pustaka pihak ketiga yang tidak diperbarui meningkatkan risiko eksploitasi oleh penyerang. Untuk mengatasi hal ini, penting bagi pengelola sistem untuk memastikan bahwa semua dependensi diperbarui secara berkala serta menghindari penggunaan skrip dari sumber yang tidak terpercaya.

Kerentanan *Application Error Disclosure dan Server Leaks Version Information* dapat dikategorikan dalam A01:2021 – *Broken Access Control* serta A07:2021 – *Identification and Authentication Failures*, karena menampilkan informasi sensitif seperti konfigurasi *server* atau pesan *error* yang dapat dimanfaatkan untuk melakukan serangan lebih lanjut. Oleh karena itu, pengaturan *server* harus dikonfigurasi agar tidak menampilkan informasi versi serta detail kesalahan yang dapat diakses oleh pengguna umum.

Selain itu, kelemahan dalam mekanisme keamanan seperti *Missing Anti-Clickjacking Header* dan *Content Security Policy* (CSP) *Header Not Set* termasuk dalam kategori A05:2021 – *Security Misconfiguration*, yang menyoroti pentingnya penerapan *header* keamanan yang sesuai untuk melindungi aplikasi dari serangan *clickjacking*, XSS, dan eksploitasi lainnya. Untuk mengatasi hal ini, penerapan *X-Frame-Options* serta *Content-Security-Policy* (CSP) harus segera dikonfigurasi pada *web server* guna membatasi sumber daya eksternal yang dapat dimuat di dalam aplikasi.

Kerentanan *Buffer Overflow*, yang ditemukan dalam sistem, berhubungan erat dengan A09:2021 – *Security Logging* and *Monitoring Failures*, karena serangan berbasis *buffer overflow* sering kali tidak terdeteksi jika sistem tidak memiliki mekanisme pemantauan yang memadai. Oleh karena itu, selain menerapkan *return length checking* untuk mencegah eksploitasi memori, disarankan agar sistem *logging* dan *monitoring* diperkuat untuk mendeteksi aktivitas mencurigakan yang dapat mengarah pada eksploitasi lebih lanjut.

Terakhir, *Cookie Without Secure Flag* dan *Cookie No HttpOnly Flag* termasuk dalam A07:2021 – *Identification and Authentication Failures*, karena kelemahan dalam pengaturan *cookie* dapat dimanfaatkan dalam serangan *session hijacking* atau *cross-site scripting* (XSS). Untuk memitigasi risiko ini, atribut *HttpOnly* dan *Secure* harus diterapkan pada semua *cookie*

yang dikirimkan melalui jaringan, memastikan bahwa *cookie* hanya dapat diakses oleh *server* dan tidak dapat dibaca oleh skrip di sisi klien.

Oleh karena itu, integrasi antara CIA Triad dan OWASP Top 10 dalam strategi keamanan ini akan membantu dalam membangun sistem yang lebih aman, andal, dan tahan terhadap serangan siber.

4.3.2.2 Rekomendasi Perbaikan Berdasarkan CIA Triad

Dalam konteks CIA Triad (*Confidentiality, Integrity, Availability*), setiap kerentanan yang ditemukan memiliki dampak yang signifikan terhadap keamanan sistem, terutama bagi pengelola situs elektro.unud.ac.id. Oleh karena itu, mitigasi harus dilakukan tidak hanya untuk menghilangkan celah keamanan, tetapi juga untuk meminimalkan dampak langsung yang dapat merugikan institusi.

Dari aspek *Confidentiality* (Kerahasiaan), beberapa celah seperti *Information Disclosure, Server Leaks Version Information, dan Timestamp Disclosure* dapat menyebabkan kebocoran informasi sensitif. Jika dibiarkan, informasi teknis mengenai struktur *server* dapat dimanfaatkan oleh penyerang untuk menyusun strategi eksploitasi lebih lanjut. Dampak langsung yang dapat terjadi meliputi akses tidak sah ke data internal, penyalahgunaan informasi pengguna, serta meningkatnya risiko serangan *phishing* dan *identity theft* terhadap pengguna yang terdaftar di dalam sistem. Untuk mengatasi masalah ini, *server* harus dikonfigurasi agar tidak menampilkan informasi teknis kepada publik, serta melakukan pengamanan terhadap direktori dan file penting dengan pembatasan izin akses yang ketat.

Dari sisi *Integrity* (Integritas), kerentanan seperti *Vulnerable JavaScript Library* dan *Cross-Domain JavaScript Inclusion* dapat memungkinkan manipulasi kode yang berjalan dalam sistem. Jika tidak diperbaiki, celah ini dapat dimanfaatkan oleh penyerang untuk menyisipkan skrip berbahaya yang berpotensi mengubah data pengguna, mencuri informasi pribadi, atau bahkan merusak tampilan situs (*defacement*). Dampak langsung yang dapat dirasakan oleh pengelola situs adalah berkurangnya kredibilitas institusi akibat serangan semacam ini, yang dapat mempengaruhi kepercayaan mahasiswa dan dosen terhadap sistem yang digunakan. Untuk menghindari kejadian ini, semua pustaka JavaScript harus diperbarui ke versi terbaru, dan penggunaan skrip dari sumber eksternal yang tidak terpercaya harus dihentikan.

Dari aspek *Availability* (Ketersediaan), beberapa celah seperti *Buffer Overflow, Big Redirect Detected,* dan *Security Misconfiguration* dapat menyebabkan gangguan operasional yang serius, termasuk *server crash* atau *downtime* yang berkepanjangan. Jika serangan *buffer*

overflow berhasil dilakukan, *server* dapat mengalami kegagalan sistem yang memerlukan pemulihan teknis yang memakan waktu dan biaya tambahan.

Selain itu, redireksi berlebihan dalam sistem dapat menyebabkan pengguna mengalami kesulitan dalam mengakses layanan yang mereka butuhkan.

Dampak langsung yang dapat terjadi adalah gangguan terhadap aktivitas akademik dan administrasi, di mana layanan seperti sistem informasi akademik atau portal mahasiswa tidak dapat diakses. Untuk mengatasi masalah ini, pengelola sistem harus menerapkan *return length checking* untuk mencegah *buffer overflow*, serta melakukan evaluasi terhadap penggunaan redireksi guna memastikan efisiensi dan keamanan sistem tetap terjaga.

4. KESIMPULAN

a) Instalasi dan Konfigurasi Snort sebagai *Intrusion Detection System* (IDS) pada *web server* ee.unud.ac.id dilakukan melalui serangkaian tahapan. Proses diawali dengan pembaruan sistem, instalasi Snort dan pemeriksaan konfigurasi alamat IP. Snort dikonfigurasi untuk memantau lalu lintas jaringan *interface* ens192 dengan alamat IP Publik 103.29.xxx.xxx, yang berfungsi sebagai titik akses eksternal menuju *web server*. File konfigurasi utama, yaitu snort.conf, disesuaikan untuk menentukan jaringan yang dipantau (HOME_NET), Snort diatur untuk mendeteksi berbagai jenis ancaman, seperti serangan DDOS, SQL *injection*, dan eksploitasi mode kerentanan, dengan mengaktifkan aturan-aturan yang relevan. Setelah konfigurasi divalidasi dan dipastikan bebas dari kesalahan, snort dijalankan dalam *real-time* untuk memantau lalu lintas jaringan dan menghasilkan peringatan (*alert)* yang ditampilkan langsung di konsol.

b) Pengujian *penetration testing* pada ee.unud.ac.id dilakukan berdasarkan *framework* OWASP Top 10 dan CIA Triad untuk mengidentifikasi kerentanan keamanan. Proses ini mencakup *information gathering* menggunakan WhatWeb, Whois, dan Wappalyzer, diikuti oleh *scanning* dengan Nikto, Nessus, dan OWASP ZAP guna mengidentifikasi kerentanan. Tahap *exploitation* kemudian menguji pemanfaatan kerentanan menggunakan SQLMap, sebelum akhirnya tahap *reporting* mendokumentasikan temuan beserta rekomendasi perbaikannya. Pengujian ini menggunakan Nessus dan OWASP ZAP sebagai alat utama dalam menganalisis kerentanan pada ee.unud.ac.id. Hasil analisis menunjukkan bahwa tingkat kerentanan diklasifikasikan menjadi empat kategori: *High*, dengan dampak besar terhadap keamanan sistem; *Medium*, yang dapat dieksploitasi dalam kondisi tertentu; *Low*, dengan dampak kecil dan tidak mengancam

langsung; serta *Information*, yang hanya berisi informasi bermanfaat tanpa ancaman terhadap sistem.

c) Hasil eksekusi alat pemindai OWASP ZAP dengan *Attack Mode* dan *Manual Explore* mengidentifikasi 25 jenis kemungkinan ancaman keamanan. Dari hasil tersebut, ditemukan bahwa 3 kategori memiliki tingkat ancaman *High*, 5 kategori memiliki tingkat ancaman *Medium*, 11 kategori lainnya masuk dalam tingkat ancaman *Low*, dan 6 jenis *alert* hanya bersifat informasional. Sementara itu, pengujian menggunakan *mode Web Application Test* dengan Nessus menunjukkan adanya kerentanan dengan tingkat keparahan *Medium*, dengan 3 kerentanan yang memiliki nilai CVSS sebesar 5.3 dan 4.3. Berdasarkan temuan ini, *website* ee.unud.ac.id belum dapat dikategorikan sebagai sistem yang sepenuhnya aman. Masih terdapat celah keamanan yang perlu diperhatikan oleh USDI selaku pengelola *website* agar dapat meminimalkan risiko ancaman. Langkah mitigasi yang tepat diperlukan untuk mencegah potensi akses tidak sah yang dapat membahayakan integritas sistem. Sementara Snort IDS yang penulis terapkan memantau seluruh *traffic* atau lalu lintas yang terjadi pada *web server*, di mana *log* pada Snort dapat mengetahui dari mana serangan berasal dan protokol apa yang digunakan. Implementasi Snort berhasil mendeteksi sebanyak 24 jenis *alert type* dan 5 jenis klasifikasi *alert*. Dengan distribusi lima belas jenis *alert type* berasal dari pengujian yang dihasilkan oleh IP penulis (103.29.196.113) dan sembilan jenis serangan berasal dari alamat IP luar yang tidak terkait dengan pengujian. Hasil ini menunjukkan bahwa selain adanya serangan simulatif yang dilakukan dalam rangka evaluasi keamanan, sistem web Elektro juga menghadapi ancaman nyata dari aktor luar yang berpotensi mengeksploitasi kelemahan yang ada.

d) Dengan menerapkan rekomendasi berdasarkan OWASP Top 10 2021 dan CIA Triad, sistem ee.unud.ac.id dapat lebih terlindungi dari berbagai ancaman keamanan yang berpotensi merugikan institusi. Implementasi perbaikan yang tepat tidak hanya akan meningkatkan keamanan dan stabilitas sistem, tetapi juga akan mengurangi risiko eksploitasi yang dapat menyebabkan kerugian reputasi, finansial, dan operasional bagi pihak pengelola USDI. Oleh karena itu, sangat penting untuk segera mengambil langkah-langkah mitigasi yang telah direkomendasikan guna memastikan bahwa sistem tetap aman dan dapat beroperasi secara optimal.

DAFTAR PUSTAKA

Akmal, M. (2023). Analisis dan uji coba tingkat keamanan website UIN Ar-Raniry menggunakan Acunetix Web Vulnerability Scanner (Tesis doktoral, UIN Ar-Raniry Banda Aceh).

Andriyani, S., Sidiq, M. F., & Zen, B. P. (2023). Analisis celah keamanan pada website dengan menggunakan metode penetration testing dan framework ISSAF pada website SMK Al-Kautsar. LEDGER: Journal Informatic and Information Technology, 2(1), 1–13.

Aqsa, M., Anwar, A., & Davi, M. (2024). Pengujian kerentanan celah keamanan website menggunakan threat modelling pada website Prodi Teknologi Rekayasa Komputer Jaringan. Proceeding of TIK, 4(2), 198–208.

Arnaldy, D., & Perdana, A. R. (2019). Implementation and analysis of penetration techniques using the man-in-the-middle attack. In 2019 2nd International Conference of Computer and Informatics Engineering (IC2IE) (pp. 188–192). IEEE. https://doi.org/10.1109/IC2IE47452.2019.8940872

Badan Siber dan Sandi Negara. (2022). Informasi serangan siber. https://honeynet.bssn.go.id/

Bitzer, M., Brinz, N., & Ollig, P. (2021). Disentangling the concept of information security properties: Enabling effective information security governance. ECIS 2021 Research Papers, 134, 1–18. https://aisel.aisnet.org/ecis2021_rp/134

Dewanto, A. P. (2018). Penetration testing pada domain uii.ac.id menggunakan OWASP. https://dspace.uii.ac.id/bitstream/handle/123456789/11281/13523025-AdetyaPutraD-laporanskripsi.pdf?sequence=1&isAllowed=y

Fata, D. (2023). Evaluasi risiko celah keamanan menggunakan metodologi Open Web Application Security Project (OWASP) pada aplikasi web sistem informasi akademik (SIAKAD) UIN Ar-Raniry.

Fikri, M. N., Zen, B. P., Adhitama, R., & Firdaus, E. A. (2023). Analisis keamanan sistem informasi website SMA Negeri 1 Sokaraja menggunakan metode Penetration Testing Execution Standard (PTES). Jurnal Informatika, 2(2), 19–27.

Fronita, M. (2016). Analisis celah keamanan website Sitasi menggunakan vulnerability assessment. Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi, 9(1), 1–7.

Hidayatulloh, S., & Saptadiaji, D. (2021). Penetration testing pada website Universitas ARS menggunakan Open Web Application Security Project (OWASP). Jurnal Algoritma, 18(1), 77–86.

Mulyanto, Y., Haryanti, E., & Jumirah, J. (2021). Analisis keamanan website SMAN 1 Sumbawa menggunakan metode vulnerability assessment. Jurnal Informatika Teknologi dan Sains (Jinteks), 3(3), 394–400.

Robbani, S. A. (2023). Analisa kerentanan keamanan aplikasi manajemen aset berbasis web menggunakan metode OWASP (Open Web Application Security Project): Studi kasus PT. XYZ (Tesis doktoral, Sekolah Tinggi Teknologi Terpadu Nurul Fikri).

Sanjaya, I. G. A. S., Sasmita, G. M. A., & Arsa, D. M. S. (2020). Evaluasi keamanan website Lembaga X melalui penetration testing menggunakan framework ISSAF. Jurnal Ilmiah Merpati, 8(2), 113–124.

Shafira, K. (2022). Analisis keamanan website repository Institut Teknologi Telkom Purwokerto menggunakan metode vulnerability assessment.

Sholeh, A. N., & Wardaya, M. S. S. (2019). Analisis dan pengujian kerentanan sistem informasi perpustakaan. Jurnal Mandiri: Ilmu Pengetahuan, Seni, dan Teknologi, 3(1), 116–131.

Stallings, W., & Brown, L. (2018). Computer security: Principles and practice. Pearson Education.

Syafaat, A. (2024, January). Identifikasi kerentanan keamanan pada website Fakultas Ilmu Komputer Universitas Subang menggunakan metodologi OWASP. In Global, 11(1), 84–99.