Walisongo Journal of Information Technology
Vol 6, No 1 (2024): 27-36. DOI: 10.21580.wjit.2024.6.1.25390
ISSN 2715-0143 (online) | 2714-9048 (printed)

Implementasi Owasp Zap Untuk Menguji Kerentanan Sistem Informasi Akademik (Sia Universitas Teknologi Yogyakarta Domain: Sia.Uty.Ac.Id)

Iin Rohmatika Aulia[1], Rianto[2], EIH Ujianto[3]
[123] Magister Teknologi Informasi, Universitas Teknologi Yogyakarta
Corresponding Author: iin.6240211012@student.uty.ac.id

Abstract: *The security of academic information systems is essential to maintaining the confidentiality, integrity, and availability of academic data. This study aims to identify web application vulnerabilities in an academic information system using the OWASP framework. Vulnerability testing identified three main findings: NGINX server misconfigurations, potential exposure of sensitive files, and the need to strengthen authentication and access controls. The identified misconfigurations and potential file exposure indicate security risks that require further verification to determine their actual exploitability and impact. Recommended mitigation measures include correcting server configurations, strengthening authentication mechanisms, and restricting unauthorized access. These findings highlight the importance of systematic vulnerability assessment and appropriate security controls to improve the protection and reliability of academic information systems.*

Keyword: *Academic Information System; OWASP; Vulnerability Assessment; Web Application Security.*

Copyright ©2024 Walisongo Journal of Information Technology

Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

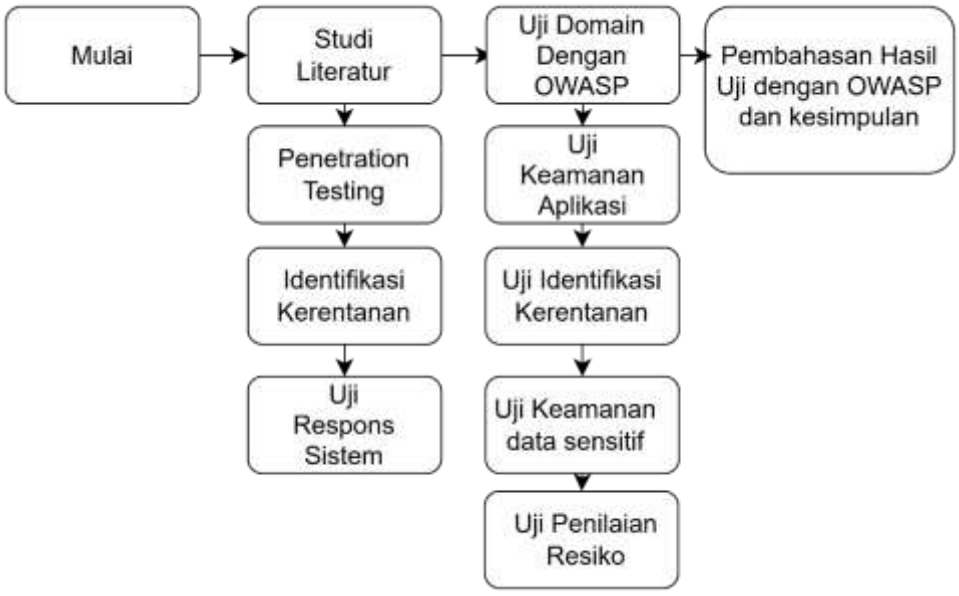
1. Pendahuluan

Informasi akademik harus terjaga atau terintegrasi. Agar terhindar dari penyalahgunaan data oleh pihak yang tidak berwenang.2. Mahasiswa yang satu tidak dapat melihat data yang dimiliki mahasiswa lain, jadi setiap mahasiswa memiliki data yang bersifat rahasia, 3. Setiap mahasiswa pasti memiliki akun pengguna untuk masuk dalam system informasi akademik, selama mahasiswa merupakan entitas yang berwenang maka system informasi akademik harus tersedia untuk diakses dalam hak akses sesuai kewenangannya. Tiga aspek ini harus dipenuhi dengan baik, agar terhindar dari ancaman keamanan system, maka diperlukan keamanan system informasi (Riyan Farismana & Dian Pramadhana, 2023). Network Security Essentials: Applications and Standards (7th ed.). Pearson. (Yunus, 2019), Stallings menegaskan bahwa aspek dasar dari keamanan informasi adalah menjaga kerahasiaan, integritas, dan ketersediaan data. Selain itu, sistem harus tetap tangguh menghadapi ancaman dan serangan yang terus berkembang. Ancaman keamanan system informasi akademik tidak hanya datang dari internal, juga dapat datang dari factor eksternal. Seperti seraangan siber, pencurian data, hingga kegagalan sistem maka perlu dilakukan keamanan isistem informasi. Keamanan system informasi merupakan suatu upaya untuk melindungi informasi dari ancaman yang dapat merusak kerahasiaan, integritas, dan ketersediaan data serta menjaga agar sistem informasi berfungsi dengan baik tanpa adanya gangguan. Keamanan ini mencakup perlindungan terhadap perangkat keras (hardware), perangkat lunak (software), serta data dan informasi yang ada dalam sistem. Keamanan sistem informasi bertujuan untuk memastikan bahwa data hanya dapat diakses

oleh pihak yang berwenang, data yang ada tetap akurat dan tidak dimodifikasi tanpa izin, serta data dapat diakses saat dibutuhkan (Book, 2023). Principles of Information Security (6th ed.). Cengage Learning. "Information security is the protection of information systems from theft or damage to the hardware, software, or information contained within them, as well as from disruption or misdirection of the services they provide." (Whitman & Mattord, 2020, p. 4) bahwa keamanan sistem informasi melibatkan perlindungan terhadap seluruh komponen dalam sistem informasi baik perangkat keras, perangkat lunak, maupun data yang terkandung di dalamnya serta memastikan bahwa layanan yang diberikan oleh sistem tidak terganggu atau salah arah. Sebelum menentukan Langkah-langkah keamanan system, perlu dilakukan uji kerentanan system informasi, dalam penelitian ini akan melakukan penerapan aplikasi OWASP untuk mengetahui kerentanan system informasi akademik, OWASP merupakan Universitas Teknologi Yogyakarta non-profit yang berfokus pada meningkatkan keamanan aplikasi perangkat lunak. Salah satu proyek OWASP yang terkenal adalah OWASP Top 10, yang merangkum sepuluh risiko keamanan terpenting yang sering ditemukan pada aplikasi web. Aplikasi OWASP terdiri dari berbagai alat dan pedoman yang dirancang untuk membantu pengembang, penguji keamanan, dan profesional TI lainnya dalam mengidentifikasi dan mengatasi masalah keamanan dalam aplikasi perangkat lunak. OWASP Foundation. (2022). OWASP Top 10 - 2021. Open Web Application Security Project. "OWASP Top 10 is the reference standard for identifying the most critical risks to web applications and provides foundational security practices for organizations to follow." (Elanda & Lintang Buana, 2021)

2. Metode

Gambar 1. Alur Penelitian



a. Studi Literatur

Penetration Test (atau sering disingkat *pentest*) adalah proses yang dilakukan oleh profesional keamanan (*pentester*) untuk mengidentifikasi dan mengeksploitasi celah keamanan dalam suatu sistem. Tujuan utamanya adalah untuk mengetahui potensi risiko dan kerentanannya sebelum bisa dimanfaatkan oleh pihak yang tidak bertanggung jawab. Metode ini dapat dilakukan secara

manual maupun otomatis dengan menggunakan alat-alat tertentu. *Penetration testing* dilakukan untuk mengidentifikasi kerentanannya.(Riandhanu, 2022),

Identifikasi kerentanan bertujuan untuk mengetahui apakah ada kelemahan dalam sistem yang bisa dimanfaatkan oleh peretas. Menguji respons sistem: untuk memeriksa bagaimana sistem merespon ketika diserang, apakah sistem dapat mendeteksi dan mengatasi serangan tersebut.(Dewangkara et al., n.d.) Meningkatkan keamanan: setelah celah ditemukan, rekomendasi perbaikan diberikan agar sistem lebih aman dan terhindar dari potensi serangan. Kendall, R. (2020). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. Wiley. "*Penetration testing typically comes in the form of black box, white box, or gray box testing, each of which offers a different level of insight into the security posture of the application or system.*" (Lee, 2023)

Black box testing: di mana *pentester* tidak memiliki informasi apa pun tentang sistem yang akan diuji. *White box testing*: di mana *pentester* memiliki akses penuh ke kode sumber atau konfigurasi sistem yang diuji. *Gray box testing*: di mana *pentester* memiliki sebagian informasi tentang sistem yang diuji, seperti akses terbatas atau data pengguna. (Tinambunan et al., 2024)*Penetration testing* pada aplikasi web (*web application*): untuk mencari kelemahan dalam aplikasi berbasis web. Sistem operasi untuk mengeksploitasi kerentanannya. Perangkat keras (*hardware*), terkadang perangkat keras juga diuji untuk potensi kerentanannya.

b. Uji Domain dengan OWASP ZAP

Penerapan framework *OWASP pada sistem informasi akademik adalah langkah penting untuk memastikan bahwa sistem tersebut aman dari potensi ancaman siber, baik itu serangan dari luar maupun dari dalam (*insider threats*). Sistem informasi akademik yang melibatkan pengelolaan data mahasiswa, dosen, jadwal kuliah, nilai, dan informasi sensitif lainnya, harus dilindungi dari kerentanannya yang dapat mengekspos data pribadi dan informasi penting.(Hariyadi & Nastiti, 2021)*

*OWASP (*Open Web Application Security Project*) menyediakan sejumlah pedoman dan alat untuk mendeteksi dan mengurangi risiko keamanan pada aplikasi web, yang sangat relevan untuk sistem informasi akademik yang biasanya berbasis web. Penerapan* framework *OWASP dalam pengembangan dan pemeliharaan sistem informasi akademik:(Hariyadi & Nastiti, 2021)*

1. Keamanan Aplikasi Web dalam Sistem Informasi Akademik

Keamanan aplikasi web adalah disiplin yang sangat penting dalam pengembangan sistem informasi akademik. Mengingat banyaknya data sensitif yang dikelola oleh sistem tersebut, dari data pribadi mahasiswa, nilai akademik, hingga informasi keuangan, penting untuk memastikan bahwa aplikasi web yang digunakan aman dan bebas dari kerentanannya. OWASP *memberikan pedoman untuk mengidentifikasi dan menangani potensi kerentanannya yang ada pada aplikasi web, dengan fokus pada serangan yang sering terjadi seperti* SQL Injection*,* Cross-Site Scripting (XSS)*, dan* Cross-Site Request Forgery (CSRF)*.*

2. OWASP Top Ten untuk Mengidentifikasi Kerentanannya

OWASP Top Ten *adalah salah satu inisiatif penting yang dapat digunakan untuk memitigasi kerentanannya pada sistem informasi akademik. Dalam konteks ini, daftar* OWASP Top Ten *memberikan kerentanannya yang sering ditemukan dalam aplikasi web yang dapat*

dimanfaatkan oleh peretas.(Kerentanan Keamanan et al., 2021) Contoh penting yang harus diperhatikan dalam sistem informasi akademik adalah:

- ***Injection Attacks (*SQL Injection*):** Penyalahgunaan input untuk menyisipkan perintah berbahaya ke dalam sistem basis data.*
- ***Broken Authentication:** Kelemahan dalam mekanisme otentikasi yang memungkinkan penyerang mendapatkan akses tidak sah ke akun pengguna.*
- ***Sensitive Data Exposure:** Kelemahan dalam mengelola data sensitif seperti informasi pribadi mahasiswa, yang dapat terpapar jika tidak dilindungi dengan baik (misalnya, tanpa enkripsi).(Irawadi Alwi & Budi Ilmawan, 2021)*

Implementasi dari pedoman *OWASP Top Ten* dalam sistem informasi akademik sangat penting untuk mengidentifikasi dan mengatasi kerentanannya yang bisa dieksploitasi oleh pihak yang tidak bertanggung jawab.

3. Integrasi OWASP ZAP untuk Pengujian Keamanan

OWASP ZAP (Zed Attack Proxy) adalah alat keamanan yang dapat digunakan untuk mengidentifikasi kerentanannya dalam sistem informasi akademik melalui pengujian penetrasi otomatis.(Bagas et al., 2022) *ZAP* memanfaatkan berbagai teknik seperti *static analysis* dan *dynamic analysis* untuk mencari potensi celah keamanan dalam aplikasi web. Untuk sistem informasi akademik, *ZAP* dapat digunakan untuk mengidentifikasi kerentanannya seperti *XSS*, *CSRF*, dan *broken authentication*, serta memberikan laporan terperinci mengenai titik-titik rentan yang harus diperbaiki. Dengan *ZAP*, pengembang dapat memverifikasi apakah aplikasi akademik mereka aman dari eksploitasi kerentanannya yang dapat terjadi selama proses autentikasi atau pengiriman data sensitif.

4. Penerapan Keamanan dengan Fokus pada Data Sensitif

Sensitive Data Exposure merupakan salah satu risiko utama dalam sistem informasi akademik, di mana data mahasiswa dan dosen bisa terekspos apabila tidak dilindungi dengan benar. Dalam hal ini, *OWASP* mengajarkan pentingnya enkripsi data pada data yang sedang dalam perjalanan (*in-transit*) dan data yang disimpan (*at-rest*). (Rochman et al., 2021)Penerapan *Transport Layer Security (TLS)* untuk mengamankan komunikasi antara *client* (misalnya, browser mahasiswa) dan server aplikasi akademik adalah hal mendasar. *OWASP* juga menyarankan untuk selalu menggunakan *hashing* yang kuat untuk penyimpanan kata sandi dan teknik enkripsi modern untuk melindungi data pribadi yang disimpan di dalam basis data.

5. Pengujian Keamanan dan Penilaian Risiko Sistem

Dalam konteks sistem informasi akademik, pengujian penetrasi (*pen test*) merupakan metode yang digunakan untuk mengevaluasi kerentanannya dan mengidentifikasi potensi ancaman. Menggunakan *OWASP ZAP* atau *OWASP Dependency-Check*, pengembang dan administrator sistem dapat menjalankan pengujian untuk memastikan bahwa tidak ada celah keamanan yang dapat dieksploitasi oleh pihak yang tidak sah.

6. Peran devsecops dalam Pengembangan Sistem

Devsecops adalah integrasi praktik keamanan ke dalam siklus pengembangan perangkat lunak yang lebih besar (*SDLC*).(Febriana, n.d.) Dalam sistem informasi akademik, penerapan prinsip *devsecops* membantu memastikan bahwa pengujian keamanan dan penilaian risiko dilakukan secara terus-menerus sepanjang siklus hidup aplikasi, dari fase pengembangan hingga pemeliharaan. *OWASP* mendukung pendekatan *devsecops* dengan menyediakan alat dan sumber daya untuk pengujian otomatis, seperti *OWASP Dependency-Check* dan *OWASP ZAP*, yang dapat diintegrasikan ke dalam *pipeline CI/CD* untuk memastikan bahwa aplikasi aman dari kerentanannya setiap kali diperbarui atau diubah.(Pendidikan & Konseling, n.d.)

3. Hasil dan Pembahasan

Setelah dilakukan pengujian dengan AWASP ditemukan tiga allert beserta resikonya : 1. Cloud Metadata Attack pada NGINX Server dengan resiko tinggi, 2. File sensitif ditemukan dapat diakses atau tersedia dengan resiko medium, 3.

1. Cloud Metadata Attack pada NGINX Server merupakan Serangan Cloud Metadata mencoba mengeksploitasi konfigurasi yang salah pada server NGINX untuk mengakses metadata instansi yang dikelola oleh penyedia layanan cloud. Metadata cloud ini adalah informasi yang disediakan oleh penyedia layanan cloud mengenai instansi virtual, seperti data konfigurasi, kredensial, dan informasi sistem lainnya. Metadata ini dapat dimanfaatkan oleh aplikasi yang berjalan di instansi cloud untuk berinteraksi dengan layanan cloud lainnya. Semua penyedia layanan cloud utama menyediakan metadata melalui alamat IP internal yang tidak dapat dijangkau dari luar (unroutable IP) yaitu 169.254.169.254. Alamat IP ini biasanya digunakan untuk mengakses metadata instansi yang berjalan di cloud. Masalah Terjadi ketika NGINX Dikonfigurasi dengan Salah: Jika server NGINX dikonfigurasi dengan cara yang salah, server ini dapat mengekspose (memperlihatkan) alamat IP internal tersebut kepada luar, yang memungkinkan penyerang untuk mengakses metadata tersebut melalui permintaan HTTP. Serangan ini terjadi ketika penyerang dapat memanipulasi header host dalam permintaan HTTP, di mana nilai host header ini digunakan untuk mengarahkan permintaan ke alamat IP yang salah atau ke server yang seharusnya tidak terjangkau oleh pengguna luar. Contoh serangan ini adalah ketika penyerang bisa memanipulasi header host dalam HTTP untuk mengarahkan server NGINX ke IP internal 169.254.169.254, yang dapat mengungkapkan metadata terkait instansi cloud. Metadata ini bisa berisi informasi yang sangat sensitif, seperti kredensial API, token akses, dan informasi konfigurasi lainnya yang dapat memungkinkan penyerang untuk mengakses atau bahkan mengambil alih sistem sepenuhnya.
   Proses Serangan:
   a. Penyerang mengirimkan permintaan HTTP ke server NGINX yang dikonfigurasi dengan salah.
   b. Dalam permintaan tersebut, penyerang memanipulasi header host untuk mengarahkan NGINX ke IP internal 169.254.169.254.
   c. NGINX yang salah dikonfigurasi kemudian mengarahkan permintaan ke alamat IP tersebut dan metadata instansi cloud yang sensitif dapat diekspos.
   d. Penyerang dapat melihat metadata yang dikembalikan melalui kode status respons yang berhasil (misalnya 200 OK).
   e. Dengan informasi yang diperoleh dari metadata, penyerang bisa mendapatkan akses ke sistem atau bahkan mengakses layanan lain di cloud yang menggunakan kredensial yang terungkap.

   Solusi yang Disarankan: Jangan percayakan data pengguna dalam konfigurasi NGINX: Konfigurasi NGINX seharusnya tidak menggunakan data dari header host yang dapat dikendalikan oleh pengguna, terutama untuk tujuan mengarahkan permintaan ke IP internal atau informasi yang sensitif. Periksa konfigurasi NGINX secara hati-hati: Pastikan bahwa NGINX tidak mengizinkan atau mengekspos IP internal seperti 169.254.169.254 melalui pengaturan yang tidak aman. Pastikan konfigurasi server memblokir akses ke alamat tersebut dari luar jaringan internal. Validasi header permintaan dengan benar: Pastikan bahwa server hanya menerima permintaan dengan header host yang valid dan telah diautentikasi, serta tidak mengandalkan data header tersebut untuk mengakses informasi sensitif.

   Serangan ini menyoroti pentingnya keamanan konfigurasi server web seperti NGINX, terutama ketika server tersebut berinteraksi dengan layanan cloud. Sebuah konfigurasi yang salah dapat dengan mudah dieksploitasi oleh penyerang yang memanfaatkan celah tersebut untuk mengakses metadata instansi cloud, yang biasanya sangat sensitif dan dapat memberikan akses penuh ke sistem dan layanan cloud. Oleh karena itu, memastikan bahwa

konfigurasi server NGINX sangat aman dan tidak mempercayakan data dari sumber yang dapat dikendalikan pengguna sangat penting dalam melindungi aplikasi dan infrastruktur cloud.

2. File sensitif ditemukan dapat diakses atau tersedia Mengindikasikan adanya potensi kebocoran informasi yang sangat penting. File yang sensitif ini bisa berisi data administratif, konfigurasi sistem, atau kredensial yang digunakan untuk mengakses sistem atau layanan lainnya. Jika file semacam ini dapat diakses tanpa kontrol yang tepat, maka individu berbahaya (seperti penyerang) dapat memanfaatkannya untuk melanjutkan serangan atau bahkan melakukan social engineering (manipulasi psikologis terhadap individu untuk mendapatkan informasi lebih lanjut atau akses ke sistem).
   Contoh file sensitif yang mungkin ditemukan termasuk:
   a. File konfigurasi yang berisi pengaturan sistem penting.
   b. File dengan kredensial seperti password atau API key.
   c. File administratif yang mungkin memberikan kontrol lebih lanjut terhadap sistem.

   Jika file semacam ini dapat diakses oleh orang yang tidak berwenang, maka penyerang bisa mendapatkan informasi yang memungkinkan mereka untuk mengakses lebih banyak data atau bahkan melakukan serangan lebih lanjut, termasuk serangan berbasis sosial (social engineering) yang bertujuan menipu pengguna untuk memberikan informasi sensitif atau akses ke sistem.
   Solusi yang direkomendasikan Berikut adalah solusi yang bisa diterapkan,
   a. Pertimbangkan apakah komponen tersebut benar-benar diperlukan di lingkungan produksi, Jika file atau komponen tersebut tidak diperlukan dalam produksi, maka sebaiknya nonaktifkan atau hapus file tersebut. Semakin sedikit file atau komponen yang tersedia di lingkungan produksi, semakin kecil kemungkinan terjadinya kebocoran informasi atau eksploitasi.
   b. Jika file atau komponen tersebut diperlukan, pastikan aksesnya dilindungi dengan autentikasi dan otorisasi yang tepat:Pastikan hanya pihak yang berwenang yang dapat mengakses file tersebut. Hal ini bisa dilakukan dengan memastikan bahwa akses ke file sensitif hanya bisa dilakukan setelah proses autentikasi (misalnya login dengan username dan password yang sah) dan otorisasi (memastikan pengguna yang mengakses memiliki hak akses yang sesuai).
   c. Batasi eksposur ke sistem internal atau alamat IP sumber tertentu: Untuk meningkatkan keamanan, batasi akses ke file sensitif hanya untuk sistem internal atau hanya dari alamat IP tertentu yang dipercaya. Dengan membatasi siapa saja yang bisa mengakses file ini, Anda dapat mengurangi risiko file tersebut jatuh ke tangan yang salah.
   d. Gunakan kontrol akses berbasis peran (role-based access control/RBAC): Dengan menggunakan RBAC, Anda dapat menentukan peran apa saja yang bisa mengakses file sensitif tersebut. Misalnya, hanya administrator atau pengguna dengan peran tertentu yang boleh mengakses file konfigurasi atau kredensial.
   e. Audit dan monitor akses: Lakukan audit akses secara berkala untuk memastikan bahwa tidak ada akses yang tidak sah ke file-file sensitif. Jika memungkinkan, aktifkan logging dan monitoring untuk memantau setiap upaya akses ke file-file tersebut.

1. *Check for differences in response based on fuzzed user agent* (misalnya, situs mobile, akses sebagai crawler mesin pencari)" menunjukkan bahwa pemindaian ini bertujuan untuk memeriksa apakah aplikasi web memberikan respons yang berbeda berdasarkan informasi yang ada dalam User-Agent header pada permintaan HTTP. User-Agent adalah sebuah header dalam permintaan HTTP yang memberi tahu server tentang jenis perangkat, browser, atau aplikasi yang digunakan oleh pengguna untuk mengakses situs web.
   Beberapa jenis User-Agent yang umum digunakan adalah:
   a. *Mobile User-Agent*: Digunakan untuk mengakses situs melalui perangkat mobile (smartphone atau tablet).

b. Crawler User-Agent: Digunakan oleh mesin pencari seperti Googlebot atau Bingbot untuk mengindeks konten situs web. Pemindaian ini mengevaluasi apakah aplikasi web merespons secara berbeda ketika permintaan datang dengan User-Agent yang berbeda (misalnya, mengakses situs menggunakan perangkat mobile atau sebagai crawler mesin pencari). Pemindaian ini mengidentifikasi apakah aplikasi memberikan konten yang berbeda untuk jenis pengguna atau agen yang berbeda. Ini bisa berisiko karena aplikasi dapat mengungkapkan lebih banyak data atau informasi kepada crawler atau pengguna yang tidak sah dibandingkan dengan pengguna biasa. Cara Kerja Pemindaian: Fuzzing User-Agent: Pemindaian akan mencoba fuzzing (menguji) aplikasi dengan User Agent yang dimanipulasi atau berbeda, seperti User-Agent untuk perangkat mobile atau untuk mesin pencari. Tujuannya adalah untuk melihat apakah server memberikan respons yang berbeda berdasarkan jenis User-Agent yang digunakan. Perbandingan Respons: Respons dari server kemudian dibandingkan dengan dua hal: Status Code: Kode status HTTP yang diberikan oleh server (misalnya, 200 OK, 404 Not Found, dll.). Hashcode dari Body Respons: Hashcode (sebuah representasi unik) dari isi respons yang dikembalikan oleh server. Hashcode ini digunakan untuk membandingkan apakah konten yang dikirimkan dalam respons berbeda antara dua permintaan dengan User-Agent yang berbeda. Jika terdapat perbedaan dalam status code atau isi respons berdasarkan User-Agent, ini bisa menunjukkan adanya potensi masalah keamanan atau kebijakan pengaturan akses yang salah. Perbedaan respons berdasarkan User-Agent dapat menunjukkan beberapa masalah atau kerentanannya, seperti: Pengungkapan Informasi Sensitif: Server mungkin memberikan data yang lebih rinci atau lebih sensitif ketika mengidentifikasi permintaan sebagai datang dari mesin pencari atau perangkat mobile. Misalnya, server bisa mengungkapkan lebih banyak informasi tentang halaman atau data yang tidak boleh diakses oleh pengguna biasa. Pengaturan yang Salah atau Tidak Konsisten: Aplikasi mungkin tidak mengelola akses secara konsisten atau benar berdasarkan tipe pengguna atau perangkat. Ini bisa menjadi celah yang dapat dimanfaatkan oleh penyerang. Serangan Berbasis User-Agent: Penyerang bisa memanfaatkan perbedaan respons untuk mengeksploitasi kerentanannya, misalnya dengan mencoba mengekspos data atau mengambil keuntungan dari pengaturan yang kurang aman untuk crawler atau perangkat mobile.

Solusi yang Disarankan:

a. Jika pemindaian menunjukkan adanya perbedaan respons yang mencurigakan, beberapa langkah mitigasi yang dapat dilakukan antara lain: Konsistensi dalam Respons Server: Pastikan bahwa server memberikan respons yang konsisten tanpa memperhatikan jenis User-Agent. Pengaturan respons server yang berbeda berdasarkan User Agent dapat membuka celah bagi serangan yang memanfaatkan perbedaan ini.
b. Batasi Pengungkapan Informasi Sensitif: Pastikan bahwa tidak ada informasi sensitif atau data penting yang hanya tersedia untuk crawler atau perangkat mobile. Semua pengguna, terlepas dari perangkat yang digunakan, harus memiliki akses yang serupa terhadap data yang sama, kecuali jika secara eksplisit dibatasi oleh kebijakan atau izin.
c. Penerapan Kebijakan Keamanan yang Tepat: Terapkan kontrol akses yang ketat dan pastikan bahwa hanya pengguna yang sah yang dapat mengakses data sensitif. Jangan memberikan akses yang lebih besar kepada agen yang tidak dikenal, seperti crawler, jika tidak diperlukan.
d. Verifikasi Respons Berdasarkan User-Agent: Selalu verifikasi bahwa respons dari server tidak mengungkapkan informasi yang tidak perlu berdasarkan jenis perangkat atau aplikasi yang digunakan untuk mengakses situs. Setiap perbedaan dalam respons harus diperiksa untuk memastikan tidak ada kebocoran data atau pengaturan yang salah.

e. Pemindaian OWASP yang memeriksa perbedaan respons berdasarkan User-Agent bertujuan untuk mengidentifikasi kerentanannya terkait pengungkapan informasi yang tidak sah atau pengaturan akses yang salah. Jika aplikasi memberikan respons berbeda tergantung pada jenis perangkat atau agen yang digunakan, ini bisa membuka potensi bagi penyerang untuk mengeksploitasi data atau melakukan serangan berbasis social engineering. Untuk itu, sangat penting untuk menjaga konsistensi dalam respons server dan memastikan bahwa tidak ada data yang bocor hanya karena perbedaan User Agent.

4. Conclusion

Dari hasil pengujian dengan OWASP ZAP ditemukan tiga allert beserta resikonya : 1. Cloud Metadata Attack pada NGINX Server dengan resiko tinggi, 2. File sensitif ditemukan dapat diakses atau tersedia dengan resiko medium, 3. File sensitif ditemukan dapat diakses atau tersedia dengan resiko medium.

Cloud Metadata Attack pada NGINX Server dengan resiko tinggi. Serangan ini terjadi jika server NGINX dikonfigurasi secara salah dan memperbolehkan akses ke alamat IP internal 169.254.169.254 yang digunakan untuk mengakses metadata. Penyerang dapat memanipulasi header host dalam permintaan HTTP untuk mengekspos metadata tersebut. Adapun usulan soolusi :1. Menonaktifkan atau menghapus komponen yang tidak diperlukan dalam konfigurasi NGINX,2. Memastikan bahwa akses ke metadata hanya dapat dilakukan oleh pihak yang sah, dengan menerapkan autentikasi dan otorisasi yang tepat, 3. Membatasi eksposur metadata hanya untuk sistem internal atau alamat IP yang dipercaya, Dengan memastikan konfigurasi NGINX yang benar, kita dapat menghindari kebocoran informasi sensitif dan melindungi sistem dari potensi eksploitasi.

File sensitif ditemukan dapat diakses atau tersedia dengan resiko medium. Pemindaian OWASP yang menemukan bahwa file sensitif dapat diakses atau tersedia mengindikasikan potensi kebocoran informasi yang sangat penting, seperti kredensial, konfigurasi, atau data administratif. Akses yang tidak sah ke file ini bisa dimanfaatkan oleh penyerang untuk melanjutkan serangan atau melakukan social engineering. Adapun Solusi yang diusulkan : 1. Menonaktifkan atau menghapus file sensitif yang tidak diperlukan di lingkungan produksi, 2.Jika file diperlukan, memastikan bahwa aksesnya dilindungi dengan autentikasi dan otorisasi yang tepat, 3. Membatasi akses hanya untuk sistem internal atau alamat IP yang tepercaya, dan menggunakan kontrol akses berbasis peran untuk membatasi siapa yang dapat mengakses file tersebut.

Pemindaian OWASP yang memeriksa perbedaan respons berdasarkan User-Agent bertujuan untuk mendeteksi potensi kerentanannya dalam aplikasi web, yang bisa mengekspos informasi sensitif atau memungkinkan eksploitasi melalui pengaturan akses yang tidak konsisten. Tiga aspek utama yang perlu diperhatikan adalah: 1. Konsistensi Respons: Aplikasi web sebaiknya memberikan respons yang konsisten, terlepas dari jenis perangkat atau agen yang digunakan untuk mengakses situs. Perbedaan respons yang tidak perlu bisa membuka celah bagi penyerang. 2.Pengungkapan Informasi Sensitif: Pengungkapan data lebih rinci kepada crawler atau perangkat mobile dapat memberikan informasi sensitif yang seharusnya hanya diakses oleh pengguna yang sah. Pengaturan akses harus memastikan informasi sensitif tidak jatuh ke tangan yang salah. 3. Keamanan dan Pengaturan Akses: Perbedaan respons berdasarkan User Agent menunjukkan adanya potensi kelemahan dalam pengaturan kontrol akses. Penggunaan kebijakan yang tepat dan konsisten untuk memastikan hanya pengguna yang sah yang dapat mengakses data sensitif sangat penting untuk mencegah eksploitasi. Secara keseluruhan, untuk menghindari potensi serangan, sangat penting untuk memastikan konsistensi respons server, menjaga kerahasiaan informasi sensitif, dan mengelola akses secara tepat berdasarkan kebijakan keamanan yang ketat.

5. Referensi

Bagas, N. A., Hendradi dan Sasongko, P. D., Bagas Adinugroho, N., Hendradi, P., & Sasongko, D. (2022). ANALISIS KEAMANAN E-LEARNING MENGGUNAKAN OPEN WEB APPLICATION SECURITY PROJECT (OWASP) (STUDI KASUS: MOCA UNIMMA). In *Jurnal Informatika* (Vol. 22).

Book, ·. (2023). *AUDIT SISTEM INFORMASI TEORI, FRAMEWORK DAN STUDI KASUS MENGGUNAKAN FRAMEWORK*. Https://www.researchgate.net/publication/366806651

Dewangkara, I. B. I., Santi, K. S., Putri, V. A., & Listartha, I. M. E. (n.d.). Penerapan Analisis Kerentanan XSS dan Rate Limiting pada Situs Web mtsn 3 Negara Menggunakan OWASP ZAP. In *JURNAL INFORMATIKA UPGRIS* (Vol. 8, Issue 1). Https://www.zaproxy.org/download/.

Elanda, A., & Lintang Buana, R. (2021). *ANALISIS KUALITAS KEAMANAN SISTEM INFORMASI E-OFFICE BERBASIS WEBSITE PADA STMIK ROSMA DENGAN MENGGUNAKAN OWASP TOP 10* (Vol. 6, Issue 2).

Febriana, R. (n.d.). Blackbox Testing Sistem Informasi Absensi Pegawai Karawang Dengan Metode Top 10 Owasp Attack. *Jurnal Ilmiah Wahana Pendidikan*, *2022*(12), 327–334. Https://doi.org/10.5281/zenodo.6945632

Hariyadi, D., & Nastiti, F. E. (2021a). Analisis Keamanan Sistem Informasi Menggunakan Sudomy dan OWASP ZAP di Universitas Duta Bangsa Surakarta. *Jurnal Komtika (Komputasi Dan Informatika)*, *5*(1), 35–42. Https://doi.org/10.31603/komtika.v5i1.5134

Hariyadi, D., & Nastiti, F. E. (2021b). Analisis Keamanan Sistem Informasi Menggunakan Sudomy dan OWASP ZAP di Universitas Duta Bangsa Surakarta. *Jurnal Komtika (Komputasi Dan Informatika)*, *5*(1), 35–42. Https://doi.org/10.31603/komtika.v5i1.5134

Irawadi Alwi, E., & Budi Ilmawan, L. (2021). Analisis Keamanan Sistem Informasi Akademik (SIAKAD) Universitas XYZ Menggunakan Metode Vulnerability Assessment. In *Informatics Journal* (Vol. 6, Issue 3).

Kerentanan Keamanan, A., Menggunakan, W., Aryanti, D., Dan, N., & Utamajaya, J. N. (2021). *METODE OWASP (OPEN WEB APPLICATION SECURITY PROJECT) PADA DINAS TENAGA KERJA* (Vol. 1, Issue 3).

Lee, D. (2023). *Analisis Kerentanan Aplikasi Akademik Berbasis Website XYZ Menggunakan OWASP*. *11*(2).

Pendidikan, J., & Konseling, D. (n.d.). *Penetration Testing pada Website Universitas Singaperbangsa Karawang Menggunakan Open Web Application Security Project (OWASP)* (Vol. 4).

Riandhanu, I. O. (2022). Analisis Metode Open Web Application Security Project (OWASP) Menggunakan Penetration Testing pada Keamanan Website Absensi. *Jurnal Informasi Dan Teknologi*. Https://doi.org/10.37034/jidt.v4i3.236

Riyan Farismana, & Dian Pramadhana. (2023). Perbandingan Vulnerability Assesment Menggunakan Owasp Zap dan Acunetix Pada Sistem Informasi Repositori Politeknik Negeri Indramayu. *Jurnal Teknik Informatika Dan Teknologi Informasi*, *3*(2), 26–32. Https://doi.org/10.55606/jutiti.v3i2.2853

Rochman, A., Rohian Salam, R., Sandi Agus Maulana Sekolah Tinggi Manajemen Ilmu Komputer, dan, & Likmi, S. (2021). DI RUMAH SAKIT XYZ. *ANALISIS KEAMANAN WEBSITE DENGAN INFORMATION SYSTEM SECURITY ASSESSMENT FRAMEWORK (ISSAF) DAN OPEN WEB APPLICATION SECURITY PROJECT*, *2*(4).

Tinambunan, F., Junaidi, A., Mustika Rizki Informatika, A., Timur Jl Rungkut Madya No, J., Anyar, G., Gn Anyar, K., & Timur, J. (2024). PENGUJIAN SISTEM INFORMASI AKADEMIK UNIVERSITAS X MELALUI

PENDEKATAN PENETRATION TESTING BERDASARKAN OWASP TOP 10. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 1).

Yunus, M. (2019). ANALISIS KERENTANAN APLIKASI BERBASIS WEB MENGGUNAKAN KOMBINASI SECURITY TOOLS PROJECT BERDASARKAN FRAMEWORK OWASP VERSI 4. *Jurnal Ilmiah Informatika Komputer*, *24*(1), 37–48. Https://doi.org/10.35760/ik.2019.v24i1.1988