LINIER
LITERATUR INFORMATIKA & KOMPUTER
e-ISSN: 3063-2218
DOI : 10.33096/linier.v1i4.2537
Literatur Informatika & Komputer
Vol 1, No 4 (2024), pp 379-388

Analisis Keamanan Website SIAKAD menggunakan Pentest Tools

Gery Ardiansyah Saputra[a], Erick Irawadi Alwi[b], Andi Widya Mufila Gaffar[c]
Universitas Muslim Indonesia, Makassar, Indonesia
[a]13020200056@student.umi.ac.id, [b]erick.alwi@umi.ac.id; [c]widya.mufila@umi.ac.id

Received: 12-08-2024 | Revised: 01-10-2024 | Accepted: 28-11-2024 | Published: 29-12-2024

Abstrak
Penggunaan Sistem Informasi Akademik (SIAKAD) telah menjadi sebuah keharusan bagi perguruan tinggi dalam memberikan kemudahan pengguna untuk menjalankan kegiatan administrasi akademik secara online. Namun, terkadang SIAKAD perguruan tinggi memiliki celah keamanan yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab untuk melakukan peretasan. Penelitian ini bertujuan untuk mengidentifikasi, mengklasifikasi, dan memberikan rekomendasi keamanan pada website SIAKAD di Universitas Muslim Indonesia. Metode yang digunakan dalam penelitiaan ini merupakan metode penilaian kerentanan, dimulai dengan *footprinting* untuk mengumpulkan informasi awal tentang SIAKAD, diikuti oleh pemindaian kerentanan menggunakan *tools PenTest* untuk mengidentifikasi kerentanan dan tingkat risiko yang ada. Berdasarkan hasil penelitian, ditemukan beberapa kerentanan dengan 1 risiko tinggi terkait dengan kerentanan pada versi PHP, 2 risiko sedang melibatkan pengaturan *cookie* yang tidak aman dan *header* keamanan yang hilang, serta 4 risiko rendah berupa pengaturan cookie tanpa *flag HttpOnly*. Adapun rekomendasi perbaikan yang diberikan meliputi peningkatan pengamanan terhadap serangan *Remote Code Execution (RCE), Cross-Site Scripting (XSS), Cookie Hijacking, dan Clickjacking.*

Kata Kunci : *Fooprinting*, Sistem Informasi, Siakad, *Vulnerabity Scenning*, *Website*

Pendahuluan

Website merupakan sebuah halaman data berbasis *web* yang menyediakan berbagai informasi, dokumen ataupun tautan yang menghubungkan halaman data satu dengan halaman data lainnya yang dapat kita akses saat terkoneksi dengan internet kapanpun dan dimanapun melalui *browser* [1]. Selain itu, *website* sangatlah dibutuhkan dalam penyampaian informasi yang begitu luas dan tanpa batas [2]. Terkhususnya pada Institusi Pendidikan, contohnya Sistem Informasi Akademik.

Dalam dunia Pendidikan modern sistem informasi akademik merupakan sebuah sistem yang digunakan oleh institusi pendidikan yang dimanfaatkan untuk meningkatkan pelayanan kepada mahasiswanya [3], sistem informasi akademik kampus berbentuk sebuah *website.* Universitas Muslim Indonesia merupakan salah satu Univeristas yang juga memanfaatkan sistem informasi akademik yaitu SIAKAD. *Website* merupakan kumpulan komponen yang terdiri dari teks, gambar, suara animasi merupakan media informasi yang menarik dan diminatidigunakan sebagai media berbagai informasi [4]. *Website* tersebut tidak hanya menjadi sumber informasi penting bagi mahasiswa, dosen, dan staf administrasi, tetapi juga menjadi *platform* vital untuk pengelolaan data mahasiswa, jadwal kuliah, registrasi kursus, dan informasi lainnya yang berkaitan dengan kegiatan akademik [5].

Vulnerability Assessment merupakan proses mendefinisikan, mengidentifikasi, mengklasifikasikan, dan memprioritaskan kerentanan dalam sistem komputer, aplikasi, dan infrastruktur jaringan dan memberikan organisasi melakukan penilaian dengan pengetahuan, kesadaran, dan latar belakang risiko yang diperlukan untuk memahami ancaman terhadap lingkungannya dan bereaksi dengan tepat. Proses *vulnerability assessment* yang dimaksudkan untuk mengidentifikasi ancaman dan risiko yang ditimbulkannya biasanya melibatkan penggunaan alat pengujian otomatis, seperti pemindai keamanan jaringan, yang hasilnya terdaftar dalam laporan *vulnerability assessment* [6]. Pengetahuan tentang aspek keamanan sangat penting diketahui oleh seorang *master web*. Metode *vulnerability assessment* dapat membantu mendeteksi kerentanan dalam sebuah aplikasi *web*. Hasil dari assessment tersebut menjadi pertimbangan bagi master web untuk mengambil tindakan pencegahan serta mengetahui kinerja serangan saat melakukan serangan [7].Dalam melindungi keamanan informasi dan data, dalam tulisan ini membahas Spesifik indikator-indikator keamanan organisasi atau perusahaan. Indikator keamanan ini dikenal dengan “CIA Triad” yaitu Confidentiality (Kerahasiaan), Integrity (Integritas), Availability (Ketersediaan). Dalam penelitian ini membahas CIA TRIAD dalam melindungi keamanan informasi dan data dalam suatu sistem yang diterapkan. Ketiga faktor ini saling berkaitan dan saling menjaga ikatan satu sama lain, dengan kata lain jika salah satu faktor tersebut dihilangkan Keamanan informasi

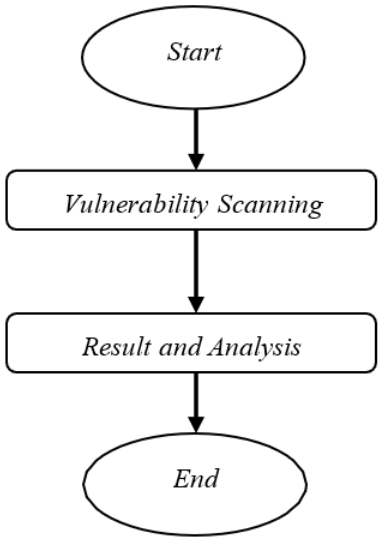
dan data akan sangat beresiko.. *Availability* yakni ketersediaan data dan informasi ketika dibutuhkan dalam hal ini informasi atau data harus tersedia untuk pihak yang memiliki aksestanpa terhambat ataupun kesulitan [8], [9].

Pentest Tools merupakan serangkaian perangkat yang digunakan untuk mensimulasikan serangan terhadap jaringan suatu organisasi atau perusahaan dengan tujuan menemukan kelemahan pada sistem tersebut. Kegiatan ini dilakukan oleh individu yang dikenal sebagai *penetration tester,* atau *pentester*. *Pentest Tools* memiliki standar resmi yang digunakan sebagai acuan dalam pelaksanaannya. Terdapat dua jenis pendekatan dalam penggunaan *Pentest Tools,* yaitu *overt,* di mana pengujian dilakukan dengan sepengetahuan pihak yang diuji, dan *covert,* di mana pengujian dilakukan tanpa sepengetahuan pihak yang diuji untuk mensimulasikan serangan yang lebih realistis [10].

Keamanan aplikasi web, termasuk SIAKAD, sering kali menjadi target serangan karena potensi kerentanannya yang dapat dieksploitasi oleh peretas. *Penetration Testing (pentest)* merupakan pendekatan yang efektif untuk mengidentifikasi dan mengevaluasi kelemahan dalam sistem keamanan. Dengan menggunakan *pentest tools*, kita dapat mensimulasikan serangan untuk menguji ketahanan sistem terhadap ancaman-ancaman potensial. *Pentest tools* ini memberikan analisis mendalam tentang kelemahan-kelemahan yang mungkin ada dalam aplikasi *web*, dan memungkinkan kita untuk mengambil tindakan pencegahan yang sesuai. Tujuan dari penelitian ini merupakan untuk menganalisis keamanan *website* SIAKAD dengan menggunakan *pentest tools* untukmengidentifikasi dan mengevaluasi kerentanan yang ada. Serta bertujuan untuk memberikan rekomendasi yangbermanfaat dalam memperbaiki dan meningkatkan keamanan *website* SIAKAD agar dapat melindungi data akademik dengan lebih baik.

Metode

Metode yang digunakan pada penelitian ini merupakan metode *Vulnerability Assessment*. *Vulnerability assessment* merupakan sebuah metode mencari celah kerentanan dari *website* target (SIAKAD) yang dapat diaksessecara *online* dengan menggunakan *tools vulnerability scanning* [11]. Adapun tahapan penelitian dapat dilihatpada Gambar 1.



Gambar 1. Tahapan Penelitian

A. *Vulnerability Scanning* merupakan tahapan dilakukannya *vulnerability scanning* dengan menggunakan *tools vulnerability scanning* (*Pentest*). Dengan tujuan mencari informasi keamanan yang terdapat pada target mencakup beberapa hal seperti *Vulnerabilities Server, Insecure cookie, security header, Server software and technology,* dan *Security.txt file* pada suatu sistem operasi atau aplikasi.
B. Tahapan *Result and Analysis* ini akan memberikan hasil analisis terkait celah keamanan yang ditemukan dan memberikan rekomendasi perbaikan dari celah keamanan tersebut. Hasil yang didapatkan biasanya terdiri dari beberapa kategori risiko keamanan, mulai dari yang kritis hingga yang bersifat informasional

Perancangan

A. Analisis Permasalahan

Penetration dan *Vulnerability testing* pada sistem informasi akademik "SIAKAD" merupakan langkah penting untuk mengidentifikasi dan mengevaluasi kerentanan sistem yang berperan krusial dalam pengelolaan data akademik di institusi pendidikan. Proses ini menggunakan serangkaian alat dari situs "*Pentest Tools*" yang memungkinkan deteksi awal celah keamanan yang mungkin tidak terungkap melalui audit rutin. Pengujian ini sangat vital mengingat sistem ini menyimpan informasi sensitif yang berkaitan dengan mahasiswa, dosen, dan staf administrasi, membuatnya menjadi target potensial bagi para peretas yang ingin mengakses dan mengeksploitasi data tersebut.

Setelah identifikasi awal menggunakan *"Pentest Tools",* alamat IP yang terkait dengan SIAKAD akan diuji lebih lanjut untuk mendeteksi kerentanan spesifik, termasuk *scanning* untuk kerentanan umum, *fingerprinting* untuk mendetailkan informasi sistem, dan *enumeration* untuk mendapatkan data lebih lanjut tentang sumber daya yang terbuka. Proses ini tidak hanya menyederhanakan pengujian keamanan melalui penggunaan alat berbasis *web* yang dapat diakses langsung melalui *browser* tetapi juga menyediakan hasil yang akurat yang dapat langsung diolah untuk mendapatkan *insight* mengenaikeamanan sistem. Laporan dari hasil pengujian ini akan memberikan evaluasi yang bermanfaat bagi pengelola jaringan komputer untuk lebih meningkatkan kewaspadaan dan melakukan perbaikan keamanan yang diperlukan.

B. Analisis Kebutuhan

Penetration testing pada sistem informasi akademik "SIAKAD" merupakan untuk secara proaktif mengidentifikasi dan menanggulangi titik kelemahan serta kerentanan yang ada sebelum dapat dieksploitasi oleh peretas. Hal ini sangat penting mengingat sistem menyimpan data vital yang berhubungan dengan kegiatan akademik, yang apabila dikompromikan dapat mengakibatkan kerugian signifikan baik dari segi integritas data maupun privasi pengguna. Kerentanan dalam sistem ini tidak hanya berpotensi merugikan reputasi institusi tetapi juga dapat mengganggu proses belajar mengajar yang berlangsung.

Selanjutnya, pelaksanaan *penetration testing* ini juga bertujuan untuk mengevaluasi dan memverifikasi bahwa manajemen keamanan yang diimplementasikan sudah sesuai dengan standar keamanan sistem informasi yang berlaku. Melalui *penetration testing*, institusi dapat memahami dengan lebih baik mengenai efektivitas mekanisme keamanan yang sudah berjalan, serta mengidentifikasi kebutuhan-kebutuhan perbaikan dalam sistem keamanan yang ada. Ini memastikan bahwa semua aspek keamanan dikelola dengan benar, dan sistem informasi akademik beroperasi dalam lingkungan yang aman dan terkontrol.

C. Analisis Sistem

Dalam penelitian ini, aplikasi yang digunakan untuk melakukan *penetration testing* merupakan program yangsesuai dengan langkah-langkah *pentesting* dan bersumber dari situs *web* yang dikenal dengan nama *"Pentest Tools".* Situs ini menyediakan berbagai alat berbasis *web* yang memungkinkan pengguna untuk melakukan tes keamanan secara langsung melalui *browser*, yang sangat memudahkan dalam proses pengujian keamanan. Pada Tabel 1. dapat dilihat sistem (*tool*) yang digunakan untuk *penetration testing* dalam pengerjaan penelitian

Tabel 1. Tahapan yang Digunakan dalam Penelitian dengan *"Pentest Tools"*

No.	Tahapan	***Pentest Tools***
1	*Footprinting*	*Online Port Scanner*
2	*Scanning*	*Website Vulnerability Scanner*
3	*Reporting*	*Manual Reporting via Pentest Tools' reports*

Tahapan yang digunakan untuk melakukan *penetration testing* menggunakan serangkaian alat dari platform "*Pentest Tools*," yang telah dipilih karena kemampuannya dalam menyediakan tes keamanan yang komprehensif secara *online*. Setiap tahapan penelitian dilakukan dengan alat yang spesifik untuk memastikan analisis yang mendalam dan akurat terhadap keamanan sistem yang diuji.*Footprinting* dilakukan dengan menggunakan *Online Port Scanner*, yang merupakan alat penting dalam menentukan

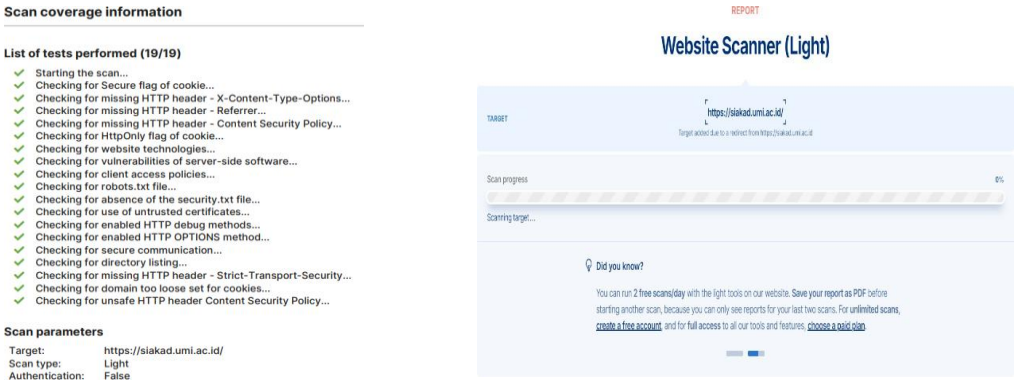
port yang terbuka pada sistem target. Ini merupakan langkah awal krusial karena memberikan gambaran awal tentang potensi titik masuk untuk serangan lebih lanjut. Selanjutnya, tahapan Scanning menggunakan *Website Vulnerability Scanner* dari "*Pentest Tools*" untuk mengidentifikasi kerentanan umum yang mungkin dimiliki oleh aplikasi *web*. Scanner ini secara otomatis menilai aplikasi *web* untukberbagai kerentanan keamanan, memberikan dasar yang kuat untuk tindakan perbaikan keamanan yang akan datang.

Akhirnya, tahapan Reporting dilakukan secara manual berdasarkan laporan yang dihasilkan oleh alat-alat dari "*Pentest Tools*." Laporan ini meliputi rincian lengkap dari temuan, analisis kerentanan, dan rekomendasi untuk perbaikan. Pendekatan ini memastikan bahwa semua temuan dan saran perbaikan dapat dikomunikasikan dengan efektif kepada pemangku kepentingan dan tim pengembang untuktindakan selanjutnya. Secara keseluruhan, penggunaan "*Pentest Tools*" dalam penelitian ini memastikan bahwa semua aspek keamanan sistem diuji secara menyeluruh, dari identifikasi awal kerentanan hingga evaluasi dampak dan pelaporan, sehingga memberikan keamanan yang maksimal terhadap serangan siber.

Pemodelan

A. Footprinting

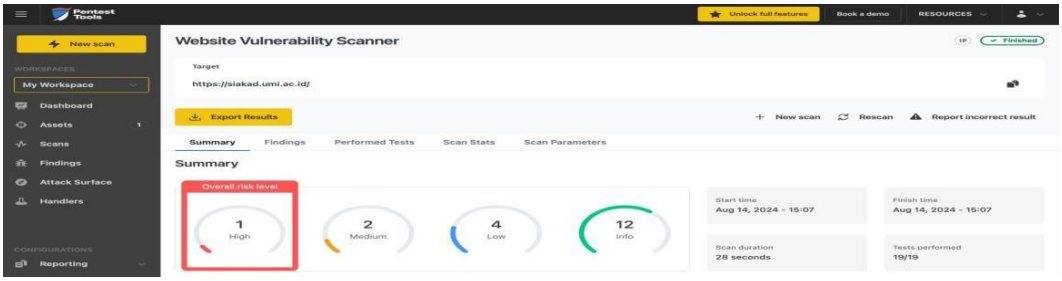
Footprinting merupakan kegiatan mengumpulkan informasi sebanyak-banyaknya yang terkait dengan target, seperti perangkat yang digunakan, merek, tipe, nomor versi OS, topologi fisik *network*, perangkat *security*, *network address*, *subnetting,* dan lain-lain [12]. *Footprinting* mengacu pada aktivitas apa pun yang bertujuan mengumpulkan data pada target yang sistemnya akan diretas sebelum melakukan proses pembobolan sistem sesungguhnya [13]. Adapun proses *footprinting* menggunakan aplikasi (*Pentest Tool*) Untuk mengambil informasi sebuah alamat *domain* dari *website* SIAKAD, maka penulis menggunakan *website (Pentest Tool)* Adapun hasil informasi domain yang didapatkan yaitu siakad.umi.ac.id dapat dilihat pada Gambar 3.



Gambar 2. Hasil dan Proses *Footprinting*

B. *Vulnerability scanning*

Vulnerability scanning merupakan kegiatan proses memperoleh informasi *vulnerability network* dengan memanfaatkan berbagai tools network scanning dan *vulnerability scanner*, seperti port yang terbuka , bugs aplikasi dan mengetahui serangan - serangan yang akan terjadi terhadap kerentaan website yang ada, yang akan berdampak cukup buruk apabila terjadi [14]. Adapun hasil *tools vulnerability scanning* yang digunakan pada penelitian ini yaitu (*Pentest Tool*) dapat dilihat pada Gambar 3 dan Tabel 2.



Gambar 3. Hasil *Vulneability Scanning* pada website siakad

1. *High* risiko pada perangkat lunak sisi server
   Tabel 2. *Vulnerabilites found for server-side software*

Tingkat	CVSSCVE	Ringkasan	Perangkat lunak yang terpengaruh
9.8	CVE-2024-4577	Dalam PHP versi 8.1.* sebelum 8.1.29, 8.2.* sebelum 8.2.20, 8.3.*sebelum 8.3.8, saat menggunakan Apache dan PHP-CGI di Windows, jika sistem diatur untuk menggunakan halaman kode tertentu, Windows dapat menggunakan perilaku "Paling Sesuai" untuk mengganti karakter di baris perintah yang diberikan ke fungsiAPI Win32. Modul PHP CGI mungkin salah menafsirkan karakter tersebut sebagai opsi PHP, yang memungkinkan pengguna jahat meneruskan opsi ke biner PHP yang sedang dijalankan, dan dengandemikian mengungkapkan kode sumber skrip, menjalankan kode PHP sewenang-wenang di server, dll.	php 5.6.40
7.5	CVE-2017-8923	Fungsi *zend_string_extend* di *Zend/zend_string.h* di PHP hingga 7.1.5 tidak mencegah perubahan pada objek *string* yang menghasilkan panjang *negatif,* yang memungkinkan penyerang jarak jauh menyebabkan penolakan layanan (aplikasi *crash*) atau mungkin memiliki dampak lain yang tidak ditentukan oleh memanfaatkan penggunaan *skrip* .= dengan *string* yang panjang.	php 5.6.40
7.5	CVE-2017-9225	Masalah ditemukan di *Oniguruma* 6.2.0, seperti yang digunakan dalam *mod Oniguruma* di *Ruby* hingga 2.4.1 dan *mbstring* di PHP hingga 7.1.5. Penulisan tumpukan di luar batas di *onigenc_unicode_get_case_fold_codes_by_str()* terjadi selama kompilasi ekspresi *reguler*. Titik kode 0xFFFFFFFF tidak ditanganidengan benar di *unicode_unfold_key().* Ekspresi *reguler* yang salahformat dapat mengakibatkan 4 byte dihapuskan dari akhir buffer tumpukan *expand_case_fold_string()* selama panggilan ke *onigenc_unicode_get_case_fold_codes_by_str(),* yang merupakan*buffer overflow* tumpukan yang umum.	php 5.6.40
7.5	CVE-2019-9641	Masalah ditemukan pada komponen *EXIF* di PHP sebelum 7.1.27, 7.2.x sebelum 7.2.16, dan 7.3.x sebelum 7.3.3. Ada pembacaan yangtidak diinisialisasi di *exik*_process_IFD_in_TIFF.	php 5.6.40
6.8	CVE-2015-9253	Masalah ditemukan di PHP 7.3.x sebelum 7.3.*0alpha3*, 7.2.x sebelum 7.2.8, dan sebelum 7.1.20. Proses master php-fpm me-*restart* proses anak dalam *loop* tanpa akhir ketika menggunakan fungsi eksekusi program (misalnya, *passthru*, *exec, shell_exec,* atau *system*) dengan aliran STDIN yang tidak memblokir, menyebabkanproses *master* ini mengkonsumsi 100% CPU, dan menggunakan ruang *disk* dengan *log* kesalahan dalam jumlah besar, seperti yang ditunjukkan oleh serangan yang dilakukan oleh pelanggan terhadapfasilitas *hosting* bersama.	php 5.6.40

Pada Tabel 2 memiliki risiko yang diamana penyerang dapat mencari eksploitasi yang sesuai (atau membuatnya sendiri) untuk setiap kerentanan ini dan menggunakannya untuk menyerang sistem. Kerentanan ini dapat memberikan akses kepada penyerang untuk melakukan berbagai serangan yang dapat merusak integritas, kerahasiaan, dan ketersediaan sistem, termasuk serangan injeksi, eksekusi kode jarak jauh, atau penolakan layanan *denial-of-service* (DoS). Dampaknya bisa sangat parah jika perangkat lunak yang rentan merupakan bagian inti dari infrastruktur *server*. Rekomendasi untuk masalah ini disarankan untuk mengupgrade perangkat lunak yang terpengaruh ke versi terbaru untuk menghilangkan risiko kerentanan ini. *Patch* keamanan dari *vendor* resmi harus diterapkan sesegera mungkin, dan penilaian keamanan berkala harus dilakukan untuk memastikan bahwa semua perangkat lunak sisi *server* tetap aman. Selain itu, penggunaan alat pemantauan keamanan dapat membantu mendeteksi dan merespon terhadap eksploitasi kerentanan secara *real-time*. Dengan

klasifikasi sebagai berikut:

CWE: CWE-1026 (Use of Unsafe Components) OWASP Top 10 - 2017: A9 - Using Components with Known VulnerabilitiesOWASP Top 10 - 2021: A6 - Vulnerable and Outdated Components

2. *Medium* Risiko Pada Pengaturan *Cookies*
Tabel 3. *Insecure cookie setting: missing Secure flag*

URL	Nama Kue	Bukti
https://siakad.umi.ac.id/	CRC	*Set-Cookie: PHPSESSID=d323fc7fb0ad4dd19e751c26da915667; path=/,CRC=5e55dab17d3d42fb1e97f8318000d2a4.*

Pada table 3 Ada desktipsi risiko bahwa penyerang akan menyadap komunikasi teks-jelas antara *browser* dan *server* yang akan mencuri *cookie* pengguna. Jika ini merupakan *cookie* sesi, penyerang dapat memperoleh akses tidak sah ke sesi *web* korban. Hal ini dapat menyebabkan pengambil alihan akun, pencurian data pribadi, atau penyalahgunaan sesi pengguna. Dalam lingkungan *web* yang sensitif terhadap keamanan, kehilangan kontrol terhadap sesi pengguna bisa berakibat fatal, terutama jika melibatkan data sensitif atau transaksi keuangan. Rekomendasi untuk permasalahan ini, Setiap kali *cookie* berisi informasi sensitif atau merupakan token sesi, maka *cookie* tersebut harus selalu diteruskan menggunakan saluran terenkripsi. Pastikan tanda aman disetel untuk *cookie* yang berisi informasi sensitif tersebut. Penggunaan HTTPS di seluruh situs *web* merupakan langkah penting untuk memastikan bahwa semua data, termasuk *cookie*, ditransmisikan secara aman. *Developer* juga harus memastikan bahwa konfigurasi *server* mendukung penggunaan *flag* "*Secure*" untuk semua *cookie* penting. Dengan klasifikasi sabagai berikut:

CWE: CWE-614 (Sensitive Cookie in HTTP)

OWASP Top 10 - 2017: A6 - Security Misconfiguration OWASP Top 10 - 2021: A5 - Security Misconfiguration

Tabel 4. *Insecure cookie setting: missing HttpOnly flag*

URL	Nama Kue	Bukti
https://siakad.umi.ac.id/	CRC, PHPSESSID	Server merespons dengan *header Set-Cookie* yang tidak menentukan *flag HttpOnly:* *Set-Cookie: CRC=5e55dab17d3d42fb1e97f8318000d2a4Set-Cookie*: PHPSESSID=d323fc7fb0ad4dd19e751c26da915667.

Pada Tabel 4 terdapat *Cookie* yang tidak dilindungi dengan *flag HttpOnly* rentan terhadap pencurian melalui serangan *XSS (Cross-Site Scripting).* Penyerang dapat menyuntikkan skrip berbahaya ke dalam halaman *web*, yang kemudian dapat mengakses *cookie* dari *skrip* tersebut dan mengirimkannya ke situs berbahaya lainnya. Jika *cookie* ini berisi informasi sesi pengguna, penyerang bisa mendapatkan akses tidak sah ke sesi tersebut, yang dapat menyebabkan pengambilalihan akun atau eksfiltrasi data sensitif. Dalam konteks aplikasi *web*, ini merupakan risiko yang signifikan karena dapat mengakibatkan hilangnya kepercayaan pengguna, pelanggaran data, dan potensi kerugian finansial. Rekomendasi dalam permasalahan ini sangat penting untuk mengamankan *cookie* yang mengandung informasi sensitif atau token sesi dengan *flag HttpOnly*.Ini memastikan bahwa *cookie* hanya dapat diakses oleh *server*, bukan oleh *skrip* yang dijalankan di*browser*. Implementasi ini dapat secara signifikan mengurangi risiko serangan XSS yang bertujuan mencuri *cookie* pengguna. *Developer* harus memeriksa semua tempat di mana *cookie* disetel danmemastikan bahwa *flag HttpOnly* digunakan secara konsisten. Dengan klasifikasi sebagai berikut:

CWE: CWE-1004 (Insecure Cookie Handling)

OWASP Top 10 - 2021: A5 - Security Misconfiguration OWASP Top 10 - 2017: A6 - Security Misconfiguration Header Keamanan Tidak Ada*: X-Content-Type-Options*

3. *Low* Risiko Pada *Hearder* Keamanan

Tabel 5. *Missing security header: X-Content-Type-Options*

URL	Bukti
https://siakad.umi.ac.id/	*Header* respons tidak menyertakan *header* keamanan *HTTP X-Content-Type-Options.*

Pada Tabel 5 ketika *header X-Content-Type-Options* tidak disetel, *browser* mungkin mencoba menebak jenis MIME dari sebuah file yang diterima, yang dapat menyebabkan penanganan konten yang tidak diinginkan atau berbahaya. Hal ini dapat membuka celah bagi serangan *Cross-Site Scripting* (XSS) di mana penyerang dapat menyisipkan konten berbahaya yang kemudian dijalankan oleh browser dengan tipe MIME yang salah. Ini terutama berbahaya bagi aplikasi *web* yang menampilkan konten pengguna atau data dari sumber eksternal yang tidak tepercaya. Risiko ini dapat dimanfaatkan oleh penyerang untuk menjalankan skrip berbahaya di browser pengguna, yang berpotensi mencuri data sensitif, mengubah halaman *web* yang dilihat pengguna, atau melakukan tindakan tidak sah atas nama pengguna. Rekomendasi untuk mencegah browser menebak jenis MIME dari file, *header X-Content-Type-Options* harus disetel ke *nosniff*. Ini akan memaksa *browser* untuk mematuhi jenis MIME yang diindikasikan oleh *server*, sehingga mencegah eksekusi konten berbahaya yang mungkin salah diklasifikasikan. Pengaturan ini harus diterapkan pada semua respon HTTP yang melibatkan konten yang dihasilkan oleh pengguna atau sumber eksternal untuk memastikan bahwa hanya tipe konten yang dimaksud yang diproses oleh *browser*. Dengan klasifikasi Sebagai Berikut:

CWE: CWE-693 (Protection Mechanism Failure)

OWASP Top 10 - 2021: A5 - Security Misconfiguration OWASP Top 10 - 2017: A6 - Security Misconfiguration

Tabel 6. *Missing security header: Referrer-Policy*

URL	Bukti
https://siakad.umi.ac.id/	*Header respons* tidak menyertakan *header* keamanan HTTP *Referrer-Policy* serta *tag* <meta> dengan nama *'referrer'* tidak ada dalam *respons*.

Pada Tabel 6 *Referrer-Policy* merupakan mekanisme yang mengontrol informasi referensi yang dikirimoleh *browser* ketika pengguna mengklik tautan. Tanpa *Referrer-Policy* yang sesuai, *browser* akan mengirimkan URL asal penuh dalam *header Referrer*, yang dapat berisi informasi sensitif seperti jalur halaman atau parameter kueri. Informasi ini bisa digunakan oleh situs *eksternal* untuk melacakperilaku pengguna atau mengumpulkan data tanpa sepengetahuan pengguna. Dalam skenario terburuk, ini dapat mengarah pada kebocoran informasi sensitif yang dapat dimanfaatkan oleh pihakketiga untuk serangan lebih lanjut, seperti pengintaian atau manipulasi pengguna. Rekomendasiu untuk melindungi privasi pengguna dan mencegah kebocoran informasi, sangat disarankan untuk mengkonfigurasi *Referrer-Policy* pada *server*. Nilai *no-referrer* akan memastikan bahwa tidak ada informasi referensi yang dikirimkan, atau same-origin dapat digunakan untuk hanya mengirim informasi referensi ke halaman dalam domain yang sama. *Administrator server* harus menilai jenisdata yang mungkin disertakan dalam *header Referrer* dan mengatur kebijakan yang paling sesuai dengan kebutuhan keamanan dan privasi aplikasi mereka. Dengan klasifikasi:

CWE: CWE-693 (Protection Mechanism Failure)

OWASP Top 10 - 2021: A5 - Security Misconfiguration OWASP Top 10 - 2017: A6 - Security Misconfiguration

Tabel 7. *Missing security header: Content-Security-Policy*

URL	Bukti
https://siakad.umi.ac.id/	*Respons* tidak menyertakan *header* atau *tag* meta keamanan Konten-Keamanan-Kebijakan HTTP. .

Pada Tabel 7 *Content-Security-Policy* (CSP) merupakan mekanisme penting yang membantu mencegahserangan *Cross-Site Scripting* (XSS) dan jenis serangan injeksi lainnya. Tanpa CSP yang tepat, aplikasi *web* rentan terhadap serangan di mana penyerang dapat menyuntikkan skrip berbahaya

ataukonten lain yang dapat dieksekusi di *browser* pengguna. Serangan semacam ini dapat menyebabkanpencurian data pengguna, modifikasi halaman *web*, atau bahkan penyebaran *malware*. CSP memungkinkan *developer* untuk mengontrol sumber daya apa saja yang boleh dimuat oleh *browser*,sehingga mengurangi risiko konten berbahaya dieksekusi. Rekomendasi untuk melindungi aplikasiweb dari serangan XSS dan injeksi konten lainnya, pengaturan *Content-Security-Policy* yang tepat harus diterapkan. Ini termasuk mendefinisikan sumber yang valid untuk skrip, gaya, gambar, dan media lainnya yang dapat dimuat oleh *browser*. Misalnya, hanya mengizinkan skrip dari domain yang tepercaya dan melarang penggunaan eval() di *JavaScript* dapat secara drastis mengurangi kemungkinan serangan XSS. Pengembang harus secara teratur meninjau dan memperbarui kebijakan ini untuk mencakup semua sumber daya yang digunakan oleh aplikasi mereka. Dengan klasifikasi sebagai berikut:

CWE: CWE-693 (Protection Mechanism Failure) OWASP Top 10 - 2021: A5 - Security MisconfigurationOWASP Top 10 - 2017: A6 - Security Misconfiguration

Tabel 8. *Server software and technology found*

Perangkat Lunak/Versi	**Kategori**
Nginx	*Web server, Reverse Proxies*
Openresty	*Web server*
PHP5.6.40	*Programing Laguages*
HSTS	*Scurity*

Pada Tabel 8 penyerang dapat menggunakan informasi ini untuk melakukan serangan spesifik terhadap jenis dan versi perangkat lunak yang teridentifikasi, dengan cara mengeksploitasi kerentanan yang sudah dikenal atau belum diperbaiki. Serangan ini bisa mencakup berbagai bentuk seperti injeksi kode berbahaya, eskalasi hak akses, atau penolakan layanan (DoS). Dampaknya bisa sangat merugikan, termasuk kompromi penuh sistem, pencurian atau manipulasi data sensitif, pengambilalihan kontrol atas aplikasi, dan gangguan signifikan terhadap operasional layanan. Selain itu, serangan yang berhasil juga dapat membuka pintu bagi serangan lanjutan atau memberikan akses bagi penyerang untuk menyebarkan malware atau ransomware ke dalam jaringan yang lebih luas. Rekomendasi untuk masalah ini sangat disarankan untuk menghapus atau menyembunyikan informasi yang dapat mengungkapkan detail teknis terkait platform perangkat lunak, teknologi, server, dan sistem operasi yang digunakan. Informasi ini mencakup, namun tidak terbatas pada, header server HTTP, metadata HTML, versi perangkat lunak yang ditampilkan, dan konfigurasi server yang terlihat. Pengungkapan informasi semacam ini dapat memberikan petunjuk berharga bagi penyerang untuk mengidentifikasi potensi kerentanan yang spesifik terhadap versi perangkat lunak atau konfigurasi sistem yang Anda gunakan, sehingga memperbesar risiko serangan yang ditargetkan. Dengan menghilangkan atau menutupi informasi tersebut, Anda dapat memperkuat keamanan sistem dan mengurangi peluang bagi penyerang untuk mengeksploitasi kelemahan yang ada.

C. Reporting

Reporting merupakan penyampaian hasil penelitian dan penjelasan mengenai kesimpulan dari hasil pengujiandan penilaian keamanan, hal ini dipaparkan pada bagian Kesimpulan [15]. Adapun hasil singkat *reporting*yakni sebagai berikut:

1. Penulis menemukan 6 *threat* utama yang terdapat pada sistem informasi akademik siakad.umi.ac.id dengan berbagai macam celah serta level mulai dari *low* hingga *high*. *Threat* ini mencakup kerentanan perangkat lunak sisi *server*, pengaturan *cookie* yang tidak aman, dan absennya berbagai *header* keamanan yang penting
2. Penulis tidak menemukan celah SQL *Injection* atau celah keamanan terkait injeksi lainnya pada sistem ini selama pemindaian dilakukan. Namun, perlu dicatat bahwa pemindaian yang dilakukan tidak mencakup uji mendalam terhadap semua jenis serangan injeksi.
3. Penulis mengidentifikasi beberapa potensi risiko yang serius berdasarkan hasil pemindaian, di antaranya:
    a. Kerentanan perangkat lunak sisi *server*: Beberapa komponen perangkat lunak yang digunakan di sisi *server* rentan terhadap eksploitasi, termasuk potensi eksekusi kode jarak jauh dan serangan*denial-of-service* (DoS).

b. Pengaturan *cookie* tidak aman: *Cookie* yang tidak dilindungi dengan *flag "Secure"* dan *"HttpOnly"* membuatnya rentan terhadap serangan *man-in-the-middle* (MITM) dan pencurian melalui skrip berbahaya (XSS).
c. Kurangnya *Header* Keamanan: *Header* seperti *X-Content-Type-Options, Referrer-Policy*, dan *Content-Security-Policy* tidak ada dalam *respons server*, yang meningkatkan risiko serangan XSS, *phishing,* dan kebocoran informasi.

Kesimpulan

Secara keseluruhan hal yang dapat disimpulkan dari hasil penlitian ini, antara lain:

1. Ditemukan beberapa celah kerentanan keamanan pada *website* siakad.umi.ac.id dengan menggunakan *tools vulnerability assessment* (*Pentest Tools*) di antaranya 1 *risk level high* yaitu *Multiple vulnerabilities in PHP versions 8.1., 4 risk level medium* yaitu *Insecure cookie settings dan Missing security headers* seperti *X-Content-Type-Options, Referrer-Policy*, dan *Content-Security-Policy,* 2 *risk level medium* yaitu *Cookie(s) missing Secure flag* dan *HttpOnly flag set,* 4 *Risk level low* yaitu *Missing security header: X-Content-Type-Options, Missing security header: Referrer-Policy, Missing security header: Content-Security-Policy*
2. Berdasarkan hasil analisa, celah kerentanan pada *website* siakad.umi.ac.id cukup serius, dengan temuan 1 *risk level high,* 2 *risk level medium*, dan 4 *risk level low*. Peneliti memberikan rekomendasi perbaikan terkait temuan celah keamanan pada siakad.umi.ac.id untuk melindungi sistem dari serangan *Remote Code Execution (RCE), Cross-Site Scripting (XSS), Cookie Hijacking, Clickjacking*, dan Informasi yang Bocor melalui *Header* yang Hilang.

Daftar Pustaka

[1] A. W. Wardhana and H. B. Seta, "Analisis Keamanan Sistem Pembelajaran Online Menggunakan Metode ISSAF pada Website Universitas XYZ," *Informatik : Jurnal Ilmu Komputer*, vol. 17, no. 3, p. 226, Dec. 2021, doi: 10.52958/iftk.v17i3.3653.

[2] S. Eko Prasetyo and N. Hassanah, "Analisis Keamanan Website Universitas Internasional Batam Menggunakan Metode ISSAF," *Jurnal Ilmiah Informatika*, vol. 9, no. 02, pp. 82–86, Sep. 2021, doi: 10.33884/jif.v9i02.3758.

[3] A. Zakir and D. Irwan, "Perancangan Sitem Informasi Pengajuan Kerja Praktek Pada Program Studi Sistem Informasi Menggunakan UML," *Jurnal Ilmiah Teknologi Informasi dan Robotika*, vol. 2, no. 2, Nov. 2023, doi: 10.33005/jifti.v2i2.34.

[4] J. Manajemen, M. Pendidikan, A. R. Hasan, C. Chotimah, and I. Junaris, "Analisis Manajemen Metode Perancangan Sistem Informasi Akademik Berbasis Web: Systematic Literatur Review," vol. 11, no. 2, pp. 47–55, doi: 10.23960/jmmp.v11.i2.2023.05.

[5] P. H. Marpaung, N. Dahri, and W. Yahyan, "Perancangan Sistem Informasi Pengolahan Data Mahasiswa Magang Di Perusahaan Berbasis WEB," *Jurnal Manajemen Teknologi Informatika*, vol. 1, no. 2, pp. 109–116, Aug. 2023, doi: 10.70038/jentik.v1i2.11.

[6] D. Laksmiati, "Vulnerability Assesment pada Situs WWW.HATSEHAT.COM Menggunakan OPENVAS," no. 3, pp. 240–246, Aug. 2020.

[7] Mira Orisa and M. Ardita, "V*ulnerability Assesment* Untuk Meningkatkan Kualitas Kamanan WEB," *Jurnal Mnemonic*, vol. 4, no. 1, pp. 16–19, Feb. 2021, doi: 10.36040/mnemonic.v4i1.3213.

[8] A. H. Harahap, C. Difa Andani, A. Christie, D. Nurhaliza, and A. Fauzi, "Pentingnya Peranan CIA Triad Dalam Keamanan Informasi dan Data Untuk Pemangku Kepentingan atau Stakholder," 2023.

[9] Muh. A. Mu'min, A. Fadlil, and I. Riadi, "Analisis Keamanan Sistem Informasi Akademik Menggunakan Open Web Application Security Project Framework," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 3, p. 1468, Jul. 2022, doi: 10.30865/mib.v6i3.4099.

[10] B. Arkin, S. Stender, and G. McGraw, "Software penetration testing," *IEEE Security and Privacy Magazine*, vol. 3, no. 1, pp. 84–87, Jan. 2005, doi: 10.1109/MSP.2005.23.

[11] B. P. Zen, R. A. G. Gultom, A. H. S. Reksoprodjo, P. T. Penginderaan, T. Pertahanan, and U. Pertahanan, "Analisis *Security Assesment* Menggunakan Metode *Penetration Testing* Dalam Menjaga Kapabilitas Keamanan Teknologi Informasi Pertahanan Negara," 2020.

[12] M. Yunus, "Analisis Kerentanan Aplikasi Berbasis WEB Menggunakan Kombinasi *Security Tools Project* Berdasarkan *Framework* OWASP Versi 4," *Jurnal Ilmiah Informatika Komputer*, vol. 24, no. 1, pp. 37–48, Apr. 2019, doi: 10.35760/ik.2019.v24i1.1988.

[13] E. I. Alwi, H. Herdianti, and F. Umar, "Analisis Keamanan Website Menggunakan Teknik Footprinting dan Vulnerability Scanning," *INFORMAL: Informatics Journal*, vol. 5, no. 2, p. 43, Aug. 2020, doi: 10.19184/isj.v5i2.18941.

[14] M. Fatkhurozzi, "Seminar Nasional Informatika Bela Negara (SANTIKA) Analisa Keamanan Website Menggunakan Metode Footprinting dan Vulnerability Scanning pada Website Kampus," 2021.
[15] E. Z. Darojat, E. Sediyono, and I. Sembiring, "Vulnerability Assessment Website E-Government dengan NIST SP 800-115 dan OWASP Menggunakan Web Vulnerability Scanner," *JURNAL SISTEM INFORMASI BISNIS*, vol. 12, no. 1, pp. 36–44, Sep. 2022, doi: 10.21456/vol12iss1pp36-44.